

uCosminexus DocumentBroker Version 3
統計解析ツール

文法書

3000-3-F11-20

対象製品

R-1M95D-13 uCosminexus DocumentBroker Server Version 3 03-60 (適用 OS : AIX 5L V5.1 , AIX 5L V5.2 , AIX 5L V5.3)

R-1595D-13 uCosminexus DocumentBroker Server Version 3 03-70 (適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 R2 x64 Edition , Windows Server 2008 x86 , Windows Server 2008 R2 , Windows Server 2012)

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

印の製品については、サポート時期をご確認ください。

輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

商標類

AIX は、米国およびその他の国における International Business Machines Corporation の商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

GIF は、米国 CompuServe Inc. が開発したフォーマットの名称です。

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

OLE は、米国 Microsoft Corporation が開発したソフトウェア名称です。

TIFF は、米国 Aldus Corp. が開発したフォーマットの名称です。

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

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

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

活文、PDFstaff は、株式会社日立ソリューションズの登録商標です。

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

2013 年 9 月 3000-3-F11-20

著作権

All Rights Reserved. Copyright (C) 2006, Hitachi, Ltd.

All Rights Reserved. Copyright (C) 2006, 2013, Hitachi Solutions, Ltd.

変更内容

変更内容 (3000-3-F11-20) DocumentBroker Server Version 3 03-70 (適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 R2 x64 Edition , Windows Server 2008 x86 , Windows Server 2008 R2 , Windows Server 2012)

追加・変更内容	変更箇所
前提 OS に Windows Server 2012 をサポートしました。	3.2(2)

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、次に示すプログラムプロダクトで提供する統計解析ツールのユーザインターフェースであるコマンドおよびコマンドが出力する情報について説明したものです。

- R-1M95D-13 uCosminexus DocumentBroker Server Version 3
- R-1595D-13 uCosminexus DocumentBroker Server Version 3

対象読者

このマニュアルは、uCosminexus DocumentBroker で構築した文書管理システムへのアクセス状況や、uCosminexus DocumentBroker サーバおよびクライアントのプログラムの処理内容を把握することで、システムを管理・運用する方を対象にしています。なお、次の内容を理解されていることを前提としています。

- uCosminexus DocumentBroker に関する知識
- UNIX または Windows に関する知識
- HiRDB に関する知識
- SQL 言語に関する知識

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 統計解析ツールの機能概要

統計解析ツールの機能概要と統計解析ツールが提供する EDMRptAclog コマンド、EDMRptTrace コマンドの機能概要について説明しています。

第 2 章 統計解析ツールの処理概要

EDMRptAclog コマンド、EDMRptTrace コマンドの処理概要について説明しています。

第 3 章 実行環境の設定

統計解析ツールの実行環境の設定について説明しています。

第 4 章 コマンドの実行に必要なファイル

EDMRptAclog コマンド、EDMRptTrace コマンドの実行に必要なファイルについて説明しています。

第 5 章 トラブルシュート機能

EDMRptAclog コマンド、EDMRptTrace コマンドを運用するためのトラブルシュート機能（コマンドの実行によって出力されるメッセージやエラー処理）について説明しています。

第 6 章 コマンドリファレンス

EDMRptAclog コマンド、EDMRptTrace コマンドの文法、機能の処理方式、実行結果の出力形式および出力例について説明しています。

付録 A 用語解説

このマニュアルで使用している用語について説明しています。

関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。なお、本文に記載のマニュアル名称は、「uCosminexus DocumentBroker」を「DocumentBroker」と表記しています。

uCosminexus DocumentBroker のマニュアル

DocumentBroker Version 3 システム導入・運用ガイド (3000-3-D01)

uCosminexus DocumentBroker Version 3 システム導入・運用ガイド (3020-3-U71)

uCosminexus DocumentBroker を使用する環境を定義, 管理および運用する場合に参照してください。

UNIX の場合は, 資料番号が 3000-3-D01 のマニュアルを参照してください。

Windows の場合は, 資料番号が 3020-3-U71 のマニュアルを参照してください。

uCosminexus DocumentBroker Version 3 クラスライブラリ C++ 解説 (3000-3-F13)

uCosminexus DocumentBroker Development Kit が提供するクラスライブラリの機能と, クラスライブラリを使用するために必要なオブジェクトモデルについて知りたい場合に参照してください。

uCosminexus DocumentBroker Version 3 クラスライブラリ C++ リファレンス 基本機能編 (3000-3-F14)

uCosminexus DocumentBroker Development Kit が提供するクラスライブラリのクラスの詳細とメソッドの文法について知りたい場合に参照してください。

uCosminexus DocumentBroker Version 3 オブジェクト操作ツール (3000-3-F15)

uCosminexus DocumentBroker Development Kit または uCosminexus DocumentBroker Runtime で提供するオブジェクト操作ツールの機能について知りたい場合に参照してください。

uCosminexus DocumentBroker Version 3 メッセージ (3000-3-F12)

uCosminexus DocumentBroker が出力するメッセージについて知りたい場合に参照してください。

関連製品のマニュアル (HiRDB)

- スケーラブルデータベースサーバ HiRDB Version 6 解説 (UNIX(R) 用) (3000-6-231)
- スケーラブルデータベースサーバ HiRDB Version 6 解説 (Windows(R) 用) (3020-6-121)
- スケーラブルデータベースサーバ HiRDB Version 7 解説 (UNIX(R) 用) (3000-6-271)
- スケーラブルデータベースサーバ HiRDB Version 7 解説 (Windows(R) 用) (3020-6-271)
- スケーラブルデータベースサーバ HiRDB Version 8 解説 (UNIX(R) 用) (3000-6-351)
- スケーラブルデータベースサーバ HiRDB Version 8 解説 (Windows(R) 用) (3020-6-351)
- スケーラブルデータベースサーバ HiRDB Version 6 システム導入・設計ガイド (UNIX(R) 用) (3000-6-232)
- スケーラブルデータベースサーバ HiRDB Version 6 システム導入・設計ガイド (Windows(R) 用) (3020-6-122)
- スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (UNIX(R) 用) (3000-6-272)
- スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (Windows(R) 用) (3020-6-272)
- スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (UNIX(R) 用) (3000-6-352)
- スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用) (3020-6-352)
- スケーラブルデータベースサーバ HiRDB Version 6 システム定義 (UNIX(R) 用) (3000-6-233)
- スケーラブルデータベースサーバ HiRDB Version 6 システム定義 (Windows(R) 用) (3020-6-123)
- スケーラブルデータベースサーバ HiRDB Version 7 システム定義 (UNIX(R) 用) (3000-6-273)
- スケーラブルデータベースサーバ HiRDB Version 7 システム定義 (Windows(R) 用) (3020-6-273)
- スケーラブルデータベースサーバ HiRDB Version 8 システム定義 (UNIX(R) 用) (3000-6-353)
- スケーラブルデータベースサーバ HiRDB Version 8 システム定義 (Windows(R) 用) (3020-6-353)
- スケーラブルデータベースサーバ HiRDB Version 6 システム運用ガイド (UNIX(R) 用) (3000-6-234)
- スケーラブルデータベースサーバ HiRDB Version 6 システム運用ガイド (Windows(R) 用) (3020-6-124)
- スケーラブルデータベースサーバ HiRDB Version 7 システム運用ガイド (UNIX(R) 用) (3000-6-274)
- スケーラブルデータベースサーバ HiRDB Version 7 システム運用ガイド (Windows(R) 用) (3020-6-274)
- スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド (UNIX(R) 用) (3000-6-354)
- スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド (Windows(R) 用) (3020-6-354)
- スケーラブルデータベースサーバ HiRDB Version 6 コマンドリファレンス (UNIX(R) 用) (3000-6-235)

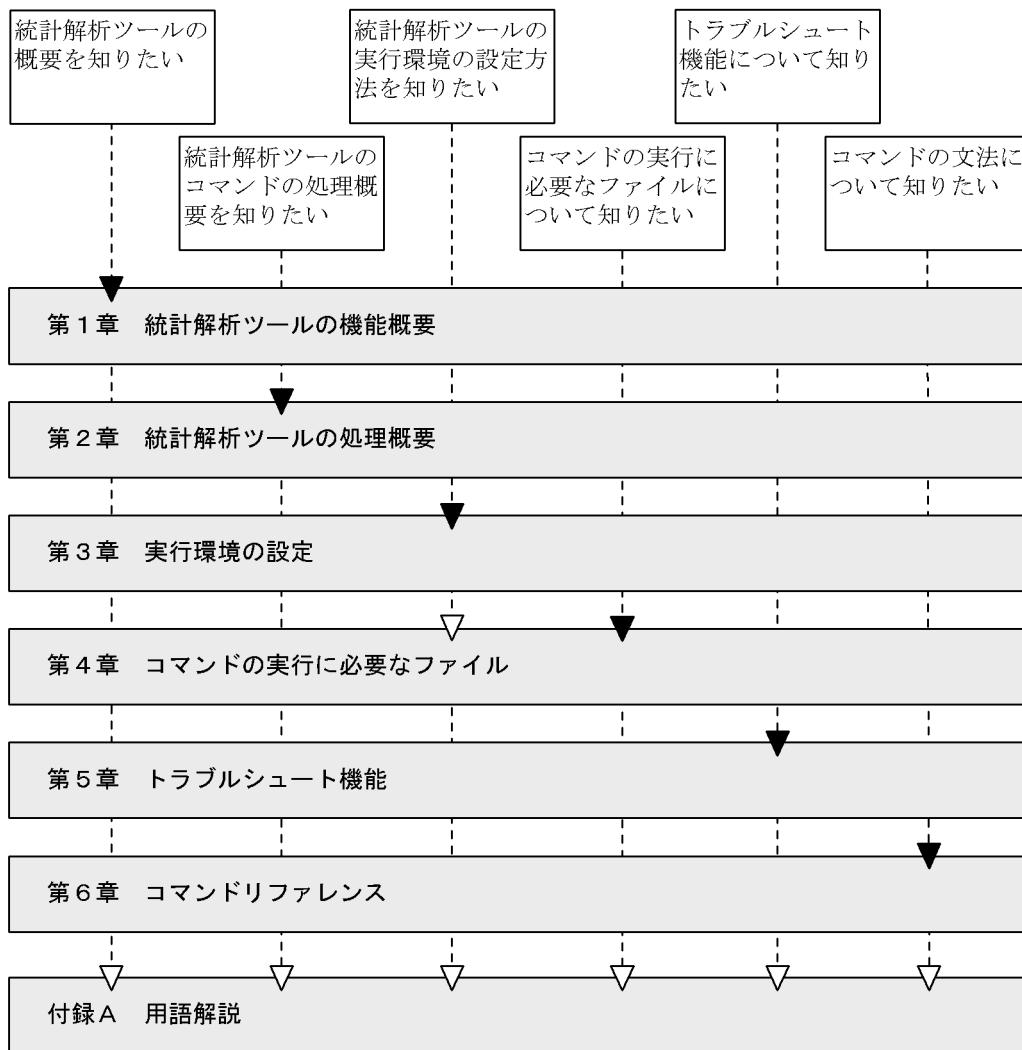
- スケーラブルデータベースサーバ HiRDB Version 6 コマンドリファレンス (Windows(R) 用)(3020-6-125)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (UNIX(R) 用)(3000-6-275)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (Windows(R) 用)(3020-6-275)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (UNIX(R) 用)(3000-6-355)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (Windows(R) 用)(3020-6-355)
- スケーラブルデータベースサーバ HiRDB Version 6 UAP 開発ガイド (UNIX(R)/Windows(R) 用)(3000-6-236)
- スケーラブルデータベースサーバ HiRDB Version 6 UAP 開発ガイド (Windows(R) 用)(3020-6-126)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R) 用)(3000-6-276)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (Windows(R) 用)(3020-6-276)
- スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド (3020-6-356)
- スケーラブルデータベースサーバ HiRDB Version 6 SQL リファレンス (UNIX(R)/Windows(R) 用)
(3000-6-237)
- スケーラブルデータベースサーバ HiRDB Version 6 SQL リファレンス (Windows(R) 用)(3020-6-127)
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (UNIX(R)/Windows(R) 用)
(3000-6-277)
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (Windows(R) 用)(3020-6-277)
- スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス (3020-6-357)
- スケーラブルデータベースサーバ HiRDB Version 6 メッセージ (UNIX(R)/Windows(R) 用)(3000-6-238)
- スケーラブルデータベースサーバ HiRDB Version 6 メッセージ (Windows(R) 用)(3020-6-128)
- スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (UNIX(R)/Windows(R) 用)(3000-6-278)
- スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (Windows(R) 用)(3020-6-278)
- スケーラブルデータベースサーバ HiRDB Version 8 メッセージ (3020-6-358)

注

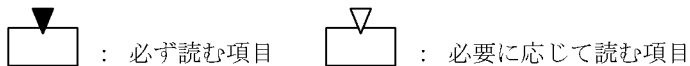
このマニュアルでは、これらのマニュアルを「HiRDB メッセージ」と表記しています。

読書手順

このマニュアルは、利用目的に合わせて章を選択してお読みいただけます。利用目的別の流れに従ってお読みいただくことをお勧めします。



(凡例)



このマニュアルでの表記

このマニュアルでは、製品名称を次に示す略称で表記しています。

製品名称	略称
AIX 5L V5.1	AIX
AIX 5L V5.2	
AIX 5L V5.3	
uCosminexus DocumentBroker Development Kit Version 3	DocumentBroker
uCosminexus DocumentBroker Runtime Version 3	
uCosminexus DocumentBroker Server Version 3	
uCosminexus DocumentBroker Development Kit Version 3	DocumentBroker Development Kit
uCosminexus DocumentBroker Runtime Version 3	DocumentBroker Runtime

製品名称	略称
uCosminexus DocumentBroker Server Version 3	DocumentBroker Server または DocumentBroker サーバ
uCosminexus DocumentBroker Development Kit Version 3	DocumentBroker クライアント
uCosminexus DocumentBroker Runtime Version 3	
HiRDB/Single Server Version 6	HiRDB
HiRDB/Single Server Version 7	
HiRDB/Single Server Version 8	
HiRDB/Parallel Server Version 6	
HiRDB/Parallel Server Version 7	
HiRDB/Parallel Server Version 8	
HiRDB/Run Time Version 6	
HiRDB/Run Time Version 7	
HiRDB/Run Time Version 8	
HiRDB/Single Server Version 6	
HiRDB/Single Server Version 7	
HiRDB/Single Server Version 8	
HiRDB/Parallel Server Version 6	HiRDB/Parallel Server
HiRDB/Parallel Server Version 7	
HiRDB/Parallel Server Version 8	
HiRDB/Run Time Version 6	HiRDB/Run Time
HiRDB/Run Time Version 7	
HiRDB/Run Time Version 8	
HiRDB/Single Server Version 6	HiRDB サーバ
HiRDB/Single Server Version 7	
HiRDB/Single Server Version 8	
HiRDB /Parallel Server Version 6	
HiRDB /Parallel Server Version 7	
HiRDB /Parallel Server Version 8	
Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版	
Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版	
Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition 日本語版	Windows Server 2003 R2
Microsoft(R) Windows Server(R) 2003 R2, Standard Edition 日本語版	
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition 日本語版	Windows Server 2003 R2 または Windows Server 2003 R2 x64 Edition
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition 日本語版	
Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版	Windows Server 2008 または Windows Server 2008 x86
Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版	
Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版	Windows Server 2008 R2
Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版	

製品名称	略称
Microsoft(R) Windows Server(R) 2012 Standard 日本語版	Windows Server 2012
Microsoft(R) Windows Server(R) 2012 Datacenter 日本語版	

このほか、このマニュアルでは、次に示す表記方法を使用しています。

- AIX を UNIX と表記することがあります。
- Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 R2 x64 Edition , Windows Server 2008 x86 , Windows Server 2008 R2 , および Windows Server 2012 を合わせて Windows と表記することがあります。
- 説明に使用する用語として「XXXX ログ」や「エントリ」(XXXX エントリも含む)を使用しています。原則として、ファイルに残された一つの記録を指す場合は「XXXX ログ」、ソフトウェアで扱う単位を指す場合は「エントリ」(XXXX エントリも含む)と表記します。

uCosminexus DocumentBroker のマニュアルで使用する略語

uCosminexus DocumentBroker のマニュアルで使用する英略語を次に示します。

英略語	英字での表記
ACE	<u>A</u> ccess <u>C</u> ontrol <u>E</u> lement
ACFlag	<u>A</u> ccess <u>C</u> ontrol <u>F</u> lag
ACL	<u>A</u> ccess <u>C</u> ontrol <u>L</u> ist
AIIM	<u>A</u> ssociation for <u>I</u> nformation and <u>I</u> mage <u>M</u> anagement International
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASCII	<u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange
BES	<u>B</u> ack <u>E</u> nd <u>S</u> erver
BLOB	<u>B</u> inary <u>L</u> arge <u>O</u> bject
BMP	<u>B</u> it <u>M</u> ap
BNF	<u>B</u> ackus <u>N</u> ormal <u>F</u> orm
BOA	<u>B</u> asic <u>O</u> bject <u>A</u> dapter
CD-ROM	<u>C</u> ompact <u>D</u> isc <u>R</u> ead <u>O</u> nly <u>M</u> emory
CGI	<u>C</u> ommon <u>G</u> ateway <u>I</u> nterface
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CPU	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
CR	<u>C</u> arriage <u>R</u> eturn
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
DAP	<u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
DAT	<u>D</u> igital <u>A</u> udio <u>T</u> ape
DB	<u>D</u> atab <u>a</u> se
DBMS	<u>D</u> atab <u>a</u> se <u>M</u> anagement <u>S</u> ystem
DCD	<u>D</u> ocument <u>C</u> ontent <u>D</u> escription
DDE	<u>D</u> ynamic <u>D</u> ata <u>E</u> xchange
DIT	<u>D</u> irectory <u>I</u> nformation <u>T</u> ree

英略語	英字での表記
DLL	<u>D</u> ynamic <u>L</u> inking <u>L</u> ibrary
DMA	<u>D</u> ocument <u>M</u> anagement <u>A</u> lliance
DN	<u>D</u> istinguished <u>N</u> ame
EOF	<u>E</u> nd of <u>F</u> ile
ESIS-B	<u>E</u> lement <u>S</u> tructure <u>I</u> nformation <u>S</u> et- <u>B</u> inary Format
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
FAM	<u>F</u> ile <u>A</u> ccess <u>M</u> odule
GIF	<u>G</u> raphics <u>I</u> nterchange <u>F</u> ormat
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
GUID	<u>G</u> lobally <u>U</u> nique <u>I</u> dentifier
HTML	<u>H</u> ypertext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> ypertext <u>T</u> ransfer <u>P</u> rotocol
IANA	<u>I</u> nternet <u>A</u> ssigned <u>N</u> umbers <u>A</u> uthority
ID	<u>I</u> dentifier
IPF	<u>I</u> tanium(R) <u>P</u> rocessor <u>F</u> amily
ISO	<u>I</u> nternational <u>O</u> rganization for Standardization
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandards
JPEG	<u>J</u> oint <u>P</u> hotographic <u>E</u> xpert <u>G</u> roup
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
LDAP	<u>L</u> ightweight <u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
LF	<u>L</u> ine <u>F</u> eed
MFC	<u>M</u> icrosoft <u>F</u> oundation <u>C</u> lass
MIME	<u>M</u> ultipurpose <u>I</u> nternet <u>M</u> ail <u>E</u> xtensions
OCR	<u>O</u> ptical <u>C</u> haracter <u>R</u> eaders
OIID	<u>O</u> bject <u>I</u> nstance <u>I</u> dentifier
OLE	<u>O</u> bject <u>L</u> inking and <u>E</u> mbedding
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
ORDB	<u>O</u> bject <u>R</u> elational <u>D</u> atabase
OS	<u>O</u> perating <u>S</u> ystem
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
PC	<u>P</u> ersonal <u>C</u> omputer
PDF	<u>P</u> ortable <u>D</u> ocument <u>F</u> ormat
RDB	<u>R</u> elational <u>D</u> atabase
RDN	<u>R</u> elative <u>D</u> istinguished <u>N</u> ame
RFC	<u>R</u> equest for <u>C</u> omment
RTF	<u>R</u> ich <u>T</u> ext <u>F</u> ormat
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage

英略語	英字での表記
TCP/IP	Transmission Control Protocol/Internet Protocol
TIFF	Tag Image File Format
UOC	User Own Coding
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

このマニュアルで使用する記号

このマニュアルで使用する記号を次に示します。

記号	意味
	横に並べられた複数の項目に対する項目間の区切りを示し、「または」を意味します。 (例) A B A または B を指定することを示します。
—	括弧で囲まれた複数項目のうち 1 項目に対し使用され、括弧内のすべてを省略したときシステムが取る標準値を示します。 (例) [A B] 「何も指定しない」か「A または B を指定する」ことを示します。何も指定しない場合は A が仮定されます。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号 で区切られている場合は、そのうちの一つを選択します。 (例) {A B C} A, B または C のどれかを指定することを示します。
[]	この記号で囲まれている項目は省略してもよいことを示します。複数の項目が横に並べて記述されている場合には、すべてを省略するか、記号 { } と同じくどれか一つを選択します。 (例 1) [A] 「何も指定しない」か「A を指定する」ことを示します。 (例 2) [B C] 「何も指定しない」か「B または C を指定する」ことを示します。
: :=	この記号の左にあるものを右にあるもので定義することを示します。 (例) A : := B 「A とは B である」と定義することを示します。
...	記述が省略されていることを示します。 (例) ABC... ABC の後ろに記述があり、その記述が省略されていることを示します。
< >	この記号で囲まれている項目は、該当する要素を指定することを示します。 (例) <プロパティ> プロパティを記述します。
()	この記号で囲まれている項目を省略しないで記述することを示します。 (例) (<プロパティ>) 項目 <プロパティ> は省略しないで記述します。
...	この記号の直前に示す記号を繰り返し、複数個指定できることを示します。 (例) (<プロパティ>)... プロパティは一つ以上で複数個、繰り返して指定できます。

このマニュアルで使用する構文要素

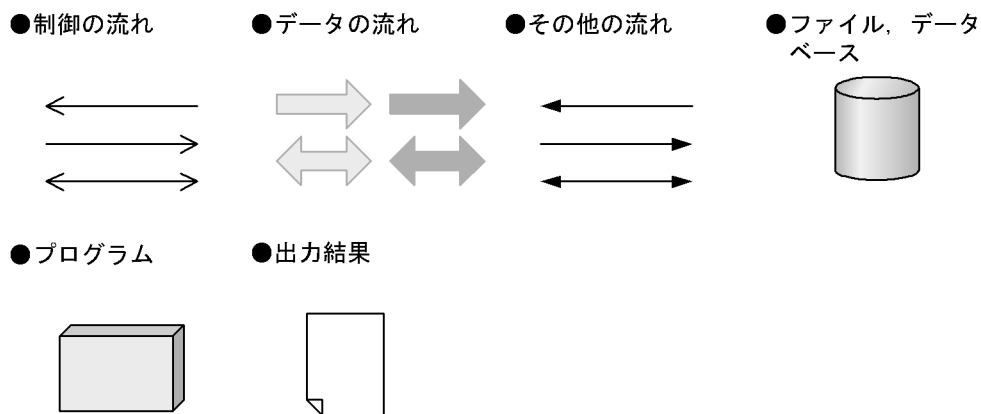
このマニュアルで使用する構文要素の種類を次に示します。

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

注 すべて半角文字を使用してください。

このマニュアルの図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。



常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については、常用漢字以外の漢字を使用しています。

個所(かしよ)

KB(キロバイト)などの単位表記について

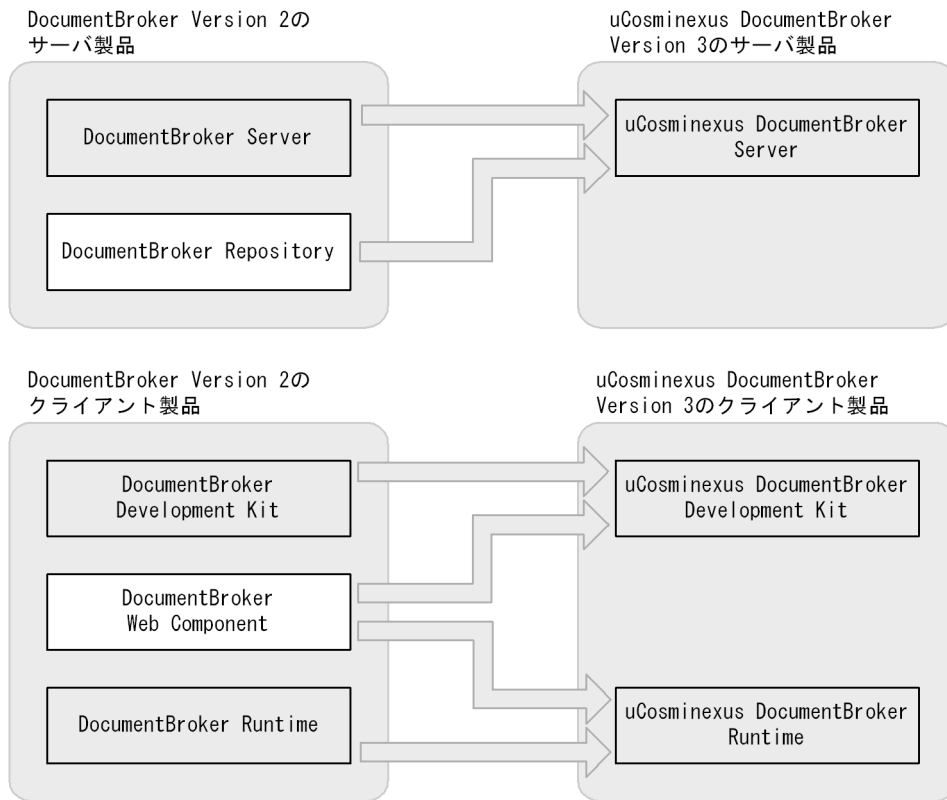
1KB(キロバイト), 1MB(メガバイト), 1GB(ギガバイト), 1TB(テラバイト)はそれぞれ 1,024 バイト, 1,024² バイト, 1,024³ バイト, 1,024⁴ バイトです。

DocumentBroker Version 2 と uCosminexus DocumentBroker Version 3 の製品体系の違い

uCosminexus DocumentBroker Version 3 では次のように製品の統合を行いました。

- DocumentBroker Repository を uCosminexus DocumentBroker Server に統合しました。
- DocumentBroker Web Component を uCosminexus DocumentBroker Development Kit および uCosminexus DocumentBroker Runtime に統合しました。

DocumentBroker Version 2 と uCosminexus DocumentBroker Version 3 の製品体系の違いを次に示します。



DocumentBroker Version 2 と uCosminexus DocumentBroker Version 3 のマニュアルの対応

バージョンアップおよび製品体系の変更に伴い、uCosminexus DocumentBroker Version 3 では次に示すようにマニュアル名称を変更しました。

Version 2 のマニュアル名称	Version 3 のマニュアル名称
DocumentBroker Version 2 システム導入・運用ガイド	uCosminexus DocumentBroker Version 3 システム導入・運用ガイド
DocumentBroker Version 2 クラスライブラリ 解説	uCosminexus DocumentBroker Version 3 クラスライブラリ C++ 解説
DocumentBroker Version 2 クラスライブラリ リファレンス 基本機能編	uCosminexus DocumentBroker Version 3 クラスライブラリ C++ リファレンス 基本機能編
DocumentBroker Version 2 クラスライブラリ リファレンス 概念 SGML 文書管理機能編	廃版
DocumentBroker Version 2 オブジェクト操作ツール	uCosminexus DocumentBroker Version 3 オブジェクト操作ツール
DocumentBroker Version 2 統計解析ツール	uCosminexus DocumentBroker Version 3 統計解析ツール
DocumentBroker Version 2 メッセージ	uCosminexus DocumentBroker Version 3 メッセージ
DocumentBroker Web Component Version 2 解説	uCosminexus DocumentBroker Version 3 クラスライブラリ Java 解説
DocumentBroker Web Component Version 2 リファレンス	uCosminexus DocumentBroker Version 3 クラスライブラリ Java リファレンス

Version 2 のマニュアル名称	Version 3 のマニュアル名称
DocumentBroker Web Component Version 2 サンプル Web アプリケーション	uCosminexus DocumentBroker Version 3 クラスライブラリ Java サンプル Web アプリケーション
DocumentBroker Text Search Index Loader Version 2	uCosminexus DocumentBroker Text Search Index Loader Version 3
DocumentBroker Rendering Option システム導入・運用ガイド	uCosminexus DocumentBroker Rendering Option Version 3
DocumentBroker Object Loader Version 2	uCosminexus DocumentBroker Object Loader Version 3

目次

1	統計解析ツールの機能概要	1
1.1	統計解析ツールとは	2
1.2	EDMRptAclog コマンドの機能概要	3
1.3	EDMRptTrace コマンドの機能概要	5
2	統計解析ツールの処理概要	7
2.1	EDMRptAclog コマンドの処理概要	8
2.2	EDMRptTrace コマンドの処理概要	9
3	実行環境の設定	11
3.1	実行環境設定の流れ	12
3.2	環境変数の設定	13
3.3	統計解析ツールのディレクトリ構成	15
4	コマンドの実行に必要なファイル	17
4.1	EDMRptAclog コマンドに必要なファイル	18
4.1.1	アクセスログファイル	18
4.1.2	プロパティ拡張情報ファイル	21
4.1.3	DocumentSpace 構成定義ファイル	24
4.1.4	集計キー定義ファイル	25
4.1.5	抽出エントリ定義ファイル	28
4.1.6	動作環境定義ファイル	31
4.2	EDMRptTrace コマンドに必要なファイル	35
4.2.1	トレースファイル	35
4.2.2	抽出エントリ定義ファイル	39
5	トラブルシューティング機能	41
5.1	EDMRptAclog コマンドのトラブルシューティング機能	42
5.1.1	メッセージ	42
5.1.2	OIID 変換, 操作対象識別子変換時のエラー処理	42
5.2	EDMRptTrace コマンドのトラブルシューティング機能	44
5.2.1	メッセージ	44
6	コマンドリファレンス	45
6.1	コマンドの記述形式	46

6.2	EDMRptAclog コマンドの詳細	48
6.2.1	EDMRptAclog コマンド (アクセスログの解析)	48
6.2.2	EDMRptAclog コマンドの機能の処理方式	52
6.2.3	EDMRptAclog コマンドの出力形式	59
6.2.4	EDMRptAclog コマンドの出力例	62
6.3	EDMRptTrace コマンドの詳細	70
6.3.1	EDMRptTrace コマンド (トレースの解析)	70
6.3.2	EDMRptTrace コマンドの機能の処理方式	75
6.3.3	EDMRptTrace コマンドの出力形式	79
6.3.4	EDMRptTrace コマンドの出力例	84

付録		91
付録 A	用語解説	92

索引		97
-----------	--	----

1

統計解析ツールの機能概要

この章では、統計解析ツールの機能概要と統計解析ツールが提供する EDMRptAclog コマンド、EDMRptTrace コマンドの機能概要について説明します。

1.1 統計解析ツールとは

1.2 EDMRptAclog コマンドの機能概要

1.3 EDMRptTrace コマンドの機能概要

1.1 統計解析ツールとは

DocumentBroker は、システムの管理・運用に有効な情報として、DocumentBroker で構築した文書管理システムへのユーザのアクセスの履歴や、DocumentBroker サーバおよびクライアントのプログラムの処理内容をファイルに出力しています。これらのファイルをそれぞれ、アクセスログファイルおよびトレースファイルといいます。

統計解析ツールは、DocumentBroker が出力したアクセスログファイルおよびトレースファイルの内容を解析・分類・集計して、レポートを出力するプログラムです。DocumentBroker の管理者は、統計解析ツールが出力するレポートを参照することによって、文書管理システムへのユーザのアクセス状況や DocumentBroker サーバおよびクライアントのプログラムごとの処理内容を把握できます。

なお、トレースファイルは、サーバのプログラムの場合はサーバトレース、クライアントのプログラムの場合はクライアントトレースといいます。

アクセスログファイルについては、「4.1.1 アクセスログファイル」を参照してください。トレースファイルについては、「4.2.1 トレースファイル」を参照してください。

統計解析ツールは、解析対象の違いによってユーザインターフェースとして次の表に示すコマンドを提供しています。

表 1-1 統計解析ツールが提供するコマンドの一覧

コマンド	機能
EDMRptAclog	アクセスログファイルの内容の解析・分類・集計
EDMRptTrace	トレースファイルの内容の解析・分類・集計

各コマンドについて次に説明します。

1.2 EDMRptAclog コマンドの機能概要

この節では、EDMRptAclog コマンドの目的と機能について説明します。

(1) EDMRptAclog コマンドの目的

EDMRptAclog コマンドは、DocumentBroker で構築した文書管理システムへのユーザのアクセス状況を把握することで、システムの管理・運用に役立てていただくことを目的としています。このため、EDMRptAclog コマンドは、DocumentBroker が出力したアクセスログファイルの内容を解析・分類・集計して、レポートを出力します。

(2) EDMRptAclog コマンドの機能

EDMRptAclog コマンドは次に示す機能を提供しています。

- 集計機能
- OIID 変換機能
- 操作対象識別子変換機能
- エントリ抽出機能

各機能の概要を次に示します。

(a) 集計機能

アクセスログファイルに残された記録を、アクセスログといいます。集計機能は、このアクセスログの内容を、ユーザ名およびオブジェクト名で分類・集計して、その結果を出力します。

集計機能の詳細については、「6.2.2(1) 集計機能の処理方式 (EDMRptAclog コマンドの集計方法)」を参照してください。また、集計機能の出力形式については、「6.2.3(1) 集計機能の出力形式」を参照してください。

なお、キーとなる項目で集計する場合、キーとなる項目を、集計キー定義ファイルにあらかじめ定義しておく必要があります。集計キー定義ファイルについては、「4.1.4 集計キー定義ファイル」を参照してください。

(b) OIID 変換機能、操作対象識別子変換機能

アクセスログには、アクセス対象となったオブジェクトの情報が、次に示す DocumentBroker 内部で生成された識別子で出力されます。

OIID

バージョン識別子

リレーション識別子

リンク識別子

なお、このマニュアルでは、アクセスログに出力される OIID 以外の識別子を操作対象識別子と呼びます。

OIID 変換機能は、アクセスログに出力された OIID を、その OIID が示すオブジェクトのプロパティの値（文書名など）に変換します。また、操作対象識別子変換機能は、アクセスログに出力された操作対象識別子を、その操作対象識別子が示すオブジェクトのプロパティの値（文書名など）に変換します。

OIID や操作対象識別子は DocumentBroker 内部で生成される識別子であるため、ユーザは OIID や操作対象識別子を見ても、その識別子がどのオブジェクトを示しているかわかりません。OIID 変換、操作対

1. 統計解析ツールの機能概要

象識別子変換をすることで、アクセスログ中のアクセス対象となったオブジェクトがわかるようになります。

OIID 変換，操作対象識別子変換の結果については、「表 6-4 アクセスログ表の例」または「6.2.3(2) OIID 変換機能，操作対象識別子変換機能 (-c オプション指定) の出力形式」を参照してください。

なお，一つのオブジェクトには，複数のプロパティが含まれているのが普通です。したがって，OIID 変換，操作対象識別子変換をするためには，オブジェクトに含まれている複数のプロパティのうち OIID 変換，操作対象識別子変換をするプロパティを，プロパティ拡張情報ファイルにあらかじめ定義しておく必要があります。プロパティ拡張情報ファイルについては、「4.1.2 プロパティ拡張情報ファイル」を参照してください。

また，OIID 変換を実行するかどうか，どの操作対象識別子に対して操作対象識別子変換を実行するかどうかを，動作環境定義ファイルで定義します。動作環境定義ファイルについては、「4.1.6 動作環境定義ファイル」を参照してください。

(c) エントリ抽出機能

エントリとは，ソフトウェアで扱う一つの単位となる情報です。EDMRptAclog コマンドは，アクセスログファイルに記録されている一つ一つのアクセスログをエントリとして扱います。

アクセスログエントリには，アクセスの内容を表すメソッド名と引数が含まれています。エントリ抽出機能は，メソッド名と引数によってアクセスログエントリを抽出して，その結果を出力します。

エントリ抽出機能の出力形式については、「6.2.3(3) エントリ抽出機能の出力形式」を参照してください。

なお，エントリを抽出する場合，抽出するエントリのメソッド名と引数を，抽出エントリ定義ファイルにあらかじめ定義しておく必要があります。抽出エントリ定義ファイルについては、「4.1.5 抽出エントリ定義ファイル」を参照してください。

1.3 EDMRptTrace コマンドの機能概要

この節では、EDMRptTrace コマンドの目的と機能について説明します。

(1) EDMRptTrace コマンドの目的

EDMRptTrace コマンドは、DocumentBroker サーバおよびクライアントのプログラムの処理内容を把握することで、システムの管理・運用に役立てていただくことを目的としています。このため、EDMRptTrace コマンドは、DocumentBroker が出力したトレースファイルの内容を解析・分類・集計して、レポートを出力します。

(2) EDMRptTrace コマンドの機能

EDMRptTrace コマンドは次に示す機能を提供しています。

- 集計機能
- トレースファイル分割機能
- エントリ抽出機能

各機能の概要を次に示します。

(a) 集計機能

トレースファイルに残された記録を、トレースログといいます。集計機能は、このトレースログの内容を、ユーザプログラムのプログラム名、プロセス ID およびスレッド ID で分類・集計して、その結果を出力します。

集計機能の出力形式については、「6.3.3(1) 集計機能の出力形式」を参照してください。

(b) トレースファイル分割機能

トレースファイル分割機能は、トレースファイルの内容を、ユーザプログラムのプログラム名、プロセス ID およびスレッド ID ごとに分類し、分類された内容を分割して別々のファイルに出力します。したがって、プログラムごと、プロセスごと、スレッド ID ごとの処理内容を把握できます。

トレースファイル分割機能の出力形式については、「6.3.3(2) トレースファイル分割機能の出力形式」を参照してください。

(c) エントリ抽出機能

EDMRptTrace コマンドは、トレースファイルに記録されている一つ一つのトレースログをエントリとして扱います。

トレースログエントリには、プログラムの処理内容を表す処理の種別が含まれています。エントリ抽出機能は、処理の種別によってトレースログエントリを抽出して、その結果を出力します。

エントリ抽出機能の出力形式については、「6.3.3(3) エントリ抽出機能の出力形式」を参照してください。

なお、エントリを抽出する場合、抽出するエントリの処理の種別を、抽出エントリ定義ファイルにあらかじめ定義しておく必要があります。抽出エントリ定義ファイルについては、「4.1.5 抽出エントリ定義ファイル」を参照してください。

2

統計解析ツールの処理概要

この章では，EDMRptAclog コマンド，EDMRptTrace コマンドの処理概要について説明します。

2.1 EDMRptAclog コマンドの処理概要

2.2 EDMRptTrace コマンドの処理概要

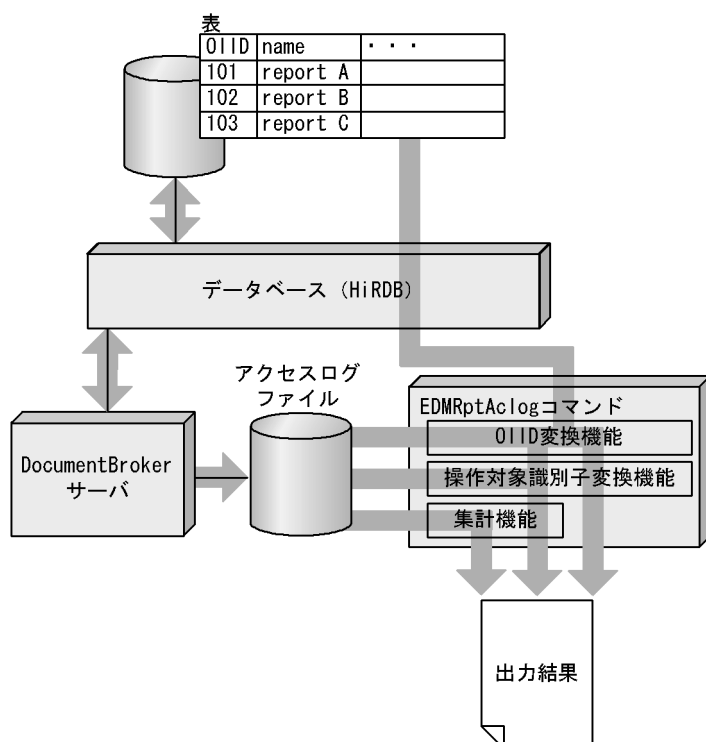
2.1 EDMRptAclog コマンドの処理概要

この節では、EDMRptAclog コマンドの処理概要について説明します。

EDMRptAclog コマンドは、DocumentBroker が出力したアクセスログファイルを入力として、レポートを出力します。OIDID、操作対象識別子をプロパティの値に変換するためにデータベース（HiRDB）にアクセスしますが、DocumentBroker にはアクセスしません。

EDMRptAclog コマンドによる処理の概要を次の図に示します。

図 2-1 EDMRptAclog コマンドによる処理の概要



EDMRptAclog コマンドによる処理の詳細については、「6.2.2 EDMRptAclog コマンドの機能の処理方式」を参照してください。

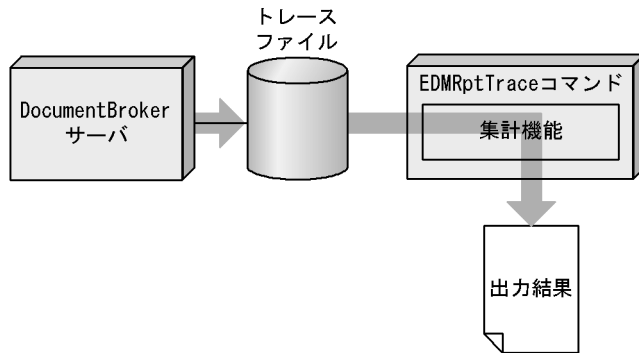
2.2 EDMRptTrace コマンドの処理概要

この節では、EDMRptTrace コマンドの処理概要について説明します。

EDMRptTrace コマンドは、DocumentBroker が出力したトレースファイルを入力として、レポートを出力します。EDMRptAclog コマンドと異なり、データベース（HiRDB）にはアクセスしません。

EDMRptTrace コマンドによる処理の概要を次の図に示します。

図 2-2 EDMRptTrace コマンドによる処理の概要



EDMRptTrace コマンドによる処理の詳細については、「6.3.2 EDMRptTrace コマンドの機能の処理方式」を参照してください。

3

実行環境の設定

この章では，統計解析ツールの実行環境の設定について説明します。

3.1 実行環境設定の流れ

3.2 環境変数の設定

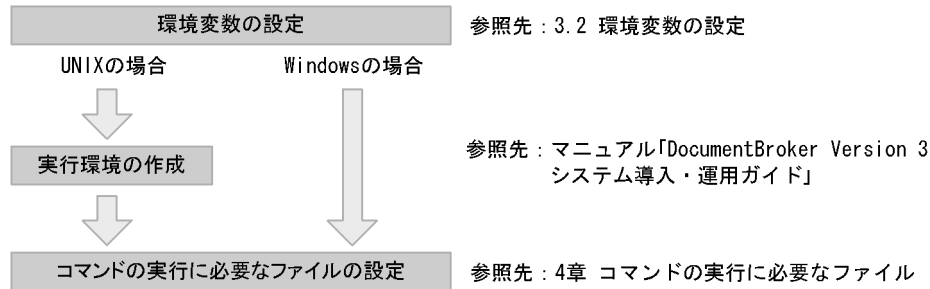
3.3 統計解析ツールのディレクトリ構成

3.1 実行環境設定の流れ

この節では、統計解析ツールの実行環境設定の流れについて説明します。

実行環境設定の流れを次の図に示します。

図 3-1 実行環境設定の流れ



説明：

1. 環境変数を設定します。

統計解析ツールでは、次の環境変数の設定が必要です。

AIX の場合

- LANG
- TZ
- DOCBROKERDIR
- LIBPATH
- EDMARUMSG

Windows の場合

- TZ
- DOCBROKERDIR
- PATH
- EDMARUMSG

なお、EDMARUMSG は、必要に応じて設定してください。

環境変数の設定については、「3.2 環境変数の設定」を参照してください。

2. 統計解析ツールの実行環境を作成します。(UNIX の場合)

統計解析ツールの実行環境は、DocumentBroker の実行環境作成コマンド (EDMSetup) で作成します。DocumentBroker の実行環境作成コマンドについては、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

統計解析ツールのディレクトリ構成については、「3.3 統計解析ツールのディレクトリ構成」を参照してください。

3. コマンドの実行に必要なファイルを設定します。

EDMRptAclog コマンド、EDMRptTrace コマンドの実行に必要なファイルを設定します。ファイルの設定については、「4. コマンドの実行に必要なファイル」を参照してください。

3.2 環境変数の設定

この節では、統計解析ツールを使用するために必要な環境変数の設定について説明します。

設定する環境変数は、ご使用の OS ごとに異なります。

(1) 環境変数の設定 (AIX の場合)

環境変数は、各サーバマシンのログインシェル環境に合わせて次のファイルに設定します。なお、「\$HOME」は、ログインユーザのホームディレクトリを示します。

Bourne シェル環境のとき

\$HOME/.profile

C シェル環境のとき

\$HOME/.cshrc または \$HOME/.login

(a) LANG

使用する文字コードセットを設定します。この環境変数に設定する値によって、メッセージの言語種別が変わります。使用できる文字コードセットを次に示します。

日本語の場合

使用できる文字コードセットは、Shift-JIS コードです。設定する値は「Ja_JP」です。

英語の場合

使用できる文字コードセットは、ASCII コードです。設定する値は「C」です。

(b) TZ

マシン時間のタイムゾーンを設定します。設定する値は「JST-9」です。

(c) DOCBROKERDIR

DocumentBroker の実行環境を作成するディレクトリを設定します。

(d) LIBPATH

関連プログラムのライブラリのインストールディレクトリを設定します。次の値を追加してください。

- /opt/hitachi/common/lib

また、EDMRptAclog コマンドを実行するには、HiRDB のライブラリのインストールディレクトリも設定しておく必要があります。次の値を追加してください。

- /HiRDB/Run Time インストールディレクトリ /client/lib

(e) EDMARUMSG (メッセージトレースを出力する場合)

EDMRptAclog コマンドおよび EDMRptTrace コマンドのメッセージトレースを出力するディレクトリを設定します。この環境変数を設定しない場合、メッセージトレースは出力されません。

(2) 環境変数の設定 (Windows の場合)

環境変数は、コントロールパネルの [システム] - [詳細設定] タブの [環境変数] ボタンをクリックして表示される [環境変数] ダイアログ (Windows Server 2003, Windows Server 2003 R2 の場合), またはコントロールパネルの [システム] - [システムの詳細設定] - [詳細設定] タブの [環境変数] ボタ

3. 実行環境の設定

ンをクリックして表示される [環境変数] ダイアログ (Windows Server 2008 , Windows Server 2008 R2 , Windows Server 2012 の場合) で設定します。

(a) TZ

マシン時間のタイムゾーンを設定します。設定する値は「JST-9」です。

(b) DOCBROKERDIR

DocumentBroker の実行環境を作成するディレクトリとして、DocumentBroker Server のディレクトリが設定されていることを確認してください。

(c) PATH

EDMRptAclog コマンドを実行するには、HiRDB のライブラリのインストールディレクトリを設定しておく必要があります。次の値のうちどれか一つが設定されていることを確認してください。

- :HiRDB/Parallel Server インストールディレクトリ %client%\utl
- :HiRDB/Single Server インストールディレクトリ %client%\utl
- :HiRDB/Run Time インストールディレクトリ %client%\utl

(d) EDMARUMSG (メッセージトレースを出力する場合)

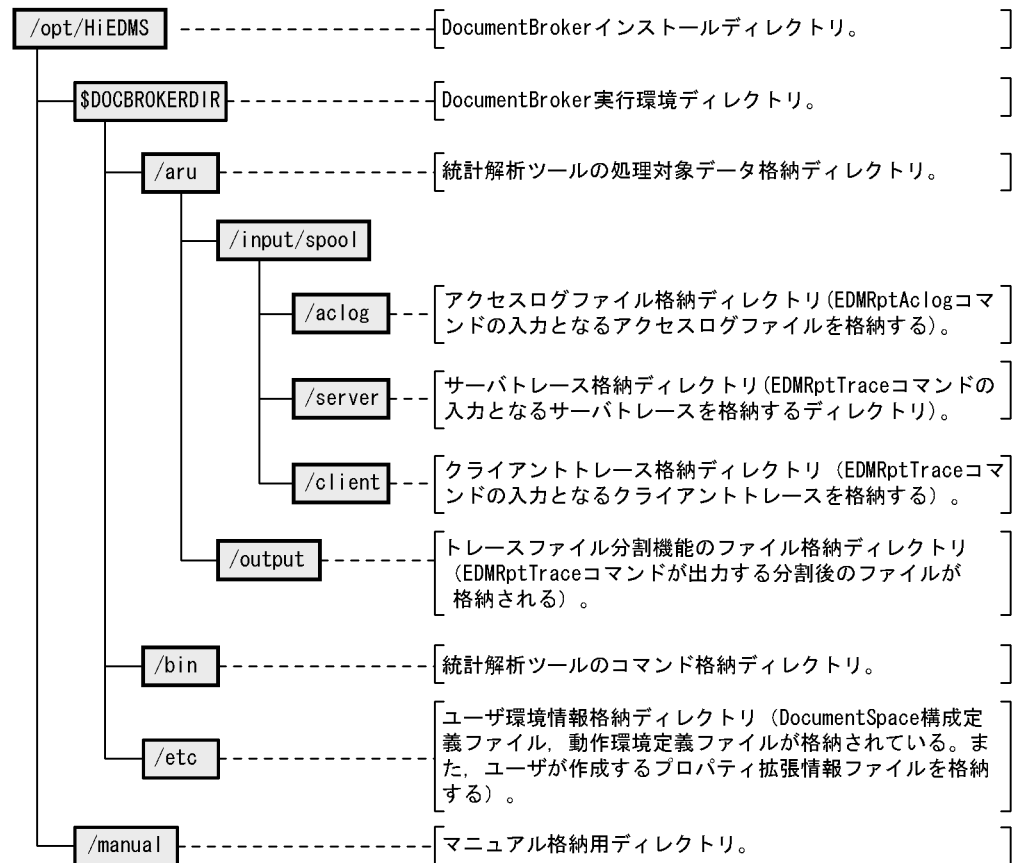
EDMRptAclog コマンドおよび EDMRptTrace コマンドのメッセージトレースを出力するディレクトリを設定します。この環境変数を設定しない場合、メッセージトレースは出力されません。

3.3 統計解析ツールのディレクトリ構成

この節では、統計解析ツールのディレクトリ構成について説明します。

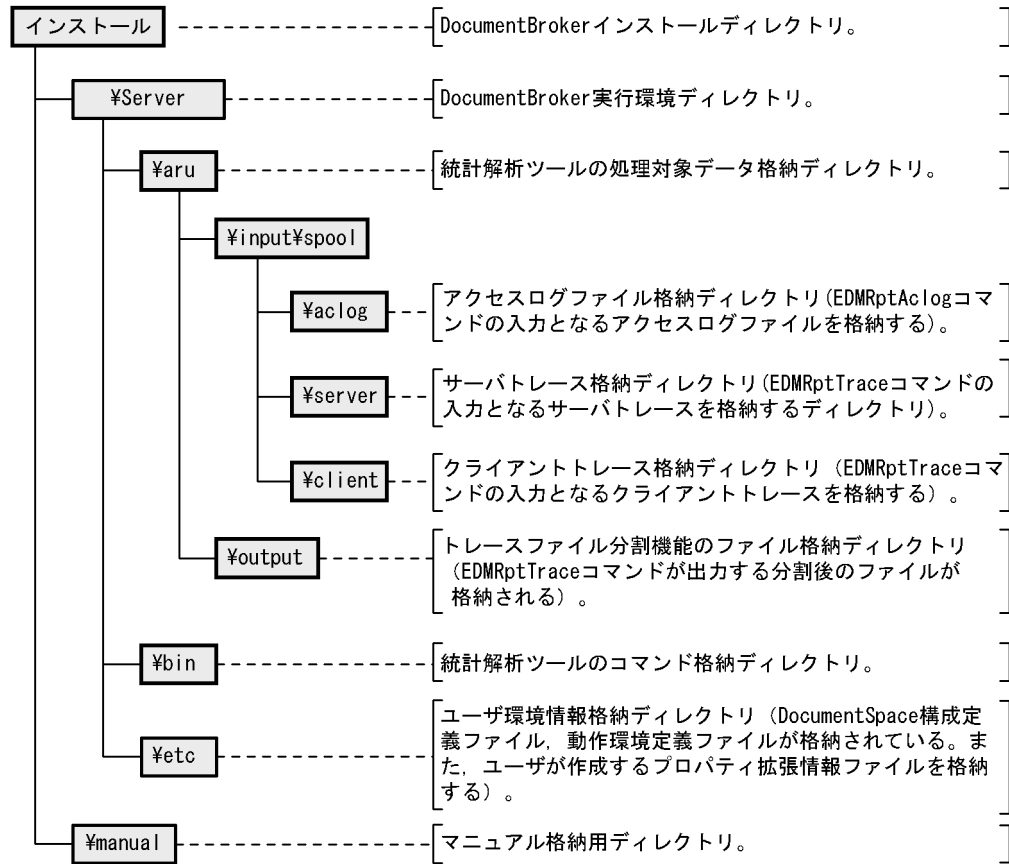
統計解析ツールのディレクトリ構成（UNIX の場合）を次の図に示します。

図 3-2 統計解析ツールのディレクトリ構成（UNIX の場合）



統計解析ツールのディレクトリ構成（Windows の場合）を次の図に示します。

図 3-3 統計解析ツールのディレクトリ構成 (Windows の場合)



4

コマンドの実行に必要なファイル

この章では、EDMRptAclog コマンド、EDMRptTrace コマンドの実行に必要なファイルについて説明します。

4.1 EDMRptAclog コマンドに必要なファイル

4.2 EDMRptTrace コマンドに必要なファイル

4.1 EDMRptAclog コマンドに必要なファイル

この節では、EDMRptAclog コマンドの実行に必要なファイルについて説明します。EDMRptAclog コマンドの実行に必要なファイルを次に示します。

- アクセスログファイル
- プロパティ拡張情報ファイル
- DocumentSpace 構成定義ファイル
- 集計キー定義ファイル
- 抽出エントリ定義ファイル
- 動作環境定義ファイル

4.1.1 アクセスログファイル

アクセスログファイルは、DocumentBroker で構築された文書管理システムへのユーザのアクセスの履歴を記録したファイルです。EDMRptAclog コマンドは、アクセスログファイルに記録された内容を入力として、オプションで指定された処理を実行します。なお、アクセスログ取得機能の詳細については、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

アクセスログファイルは、EDMRptAclog コマンドの引数で指定できます。指定を省略する場合は、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリにあるファイルがコマンドの対象になります。

EDMRptAclog コマンドは、DocumentBroker がアクセスログファイルを生成するディレクトリにあるファイルを対象にして実行できません。このため、EDMRptAclog コマンドを実行する前に、DocumentBroker が生成したアクセスログファイルを、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリにコピーして、コマンド引数でのアクセスログファイルの指定を省略することをお勧めします。

(1) ファイルのコピー

アクセスログファイルのコピー手順は、ご使用の OS ごとに異なります。

(a) UNIX の場合

DocumentBroker は「DocumentBroker 実行環境ディレクトリ /spool/aclog」に「EDMAccess_"NO".log」という名称でアクセスログファイルを生成します。ファイル名の "NO" は 1 ~ 16 の通し番号です。

このアクセスログファイルを、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリ「DocumentBroker 実行環境ディレクトリ /aru/input/spool/aclog」にコピーしてください。

cp コマンドを使用したコピー手順の例を次に示します。なお、\$DOCBROKERDIR は、DocumentBroker 実行環境ディレクトリです。

```
cp $DOCBROKERDIR/spool/aclog/* /$DOCBROKERDIR/aru/input/spool/aclog
```

(b) Windows の場合

DocumentBroker は「DocumentBroker インストールディレクトリ %Server%\spool\aclog」に「EDMAccess_"NO".log」という名称でアクセスログファイルを生成します。ファイル名の "NO" は 1 ~ 16 の通し番号です。

このアクセスログファイルを、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリ「DocumentBroker インストールディレクトリ ¥Server¥aru¥input¥spool¥aclog」にコピーしてください。

(2) 出力形式

アクセスログの出力形式およびアクセスログの項目について説明します。

(a) アクセスログの出力形式

アクセスログの出力形式について説明します。

アクセスログファイル中のアクセスログは、アクセスの情報を示す複数の項目で構成されます。アクセスログの出力形式を次に示します。

yyyy/mm/dd hh:mm:ss.sss UserID LogInformation

- アクセスログファイルは、ASCII コードで構成されます。
- エントリは改行で区切られます。

なお、EDMRptAclog コマンドは、yyyy/mm/dd hh:mm:ss.sss で始まらないエントリを無視します。

(b) アクセスログの項目

アクセスログの各項目について説明します。

yyyy/mm/dd

アクセスした日付（年 / 月 / 日）が出力されます。

hh:mm:ss.sss

アクセスした時刻（時 : 分 : 秒 . ミリ秒）が出力されます。

UserID

アクセスしたユーザのユーザ ID が 8 ~ 32 バイトで出力されます。8 バイトに満たないユーザ ID は、満たない部分に空白が出力されます。

LogInformation

アクセスの内容を表す複数の文字列が、半角空白で区切られて出力されます。

このアクセスの内容にはメソッド名と引数が含まれています。EDMRptAclog コマンドの集計機能は、そのメソッド名と引数を基に、次に示す七つの項目（アクセス種別）に分類して集計します。

- connect
文書空間への接続
- create
オブジェクトの生成
- delete
オブジェクトの削除
- error
エラー
- refer
オブジェクトの参照
- search
検索の実行（検索結果の取得も含む）
- update
オブジェクトの更新

4. コマンドの実行に必要なファイル

集計機能で分類されるアクセスの内容と集計項目（アクセス種別）の関係を次の表に示します。

表 4-1 集計機能で分類されるアクセスの内容と集計項目（アクセス種別）の関係

アクセスの内容		集計項目（アクセス種別）						
メソッド名	引数	connect	create	delete	error	refer	search	update
BIND	ACLS							
CHANGE	ACL MODE							
	MASTER RENDITION							
	SEARCH MODE							
CHECKIN	VERSION							
CHECKOUT	VERSION							
CONNECT	なし							
	OBJECT	-	-	-	-	-	-	-
CREATE	OBJECT							
	INDEX							
	RELATION							
	RENDITION							
DELETE	OBJECT							
	INDEX							
	RELATION							
	RENDITION							
	RESULT							
	VERSION							
DISCONNECT	なし	-	-	-	-	-	-	-
DOWNLOAD	CONTENT							
ERROR	なし							
EXECUTE	SEARCH							
	SEARCH & GET RESULT							
	なし	-	-	-	-	-	-	-
GET	RESULT	-	-	-	-	-	-	-
	ACL MODE							
	PROPERTIES							
	RESERVATION							
LINK	なし							
LIST	ACLS							
	CHILDREN							
	OBJECTS							
	PARENTS							
	RELATIONS							
	RENDITIONS							
	VERSIONABLES							

アクセスの内容		集計項目 (アクセス種別)						
メソッド名	引数	connect	create	delete	error	refer	search	update
	VERSIONS							
PREPARE	なし	-	-	-	-	-	-	-
RELEASE	OBJECT	-	-	-	-	-	-	-
REVOKE	VERSION							
SET	MODE FIX							
	MODE FLOATING							
	PROPERTIES							
UNBIND	ACLS							
UNLINK	なし							
UPLOAD	CONTENT							

(凡例)

空欄：分類されない集計項目であることを示す。

：分類される集計項目であることを示す。

-：分類・集計の対象ではないアクセスの内容であることを示す。

注

個々のアクセスログを区別する場合、アクセスの内容に含まれるメソッド名(引数を含めることもあります)を使用して「CONNECT エントリ」と表記することがあります。

(3) アクセスログファイルの出力例

アクセスログファイルの出力例を次に示します。

```
2000/06/19 02:18:19.388 ACUser6 CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d
2000/06/19 02:18:52.261 ACUser6 DISCONNECT
2000/06/19 02:19:16.420 ACUser6 CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d
2000/06/19 02:21:50.877 ACUser6 CREATE OBJECT dma:///07a17522-a626-11d0-b11f-0020af27a837/
673d2be0-d1fd-11d0-ab59-08002be29e1d/f5d8e3a1-284c-11d2-9177-0000e2130367000000000001216
2000/06/19 02:22:37.607 ACUser6 GET PROPERTIES dma:///07a17522-a626-11d0-b11f-0020af27a837/
673d2be0-d1fd-11d0-ab59-08002be29e1d/f5d8e3a1-284c-11d2-9177-0000e2130367000000000001216
2000/06/19 02:22:54.537 ACUser6 RELEASE OBJECT dma:///07a17522-a626-11d0-b11f-0020af27a837/
673d2be0-d1fd-11d0-ab59-08002be29e1d/f5d8e3a1-284c-11d2-9177-0000e2130367000000000001216
2000/06/19 02:36:13.963 ACUser6 DISCONNECT
```

4.1.2 プロパティ拡張情報ファイル

プロパティ拡張情報ファイルは、DocumentBroker の文書モデルを構成するクラスに定義されている複数のプロパティの中から、OID変換、操作対象識別子変換の対象となるプロパティを特定するために使用するファイルです。

プロパティ拡張情報ファイルに、OID変換、操作対象識別子変換の対象となるプロパティを指定しておきます。EDMRptAclog コマンドは、データベース (HiRDB) からアクセス対象のオブジェクトのプロパティの値を取得して、アクセスログ中の OID、操作対象識別子をプロパティの値 (文書名など) に変換する際に、プロパティ拡張情報ファイルを参照します。プロパティ拡張情報ファイルがない場合、EDMRptAclog コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されます。

プロパティ拡張情報ファイルは、ユーザが作成する必要があります。

4. コマンドの実行に必要なファイル

(1) 記述規則

プロパティ拡張情報ファイルの記述規則は次のとおりです。

次のディレクトリに次のファイル名で作成してください。

ディレクトリ

- UNIX の場合

DocumentBroker 実行環境ディレクトリ /etc

- Windows の場合

DocumentBroker インストールディレクトリ %Server%etc

ファイル名

propex.ini

テキストファイルとして作成してください。

ASCII コードで記述してください。ただし、コメント行では Shift-JIS コードも使用できます。

シャープ (#) およびセミコロン (;) で始まる行はコメント行として扱われます。記述内容を説明する場合やプロパティ拡張情報ファイルを修正した場合のコメントなど、必要に応じて記述してください。

先頭や末尾の空白・タブは無視されます。

空白・改行だけの行は無視されます。

ファイルの先頭の行 (コメント行を除く) には、セクションを記述してください。

(2) 記述形式

プロパティ拡張情報ファイルの記述形式を次に示します。

# property extension information File	コメント
[<クラス名>]	セクション
ClassId=<GUID文字列>	エントリ
PrepresentProperties=<プロパティ名, プロパティ名, プロパティ名>	エントリ
:	
[<クラス名>]	セクション
ClassId=<GUID文字列>	エントリ
PrepresentProperties=<プロパティ名>	エントリ

プロパティ拡張情報ファイルは、複数のクラス名セクションと、各セクションに指定する ClassId エントリ、PrepresentProperties エントリで構成されます。

各セクションとエントリの記述について説明します。

(a) セクション

OID 変換、操作対象識別子変換をするプロパティを定義している DocumentBroker のクラスまたはそのサブクラスのクラス名を、次に示す形式で記述してください。

[<クラス名>]

クラス名セクションは、次に示す文法に従って記述してください。

クラス名は角括弧 ([]) で囲んでください。

なお、次の行は不正データとして扱われます。コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

- 角括弧 ([) で始まり、角括弧 (]) で終わらない行
- 角括弧 ([) で始まらないで、角括弧 (]) で終わる行

クラス名の英字の大文字、小文字は区別されるので、クラス名には DocumentBroker のクラスまたはそのサブクラスとして定義されているとおりの文字列を記述してください。

(b) エントリ

クラス名セクションを構成する ClassId エントリ、PrepresentProperties エントリの記述について説明します。

ClassId エントリ

UUID 変換、操作対象識別子変換をするプロパティのクラス識別子を表す GUID 文字列を、次に示す形式で記述してください。

```
ClassId=<GUID文字列>
```

なお、GUID 文字列は、「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8けた -4けた -4けた -4けた -12けた)」という形式の文字列で、x は 16 進数の 0 ~ 9, a ~ f です。クラス識別子を表す GUID 文字列は、クラス定義情報ファイルに出力されるので、クラス定義情報ファイルを参照して指定することをお勧めします。クラス定義情報ファイルについては、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

PrepresentProperties エントリ

UUID 変換、操作対象識別子変換をするプロパティのプロパティ名を、次に示す形式で記述してください。複数のプロパティ名を指定する場合、コンマ (,) で区切って指定します。プロパティ名は 3 個まで指定できます。4 個以上指定した場合、EDMRptAcllog コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

```
PrepresentProperties=<プロパティ名,プロパティ名,プロパティ名>
```

ここで指定したプロパティの値が UUID 変換、操作対象識別子変換の結果として出力されます。したがって、UUID 変換、操作対象識別子変換をするプロパティとしては、クラスを代表するプロパティを選択することをお勧めします。

プロパティ名の英字の大文字、小文字は区別されるので、プロパティ名には DocumentBroker のクラスまたはそのサブクラスのプロパティとして定義されているとおりの文字列を記述してください。

各エントリは、次に示す文法に従って記述してください。

エントリは「エントリ名 = 値」の形式で、1 行で記述してください。

エントリ名の英字の大文字、小文字は区別されるので、エントリ名には「ClassId」または「PrepresentProperties」と記述してください。

ClassId エントリは、クラス名セクションの次の行に記述してください。クラス名セクションの次の行に ClassId エントリがない場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

PrepresentProperties エントリは、ClassId エントリの次の行に記述してください。ClassId エントリの次の行に PrepresentProperties エントリがない場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

(3) プロパティ拡張情報ファイルの記述例

プロパティ拡張情報ファイルの記述例を次に示します。

```
# property extension information

[usrClass_Folder]
ClassId=01a3a8c2-7aec-11d1-a31b-0020af9fbb1c
PrepresentProperties=usrProp_Name

[usrClass_Document]
ClassId=01a3a8c2-7aec-11d1-a31b-0020af9fbb1d
PrepresentProperties=usrProp_Title,usrProp_Author
```

この例では、DocumentBroker で用意しているクラスのサブクラスとして usrClass_Folder と usrClass_Document というクラスが定義されていると仮定しています。また、usrClass_Folder クラスのプロパティとして usrProp_Name、usrClass_Document クラスのプロパティとして usrProp_Title と usrProp_Author が定義されていると仮定しています。

4.1.3 DocumentSpace 構成定義ファイル

DocumentSpace 構成定義ファイルは、文書空間の構成を定義するためのファイルで、EDMRptAclog コマンドのデータベース (HiRDB) への接続に使用します。

DocumentSpace 構成定義ファイルに、データベースへの接続に必要な情報を指定しておきます。EDMRptAclog コマンドは、データベースに接続する際に、DocumentSpace 構成定義ファイルを参照します。DocumentSpace 構成定義ファイルにデータベースへの接続に必要な情報が指定されていない場合、コマンドの実行時にデータベースに接続できないため、標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

DocumentSpace 構成定義ファイルは、次のディレクトリに次のファイル名で格納されています。

ディレクトリ

- UNIX の場合
DocumentBroker 実行環境ディレクトリ /etc
- Windows の場合
DocumentBroker インストールディレクトリ %Server%etc

ファイル名

docspace.ini

DocumentSpace 構成定義ファイルの [Entry0001] セクションを構成する次の三つのエントリに、データベースへの接続に必要な情報を指定してください。

- PdHost エントリ
データベースを構築しているサーバ (HiRDB サーバ) のホスト名を指定します。
- PdNamePort エントリ
データベースを構築しているサーバ (HiRDB サーバ) のポート番号を指定します。
- PdUser エントリ
データベースに接続するユーザのユーザ名とパスワードを指定します。

ファイル内で同じセクションまたはエントリを重複して指定した場合、最初のセクションまたはエントリが有効になります。

各エントリに指定する内容の詳細については、マニュアル「DocumentBroker Version 3 システム導入・

運用ガイド」を参照してください。

4.1.4 集計キー定義ファイル

集計キー定義ファイルは、EDMRptAclog コマンドの集計機能で、`-k` オプションを指定してキーとなる項目によって集計する場合に、キーとなる項目を指定しておくファイルです。EDMRptAclog コマンドの `-k` オプションで、集計キー定義ファイルのファイル名を指定します。

集計キー定義ファイルは、ユーザが作成する必要があります。

なお、集計キー定義ファイルの内容と抽出エントリ定義ファイルの内容を、一つのファイルにまとめて記述してもかまいません。一つのファイルにまとめて記述する場合の記述規則については、「4.1.5(4) 集計キー定義ファイルと抽出エントリ定義ファイルを一つのファイルに記述する場合の記述規則」を参照してください。

(1) 記述規則

集計キー定義ファイルの記述規則は次のとおりです。

作成先のディレクトリおよびファイル名は任意です。

テキストファイルとして作成してください。

ASCII コードで記述してください。ただし、コメント行では Shift-JIS コードも使用できます。

シャープ (#) およびセミコロン (;) で始まる行はコメント行として扱われます。記述内容を説明する場合や集計キー定義ファイルを修正した場合のコメントなど、必要に応じて記述してください。

先頭や末尾の空白・タブは無視されます。

空白・改行だけの行は無視されます。

ファイルの先頭の行（コメント行を除く）には、セクションを記述してください。

(2) 記述形式

集計キー定義ファイルの記述形式を次に示します。

# Extract Key Definition File	コメント
[Key User]	セクション
<ユーザ名>	エントリ
<ユーザ名>	エントリ
[Key Object]	セクション
<オブジェクト名>	エントリ
<オブジェクト名>	エントリ

集計キー定義ファイルは、[Key User] セクション、[Key Object] セクションと、各セクションに指定するエントリで構成されます。

各セクションとエントリの記述について説明します。

(a) セクション

[Key User] セクション、[Key Object] セクションの記述について説明します。

[Key User] セクション

集計対象のユーザ名を定義する [Key User] セクションを、次に示す形式で記述してください。

4. コマンドの実行に必要なファイル

[Key User]

[Key Object] セクション

集計対象のオブジェクト名を定義する [Key Object] セクションを、次に示す形式で記述してください。

[Key Object]

各セクションは、次に示す文法に従って記述してください。

セクション名は角括弧 ([]) で囲んでください。

なお、次の行は不正データとして扱われます。コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

- 角括弧 ([]) で始まり、角括弧 (]) で終わらない行
- 角括弧 ([]) で始まらないで、角括弧 (]) で終わる行

[Key User] セクション , [Key Object] セクション以外のセクション名は指定しないでください。不正なセクション名を指定した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

[Key User] セクション , [Key Object] セクションは省略できます。ただし , [Key User] セクション , [Key Object] セクションをどちらも省略した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

ファイル内で同じセクションを重複して指定した場合、最初のセクションのエントリが有効になり、2 回目以降のセクションのエントリは不正データとして扱われて無視されます。

(b) エントリ

[Key User] セクションに定義するユーザ名 , [Key Object] セクションに定義するオブジェクト名の記述について説明します。

ユーザ名

集計対象のユーザ名を、次に示す形式で記述してください。

```
[Key User]
<ユーザ名>
<ユーザ名>
```

ユーザ名は、次に示す文法に従って記述してください。

- ユーザ名の指定は省略できます。省略した場合、すべてのユーザが集計対象になります。
- 複数のユーザ名を指定する場合、改行で区切って指定します。指定できるバイト数は 32 バイトまでです。指定できる個数の上限はありません。
- 同じユーザ名を重複して指定した場合、最初のユーザ名が有効になり、2 回目以降のユーザ名は不正データとして扱われて無視されます。

オブジェクト名

集計対象のオブジェクト名を、次に示す形式で記述してください。

```
[Key Object]
<オブジェクト名>
<オブジェクト名>
```

オブジェクト名は、次に示す文法に従って記述してください。

- オブジェクト名の指定は省略できます。省略した場合、すべてのオブジェクトが集計対象になります。
- 複数のオブジェクト名を指定する場合、改行で区切って指定します。指定できるバイト数および個数の上限はありません。
- オブジェクト名をダブルクォーテーション (") で囲んだ場合、ダブルクォーテーション (") は無視されます。例えば、「"Plan"」と指定した場合、オブジェクト名は「Plan」になります。
- 同じオブジェクト名を重複して指定した場合、最初のオブジェクト名が有効になり、2回目以降のオブジェクト名は不正データとして扱われて無視されます。
- プロパティ拡張情報ファイルの PrepresentProperties エントリに複数の値を指定している場合、オブジェクト名として PrepresentProperties エントリの値の一部を指定できます。指定例を次に示します。

指定例

例えば、データベースとプロパティ拡張情報ファイルが次のように設定されていると仮定します。

データベース	プロパティ拡張情報ファイル								
<table border="1"> <thead> <tr> <th>OID</th> <th>column1</th> <th>column2</th> <th>column3</th> </tr> </thead> <tbody> <tr> <td>...</td> <td>Design</td> <td>Plan</td> <td>Specs</td> </tr> </tbody> </table>	OID	column1	column2	column3	...	Design	Plan	Specs	PrepresentProperty=column1, column2, column3
OID	column1	column2	column3						
...	Design	Plan	Specs						

この条件で、次のように集計キー定義ファイルを記述してコマンドを実行します。

```
[Key Object]
Plan
```

このとき、実行結果には次のよう出力されます。

user	object	access	times
ACUser1	Design. Plan. Specs. /TableName	create	1
		delete	1

(3) 集計キー定義ファイルの記述例

集計キー定義ファイルの記述例を次に示します。

すべてのユーザを集計対象とする場合の例

```
[Key User]
```

この例では、集計結果に出力される項目は、user, access, times です。

ユーザ「ACUser1」のすべてのオブジェクトを集計対象とする場合の例

```
[Key User]
ACUser1
[Key Object]
```

この例では、集計結果に出力される項目は、user, object, access, times です。

ユーザ「ACUser1」と「ACUser2」のオブジェクト「Object1」と「Object2」を集計対象とする場合の例

```
[Key User]
ACUser1
ACUser2
[Key Object]
Object1
Object2
```

4. コマンドの実行に必要なファイル

この例では、集計結果に出力される項目は、user、object、access、times です。

4.1.5 抽出エントリ定義ファイル

抽出エントリ定義ファイルは、EDMRptAclog コマンド、EDMRptTrace コマンドのエントリ抽出機能で、-x オプションを指定してエントリを抽出する場合に、抽出するエントリのメソッド名、引数および処理の種別を指定しておくファイルです。EDMRptAclog コマンド、EDMRptTrace コマンドの -x オプションで、抽出エントリ定義ファイルのファイル名を指定します。

抽出エントリ定義ファイルは、ユーザが作成する必要があります。

なお、集計キー定義ファイルの内容と抽出エントリ定義ファイルの内容を、一つのファイルにまとめて記述してもかまいません。一つのファイルにまとめて記述する場合の記述規則については、「4.1.5(4) 集計キー定義ファイルと抽出エントリ定義ファイルを一つのファイルに記述する場合の記述規則」を参照してください。

(1) 記述規則

抽出エントリ定義ファイルの記述規則は次のとおりです。

作成先のディレクトリおよびファイル名は任意です。

テキストファイルとして作成してください。

ASCII コードで記述してください。ただし、コメント行では Shift-JIS コードも使用できます。

シャープ (#) およびセミコロン (;) で始まる行はコメント行として扱われます。記述内容を説明する場合や抽出エントリ定義ファイルを修正した場合のコメントなど、必要に応じて記述してください。

先頭や末尾の空白・タブは無視されます。

空白・改行だけの行は無視されます。

ファイルの先頭に行(コメント行を除く)には、セクションを記述してください。

(2) 記述形式

抽出エントリ定義ファイルの記述形式を次に示します。

# Extract Entry Definition File	コメント
[Extract Aclog]	セクション
<メソッド名 引数>	エントリ
<メソッド名 引数>	エントリ
[Extract Trace]	セクション
<処理の種別>	エントリ
<処理の種別>	エントリ

抽出エントリ定義ファイルは、[Extract Aclog] セクション、[Extract Trace] セクションと、各セクションに指定するエントリで構成されます。

各セクションとエントリの記述について説明します。

(a) セクション

[Extract Aclog] セクション、[Extract Trace] セクションの記述について説明します。

[Extract Aclog] セクション

[Extract Aclog] セクションには、EDMRptAclog コマンドのエントリ抽出機能で抽出するエントリ

のメソッド名と引数を指定します。[Extract Aclog] セクションは、次に示す形式で記述してください。

```
[Extract Aclog]
```

[Extract Trace] セクション

[Extract Trace] セクションには、EDMRptTrace コマンドのエントリ抽出機能で抽出するエントリの処理の種別を指定します。[Extract Trace] セクションは、次に示す形式で記述してください。

```
[Extract Trace]
```

各セクションは、次に示す文法に従って記述してください。

セクション名は角括弧 ([]) で囲んでください。

なお、次の行は不正データとして扱われます。コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

- 角括弧 ([]) で始まり、角括弧 (]) で終わらない行
- 角括弧 ([]) で始まらないで、角括弧 (]) で終わる行

[Extract Aclog] セクション , [Extract Trace] セクション以外のセクション名は指定しないでください。不正なセクション名を指定した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

[Extract Aclog] セクション , [Extract Trace] セクションは省略できます。ただし、[Extract Aclog] セクション , [Extract Trace] セクションをどちらも省略した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

ファイル内で同じセクションを重複して指定した場合、最初のセクションのエントリが有効になり、2 回目以降のセクションのエントリは不正データとして扱われて無視されます。

セクション名だけを指定してエントリを指定しなかった場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

(b) エントリ

[Extract Aclog] セクションに指定するメソッド名と引数 , [Extract Trace] セクションに指定する処理の種別の記述について説明します。

メソッド名と引数

EDMRptAclog コマンドのエントリ抽出機能で抽出するエントリのメソッド名と引数を、次に示す形式で記述してください。

```
[Extract Aclog]
<メソッド名 引数>
<メソッド名 引数>
```

指定できるメソッド名と引数については、「表 4-1 集計機能で分類されるアクセスの内容と集計項目 (アクセス種別) の関係」を参照してください。不正なメソッド名と引数を指定した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

メソッド名と引数は、次に示す文法に従って記述してください。

- メソッド名と引数の間は、半角空白で区切ります。
- 複数のメソッド名と引数を指定する場合、改行で区切って指定します。
- 同じメソッド名と引数を重複して指定した場合、最初のメソッド名と引数が有効になり、2 回目以

4. コマンドの実行に必要なファイル

降のメソッド名と引数は不正データとして扱われて無視されます。

処理の種別

EDMRptTrace コマンドのエントリ抽出機能で抽出するエントリの処理の種別を、次に示す形式で記述してください。

```
[Extract Trace]
<処理の種別>
<処理の種別>
```

指定できる処理の種別を次に示します。

- enter
- arg
- call
- return
- error
- warn
- mark
- msg
- exit

処理の種別は、次に示す文法に従って記述してください。

- 複数の処理の種別を指定する場合、改行で区切って指定します。
- 同じ処理の種別を重複して指定した場合、最初の処理の種別が有効になり、2 回目以降の処理の種別は不正データとして扱われて無視されます。

(3) 抽出エントリ定義ファイルの記述例

抽出エントリ定義ファイルの記述例を次に示します。

```
# Extract Entry Definition File

[Extract Aclog]
CREATE OBJECT
GET RESULT

[Extract Trace]
enter
msg
```

この例では、EDMRptAclog コマンドのエントリ抽出機能で、「CREATE OBJECT」と「GET RESULT」のエントリが抽出されます。また、EDMRptTrace コマンドのエントリ抽出機能で、「enter」と「msg」のエントリが抽出されます。

(4) 集計キー定義ファイルと抽出エントリ定義ファイルを一つのファイルに記述する場合の記述規則

集計キー定義ファイルと抽出エントリ定義ファイルを一つのファイルに記述する場合、集計キー定義ファイルおよび抽出エントリ定義ファイルの記述規則に加えて、次に示す記述規則に従ってください。

作成先のディレクトリおよびファイル名は任意です。

ファイルに指定できるセクション名は次のとおりです。

- [Key User]
- [Key Object]
- [Extract Aclog]

- [Extract Trace]

セクション名の指定順序に規則はありません。どのような順序でも指定できます。

すべてのセクションを省略した場合または不正なセクション名を指定した場合、コマンド実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

ファイル内で同じセクションまたはエントリを重複して指定した場合、最初のセクションまたはエントリが有効になります。

コマンド実行時にどのオプションを指定したかによってチェック対象のセクションが異なります。

- k オプションを指定した場合

[Key User] セクションおよび [Key Object] セクションのエントリはチェック対象になります。

[Extract Aclog] セクションおよび [Extract Trace] セクションのエントリはチェック対象になりません。

- x オプションを指定した場合

[Extract Aclog] セクションおよび [Extract Trace] セクションのエントリはチェック対象になります。

[Key User] セクションおよび [Key Object] セクションのエントリはチェック対象になりません。

なお、-k オプションまたは x オプションで指定したファイルがない場合、コマンド実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

4.1.6 動作環境定義ファイル

動作環境定義ファイルは、統計解析ツールが動作する時に参照する情報を定義するファイルです。動作環境定義ファイルには、次に示す情報が定義されています。

ユーザ ID の集計方法に関する情報

アクセスログファイルを解析するときに、アクセスログ中のユーザ ID の大文字、小文字を区別して集計するかどうかを指定します。

OIID 変換、操作対象識別子変換の実行に関する情報

EDMRptAclog コマンドの実行結果として、アクセス対象オブジェクトの情報を出力するときに、アクセスログに出力された OIID、操作対象識別子を、プロパティの値 (文書名など) に変換するかどうかを指定します。操作対象識別子を変換する場合は、対象となる操作対象識別子の種類を指定します。なお、集計キーを指定して EDMRptAclog コマンドを実行した場合、このファイルで指定した変換対象を含むアクセスログエントリが集計対象になります。

動作環境定義ファイルは、インストール時に、次のディレクトリに次のファイル名で格納されます。なお、必要に応じてユーザが編集できます。

ディレクトリ

- UNIX の場合
DocumentBroker の実行環境ディレクトリ /etc
- Windows の場合
DocumentBroker インストールディレクトリ %Server%etc

ファイル名

edmrpt.ini

(1) 記述規則

動作環境定義ファイルの記述規則は次のとおりです。

4. コマンドの実行に必要なファイル

テキストファイルとして作成してください。

ASCII コードで記述してください。ただし、コメント行では Shift-JIS コードも使用できます。

シャープ (#) およびセミコロン (;) で始まる行はコメント行として扱われます。記述内容を説明する場合や抽出エントリ定義ファイルを修正した場合のコメントなど、必要に応じて記述してください。

先頭や末尾の空白・タブは無視されます。

空白・改行だけの行は無視されます。

ファイルの先頭の行 (コメント行を除く) には、セクションを記述してください。

(2) 記述形式

動作環境定義ファイルの記述形式を次に示します。

[UserCountingMode] <ユーザ名集計モード>	セクション エントリ
[ObjectConvertingMode] <オブジェクト変換モード> <オブジェクト変換モード>	セクション エントリ エントリ

動作環境定義ファイルは、[UserCountingMode] セクション、[ObjectConvertingMode] セクションと、各セクションに指定するエントリで構成されます。

各セクションとエントリの記述について説明します。

(a) セクション

[UserCountingMode] セクション、[ObjectConvertingMode] セクションの記述について説明します。

[UserCountingMode] セクション

[UserCountingMode] セクションには、ユーザ ID の集計方法に関する情報を指定します。

[UserCountingMode] セクションは、次に示す形式で記述してください。

```
[UserCountingMode]
```

[ObjectConvertingMode] セクション

[ObjectConvertingMode] セクションは、OID 変換、操作対象識別子変換の実行に関する情報を指定します。[ObjectConvertingMode] セクションは、次に示す形式で記述してください。

```
[ObjectConvertingMode]
```

各セクションは、次に示す文法に従って記述してください。

セクション名は角括弧 ([]) で囲んでください。

なお、次の行は不正データとして扱われます。コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

- 角括弧 ([) で始まり、角括弧 (] で終わらない行
- 角括弧 ([) で始まらないで、角括弧 (] で終わる行

[UserCountingMode] セクション、[ObjectConvertingMode] セクション以外のセクション名は指定しないでください。不正なセクション名を指定した場合、コマンドの実行時に標準エラー出力 (stderr) にメッセージが出力されて、処理が中止されます。

ファイル内で同じセクションを重複して指定した場合、最初のセクションのエントリが有効になり、2

回目以降のセクションのエントリは不正データとして扱われて無視されます。

(b) エントリ

[UserCountingMode] セクションに指定するユーザ名集計モード, [ObjectConvertingMode] セクションに指定するオブジェクト変換モードの記述について説明します。

ユーザ名集計モード

アクセスログファイルを解析するときに, アクセスログに含まれるユーザ ID の大文字, 小文字を区別するかどうかを次に示す形式で記述してください。

```
[UserCountingMode]
ユーザ名集計モード
```

指定できるユーザ名集計モードと集計方法を次の表に示します。

表 4-2 ユーザ名集計モードと集計方法

ユーザ名集計モード	集計方法
0	大文字, 小文字を区別する。
1	大文字, 小文字を区別しない。

なお, デフォルトでは, 「0」が指定されています。

オブジェクト変換モード

EDMRptAclog コマンドの実行結果として, アクセス対象オブジェクトの情報を出力するときに, OIID 変換, 操作対象識別子変換を実行するかどうかを次に示す形式で指定してください。なお, 操作対象識別子変換を実行する場合は, 対象となる操作対象識別子の種類を指定してください。

```
[ObjectConvertingMode]
オブジェクト変換モード
オブジェクト変換モード
```

指定できるオブジェクト変換モードと変換方法を次の表に示します。

表 4-3 オブジェクト変換モードと変換方法

オブジェクト変換モード	変換方法
OIID	OIID 変換を実行する。
VD	バージョン識別子に対して, 操作対象識別子変換を実行する。
RL	リレーション識別子に対して, 操作対象識別子変換を実行する。
LD	直接型のリンク識別子に対して, 操作対象識別子変換を実行する。
LR	参照型のリンク識別子に対して, 操作対象識別子変換を実行する。
LV	構成管理型のリンク識別子に対して, 操作対象識別子変換を実行する。

なお, デフォルトでは「OIID」が指定されています。

変換対象は, 次に示す文法に従って記述してください。

- 複数の変換対象を指定する場合, 改行で区切って指定します。
- 同じ変換対象を重複して指定した場合, 最初の変換対象が有効になり, 2 回目以降の変換対象は無視されます。

4. コマンドの実行に必要なファイル

(3) 記述例

動作環境定義ファイルの記述例を次に示します。

```
[UserCountingMode]
1

[ObjectConvertingMode]
OIID
VD
```

この例では、アクセスログに含まれるユーザ ID の大文字、小文字を区別しないで集計が実行されます。また、EDMRptAclog コマンドの実行結果にオブジェクト名を出力するとき、OIID 変換が実行され、バージョン識別子に対して操作対象識別子変換が実行されます。なお、集計キーを指定して EDMRptAclog コマンドを実行した場合、OIID またはバージョン識別子を含むアクセスログエントリが集計対象になります。

4.2 EDMRptTrace コマンドに必要なファイル

この節では、EDMRptTrace コマンドの実行に必要なファイルについて説明します。EDMRptTrace コマンドの実行に必要なファイルを次に示します。

- トレースファイル
- 抽出エントリ定義ファイル

4.2.1 トレースファイル

トレースファイルは、DocumentBroker サーバおよびクライアントのプログラムの処理内容の履歴（トレース）を記録したファイルです。EDMRptTrace コマンドは、トレースファイルに記録された内容を入力として、オプションで指定された処理を実行します。なお、トレース出力機能の詳細については、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

トレースファイルは、EDMRptTrace コマンドの引数で指定できます。指定を省略する場合は、EDMRptTrace コマンドのトレースファイル格納ディレクトリにあるファイルがコマンドの対象になります。

EDMRptTrace コマンドは、DocumentBroker がトレースファイルを生成するディレクトリにあるファイルを対象にして実行できません。このため、EDMRptTrace コマンドを実行する前に、DocumentBroker が生成したトレースファイルを、EDMRptTrace コマンドのトレースファイル格納ディレクトリにコピーして、コマンド引数でのトレースファイルの指定を省略することをお勧めします。

(1) ファイルのコピー

トレースファイルのコピー手順は、ご使用の OS ごとに異なります。

(a) UNIX の場合

サーバトレース

DocumentBroker は「DocumentBroker 実行環境ディレクトリ /spool/server」に「EDMRasTracePID"_NO".log」という名称でサーバトレースを生成します。ファイル名の "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。ただし、環境変数「_HIEDMS_TRACE_DIR」が設定されている場合は、「_HIEDMS_TRACE_DIR」に設定されたディレクトリに生成します。このサーバトレースを、EDMRptTrace コマンドのサーバトレース格納ディレクトリ「DocumentBroker 実行環境ディレクトリ /aru/input/spool/server」にコピーしてください。cp コマンドを使用したコピー手順の例を次に示します。なお、\$DOCBROKERDIR は、DocumentBroker 実行環境ディレクトリです。

```
cp $DOCBROKERDIR/spool/server/* /$DOCBROKERDIR/aru/input/spool/server
```

クライアントトレース

DocumentBroker は「DocumentBroker 実行環境ディレクトリ /spool/client」に「EDMRasTraceCL"PID"_NO".log」という名称でクライアントトレースを生成します。ファイル名の "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。ただし、環境変数「_HIEDMS_TRACE_DIR」が設定されている場合は、「_HIEDMS_TRACE_DIR」に設定されたディレクトリに生成します。このクライアントトレースを、EDMRptTrace コマンドのクライアントトレース格納ディレクトリ「DocumentBroker 実行環境ディレクトリ /aru/input/spool/client」にコピーしてください。cp コマンドを使用したコピー手順の例を次に示します。なお、\$DOCBROKERDIR は、

4. コマンドの実行に必要なファイル

DocumentBroker 実行環境ディレクトリです。

```
cp $DOCBROKERDIR/spool/client/* /$DOCBROKERDIR/aru/input/spool/client
```

(b) Windows の場合

サーバトレース

DocumentBroker は「DocumentBroker インストールディレクトリ %Server%spool%server」に「EDMRasTrace"PID"_"NO".log」という名称でサーバトレースを生成します。ファイル名の "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。ただし、環境変数「_HIEDMS_TRACE_DIR」が設定されている場合は、「_HIEDMS_TRACE_DIR」に設定されたディレクトリに生成します。

このサーバトレースを、EDMRptTrace コマンドのサーバトレース格納ディレクトリ

「DocumentBroker インストールディレクトリ %Server%aru%input%spool%server」にコピーしてください。

クライアントトレース

DocumentBroker は「DocumentBroker インストールディレクトリ %Server%spool%client」に

「EDMRasTraceCL"PID"_"NO".log」という名称でクライアントトレースを生成します。ファイル名の "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。ただし、環境変数

「_HIEDMS_TRACE_DIR」が設定されている場合は、「_HIEDMS_TRACE_DIR」に設定されたディレクトリに生成します。

このクライアントトレースを、EDMRptTrace コマンドのクライアントトレース格納ディレクトリ

「DocumentBroker インストールディレクトリ %Server%aru%input%spool%client」にコピーしてください。

(2) 出力形式

トレースログの出力形式およびトレースログの項目について説明します。

(a) トレースログの出力形式

トレースログの出力形式について説明します。

トレースファイル中のトレースログは、サーバまたはクライアントのプログラムの処理内容を示す複数の項目で構成されます。トレースログの出力形式を次に示します。

```
nnnn yyyy/mm/dd hh:mm:ss.sss system_name pid tid message-id message
```

- トレースファイルは、ASCII コードと Shift-JIS コードで構成されます。ただし、enter エントリおよび msg エントリは ASCII コードで構成されます。
- エントリは改行で区切られます。

なお、EDMRptTrace コマンドは、nnnn yyyy/mm/dd hh:mm:ss.sss で始まらないエントリを無視します。

(b) トレースログの項目

トレースログの各項目について説明します。

```
nnnn
```

行番号が出力されます。

```
yyyy/mm/dd
```

サーバまたはクライアントのプログラムが処理を実行した日付（年 / 月 / 日）が出力されます。

hh:mm:ss.sss

サーバまたはクライアントのプログラムが処理を実行した時刻（時：分：秒.ミリ秒）が出力されます。

system_name

システム名が出力されます。システム名は「DocumentBroker」固定です。

pid

プロセス ID が出力されます。なお、サーバトレースかクライアントトレースかによって、次に示す違いがあります。

サーバトレースの場合

サーバプロセスのプロセス ID

クライアントトレースの場合

ユーザプログラムのプロセス ID

tid

スレッド ID が出力されます。なお、サーバトレースかクライアントトレースかによって、次に示す違いがあります。

サーバトレースの場合

サーバプロセスのスレッド ID

クライアントトレースの場合

ユーザプログラムのスレッド ID

message-id

メッセージ ID が出力されます。ほとんどの場合、11 個の半角空白が出力されます。

message

サーバまたはクライアントのプログラムの処理内容として、実行された処理の種別と処理の内容が出力されます。処理の種別と処理の内容の間は半角空白で区切られます。EDMRptTrace コマンドは、処理の種別を基にトレースログを分類して集計します。

message に出力される処理の種別と処理の内容を次の表に示します。

表 4-4 message に出力される処理の種別と処理の内容

処理の種別	処理の内容
enter	<p>実行された処理の内容が次に示す形式で出力されます。</p> <p>"クラス名::メソッド名(または関数名)" "ユーザプログラム情報"</p> <p>ユーザプログラム情報は、次に示す形式でサーバトレースだけに出力される情報です。</p> <p>"ユーザプログラムのホスト名" "ユーザプログラムのプログラム名" "ユーザプログラムのプロセスID" "ユーザプログラムのセッションID" "プログラム識別子"</p>
msg	<p>メッセージ (SQL 文などの保守情報) が出力されます。</p>

注

処理の種別には、exit や arg など、表 4-4 で説明していない種別もありますが、EDMRptTrace コマンドの集計対象ではないので説明を省略しています。なお、個々のトレースログを区別する場合、処理の種別を使用して「enter エントリ」、「msg エントリ」と表記することがあります。

4. コマンドの実行に必要なファイル

注意事項

- プログラム識別子がない enter エントリもあります。
- サーバトレースの enter エントリのうち、ユーザプログラム情報が出力されない enter エントリは、その前に同じスレッド ID の enter エントリがあるかないかによって、次に示すように処理されません。

同じスレッド ID の enter エントリが前にある場合

前にある同じスレッド ID の enter エントリのユーザプログラム情報（ユーザプログラムのプログラム名、ユーザプログラムのプロセス ID）を持つものとして処理されます。

同じスレッド ID の enter エントリが前にない場合

集計されません。

なお、ユーザプログラム情報がなく enter エントリもトレース表には出力されません。トレース表については、「6.3.2(1) 集計機能の処理方式 (EDMRptTrace コマンドの集計方法)」を参照してください。

(3) トレースファイルの出力例

トレースファイルの出力例を次に示します。

サーバトレースの出力例

```
**** Microsoft WindowsNT5.2 (Build:3790) Service Pack 2          TEST-SERV          TZ=(local)-9:00
2009/02/08 13:43:42.562
      yyyy/mm/dd hh:mm:ss.sss          pid          tid          message-id
message (LANG=0x0411)
2063 2009/02/08 13:43:42.562          DocumentBroker  00005282 0000179B          enter
CdbrDMA::GetPropertyValuesAndLock() sis01 WETG2J0 12491 000052823906310000040a0d(userdoc)
EDMS000021122 ()
2064 2009/02/08 13:43:42.562          DocumentBroker  00005282 0000179B          msg
SELECT
P9FFOR71000FC9972P521A2HD56,P9FFOR71000GC99727V7140CF56,P9FFOR71000FC9972VM01ESQF56,P9FFOR71
000GC9972E061CD0LM5 FROM T9FFOR71000FC9972O521IUPU96 WHERE
PI29HEI0K08PB0872DLUUMRGB4='000000000ABC42239'
2065 2009/02/08 13:43:42.562          DocumentBroker  00005282 0000179B          exit
CdbrDMA::GetPropertyValuesAndLock() sis01 WETG2J0 12491 000052823906310000040a0d(userdoc)
EDMS000021122 ()
```

下線で示したトレースログでは、ユーザプログラム情報は次のようになります。

- ユーザプログラムのホスト名：sis01
- ユーザプログラムのプログラム名：WETG2J0
- ユーザプログラムのプロセス ID：12491
- ユーザプログラムのセッション ID：000052823906310000040a0d(userdoc)
- プログラム識別子：EDMS000021122

クライアントトレースの出力例


```

**** Microsoft WindowsNT5.2(Build:3790) Service Pack 2          TEST-CL          TZ=(local)-9:00
2009/02/08 13:43:42.562
      yyyy/mm/dd hh:mm:ss.sss          pid          tid          message-id
message (LANG=0x0411)
0000 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          enter
CedmRasMsg::cs_setMsgFormatter()
0001 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          exit
CedmRasMsg::cs_setMsgFormatter()
0002 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          enter
CedmCmdClassDescriptionForMulti::Initialize()
0003 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
EDMCLASSDEFPATH "NULL"
0004 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          exit
CedmCmdClassDescriptionForMulti::Initialize()
0005 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          enter
dbrgetprop::main()
0006 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
argv[ 0 ] "dbrgetprop"
0007 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
argv[ 1 ] "D:¥test¥objtool¥file¥tmp¥ipoid"
0008 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
argv[ 2 ] "D:¥test¥objtool¥file¥ipgetprop"
0009 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          enter
CedmOotTools::initialize()
0010 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
DBRTOOLS DIR "D:¥test¥objtool"
0011 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          enter
CedmCmdClassDescription::InitializeForCache()
0012 2009/02/08 13:43:42.562          DocumentBroker 00006AF8 00000001          arg
pClassDescriptionPath "D:¥test¥objtool¥etc¥673d2be0-d1fd-11d0-ab59-08002be29e1d.

```

4.2.2 抽出エントリ定義ファイル

抽出エントリ定義ファイルは、EDMRptAclog コマンド、EDMRptTrace コマンドのエントリ抽出機能で、`-x` オプションを指定してエントリを抽出する場合に、抽出するエントリのメソッド名、引数および処理の種類を指定しておくファイルです。EDMRptAclog コマンド、EDMRptTrace コマンドの `-x` オプションで、抽出エントリ定義ファイルのファイル名を指定します。

抽出エントリ定義ファイルは、ユーザが作成する必要があります。

抽出エントリ定義ファイルの作成については、「4.1.5 抽出エントリ定義ファイル」を参照してください。

5

トラブルシューティング機能

この章では、EDMRptAclog コマンド、EDMRptTrace コマンドを運用するためのトラブルシューティング機能（コマンドの実行によって出力されるメッセージやエラー処理）について説明します。

5.1 EDMRptAclog コマンドのトラブルシューティング機能

5.2 EDMRptTrace コマンドのトラブルシューティング機能

5.1 EDMRptAclog コマンドのトラブルシュート機能

この節では、EDMRptAclog コマンドのメッセージおよび OIID 変換，操作対象識別子変換時のエラー処理について説明します。

5.1.1 メッセージ

EDMRptAclog コマンドのメッセージの出力先およびメッセージの言語種別について説明します。

(1) メッセージの出力先

EDMRptAclog コマンドのメッセージの出力先を次に示します。

標準エラー出力 (stderr)

syslog ファイル (UNIX の場合) またはイベントログ (Windows の場合)

メッセージトレース

出力されるメッセージについては、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。なお、データベース (HiRDB) への接続でエラーが発生した場合、出力されるメッセージの中に HiRDB からのリターンコードと HiRDB のメッセージ (SQLERRMC) が含まれていますので、必要に応じてマニュアル「HiRDB メッセージ」を参照してください。

メッセージトレースについて

環境変数「EDMARUMSG」を設定している場合、「EDMARUMSG」で設定したディレクトリに「aclog_msg.log」という名称で EDMRptAclog コマンドのメッセージトレースが出力されます。

ファイル名の規則を次に示します。

- すでに同名のファイルがある場合、そのファイル名に "_n" (n は通し番号。最大値は 179) を付加した形式で出力されます。
- "_n" を付加した形式のファイルがある場合、最終更新ファイルの n に 1 を足したファイルが出力または更新されます。なお、n が 179 のファイルが最終更新ファイルの場合、"_n" が付いてないファイルが更新されます。

例えば、n が 5 のファイルが最終更新ファイルの場合、n が 6 のファイルが更新されます。また、n が 9 のファイルしかない場合、n が 10 のファイルが出力されます。

ファイル名の例を次に示します。

例 1：同名のファイルがない場合

「aclog_msg.log」というファイルが出力されます。

例 2：同名のファイル「aclog_msg.log」がある場合

「aclog_msg_1.log」というファイルが出力されます。

例 3：ファイル「aclog_msg_5.log」が最終更新ファイルの場合

「aclog_msg_6.log」というファイルが出力されます。

例 4：ファイル「aclog_msg_9.log」しかない場合

「aclog_msg_10.log」というファイルが出力されます。

5.1.2 OIID 変換，操作対象識別子変換時のエラー処理

OIID 変換，操作対象識別子変換でエラーが発生した場合に考えられるエラーの要因と統計解析ツールの

処理内容を次の表に示します。

表 5-1 OIID 変換，操作対象識別子変換時のエラーの要因と統計解析ツールの処理内容

要因	処理内容
プロパティ拡張情報ファイルがない	標準出力 (stdout) に処理結果として OIID の末尾 16 バイトを出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
プロパティ拡張情報ファイルの形式が間違っている	標準エラー出力 (stderr) にメッセージを出力して、処理を中止します。
プロパティ拡張情報ファイルに変換対象のプロパティのクラス識別子が指定されていない	標準出力 (stdout) に処理結果として OIID の末尾 16 バイトを出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
プロパティ拡張情報ファイルに指定されている変換対象のプロパティがデータベース (HiRDB) で見つからない	標準出力 (stdout) に処理結果として「(deleted)」を出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
データベース (HiRDB) への接続，検索に失敗した	標準出力 (stdout) に処理結果として OIID の末尾 16 バイトを出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
プロパティ拡張情報ファイルに指定されている変換対象のプロパティの値のデータ型が MVARCHAR, INTEGER または CHAR 以外である	標準エラー出力 (stderr) にメッセージを出力して、処理を中止します。
DocumentSpace 構成定義ファイルの PdUser エントリに指定されているユーザにデータベース (HiRDB) の表に対するアクセス権がない	標準出力 (stdout) に処理結果として「(deleted)」を出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
OIID の末尾 16 バイトがデータベース (HiRDB) で見つからない	標準出力 (stdout) に処理結果として「(deleted)」を出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
プロパティ拡張情報ファイルに指定されている変換対象のプロパティを定義しているクラスがデータベース (HiRDB) で見つからない	標準出力 (stdout) に処理結果として「(deleted)」を出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。
操作対象識別子変換時，カラムの値が見つからない	標準出力 (stdout) に処理結果として「(deleted)」を出力して、標準エラー出力 (stderr) にメッセージを出力します。処理は継続します。

5.2 EDMRptTrace コマンドのトラブルシュート機能

この節では、EDMRptTrace コマンドのメッセージについて説明します。

5.2.1 メッセージ

EDMRptTrace コマンドのメッセージの出力先およびメッセージの言語種別について説明します。

(1) メッセージの出力先

EDMRptTrace コマンドのメッセージの出力先を次に示します。

標準エラー出力 (stderr)

syslog ファイル (UNIX の場合) またはイベントログ (Windows の場合)

メッセージトレース

出力されるメッセージについては、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

メッセージトレースについて

環境変数「EDMARUMSG」を設定している場合、「EDMARUMSG」で設定したディレクトリに「trace_msg.log」という名称で EDMRptTrace コマンドのメッセージトレースが出力されます。ファイル名の規則を次に示します。

- すでに同名のファイルがある場合、そのファイル名に "_n" (n は通し番号。最大値は 179) を付加した形式で出力されます。
- "_n" を付加した形式のファイルがある場合、最終更新ファイルの n に 1 を足したファイルが出力または更新されます。なお、n が 179 のファイルが最終更新ファイルの場合、"_n" が付いてないファイルが更新されます。

例えば、n が 5 のファイルが最終更新ファイルの場合、n が 6 のファイルが更新されます。また、n が 9 のファイルしかない場合、n が 10 のファイルが出力されます。

ファイル名の例を次に示します。

例 1：同名のファイルがない場合

「trace_msg.log」というファイルが出力されます。

例 2：同名のファイル「trace_msg.log」がある場合

「trace_msg_1.log」というファイルが出力されます。

例 3：ファイル「trace_msg_5.log」が最終更新ファイルの場合

「trace_msg_6.log」というファイルが出力されます。

例 4：ファイル「trace_msg_9.log」しかない場合

「trace_msg_10.log」というファイルが出力されます。

6

コマンドリファレンス

この章では、EDMRptAclog コマンド、EDMRptTrace コマンドの文法、機能の処理方式、実行結果の出力形式および出力例について説明します。

6.1 コマンドの記述形式

6.2 EDMRptAclog コマンドの詳細

6.3 EDMRptTrace コマンドの詳細

6.1 コマンドの記述形式

この節では、統計解析ツールで使用するコマンドの入力形式、使用方法および注意事項について説明します。

(1) 入力形式

コマンドの入力形式を次に示します。

コマンド名	オプション	コマンド引数
-------	-------	--------

コマンドの要素について示します。

コマンド名

コマンド名は実行するコマンドのファイル名です。

オプション

オプションの入力形式の規則を次に示します。なお、説明文で使用する「\$」は、UNIX の場合はシェルプロンプト、Windows の場合はコマンドプロンプトを表します。「cmd」はコマンド名を表します。

オプションの入力形式

オプションはマイナス記号 (-) で始まる文字列で、コマンド行には一つまたは複数のオプションを指定できます。オプションの形式は、次に示すように引数を取らないかまたは 1 個の引数を取ります。

形式1: -オプションフラグ

形式2: -オプションフラグ <空白またはタブ> フラグ引数

- オプションフラグ
1 文字または 2 文字の英字です。大文字と小文字は区別されます。
- フラグ引数
オプションフラグに対する引数です。

オプションの指定規則

- フラグ引数を指定しないオプションフラグを、一つのマイナス記号 (-) のあとにまとめて指定できません。

誤った指定例: \$ cmd -abc

正しい指定例: \$ cmd -a -b -c

- フラグ引数を必要とするオプションフラグのフラグ引数は省略できません。オプションフラグとフラグ引数の間には空白またはタブが必要です。

誤った指定例: \$ cmd -afile

正しい指定例: \$ cmd -a file

- 同じオプションフラグを 2 回以上指定できません。

誤った指定例: \$ cmd -a 1 -a 2

- マイナス記号 (-) だけのオプションは入力できません。次に示す例では「-」はコマンド引数とみなされます。

誤った指定例: \$ cmd -

コマンド引数

コマンド引数は、コマンド操作の直接の対象となるものを指定します。指定する内容は、コマンドによって異なります。詳細は各コマンドの説明を参照してください。

(2) 入出力

入力

入力は、すべてコマンドのオプション、引数の並びです。

出力

コマンド処理が正常に終了したときの出力は、標準出力 (stdout) または o オプションで指定したファイルに対して実行します。また、終了コードの一覧を次の表に示します。

表 6-1 終了コード一覧

終了コード	意味
0	正常終了
1	警告付き正常終了
2	引数エラー
それ以外	そのほかのエラー

6.2 EDMRptAclog コマンドの詳細

この節では、EDMRptAclog コマンドの文法、機能の処理方式、実行結果の出力形式および出力例について説明します。

6.2.1 EDMRptAclog コマンド (アクセスログの解析)

(1) 機能

EDMRptAclog コマンドは、指定されたアクセスログファイルを解析し、指定された項目でアクセス回数を分類・集計して結果を出力します。また、結果を出力する際、アクセスログ中の OIID、操作対象識別子をオブジェクトのプロパティの値に変換します。

(2) 形式

```
EDMRptAclog
  [-k 集計キー定義ファイル名]
  [-s '開始時刻']
  [-e '終了時刻']
  [-d 集計間隔]
  [-c]
  [-t]
  [-v]
  [-vt]
  [-o 出力ファイル名]
  [-x 抽出エントリ定義ファイル名]
  [アクセスログファイル名 ...]
  [ディレクトリ名 ...]
```

(3) オプション

-k 集計キー定義ファイル名

キーとなる項目によってアクセス回数を集計する場合に、キーとなる項目を指定した集計キー定義ファイルのファイル名を指定します。集計キー定義ファイルについては、「4.1.4 集計キー定義ファイル」を参照してください。

このオプションは、次に示す事項に留意して指定してください。

- ファイルパスは、フルパスまたは相対パスのどちらでも指定できます。フルパスで指定されていない場合は、カレントディレクトリ下を探します。
- ハイフン (-) で始まるファイル名は指定できません。ただし、フルパスで指定された場合は処理できます。
- ファイル名に空白が含まれている場合、次に示す記号でファイル名全体を囲めば指定できます。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- 指定されたファイルがない場合や指定されたファイルがディレクトリの場合、エラーになります。

-s '開始時刻'

アクセス回数を集計する期間の開始時刻を指定します。指定した時刻以降のアクセスログエントリが集計対象になります。指定した時刻は、集計期間の対象になります。例えば、12:00:00.000 と指定した場合、12:00:00.000 からが集計期間になります。

開始時刻は次に示す形式で指定してください。

```
'yyyy/mm/dd hh:mm [:ss.sss]'
```

```
yyyy
```

日付の年の値を指定します。

mm

日付の月の値を指定します。指定できる値の範囲は、1月を1として01～12です。範囲外の値を指定するとエラーになります。

dd

日付の日にちの値を指定します。指定できる値の範囲は01～31です。範囲外の値を指定するとエラーになります。

hh

時間の時の単位の値を指定します。指定できる値の範囲は00～23です。範囲外の値を指定するとエラーになります。

mm

時間の分の単位の値を指定します。指定できる値の範囲は00～59です。範囲外の値を指定するとエラーになります。

ss

時間の秒の単位の値を指定します。指定できる値の範囲は00～59です。範囲外の値を指定するとエラーになります。

sss

時間のミリ秒の単位の値を指定します。指定できる値の範囲は000～999です。範囲外の値を指定するとエラーになります。

このオプションは、次に示す事項に留意して指定してください。

- 開始時刻は次に示す記号で囲んで指定します。
 - UNIXの場合：シングルクォーテーション(')またはダブルクォーテーション(")
 - Windowsの場合：ダブルクォーテーション(")
- 指定を省略した場合、集計対象は最も古いアクセスログエントリからになります。
- 指定した時刻が-e オプションで指定した終了時刻以降の場合、エラーになります。
- 年/月/日および時:分は省略できません。秒.ミリ秒は省略できますが、秒だけやミリ秒だけは省略できません。秒.ミリ秒を省略した場合、デフォルト値として00.000が仮定されます。

-e '終了時刻'

アクセス回数を集計する期間の終了時刻を指定します。指定した時刻より前のアクセスログエントリが集計対象になります。指定した時刻は、集計期間の対象になりません。例えば、12:45:00.000と指定した場合、12:44:59.999までが集計期間になります。

終了時刻の指定形式は-s オプションの開始時刻と同じです。-s オプションの開始時刻の説明を参照してください。

このオプションは、次に示す事項に留意して指定してください。

- 終了時刻は次に示す記号で囲んで指定します。
 - UNIXの場合：シングルクォーテーション(')またはダブルクォーテーション(")
 - Windowsの場合：ダブルクォーテーション(")
- 指定を省略した場合、集計対象は最も新しいアクセスログエントリまでになります。
- 指定した時刻が-s オプションで指定した開始時刻以前の場合、エラーになります。
- 指定した時刻を満たさないアクセスログエントリは、集計対象から外されます。

例えば、アクセスログファイル中の最初のアクセスログエントリの時刻が12:00:00.000で、最後のアクセスログエントリの時刻が13:00:00.000だとします。このとき、-e オプションに12:45:00.000を指定してEDMRptAclog コマンドを実行した場合、12:00:00.000から12:44:59.999までのアクセスログエントリは集計・出力対象になります。しかし、12:45:00.000から13:00:00.000までのアクセスログエントリは集計・出力対象になりません。

-d 集計間隔

アクセス回数を集計する時間の幅を、分単位の整数で指定します。指定できる値の範囲は 1 ~ 43200 です。例えば、-d 15 と指定すると、15 分間ごとのアクセス回数が集計されます。

このオプションは、次に示す事項に留意して指定してください。

- 整数以外を指定した場合、および 0 以下の値を指定した場合はエラーになります。
- 集計・出力対象のアクセスログエントリは、-s オプション、-e オプションとの組み合わせによって変わります。

例えば、アクセスログファイル中の最初のアクセスログエントリの時刻が 12:00:00.000 で、最後のアクセスログエントリの時刻が 13:00:00.000 だとします。

このとき、-d オプションに 45、-e オプションに 12:45:00.000 を指定して EDMRptAclog コマンドを実行した場合、12:00:00.000 から 12:44:59.999 までのアクセスログエントリが集計・出力対象になります。しかし、12:45:00.000 から 13:00:00.000 までのアクセスログエントリは集計・出力対象になりません。

また、このとき、-d オプションに 45 を指定し、-e オプションを省略して EDMRptAclog コマンドを実行した場合、12:00:00.000 から 12:44:59.999 までのアクセスログエントリが最初のインターバルの集計・出力対象になり、12:45:00.000 から 13:00:00.000 までのアクセスログエントリは次のインターバルの集計・出力対象になります。

-c

集計はしないで、アクセスした時間順にアクセスログを並べて出力する場合に指定します。このとき、動作環境定義ファイルで指定されているオブジェクト変換モードに応じて、アクセスログ中の OIID、操作対象識別子がオブジェクトのプロパティの値に置き換えられます。

-t

アクセス回数の SUBTOTAL (-k オプションで指定した集計キー定義ファイルの第 1 キーごとの小計) と TOTAL (総計) を出力する場合に指定します。

なお、このオプションの指定は、アクセス回数を集計する際のキー項目が二つある場合 (集計キー定義ファイルの [Key User] セクションと [Key Object] セクションを指定した場合) に有効です。

例えば、集計キー定義ファイルで、[Key User] セクションと [Key Object] セクションを指定した場合、[Key User] セクションが第 1 キーになり、ユーザごとの SUBTOTAL と TOTAL が出力されます。

次の場合には、SUBTOTAL と TOTAL は出力されません。

- -k オプションを省略する場合
- キー項目が一つの場合 (集計キー定義ファイルで、[Key User] セクション、[Key Object] セクションのどちらかを指定した場合)

-v

結果を CSV 形式で出力する場合に指定します。

-vt

アクセスログ表を出力する場合に指定します。アクセスログ表については、「6.2.2(1) 集計機能の処理方式 (EDMRptAclog コマンドの集計方法)」を参照してください。

アクセスログ表は、-v オプションを指定しなくても、CSV 形式で出力されます。

-o 出力ファイル名

結果をファイルに出力する場合に、出力するファイル名を指定します。ファイルパスは、フルパス、相対パスまたはファイル名だけのどの方法でも指定できます。ファイル名だけを指定した場合、カレントディレクトリに出力されます。

指定を省略する場合、結果は標準出力 (stdout) に出力されます。

-x 抽出エントリ定義ファイル名

エントリを抽出する場合に、抽出するエントリのメソッド名と引数を指定した抽出エントリ定義ファイルのファイル名を指定します。抽出エントリ定義ファイルについては、「4.1.5 抽出エントリ定義ファイル」を参照してください。

このオプションを `-c` オプションと同時に指定した場合、エントリが抽出されると同時に、アクセスログ中の OIID は、オブジェクトのプロパティの値に変換されます。

このオプションは、次に示す事項に留意して指定してください。

- ファイルパスは、フルパスまたは相対パスのどちらでも指定できます。フルパスで指定されていない場合は、カレントディレクトリ下を探します。
- ハイフン (-) で始まるファイル名は指定できません。ただし、フルパスで指定された場合は処理できます。
- ファイル名に空白が含まれている場合、次に示す記号でファイル名全体を囲めば指定できます。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- 指定されたファイルがない場合や指定されたファイルがディレクトリの場合、エラーになります。

(4) コマンド引数

アクセスログファイル名

解析対象のアクセスログファイルを指定します。

ディレクトリ名

解析対象のディレクトリを指定します。ただし、サブディレクトリ下のファイルは解析対象に含まれません。ディレクトリパスは、1023 バイト以内で指定してください。指定したディレクトリの下にファイルがある場合は、そのファイル名も含めたパスが、1023 バイト以内になるようにしてください。

コマンド引数は、次に示す事項に留意して指定してください。

- ファイルパスおよびディレクトリパスは、フルパスまたは相対パスのどちらでも指定できます。フルパスで指定されていない場合は、カレントディレクトリ下を探します。
- ハイフン (-) で始まるファイル名およびディレクトリ名は指定できません。ただし、フルパスで指定された場合は処理できます。
- ファイル名またはディレクトリ名に空白が含まれている場合、次に示す記号でファイル名全体またはディレクトリ名全体を囲めば指定できます。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- ファイル名およびディレクトリ名の最後のパスの区切り (UNIX の場合「/」、Windows の場合「¥」) がある場合、標準エラー出力 (stderr) にメッセージが出力されます。
- 次の場合、標準エラー出力 (stderr) にメッセージが出力されます。
 - 指定したファイルまたはディレクトリがない場合
 - 指定したディレクトリが一つで、そのディレクトリに処理対象のファイルがない場合
 - 指定したディレクトリが複数で、すべてのディレクトリに処理対象のファイルがない場合
- アクセスログファイルを指定する場合、アクセスログファイルを 160 個まで指定できます。なお、複数のファイルを指定した場合、その中にアクセスできないファイルが一つでもあるときは、処理を中止してメッセージを出力します。また、同じファイルを重複して指定しても、そのファイルに対する処理は 1 回だけになります。
- ディレクトリを指定する場合、複数のディレクトリを指定できます。なお、指定したディレクトリ下のファイル総数が 160 を超えた場合、および指定したディレクトリ下にアクセスできないファイルが一つでもある場合は、処理を中止してメッセージを出力します。また、同じディレクトリを重

複して指定しても、そのディレクトリ下のファイルに対する処理は1回だけになります。

- アクセスログファイルは、ASCII コードで構成されています。
- この指定を省略する場合、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリ（UNIX の場合「DocumentBroker 実行環境ディレクトリ /aru/input/spool/aclog」、Windows の場合「DocumentBroker インストールディレクトリ %Server%\aru\input\spool\aclog」）にある「EDMAccess_「NO」.log」（「NO」は1～16の通し番号）という名称のファイルが対象になります。このとき、処理できるファイルの数は160個ではなく、上記のファイル名に該当するファイルの数になります。
- EDMRptAclog コマンドは、DocumentBroker がアクセスログファイルを生成するディレクトリにあるファイルを対象にして実行できません。EDMRptAclog コマンドを実行する前に、DocumentBroker が生成したアクセスログファイルを、EDMRptAclog コマンドのアクセスログファイル格納ディレクトリにコピーして、コマンド引数でのアクセスログファイルの指定を省略することをお勧めします。アクセスログファイルのコピーについては、「4.1.1 アクセスログファイル」を参照してください。

(5) 注意事項

EDMRptAclog コマンドに関するそのほかの注意事項を次に示します。

- オプションの指定順序に規則はありません。どのような順序でも指定できます。
- オプションには、指定できない組み合わせがあります。オプションの指定の組み合わせを次の表に示します。

表 6-2 オプションの指定の組み合わせ（EDMRptAclog コマンド）

指定するオプション	組み合わせるオプション									
	-k	-s	-e	-d	-c	-t	-v	-vt	-o	-x
-k	x				x			x		x
-s		x								
-e			x							
-d				x	x			x		x
-c	x			x	x	x		x		
-t					x	x		x		x
-v							x			
-vt	x			x	x	x		x		x
-o									x	
-x	x			x		x		x		x

(凡例)

空欄：指定できる組み合わせであることを示す。

x：指定できない組み合わせであることを示す。

6.2.2 EDMRptAclog コマンドの機能の処理方式

EDMRptAclog コマンドの次に示す機能の処理方式について説明します。

- 集計機能
- OIID 変換機能
- 操作対象識別子変換機能
- エントリ抽出機能

(1) 集計機能の処理方式 (EDMRptAclog コマンドの集計方法)

EDMRptAclog コマンドの集計機能では、まず、アクセスログファイル中のアクセスログを解析して、アクセスログエントリごとにアクセスログ表を作成します。その上で、動作環境定義ファイルの内容や、コマンドに指定するオプションに従って、アクセス種別 (アクセスログの集計項目) で分類・集計して、結果を標準出力 (stdout) またはファイルに出力します。

(a) アクセスログ表作成時の処理方式

アクセスログ表には、動作環境定義ファイルに指定されている OIID または操作対象識別子 (OIID 変換、操作対象識別子変換の変換対象) を含むアクセスログエントリだけでなく、アクセス種別で分類できるすべてのアクセスログエントリが出力されます。例えば、次のようなアクセスログエントリも出力されます。

動作環境定義ファイルに指定されていない OIID または操作対象識別子を含むアクセスログエントリ

統計解析ツールの対象ではない OIID または操作対象識別子を含むアクセスログエントリ

OIID または操作対象識別子を含んでいないアクセスログエントリ

これらのアクセスログエントリには、メソッド名の後ろに付加情報が付いている場合があります。アクセスログ表には、アクセスログエントリに含まれる付加情報の個数分の行が出力されます。OIID および操作対象識別子は付加情報の一種であり、例えば、OIID が 2 個含まれているアクセスログエントリに対して、アクセスログ表は 2 行出力されます。

アクセスログ表の項目を次の表に示します。

表 6-3 アクセスログ表の項目

項目名	説明
日付	アクセスした日付 (年 / 月 / 日)
時刻	アクセスした時刻 (時 : 分 : 秒 . ミリ秒)
ユーザ名	アクセスしたユーザのユーザ名。
アクセス種別	connect, create, delete, error, refer, search または update のどれか。アクセス種別については、「表 4-1 集計機能で分類されるアクセスの内容と集計項目 (アクセス種別) の関係」を参照してください。
OIID の末尾 16 バイト	アクセス対象オブジェクトを示す 16 進数の数値。アクセス対象オブジェクトが登録された時に、文書空間内で割り当てられたオブジェクト番号です。
OIID に対応するプロパティ	アクセス対象オブジェクトのプロパティの値 / クラス名。 ただし、プロパティ拡張情報ファイルに変換対象のプロパティのクラス識別子が指定されていない場合は、アクセス対象オブジェクトを示す 16 進数の数値が出力されます。

なお、動作環境定義ファイルに指定されている OIID または操作対象識別子を含むアクセスログエントリのアクセスログ表を作成する場合、その OIID または操作対象識別子は、OIID 変換機能、操作対象識別子変換機能によって対応するプロパティの値に置き換えられます。それ以外の場合、アクセスログ表の OIID の末尾 16 バイトおよび OIID に対応するプロパティの値は空白になります。アクセスログ表の例を次の表に示します。

表 6-4 アクセスログ表の例

日付	時刻	ユーザ名	アクセス種別	OIID の末尾 16 バイト	OIID に対応するプロパティ
2000/06/19	02:18:19.388	ACUser6	connect	0000000000001216	設計書 /usrClass
2000/06/19	02:18:52.261	ACUser6	create	0000000000001216	設計書 /usrClass
2000/06/19	02:19:16.420	ACUser6	refer	0000000000001216	設計書 /usrClass
2000/06/19	02:19:16.420	ACUser6	refer	0000000000001221	企画書 /usrClass

アクセスログ表は、EDMRptAclog コマンド実行時に `-vt` オプションを指定することによって出力できます。

(b) アクセスログエントリ集計時の処理方式

アクセスログエントリを集計するときの集計対象は次の二つです。

OIID 変換または操作対象識別子変換の変換対象を含むアクセスログエントリ

OIID 変換，操作対象識別子変換の変換対象は，動作環境定義ファイルの [ObjectConvertingMode] セクションに指定されています。なお，動作環境定義ファイルについては，「4.1.6 動作環境定義ファイル」を参照してください。

OIID および操作対象識別子を含まないアクセスログエントリ

OIID および操作対象識別子を含まないアクセスログエントリの例を次に示します。

```
2000/06/19 02:18:52.261 ACUser6 DISCONNECT
```

ただし、`-k` オプションに集計キー定義ファイルを指定したときの集計対象は，集計キー定義ファイルで指定された項目によって異なります。集計キー定義ファイルで指定された項目別の集計対象を次の表に示します。

表 6-5 集計キー定義ファイルで指定された項目別の集計対象

集計キー定義ファイルで指定された項目	集計対象	
	OIID および操作対象識別子を含まないアクセスログエントリ	OIID 変換または操作対象識別子変換の変換対象を含むアクセスログエントリ
ユーザ		
オブジェクト	×	
ユーザおよびオブジェクト	×	

(凡例)

：集計する。

×：集計しない。

なお，OIID 変換，操作対象識別子変換の変換対象を含むアクセスログエントリで，一つのアクセスログエントリに OIID または操作対象識別子が二つ出現したとき，それぞれの OIID または操作対象識別子の変換後の値によって集計のしかたが異なります。

変換後の値が同じ場合

そのアクセスログエントリは、1件として集計します。

変換後の値が異なる場合

そのアクセスログエントリは、2件として集計します。

(2) OIID 変換機能，操作対象識別子変換機能の処理方式

EDMRptAclog コマンドの OIID 変換機能，操作対象識別子変換機能では，プロパティ拡張情報ファイルを参照し，アクセスログ中の OIID および操作対象識別子に置き換えるプロパティの値をデータベースから取得して変換します。

(a) OIID 変換機能の処理の流れ

OIID 変換機能の処理の流れを次の図に示します。

図 6-1 OIID 変換機能の処理の流れ

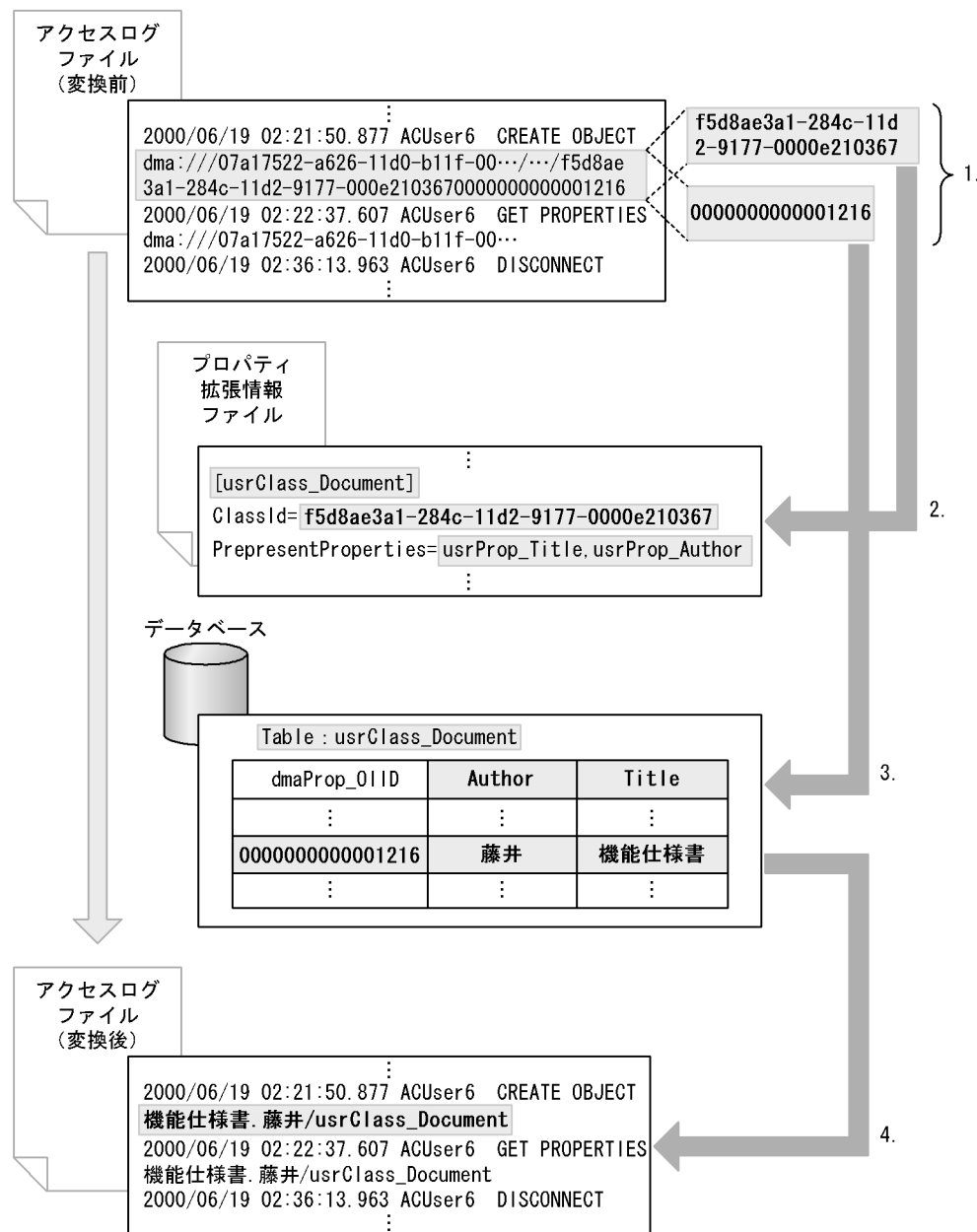


図 6-1 の 1 ~ 4 について説明します。

1. アクセスログの OIID から、クラス識別子および OIID 末尾 16 バイトを抽出する。
アクセスログの OIID から、クラス識別子（例：f5d8ae3a1-284c-11d2-9177-0000e210367）および OIID 末尾 16 バイト（例：0000000000001216）を抽出します。
2. プロパティ拡張情報ファイルを検索して、手順 1 で抽出したクラス識別子が示すクラス名、および OIID 変換後の値となるプロパティのプロパティ名を取得する。
プロパティ拡張情報ファイル（propex.ini）で、手順 1 で抽出したクラス識別子と一致する ClassID エントリを検索して、そのクラス識別子が示すクラス名（例：usrClass_Document）、および OIID 変換後の値となるプロパティのプロパティ名（例：usrProp_Title, usrProp_Author）を取得します。
3. データベースを検索して、OIID 変換後の値となるプロパティの値を取得する。
手順 2 で取得したクラス名のテーブルで、手順 1 で抽出した OIID 末尾 16 バイトと dmaProp_OIID カラムの値が一致するレコードを検索して、OIID 変換後の値となるプロパティのプロパティ名のカラム（例：usrProp_Title, usrProp_Author）に格納されている値（例：機能仕様書・藤井 / usrClass_Document）を取得します。
4. アクセスログの OIID を、プロパティの値に置き換える。
アクセスログの OIID を、手順 3 で取得したプロパティの値（例：機能仕様書・藤井 / usrClass_Document）に置き換えます。

(b) 操作対象識別子変換機能の処理の流れ

操作対象識別子変換機能の処理の流れを次の図に示します。なお、次の図では、バージョン識別子に対して操作対象識別子変換を実行する場合の処理の流れを示しています。

図 6-2 操作対象識別子変換機能の処理の流れ（バージョン識別子の場合）

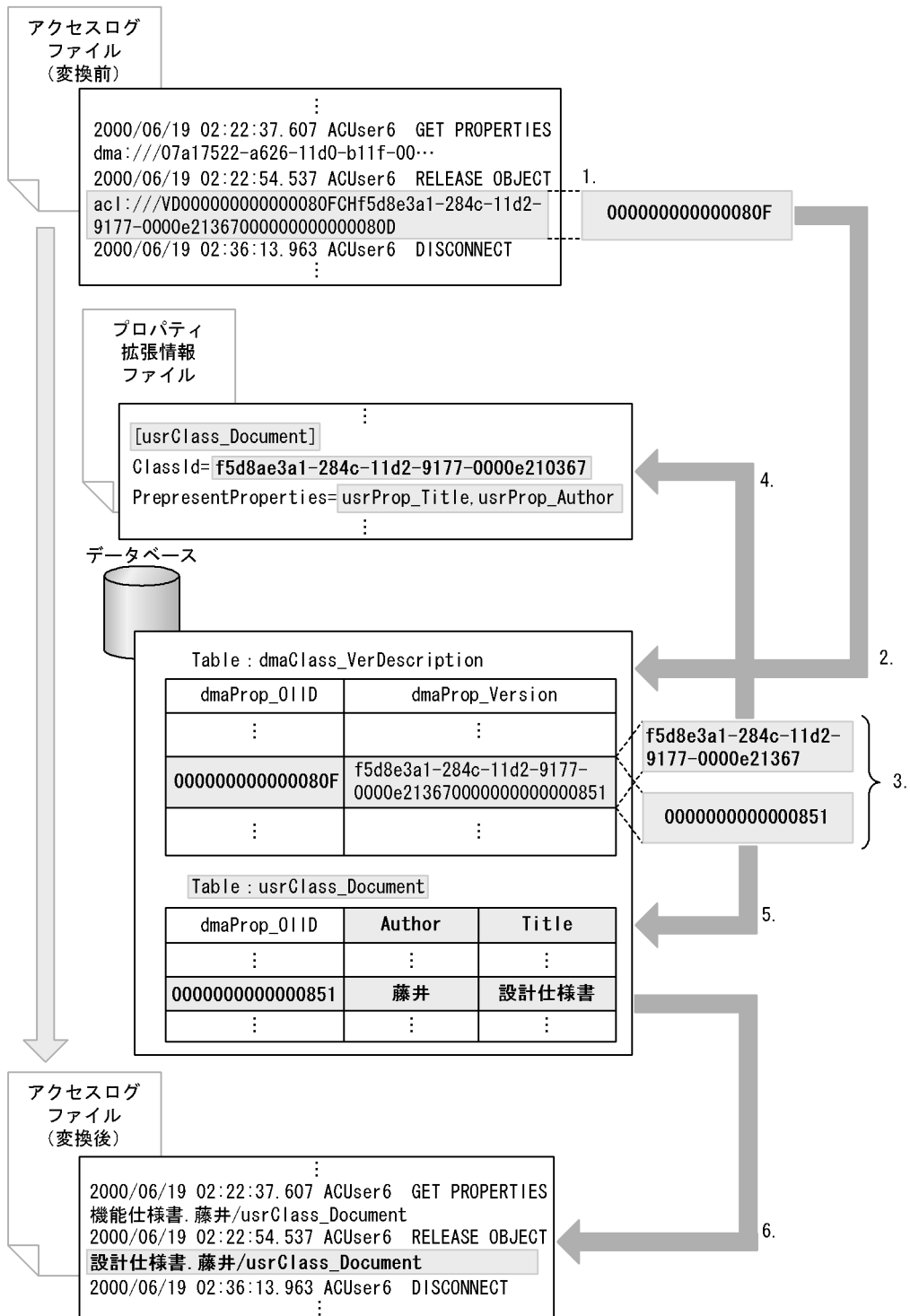


図 6-2 の 1 ~ 6 について説明します。

1. アクセスログから操作対象識別子の一部を抽出する。
アクセスログから操作対象識別子の一部（例：000000000000080F）を抽出します。
2. データベースを検索して、操作対象識別子の値をプロパティとして持つオブジェクトの OIID を取得す

る。

dmaClass_VerDescription テーブル で、手順 1 で抽出した操作対象識別子の一部と dmaProp_OIID カラムの値が一致するレコードを検索して、操作対象識別子の値をプロパティとして持つオブジェクトの OIID (dmaProp_Version カラム の値) を取得します。

3. 手順 2 で取得した OIID から、クラス識別子および OIID 末尾 16 バイトを抽出する。
手順 2 で取得した OIID から、クラス識別子 (例: f5d8ae3a1-284c-11d2-9177-0000e210367) および OIID 末尾 16 バイト (例: 0000000000000851) を抽出します。
4. プロパティ拡張情報ファイルを検索して、手順 3 で抽出したクラス識別子が示すクラス名、および操作対象識別子変換後の値となるプロパティのプロパティ名を取得する。
プロパティ拡張情報ファイル (propex.ini) で、手順 3 で抽出したクラス識別子と一致する ClassID エントリの値を検索して、そのクラス識別子が示すクラス名 (例: usrClass_Document), および操作対象識別子変換後の値となるプロパティのプロパティ名 (例: usrProp_Title, usrProp_Author) を取得します。
5. データベースを検索して、操作対象識別子変換後の値となるプロパティの値を取得する。
手順 4 で取得したクラス名のテーブルで、手順 3 で抽出した OIID 末尾 16 バイトと dmaProp_OIID カラムの値が一致するレコードを検索して、操作対象識別子変換後の値となるプロパティのプロパティ名のカラム (例: usrProp_Title, usrProp_Author) に格納されている値 (例: 設計仕様書・藤井 / usrClass_Document) を取得します。
6. アクセスログの操作対象識別子を、プロパティの値に置き換える。
アクセスログの操作対象識別子を、手順 5 で取得したプロパティの値 (例: 設計仕様書・藤井 / usrClass_Document) に置き換えます。

注

データベースで検索するテーブル、および操作対象識別子の値をプロパティとして持つオブジェクトの OIID が格納されているカラムは、操作対象識別子の種類によって異なります。操作対象識別子別のデータベースで検索するテーブル名およびカラム名を次の表に示します。

表 6-6 操作対象識別子別のデータベースで検索するテーブル名およびカラム名

操作対象識別子名 (オブジェクト変換モード)	テーブル名	カラム名
バージョン識別子 (VD)	dmaClass_VerDescription	dmaProp_Version
リレーション識別子 (RL)	edmClass_Relationship	dmaProp_Head
リンク識別子 (LD)	dmaClass_DCRelationship	dmaProp_Head
リンク識別子 (LR)	dmaClass_RCRelationship	dmaProp_Head
リンク識別子 (LV)	edmClass_VTCRelationship	dmaProp_Head

OIID 変換、操作対象識別子変換でエラーが発生した場合に考えられるエラーの要因については、「5.1.2 OIID 変換、操作対象識別子変換時のエラー処理」を参照してください。

(3) エントリ抽出機能の処理方式

アクセスログエントリには、アクセスの内容を表すメソッド名と引数が含まれています。EDMRptAclog コマンドのエントリ抽出機能では、抽出エントリ定義ファイルに指定されたメソッド名と引数によって、アクセスログエントリを抽出して、結果を標準出力 (stdout) またはファイルに出力します。

6.2.3 EDMRptAclog コマンドの出力形式

EDMRptAclog コマンド実行結果の出力形式について説明します。

(1) 集計機能の出力形式

EDMRptAclog コマンドの集計機能の出力例を次の図に示します。

図 6-3 EDMRptAclog コマンドの集計機能の出力例

# EDMRptAclog -k user_object.def -t				
# COUNT START = 2000/06/19 15:00:00				
# COUNT END = 2000/06/19 18:00:01				
# LOG START = 2000/06/19 15:00:00				
# LOG END = 2000/06/19 18:00:00				
user	object	access	times	
user01	機能仕様書/usrClass_Document	refer	1	
		update	3	
	マニュアル. 藤井/usrClass_Manual	refer	1	
		update	3	
SUBTOTAL		refer	2	
		update	6	
user02	(deleted)	refer	1	
		update	2	
SUBTOTAL		refer	1	
		update	2	
TOTAL		refer	3	
		update	8	

図 6-3 の要素 (a) ~ (e) について説明します。

(a) 共通情報

集計結果に関する共通情報が出力されます。COUNT START から INTERVAL までの共通情報の出力形式を次に示します。

```
# < 共通情報名 > = < 値 >
```

出力される共通情報の内容は、EDMRptAclog コマンドのオプションの指定によって変わります。共通情報の内容およびオプションの指定との関係を次に示します。

```
# EDMRptAclog < オプション >< 引数 >
```

実行したコマンドラインが出力されます。

```
# COUNT START = yyyy/mm/dd hh:mm:ss.sss
```

開始エントリの時刻が出力されます。-s オプションで開始時刻を指定している場合、その指定値が出力されます。-s オプションで開始時刻を指定していない場合、ログ開始時刻 (LOG START) が出力されます。

COUNT START に該当するエントリは、集計対象になります。

```
# COUNT END = yyyy/mm/dd hh:mm:ss.sss
```

終了エントリの時刻が出力されます。-e オプションで終了時刻を指定している場合、その指定値が出力されます。-e オプションで終了時刻を指定していない場合、ログ終了時刻 (LOG END) に 1 ミリ

6. コマンドリファレンス

秒を足した時刻が出力されます。

COUNT END に該当するエントリは、集計対象になりません。

LOG START = yyyy/mm/dd hh:mm:ss.sss

ログ開始時刻（ログ内の最も古いアクセスログエントリの時刻）が出力されます。

LOG END = yyyy/mm/dd hh:mm:ss.sss

ログ終了時刻（ログ内の最も新しいアクセスログエントリの時刻）が出力されます。

INTERVAL = mm minutes

-d オプションの指定値（アクセス回数の集計間隔）が出力されます。-d オプションを指定していない場合は出力されません。

(b) 分類項目

分類項目が出力され、その下に各項目の内容が出力されます。集計のキーとなる項目は、-k オプションで指定した集計キー定義ファイルに指定された項目です。分類項目とその内容を次に示します。

user

ユーザ名が 14 バイトまで出力されます。14 バイトを超える場合、15 バイト以降は出力されません。

CSV 形式では 15 バイト以降も出力されます。

user はアクセスログに出現した順に出力されます。

object

アクセス対象オブジェクトのプロパティの値とクラス名が、次に示す形式で出力されます。プロパティが二つ以上ある場合、ピリオド(.)を区切り文字として並べて出力されます。

プロパティの値/クラス名

プロパティの値とクラス名は 48 バイトまで出力されます。48 バイトを超える場合、49 バイト以降は出力されません。CSV 形式では 49 バイト以降も出力されます。

object はアクセスログに出現した順に出力されます。

なお、プロパティ拡張情報ファイルに変換対象のプロパティが指定されていても、データベース上に該当するレコードがない場合は、「(deleted)」が出力されます。また、プロパティ拡張情報ファイルに変換対象のプロパティが指定されていないため、データベースからプロパティの値を取得できない場合は、OID の末尾 16 バイトが出力されます。

(c) 集計結果

集計結果の項目が出力され、その下に各項目の内容が出力されます。集計結果の項目とその内容を次に示します。

access

アクセス種別が出力されます。アクセス種別とその意味を次の表に示します。

表 6-7 アクセス種別とその意味

アクセス種別	意味
connect	文書空間への接続
create	オブジェクトの生成
delete	オブジェクトの削除
error	エラー
refer	オブジェクトの参照

アクセス種別	意味
search	検索の実行（検索結果の取得も含む）
update	オブジェクトの更新

なお、アクセス種別は、アクセスログに含まれるメソッド名に基づいた集計項目で、メソッド名そのものではありません。アクセス種別とメソッド名の関係については、「表 4-1 集計機能で分類されるアクセスの内容と集計項目（アクセス種別）の関係」を参照してください。

times

アクセスの回数を示す数字が、アクセス種別ごとに 7 バイトまで出力されます。7 バイトを超える場合、8 バイト以降は出力されません。CSV 形式では 8 バイト以降も出力されます。

(d) SUBTOTAL

-t オプションを指定した場合、-k オプションで指定した集計キー定義ファイルの第 1 キーごとのアクセス回数の小計が、アクセス種別ごとに出力されます。

(e) TOTAL

-t オプションを指定した場合、アクセス回数の総計がアクセス種別ごとに出力されます。

(2) OIID 変換機能、操作対象識別子変換機能（-c オプション指定）の出力形式

-c オプションを指定した場合、集計はされないで、アクセスした時間順にアクセスログが出力されます。このとき、動作環境定義ファイルに指定された変換モードに応じて、OIID、操作対象識別子がオブジェクトのプロパティの値に置き換えられます。

OIID 変換機能、操作対象識別子変換機能の出力例を次の図に示します。太字で示す部分が置き換わります。なお、OIID または操作対象識別子がプロパティの値（プロパティの値 / クラス名）に置き換わる以外は、元のアクセスログファイルと同じ形式になります。

図 6-4 EDMRptAclog コマンドの OIID 変換機能、操作対象識別子変換機能の出力例

変換前のアクセスログ

```
2000/06/19 02:21:50.877 ACUser6 CREATE OBJECT dma:///07a17522-a626-11d0-b11f-00...
2000/06/19 02:22:37.607 ACUser6 GET PROPERTIES dma:///07a17522-a626-11d0-b11f-00...
2000/06/19 02:22:54.537 ACUser6 RELEASE OBJECT acl:///VD000000000000080FCHf5d8e3...
2000/06/19 02:36:13.963 ACUser6 DISCONNECT
```

変換後のアクセスログ（OIID変換の出力結果）

```
2000/06/19 02:21:50.877 ACUser6 CREATE OBJECT 機能仕様書/usrClass_Document
2000/06/19 02:22:37.607 ACUser6 GET PROPERTIES 機能仕様書/usrClass_Document
2000/06/19 02:22:54.537 ACUser6 RELEASE OBJECT 設計仕様書/usrClass_Document
2000/06/19 02:36:13.963 ACUser6 DISCONNECT
```

(3) エントリ抽出機能の出力形式

-x オプションを指定した場合、抽出エントリ定義ファイルに指定したメソッド名と引数によって、アクセスログエントリが抽出されて出力されます。

エントリ抽出機能の出力例を次の図に示します。この例では、抽出エントリ定義ファイルにメソッド名「CREATE」と引数「OBJECT」を指定して、CREATE OBJECT エントリを抽出しています。

図 6-5 EDMRptAclog コマンドのエントリ抽出機能の出力例

```
# EDMRptAclog -x extract.def Input.log
# COUNT START = 2000/06/19 02:05:00.000
# COUNT END   = 2000/06/19 02:50:00.001
# LOG START   = 2000/06/19 02:05:00.000
# LOG END     = 2000/06/19 02:50:00.000

2000/06/19 02:05:00.000 ACUser6 CREATE OBJECT dma:///07a17526-11d0-b11f-00...
```

-x オプションと -c オプションを同時に指定した場合は、エントリが抽出されると同時に、アクセスログ中の OIID は、オブジェクトのプロパティの値に変換されます。

6.2.4 EDMRptAclog コマンドの出力例

EDMRptAclog コマンド実行結果の出力例を次に示します。

なお、次に示す出力例では、集計されたアクセスログエントリには、動作環境定義ファイルで指定した、OIID 変換または操作対象識別子変換の変換対象が含まれていることが前提になっています。また、EDMRptAclog コマンドで -k オプションを指定した場合、集計キー定義ファイルに指定した項目によって集計対象になるアクセスログエントリが異なっています。集計キー定義ファイルに指定した項目別の集計対象については、「6.2.2(1) 集計機能の処理方式 (EDMRptAclog コマンドの集計方法)」を参照してください。

(1) オプション指定なしの出力例

オプションを指定しないでコマンドを実行した例を次に示します。

```
# EDMRptAclog
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

access      times
-----
connect      1
search       2
create       3
delete       4
refer        5
update       6
error        7
-----
```

(2) CSV 形式での出力例 (-v 指定)

-v オプションだけを指定して、CSV 形式で出力した例を次に示します。

```
"#", "EDMRptAclog -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"access", "times"
"connect", 1
"search", 2
"create", 3
"delete", 4
"refer", 5
"update", 6
"error", 7
```


(3) ユーザごとのアクセス回数の出力例 (-k 集計キー定義ファイル 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。すべてのユーザが集計対象になります。

```
[Key User]
```

すべてのユーザごとのアクセス回数を出力した例を次に示します。

```
# EDMRptAclog -k count.def
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

user      access      times
-----
user01    connect     1
          search     2
          create     3
          delete    4
          refer     5
          update    6
          error     7
-----
user02    connect     8
          search     9
          create    10
          delete    11
          refer     12
          update    13
          error     14
-----
user03    connect    15
          search    16
          create    17
          delete    18
          refer     19
          update    20
          error     21
-----
```

(4) ユーザごとのアクセス回数の CSV 形式での出力例 (-k 集計キー定義ファイル -v 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。特定のユーザ (user01 と user03) が集計対象になります。

```
[Key User]
user01
user03
```

特定のユーザごとのアクセス回数を CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、-k オプションで指定したキー項目の内容 (この例ではユーザ名) は、集計結果の行数分、繰り返し出力されます。

6. コマンドリファレンス

```
"#", "EDMRptAclog -k count.def -v"  
"#", "COUNT START", "2000/06/19", "15:00:00.000"  
"#", "COUNT END", "2000/06/19", "18:00:00.001"  
"#", "LOG START", "2000/06/19", "15:00:00.000"  
"#", "LOG END", "2000/06/19", "18:00:00.000"  
"user", "access", "times"  
"user01", "connect", 1  
"user01", "search", 2  
"user01", "create", 3  
"user01", "delete", 4  
"user01", "refer", 5  
"user01", "update", 6  
"user01", "error", 7  
"user03", "connect", 15  
"user03", "search", 16  
"user03", "create", 17  
"user03", "delete", 18  
"user03", "refer", 19  
"user03", "update", 20  
"user03", "error", 21
```

(5) オブジェクトごとのアクセス回数の出力例 (-k 集計キー定義ファイル 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。すべてのオブジェクトが集計対象になります。

```
[Key Object]
```

すべてのオブジェクトごとのアクセス回数を出力した例を次に示します。

プロパティ拡張情報ファイルに変換対象のプロパティは指定されていても、データベース上に該当するレコードがない場合は、「(deleted)」が出力されます。

```
# EDMRptAclog -k count.def  
# COUNT START = 2000/06/19 15:00:00.000  
# COUNT END   = 2000/06/19 18:00:00.001  
# LOG START   = 2000/06/19 15:00:00.000  
# LOG END     = 2000/06/19 18:00:00.000  
  
object                                access    times  
-----  
機能仕様書/usrClass_Document         create    1  
                                       refer     2  
                                       update    3  
-----  
マニュアル.藤井/usrClass_Manual       refer     3  
                                       update    4  
-----  
(deleted)                             refer     5  
                                       update    6  
-----
```

(6) ユーザ、オブジェクトごとのアクセス回数の出力例 (-k 集計キー定義ファイル 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。すべてのユーザのすべてのオブジェクトが集計対象になります。

```
[Key User]  
[Key Object]
```

すべてのユーザ、オブジェクトごとのアクセス回数を出力した例を次に示します。

プロパティ拡張情報ファイルに変換対象のプロパティが指定されていても、データベース上に該当するレコードがない場合は、「(deleted)」が出力されます。また、プロパティ拡張情報ファイルに変換対象のプロパティが指定されていないため、データベースからプロパティの値を取得できない場合は、OIDの末尾

16 バイトが出力されます。

```
# EDMRptAclog -k count.def
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

user      object                                     access  times
-----
user01    機能仕様書/usrClass_Document                refer    1
                                                update  2
-----
          マニュアル.藤井/usrClass_Manual        refer    3
                                                update  4
-----
user02    (deleted)                                     refer    5
                                                update  6
-----
          0000000000001216                       refer    7
                                                update  8
-----
```

(7) ユーザ、オブジェクトごとのアクセス回数の CSV 形式での出力例 (-k 集計キー定義ファイル -v 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。特定のユーザ (user01) の特定のオブジェクト (マニュアル) が集計対象になります。

```
[Key User]
user01
[Key Object]
マニュアル
```

特定のユーザ、オブジェクトごとのアクセス回数を CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、-k オプションで指定したキー項目の内容 (この例ではユーザ名とプロパティの値 / クラス名) は、集計結果の行数分、繰り返し出力されます。

```
"#", "EDMRptAclog -k count.def -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"#"
"user", "object", "access", "times"
"user01", "マニュアル.藤井/usrClass_Manual", "refer", 3
"user01", "マニュアル.藤井/usrClass_Manual", "update", 4
```

(8) OIID 変換、操作対象識別子変換の出力例 (-c 指定)

アクセスログ中の OIID、操作対象識別子をプロパティの値に変換して出力した例を次に示します。

```
# EDMRptAclog -c
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

2000/06/19 15:00:00.000 ACUser6 CREATE OBJECT 機能仕様書/usrClass_Document
2000/06/19 15:10:00.000 ACUser6 GET PROPERTIES 機能仕様書/usrClass_Document (nil)
2000/06/19 15:20:00.000 ACUser6 RELEASE OBJECT 設計仕様書/usrClass_Document
2000/06/19 18:00:00.000 ACUser6 DISCONNECT
```

(9) OIID 変換、操作対象識別子変換の CSV 形式での出力例 (-c -v 指定)

アクセスログ中の OIID、操作対象識別子をプロパティの値に変換して CSV 形式で出力した例を次に示し

6. コマンドリファレンス

ます。

```
"#", "EDMRptAclog -c -v"  
"#", "COUNT START", "2000/06/19", "15:00:00.000"  
"#", "COUNT END", "2000/06/19", "18:00:00.001"  
"#", "LOG START", "2000/06/19", "15:00:00.000"  
"#", "LOG END", "2000/06/19", "18:00:00.000"  
"2000/06/19", "15:00:00.000", "ACUser6", "CREATE OBJECT", "機能仕様書/usrClass_Document"  
"2000/06/19", "15:10:00.000", "ACUser6", "GET PROPERTIES", "機能仕様書/usrClass_Document (nil)"  
"2000/06/19", "15:20:00.000", "ACUser6", "RELEASE OBJECT", "設計仕様書/usrClass_Document"  
"2000/06/19", "18:00:00.000", "ACUser6", "DISCONNECT"
```

(10) SUBTOTAL, TOTAL の出力例 (-k 集計キー定義ファイル -t 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。

```
[Key User]  
[Key Object]
```

SUBTOTAL および TOTAL を出力した例を次に示します。

プロパティ拡張情報ファイルに変換対象のプロパティが指定されていても、データベース上に該当するレコードがない場合は、「(deleted)」が出力されます。

```
# EDMRptAclog -k count.def -t  
# COUNT START = 2000/06/19 15:00:00.000  
# COUNT END   = 2000/06/19 18:00:00.001  
# LOG START   = 2000/06/19 15:00:00.000  
# LOG END     = 2000/06/19 18:00:00.000
```

user	object	access	times
user01	機能仕様書/usrClass_Document	refer	1
		update	3
	マニュアル.藤井/usrClass_Manual	refer	1
		update	3
SUBTOTAL		refer	2
		update	6
user02	(deleted)	refer	1
		update	2
SUBTOTAL		refer	1
		update	2
TOTAL		refer	3
		update	8

(11) SUBTOTAL, TOTAL の CSV 形式での出力例 (-k 集計キー定義ファイル -t -v 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。

```
[Key User]  
[Key Object]
```

SUBTOTAL および TOTAL を CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、-k オプションで指定したキー項目の内容（この例ではユーザ名とプロパティの値 / クラス名）と -t オプションの SUBTOTAL および TOTAL は、集計結果の行数分、繰り返し出力されます。

プロパティ拡張情報ファイルに変換対象のプロパティが指定されていても、データベース上に該当するレコードがない場合は、「(deleted)」が出力されます。

```

"#", "EDMRptAclog -k count.def -t -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"user", "object", "access", "times"
"user01", "機能仕様書/usrClass_Document", "refer", 1
"user01", "機能仕様書/usrClass_Document", "update", 3
"user01", " マニュアル.藤井/usrClass_Manual", "refer", 1
"user01", " マニュアル.藤井/usrClass_Manual", "update", 3
"SUBTOTAL", "", "refer", 2
"SUBTOTAL", "", "update", 6
"user02", " (deleted)", "refer", 1
"user02", " (deleted)", "update", 2
"SUBTOTAL", "", "refer", 1
"SUBTOTAL", "", "update", 2
"TOTAL", "", "refer", 3
"TOTAL", "", "update", 8

```

(12) エントリの抽出例 (-x 抽出エントリ定義ファイル 指定)

この例では、-x オプションで、次の抽出エントリ定義ファイルを指定したと仮定します。

```

[Extract Aclog]
CREATE OBJECT

```

また、この例では、コマンド引数で、次のアクセスログファイルを指定したと仮定します。

```

Input.log:
2000/06/19 02:05:00.000 ACUser6 CREATE OBJECT dma:///07a17526-11d0-b11f-00...
2000/06/19 02:25:00.000 ACUser6 GET PROPERTIES dma:///07a626-11d0-b11f-00...
2000/06/19 02:40:00.000 ACUser6 RELEASE OBJECT dma:///07-a626-11d0-b11f-00...
2000/06/19 02:50:00.000 ACUser6 DISCONNECT

```

アクセスログファイルから CREATE OBJECT エントリを抽出して出力した例を次に示します。

```

# EDMRptAclog -x extract.def Input.log
# COUNT START = 2000/06/19 02:05:00.000
# COUNT END   = 2000/06/19 02:50:00.001
# LOG START   = 2000/06/19 02:05:00.000
# LOG END     = 2000/06/19 02:50:00.000

2000/06/19 02:05:00.000 ACUser6 CREATE OBJECT dma:///07a17526-11d0-b11f-00...

```

(13) アクセスログ表の出力例 (-vt 指定)

アクセスログ表を出力した例を次に示します。アクセスログ表は、-v オプションを指定しなくても、CSV 形式で出力されます。

```

"#", "EDMRptAclog -vt"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"date", "time", "user", "access", "end 16 bytes of OIID", "property corresponding to OIID"
"2000/06/19", "15:00:00.000", "ACUser6", "connect", "0000000000001216", "Design document/
usrClass_Folder"
"2000/06/19", "15:10:00.000", "ACUser6", "create", "0000000000001216", "Design document/
usrClass_Folder"
"2000/06/19", "15:20:00.000", "ACUser6", "refer", "0000000000001216", "Design document/
usrClass_Folder"
"2000/06/19", "18:00:00.000", "ACUser6", "refer", "0000000000001221", "Plan document.Tanaka/
usrClass_Folder"

```

6. コマンドリファレンス

(14) 集計間隔を指定した出力例 (-k 集計キ-定義ファイル -s '開始時刻' -e '終了時刻' -d 集計間隔 指定)

この例では、-k オプションで、次の集計キ-定義ファイルを指定したと仮定します。

```
[Key Object]
機能仕様書
```

特定のオブジェクト (機能仕様書)、集計の開始時刻 (2000/06/19 14:30:00.000) と終了時刻 (2000/06/19 16:30:00.000) に加えて、集計間隔 (20) を指定して出力した例を次に示します。

```
# EDMRptAclog -k count.def -s '2000/06/19 14:30:00.000' -e '2000/06/19 16:30:00.000' -d 20
# COUNT START = 2000/06/19 14:30:00.000
# COUNT END   = 2000/06/19 16:30:00.000
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 16:00:00.000
# INTERVAL    = 20 minutes

# COUNT START = 2000/06/19 14:30:00.000
# No entry found.
# COUNT START = 2000/06/19 14:50:00.000

object                                access  times
-----
機能仕様書/usrClass_Document        create   1
                                       refer    2
                                       update   3
-----

# COUNT START = 2000/06/19 15:10:00.000

object                                access  times
-----
機能仕様書/usrClass_Document        refer    4
                                       update   5
-----

# COUNT START = 2000/06/19 15:30:00.000

object                                access  times
-----
機能仕様書/usrClass_Document        refer    6
                                       update   7
-----

# COUNT START = 2000/06/19 15:50:00.000

object                                access  times
-----
機能仕様書/usrClass_Document        refer    8
                                       update   9
-----

# COUNT START = 2000/06/19 16:10:00.000
# No entry found.
```

この例では、集計対象となるのは、2000/06/19 14:30:00.000 から 2000/06/19 16:29:59.999 までのアクセスログエントリです。2000/06/19 14:30:00.000 から 20 分間隔で集計しています。

ログ終了時刻 2000/06/19 16:00:00.000 を超えて 2000/06/19 16:10:00.000 に集計したときには、該当するエントリがない、つまりオブジェクトに対するアクセスがなかったため「No entry found.」という結果が出力されています。

(15) 集計間隔を指定した CSV 形式での出力例 (-k 集計キー定義ファイル s '開始時刻' -e '終了時刻' -d 集計間隔 -v 指定)

この例では、-k オプションで、次の集計キー定義ファイルを指定したと仮定します。

```
[Key Object]
機能仕様書
```

特定のオブジェクト（機能仕様書）、集計の開始時刻（2000/06/19 14:30:00.000）と終了時刻（2000/06/19 16:30:00.000）に加えて、集計間隔（20）を指定して CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、-k オプションで指定したキー項目の内容（この例ではプロパティの値 / クラス名）は、集計結果の行数分、繰り返し出力されます。

```
"#", "EDMRptAclog -k count.def -s '2000/06/19 14:30:00.000' -e '2000/06/19 16:30:00.000' -d 20
-v"
"#", "COUNT START", "2000/06/19", "14:30:00.000"
"#", "COUNT END", "2000/06/19", "16:30:00.000"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "16:00:00.000"
"#", "INTERVAL", 20, "minutes"
"#", "COUNT START", "2000/06/19", "14:30:00.000"
"#", "No entry found."
"#", "COUNT START", "2000/06/19", "14:50:00.000"
"object", "access", "times"
"機能仕様書/usrClass_Document", "create", 1
"refer", 2
"update", 3
"#", "COUNT START", "2000/06/19", "15:10:00.000"
"object", "access", "times"
"機能仕様書/usrClass_Document", "refer", 4
"update", 5
"#", "COUNT START", "2000/06/19", "15:30:00.000"
"object", "access", "times"
"機能仕様書/usrClass_Document", "refer", 6
"update", 7
"#", "COUNT START", "2000/06/19", "15:50:00.000"
"object", "access", "times"
"機能仕様書/usrClass_Document", "refer", 8
"update", 9
"#", "COUNT START", "2000/06/19", "16:10:00.000"
"#", "No entry found."
```

6.3 EDMRptTrace コマンドの詳細

この節では、EDMRptTrace コマンドの文法、機能の処理方式、実行結果の出力形式および出力例について説明します。

6.3.1 EDMRptTrace コマンド (トレースの解析)

(1) 機能

EDMRptTrace コマンドは、指定されたトレースファイルを解析し、DocumentBroker のサーバおよびクライアントのプログラムの処理内容を集計して結果を出力したり、分割してファイルに出力したりします。サーバトレース解析の場合は、ユーザプログラムのプログラム名とユーザプログラムのプロセス ID ごとに集計・分割します。クライアントトレース解析の場合は、ユーザプログラムのプロセス ID とユーザプログラムのスレッド ID ごとに集計し、ユーザプログラムのスレッド ID ごとに分割します。

(2) 形式

```
EDMRptTrace
  { -sv | -cl }
  [ -s '開始時刻' ]
  [ -e '終了時刻' ]
  [ -d 集計間隔 ]
  [ -w ]
  [ -t ]
  [ -v ]
  [ -vt ]
  [ -o 出力ファイル名 ]
  [ -x 抽出エントリ定義ファイル名 ]
  [ トレースファイル名 ... ]
  [ ディレクトリ名 ... ]
```

(3) オプション

{ -sv | -cl }

解析するトレースファイルの種別を指定します。指定できるのは次に示すどちらかです。なお、このオプションの指定は省略できません。

-sv

サーバトレースを解析する場合に指定します。

-cl

クライアントトレースを解析する場合に指定します。

-s '開始時刻'

ユーザプログラムの処理内容を集計する期間の開始時刻を指定します。指定した時刻以降のトレースログエントリが集計対象になります。指定した時刻は、集計期間の対象になります。例えば、12:00:00.000 と指定した場合、12:00:00.000 からが集計期間になります。

開始時刻は次に示す形式で指定してください。

'yyyy/mm/dd hh:mm [:ss.sss]'

yyyy

日付の年の値を指定します。

mm

日付の月の値を指定します。指定できる値の範囲は、1月を1として01～12です。範囲外の値

を指定するとエラーになります。

dd

日付の日にちの値を指定します。指定できる値の範囲は 01 ~ 31 です。範囲外の値を指定するとエラーになります。

hh

時間の時の単位の値を指定します。指定できる値の範囲は 00 ~ 23 です。範囲外の値を指定するとエラーになります。

mm

時間の分の単位の値を指定します。指定できる値の範囲は 00 ~ 59 です。範囲外の値を指定するとエラーになります。

ss

時間の秒の単位の値を指定します。指定できる値の範囲は 00 ~ 59 です。範囲外の値を指定するとエラーになります。

sss

時間のミリ秒の単位の値を指定します。指定できる値の範囲は 000 ~ 999 です。範囲外の値を指定するとエラーになります。

このオプションは、次に示す事項に留意して指定してください。

- 開始時刻は次に示す記号で囲んで指定します。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- 指定を省略した場合、集計対象は最も古いトレースログエントリからになります。
- 指定した時刻が -e オプションで指定した終了時刻以降の場合、エラーになります。
- 年 / 月 / 日および時 : 分は省略できません。秒 . ミリ秒は省略できますが、秒だけやミリ秒だけは省略できません。秒 . ミリ秒を省略した場合、デフォルト値として 00.000 が仮定されます。

-e '終了時刻'

ユーザプログラムの処理内容を集計する期間の終了時刻を指定します。指定した時刻より前のトレースログエントリが集計対象になります。指定した時刻は、集計期間の対象になりません。例えば、12:45:00.000 と指定した場合、12:44:59.999 までが集計期間になります。

終了時刻の指定形式は -s オプションの開始時刻と同じです。-s オプションの開始時刻の説明を参照してください。

このオプションは、次に示す事項に留意して指定してください。

- 終了時刻は次に示す記号で囲んで指定します。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- 指定を省略した場合、集計対象は最も新しいトレースログエントリまでになります。
- 指定した時刻が -s オプションで指定した開始時刻以前の場合、エラーになります。
- 指定した時刻を満たさないトレースログエントリは、集計対象から外されます。

例えば、トレースファイル中の最初のトレースログエントリの時刻が 12:00:00.000 で、最後のトレースログエントリの時刻が 13:00:00.000 だとします。このとき、-e オプションに 12:45:00.000 を指定して EDMRptTrace コマンドを実行した場合、12:00:00.000 から 12:44:59.999 までのトレースログエントリは集計・出力対象になります。しかし、12:45:00.000 から 13:00:00.000 までのトレースログエントリは集計・出力対象になりません。

-d 集計間隔

ユーザプログラムの処理内容を集計する時間の幅を、分単位の整数で指定します。指定できる値の範囲は 1 ~ 43200 です。例えば、-d 15 と指定すると、15 分間ごとのユーザプログラムの処理内容が集

計されます。

このオプションは、次に示す事項に留意して指定してください。

- 整数以外を指定した場合、および 0 以下の値を指定した場合はエラーになります。
- 集計・出力対象のトレースログエントリは、`-s` オプション、`-e` オプションとの組み合わせによって変わります。

例えば、トレースファイル中の最初のトレースログエントリの時刻が 12:00:00.000 で、最後のトレースログエントリの時刻が 13:00:00.000 だとします。

このとき、`-d` オプションに 45、`-e` オプションに 12:45:00.000 を指定して `EDMRptTrace` コマンドを実行した場合、12:00:00.000 から 12:44:59.999 までのトレースログエントリが集計・出力対象になります。しかし、12:45:00.000 から 13:00:00.000 までのトレースログエントリは集計・出力対象になりません。

また、このとき、`-d` オプションに 45 を指定し、`-e` オプションを省略して `EDMRptTrace` コマンドを実行した場合、12:00:00.000 から 12:44:59.999 までのトレースログエントリが最初のインターバルの集計・出力対象になり、12:45:00.000 から 13:00:00.000 までのトレースログエントリは次のインターバルの集計・出力対象になります。

-w

トレースファイルの内容を分割して出力する場合に指定します。

サーバトレース解析の場合はユーザプログラムのプログラム名とユーザプログラムのプロセス ID ごとに、クライアントトレース解析の場合はユーザプログラムのスレッド ID ごとに、トレースファイルの内容を分割して別々のファイルに出力します。その際、出力されるすべてのエントリは、分類項目の `tid` でソートされます。同じ `tid` のエントリが複数ある場合は、その中で `date` の昇順にソートされます。出力先のディレクトリとファイル名を次に示します。

ディレクトリ

- UNIX の場合
DocumentBroker 実行環境ディレクトリ /aru/output
- Windows の場合
DocumentBroker インストールディレクトリ %Server%\aru\output

ファイル名

- サーバトレース解析の場合
" ユーザプログラムのプログラム名 "_" ユーザプログラムのプロセス ID".txt
- クライアントトレース解析の場合
" ユーザプログラムのスレッド ID".txt

なお、このオプションの対象エントリは、`enter` エントリと `msg` エントリです。トレースファイル分割機能の詳細については、「6.3.2(2) トレースファイル分割機能の処理方式」を参照してください。トレースファイル分割の分類項目については、「6.3.3(2) トレースファイル分割機能の出力形式」を参照してください。

-t

ユーザプログラムの処理内容の `SUBTOTAL` (小計) と `TOTAL` (総計) を出力する場合に指定します。

サーバトレース解析の場合は、ユーザプログラムのプログラム名ごとに `SUBTOTAL` と `TOTAL` が出力されます。クライアントトレース解析の場合は、ユーザプログラムのプロセス ID ごとに `SUBTOTAL` と `TOTAL` が出力されます。

なお、このオプションの対象エントリは、`enter` エントリです。

-v

結果を CSV 形式で出力する場合に指定します。

-vt

トレース表を出力する場合に指定します。トレース表については、「6.3.2(1) 集計機能の処理方式 (EDMRptTrace コマンドの集計方法)」を参照してください。なお、このオプションの対象エントリは、enter エントリと msg エントリです。また、ユーザプログラム情報のないサーバトレースの enter エントリも、トレース表に出力されます。

トレース表は、-v オプションを指定しなくても、CSV 形式で出力されます。

-o 出力ファイル名

結果をファイルに出力する場合に、出力するファイル名を指定します。ファイルパスは、フルパス、相対パスまたはファイル名だけのどの方法でも指定できます。ファイル名だけを指定した場合、カレントディレクトリに出力されます。

指定を省略する場合、結果は標準出力 (stdout) に出力されます。

-x 抽出エントリ定義ファイル名

エントリを抽出する場合に、抽出するエントリの処理の種別を指定した抽出エントリ定義ファイルのファイル名を指定します。抽出エントリ定義ファイルについては、「4.1.5 抽出エントリ定義ファイル」を参照してください。

このオプションは、次に示す事項に留意して指定してください。

- ファイルパスは、フルパスまたは相対パスのどちらでも指定できます。フルパスで指定されていない場合は、カレントディレクトリ下を探します。
- ハイフン (-) で始まるファイル名は指定できません。ただし、フルパスで指定された場合は処理できます。
- ファイル名に空白が含まれている場合、次に示す記号でファイル名全体を囲めば指定できます。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- 指定されたファイルがない場合や指定されたファイルがディレクトリの場合、エラーになります。

(4) コマンド引数

トレースファイル名

解析対象のトレースファイルを指定します。

ディレクトリ名

解析対象のディレクトリを指定します。ただし、サブディレクトリ下のファイルは解析対象に含まれません。ディレクトリパスは、1023 バイト以内で指定してください。指定したディレクトリの下にファイルがある場合は、そのファイル名も含めたパスが、1023 バイト以内になるようにしてください。

コマンド引数は、次に示す事項に留意して指定してください。

- ファイルパスおよびディレクトリパスはフルパスまたは相対パスのどちらでも指定できます。フルパスで指定されていない場合は、カレントディレクトリ下を探します。
- ハイフン (-) で始まるファイル名およびディレクトリ名は指定できません。ただし、フルパスで指定された場合は処理できます。
- ファイル名およびディレクトリ名に空白が含まれている場合、次に示す記号でファイル名全体を囲めば指定できます。
 - UNIX の場合：シングルクォーテーション (') またはダブルクォーテーション (")
 - Windows の場合：ダブルクォーテーション (")
- ファイル名およびディレクトリ名の最後のパスの区切り (UNIX の場合「/」、Windows の場合

「¥」)がある場合、標準エラー出力 (stderr) にメッセージが出力されます。

- 次の場合、標準エラー出力 (stderr) にメッセージが出力されます。
 - 指定したファイルまたはディレクトリがない場合
 - 指定したディレクトリが一つで、そのディレクトリに処理対象のファイルがない場合
 - 指定したディレクトリが複数で、すべてのディレクトリに処理対象のファイルがない場合
- トレースファイルを指定する場合、トレースファイルを 160 個まで指定できます。なお、複数のファイルを指定した場合、その中にアクセスできないファイルが一つでもあるときは、処理を中止してメッセージを出力します。また、同じファイルを重複して指定しても、そのファイルに対する処理は 1 回だけになります。
- ディレクトリを指定する場合、複数のディレクトリを指定できます。なお、指定したディレクトリ下のファイル総数が 160 を超えた場合、および指定したディレクトリ下にアクセスできないファイルが一つでもある場合は、処理を中止してメッセージを出力します。また、同じディレクトリを重複して指定しても、そのディレクトリ下のファイルに対する処理は 1 回だけになります。
- トレースファイルは、ASCII コードと Shift-JIS コードで構成されています。ただし、enter エン트리および msg エントリは ASCII コードで構成されます。
- この指定を省略する場合、サーバトレース解析では、EDMRptTrace コマンドのサーバトレース格納ディレクトリ (UNIX の場合「DocumentBroker 実行環境ディレクトリ /aru/input/spool/server」、Windows の場合「DocumentBroker インストールディレクトリ ¥Server¥aru¥input¥spool¥server」)にある「EDMRasTrace"PID"_"NO".log」("PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号) という名称のファイルが対象になります。このとき、処理できるファイルの数は 160 個ではなく、上記のファイル名に該当するファイルの数になります。
- この指定を省略する場合、クライアントトレース解析では、EDMRptTrace コマンドのクライアントトレース格納ディレクトリ (UNIX の場合「DocumentBroker 実行環境ディレクトリ /aru/input/spool/client」、Windows の場合「DocumentBroker インストールディレクトリ ¥Server¥aru¥input¥spool¥client」)にある「EDMRasTraceCL"PID"_"NO".log」("PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号) という名称のファイルが対象になります。このとき、処理できるファイルの数は 160 個ではなく、上記のファイル名に該当するファイルの数になります。
- EDMRptTrace コマンドは、DocumentBroker がトレースファイルを生成するディレクトリにあるファイルを対象にして実行できません。EDMRptTrace コマンドを実行する前に、DocumentBroker が生成したサーバトレースおよびクライアントトレースを、EDMRptTrace コマンドのサーバトレースまたはクライアントトレース格納ディレクトリにコピーして、コマンド引数でのトレースファイルの指定を省略することをお勧めします。トレースファイルのコピーについては、「4.2.1 トレースファイル」を参照してください。

(5) 注意事項

EDMRptTrace コマンドに関するそのほかの注意事項を次に示します。

- オプションの指定順序に規則はありません。どのような順序でも指定できます。
- オプションには、指定できない組み合わせがあります。オプションの指定の組み合わせを次の表に示します。

表 6-8 オプションの指定の組み合わせ (EDMRptTrace コマンド)

指定するオプション	組み合わせるオプション										
	-sv	-cl	-s	-e	-d	-w	-t	-v	-vt	-o	-x
-sv	x	x									

-cl	x	x									
-s			x								
-e				x							
-d					x	x			x		x
-w					x	x	x		x	x	x
-t						x	x		x		x
-v								x			
-vt					x	x	x		x		x
-o						x				x	
-x					x	x	x		x		x

(凡例)

空欄：指定できる組み合わせであることを示す。

x：指定できない組み合わせであることを示す。

- 機能ごとに処理対象のエントリが異なります。
機能と処理対象のエントリを次の表に示します。

表 6-9 機能と処理対象のエントリ

エントリ名	集計機能	トレースファイル分割機能	エントリ抽出機能
enter エントリ			
arg エントリ	-	-	
call エントリ	-	-	
return エントリ	-	-	
error エントリ	-	-	
warn エントリ	-	-	
mark エントリ	-	-	
msg エントリ			
exit エントリ	-	-	

(凡例)

：処理対象になることを示す。

-：処理対象にならないことを示す。

注

-t オプションを指定する場合、msg エントリは処理対象になりません。enter エントリだけが処理対象になります。

6.3.2 EDMRptTrace コマンドの機能の処理方式

EDMRptTrace コマンドの次に示す機能の処理方式について説明します。

- 集計機能
- トレースファイル分割機能
- エントリ抽出機能

(1) 集計機能の処理方式 (EDMRptTrace コマンドの集計方法)

EDMRptTrace コマンドの集計機能では、まず、トレースファイル中のトレースログを解析して、トレースログエントリごとにトレース表を作成します。その上で、サーバおよびクライアントのプログラムによる処理内容を、ユーザプログラムのプログラム名、プロセス ID およびスレッド ID で分類・集計して、結果を標準出力 (stdout) またはファイルに出力します。

トレース表の項目を次の表に示します。

表 6-10 トレース表の項目

項目名	説明
日付	サーバまたはクライアントのプログラムが処理を実行した日付 (年 / 月 / 日)。
時刻	サーバまたはクライアントのプログラムが処理を実行した時刻 (時 : 分 : 秒 . ミリ秒)。
プロセス ID	解析対象によって、プロセス ID が異なります。 サーバトレース解析の場合 サーバプロセスのプロセス ID クライアントトレース解析の場合 ユーザプログラムのプロセス ID
スレッド ID	解析対象によって、スレッド ID が異なります。 サーバトレース解析の場合 サーバプロセスのスレッド ID クライアントトレース解析の場合 ユーザプログラムのスレッド ID
処理の種別	enter または msg。enter または msg については、「表 4-4 message に出力される処理の種別と処理の内容」を参照してください。
ユーザプログラムのホスト名	トレースログに含まれるユーザプログラム情報の項目。クライアントトレースにはありません。
ユーザプログラムのプログラム名	
ユーザプログラムのプロセス ID	
ユーザプログラムのセッション ID	
プログラム識別子	
処理内容またはメッセージ	処理の種別によって異なります。 enter の場合 クラス名 :: メソッド名 (または関数名) msg の場合 メッセージ (SQL 文など)

トレース表の例を表 6-11 および表 6-12 に示します。なお、実際には、表 6-11 の右側に表 6-12 が続く形式の一つのトレース表として作成されます。クライアントトレースの場合、ユーザプログラム情報の項目を除いた形式のトレース表になります。

表 6-11 トレース表の例 (1/2)

日付	時刻	プロセス ID	スレッド ID	処理の種類	ユーザプログラムのホスト名	ユーザプログラムのプログラム名
2000/06/19	02:18:19.388	00002541	00005488	enter	ht01	UAP01
2000/06/19	02:18:52.261	00002541	00005488	enter	ht01	UAP01
2000/06/19	02:19:16.420	00002541	00005488	msg	-	-
2000/06/19	02:19:16.420	00002541	00005488	enter	ht01	UAP01

(凡例)

- : 該当する内容がないことを示す。

注

ユーザプログラム情報の項目です。クライアントトレースのトレース表にはありません。

表 6-12 トレース表の例 (2/2)

ユーザプログラムのプロセス ID	ユーザプログラムのセッション ID	プログラム識別子	処理内容またはメッセージ
14455	545421asd(usr)	EDMS2225	CedmRasMsg::cs_setMsgFormatter()
14455	545421asd(usr)	EDMS2225	CedmRasMsg::cs_setMsgFormatter()
-	-	-	SELECT "dmaProp_OIID" FROM "mdmClass_CfgH"
14455	545421asd(usr)	EDMS2225	Edm_sys_open_directory()

(凡例)

- : 該当する内容がないことを示す。

注

ユーザプログラム情報の項目です。クライアントトレースのトレース表にはありません。

なお、トレース表は、EDMRptTrace コマンド実行時に -vt オプションを指定することによって出力できます。

(2) トレースファイル分割機能の処理方式

トレースファイルには、複数のユーザプログラム情報が含まれています。EDMRptTrace コマンドのトレースファイル分割機能では、トレースファイルの内容を分割して別々のファイルに出力します。サーバトレース解析の場合はユーザプログラムのプログラム名とユーザプログラムのプロセス ID ごとに、クライアントトレース解析の場合はユーザプログラムのスレッド ID ごとに分割して出力します。その際、出力されるすべてのエントリは、出力結果の分類項目である tid でソートされます。同じ tid のエントリが複数ある場合は、その中で date の昇順にソートされます。出力結果の分類項目の tid および date については、「6.3.3(2) トレースファイル分割機能の出力形式」を参照してください。

トレースファイル分割の概念を図 6-6 および図 6-7 に示します。

図 6-6 トレースファイル分割の概念 (サーバトレース解析の場合)

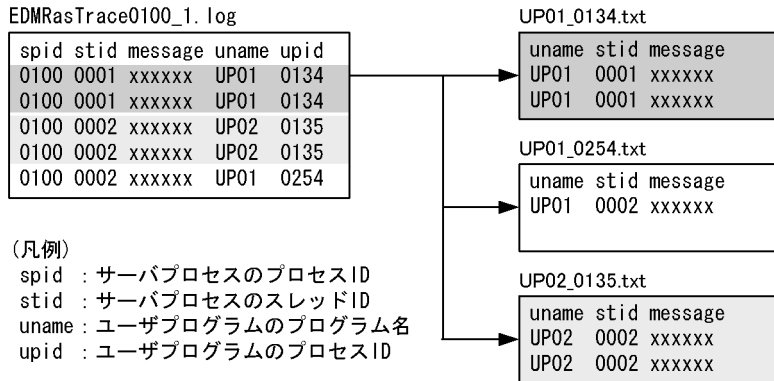
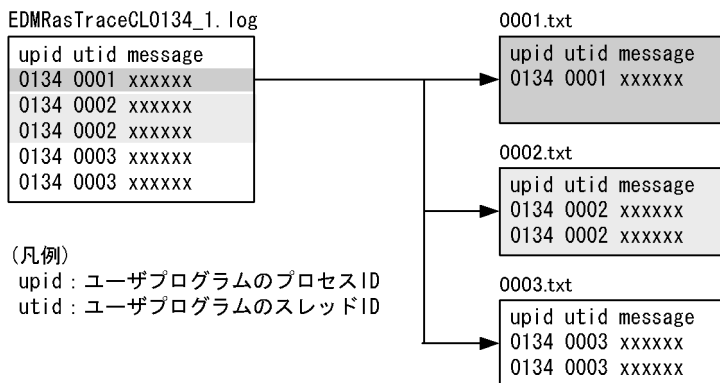


図 6-7 トレースファイル分割の概念 (クライアントトレース解析の場合)



トレースファイル分割機能で出力されるファイルの出力先のディレクトリとファイル名を次に示します。

ディレクトリ

UNIX の場合

DocumentBroker 実行環境ディレクトリ /aru/output

Windows の場合

DocumentBroker インストールディレクトリ ¥Server¥aru¥output

ファイル名

サーバトレース解析の場合

"ユーザプログラムのプログラム名 "_" ユーザプログラムのプロセス ID".txt

クライアントトレース解析の場合

"ユーザプログラムのスレッド ID".txt

ファイル名の規則を次に示します。

- すでに同名のファイルがある場合、そのファイル名に "_n" (n は通し番号。最大値は 15) を付加した形式で出力されます。
- "_n" を付加した形式のファイルがある場合、最終更新ファイルの n に 1 を足したファイルが出力または更新されます。なお、n が 15 のファイルが最終更新ファイルの場合、"_n" が付いてないファイルが更新されます。

例えば、n が 5 のファイルが最終更新ファイルの場合、n が 6 のファイルが更新されます。また、n が 9 のファイルしかない場合、n が 10 のファイルが出力されます。

ファイル名の例を次に示します。

サーバトレース解析の場合のファイル名の例

例 1：同名のファイルがない場合

「UP01_0134.txt」というファイルが出力されます。

例 2：同名のファイル「UP01_0134.txt」がある場合

「UP01_0134_1.txt」というファイルが出力されます。

例 3：ファイル「UP01_0134_5.txt」が最終更新ファイルの場合

「UP01_0134_6.txt」というファイルが出力されます。

例 4：ファイル「UP01_0134_9.txt」しかない場合

「UP01_0134_10.txt」というファイルが出力されます。

クライアントトレース解析の場合のファイル名の例

例 1：同名のファイルがない場合

「0001.txt」というファイルが出力されます。

例 2：同名のファイル「0001.txt」がある場合

「0001_1.txt」というファイルが出力されます。

例 3：ファイル「0001_5.txt」が最終更新ファイルの場合

「0001_6.txt」というファイルが出力されます。

例 4：ファイル「0001_9.txt」しかない場合

「0001_10.txt」というファイルが出力されます。

(3) エントリ抽出機能の処理方式

トレースログエントリには、プログラムの処理内容を表す処理の種別が含まれています。EDMRptTrace コマンドのエントリ抽出機能では、抽出エントリ定義ファイルに指定された処理の種別によって、トレースログエントリを抽出して、結果を標準出力 (stdout) またはファイルに出力します。

6.3.3 EDMRptTrace コマンドの出力形式

EDMRptTrace コマンド実行結果の出力形式について説明します。

(1) 集計機能の出力形式

EDMRptTrace コマンドの集計機能の出力例を次の図に示します。

図 6-8 EDMRptTrace コマンドの集計機能の出力例

program name	pid	message	times
# EDMRptTrace -sv -t # COUNT START = 2000/06/19 15:00:00.000 # COUNT END = 2000/06/19 18:00:00.001 # LOG START = 2000/06/19 15:00:00.000 # LOG END = 2000/06/19 18:00:00.000			
prog01	00005124	CdbrSession::Commit ()	2
		CdbrSession::Begin ()	3
		CdbrEqIStatement::GetResult ()	5
	00005132	CdbrSession::Commit ()	21
		CdbrSession::Begin ()	32
		CdbrEqIStatement::Execute ()	18
		CdbrEqIStatement::GetResult ()	18
SUBTOTAL		CdbrSession::Commit ()	23
		CdbrSession::Begin ()	35
		CdbrEqIStatement::Execute ()	18
		CdbrEqIStatement::GetResult ()	23
prog02	00009588	CdbrSession::Commit ()	7
		CdbrSession::Begin ()	8
		CdbrEqIStatement::GetResult ()	4
SUBTOTAL		CdbrSession::Commit ()	7
		CdbrSession::Begin ()	8
		CdbrEqIStatement::GetResult ()	4
TOTAL		CdbrSession::Commit ()	30
		CdbrSession::Begin ()	43
		CdbrEqIStatement::Execute ()	18
		CdbrEqIStatement::GetResult ()	27

図 6-8 の要素 (a) ~ (e) について説明します。

(a) 共通情報

集計結果に関する共通情報が出力されます。COUNT START から INTERVAL までの共通情報の出力形式を次に示します。

< 共通情報名 > = < 値 >

出力される共通情報の内容は、EDMRptTrace コマンドのオプションの指定によって変わります。共通情報の内容およびオプションの指定との関係を次に示します。

EDMRptTrace < オプション >< 引数 >

実行したコマンドラインが出力されます。

COUNT START = yyyy/mm/dd hh:mm:ss.sss

開始エントリの時刻が出力されます。-s オプションで開始時刻を指定している場合、その指定値が出力されます。-s オプションで開始時刻を指定していない場合、ログ開始時刻 (LOG START) が出力されます。

COUNT START に該当するエントリは、集計対象になります。

COUNT END = yyyy/mm/dd hh:mm:ss.sss

終了エントリの時刻が出力されます。-e オプションで終了時刻を指定している場合、その指定値が出力されます。-e オプションで終了時刻を指定していない場合、ログ終了時刻 (LOG END) に 1 ミリ秒を足した時刻が出力されます。

COUNT END に該当するエントリは、集計対象になりません。

LOG START = yyyy/mm/dd hh:mm:ss.sss

ログ開始時刻（ログ内の最も古いトレースログエントリの時刻）が出力されます。

LOG END = yyyy/mm/dd hh:mm:ss.sss

ログ終了時刻（ログ内の最も新しいトレースログエントリの時刻）が出力されます。

INTERVAL = *mm* minutes

-d オプションの指定値（ユーザプログラムの処理内容の集計間隔）が出力されます。-d オプションを指定していない場合は出力されません。

(b) 分類項目

分類項目が出力され、その下に各項目の内容が出力されます。分類項目とその内容は、サーバトレース解析（-sv オプション指定）かクライアントトレース解析（-cl オプション指定）かによって、次に示す違いがあります。

サーバトレース解析の場合の分類項目

program name

ユーザプログラムのプログラム名が 14 バイトまで出力されます。14 バイトを超える場合、15 バイト以降は出力されません。CSV 形式では 15 バイト以降も出力されます。

program name はトレースログに出現した順に出力されます。

pid

ユーザプログラムのプロセス ID が 8 バイトで出力されます。

pid は昇順にソートされて出力されます。

クライアントトレース解析の場合の分類項目

pid

ユーザプログラムのプロセス ID が 8 バイトで出力されます。

pid はトレースログに出現した順に出力されます。

tid

ユーザプログラムのスレッド ID が 8 バイトで出力されます。

tid は昇順にソートされて出力されます。

(c) 集計結果

集計結果の項目が出力され、その下に各項目の内容が出力されます。集計結果の項目とその内容を次に示します。

message

ユーザプログラムの処理内容が、47 バイトまで出力されます。47 バイトを超える場合、48 バイト以降は出力されません。CSV 形式では 48 バイト以降も出力されます。

message はトレースログに出現した順に出力されます。

なお、ユーザプログラムの処理内容とは、トレースログ中の message という項目に含まれる、実行されたメソッド名または関数名です。ユーザプログラムの処理内容は次に示す形式で出力されます。

クラス名::メソッド名（または関数名）

トレースログ中の message については、「4.2.1(2)(b) トレースログの項目」を参照してください。

6. コマンドリファレンス

times

メソッドまたは関数の実行回数を示す数字が、7 バイトまで出力されます。7 バイトを超える場合、8 バイト以降は出力されません。CSV 形式では 8 バイト以降も出力されます。

(d) SUBTOTAL

-t オプションを指定した場合、サーバトレース解析では分類項目の program name ごとに、クライアントトレース解析では分類項目の pid ごとに、メソッドまたは関数の実行回数の小計が出力されます。

(e) TOTAL

-t オプションを指定した場合、メソッドまたは関数の実行回数の総計が出力されます。

(2) トレースファイル分割機能の出力形式

EDMRptTrace コマンドのトレースファイル分割機能の出力例を次の図に示します。なお、次の図では 1 エントリは複数行にわたって見えますが、実際には 1 エントリは 1 行で出力されます。

図 6-9 EDMRptTrace コマンドのトレースファイル分割機能の出力例

date	tid	message
<pre># EDMRptTrace -sv -w # COUNT START = 2000/06/05 12:00:00.000 # COUNT END = 2000/06/05 18:00:00.001 # LOG START = 2000/06/05 12:00:00.000 # LOG END = 2000/06/05 18:00:00.000</pre>		
2000/06/05 12:27:48.506	00000001	enter CedmRasMsg::cs_setMsgFormatter() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()
2000/06/05 12:27:48.506	00000001	enter CedmRasMsg::cs_setMsgFormatter() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()
2000/06/05 12:27:48.507	00000002	enter CedmCmdClassDescriptionForMulti::Initialize()
2000/06/05 12:27:48.507	00000002	msg SELECT P9FFOR71000FC9972P521A2HD56, P9FFOR71000GC99727V7140CF56, P9FFOR71000FC9972VMO1ESQF56, P9FFOR71000GC9972E061CD0LM5 FROM T9FFOR71000FC99720521IUPU96 WHERE P129HE1OK08PB0872DLUUMRGB4='00000000ABC42239'
2000/06/05 12:27:48.507	00000002	enter Edm_sys_open_directory()
2000/06/05 12:27:48.514	00000003	enter Edm_sys_open_directory()
2000/06/05 12:27:48.514	00000003	enter CdbrSession::Commit() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()
2000/06/05 12:27:48.514	00000003	enter CdbrSession::Begin() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()
2000/06/05 12:27:48.514	00000003	enter CdbrSession::ExcuteSearch() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()
2000/06/05 12:27:48.515	00000003	enter CdbrSession::GetSearchResult() sis01 WETG2JO 12491 000052823906310000040a0d(userdoc) EDMS000021122 ()

図 6-9 の要素 (a), (b) について説明します。

(a) 共通情報

集計結果に関する共通情報が出力されます。COUNT START から LOG END までの共通情報の出力形式を次に示します。

```
# < 共通情報名 > = < 値 >
```

出力される共通情報の内容は、EDMRptTrace コマンドのオプションの指定によって変わります。共通情報の内容およびオプションの指定との関係を次に示します。

```
# EDMRptTrace < オプション > < 引数 >
```

実行したコマンドラインが出力されます。

COUNT START = yyyy/mm/dd hh:mm:ss.sss

開始エントリの時刻が出力されます。-s オプションで開始時刻を指定している場合、その指定値が出力されます。-s オプションで開始時刻を指定していない場合、ログ開始時刻 (LOG START) が出力されます。

COUNT START に該当するエントリは、集計対象になります。

COUNT END = yyyy/mm/dd hh:mm:ss.sss

終了エントリの時刻が出力されます。-e オプションで終了時刻を指定している場合、その指定値が出力されます。-e オプションで終了時刻を指定していない場合、ログ終了時刻 (LOG END) に 1 ミリ秒を足した時刻が出力されます。

COUNT END に該当するエントリは、集計対象になりません。

LOG START = yyyy/mm/dd hh:mm:ss.sss

ログ開始時刻 (ログ内の最も古いトレースログエントリの時刻) が出力されます。

LOG END = yyyy/mm/dd hh:mm:ss.sss

ログ終了時刻 (ログ内の最も新しいトレースログエントリの時刻) が出力されます。

(b) 分類項目

分類項目が出力され、その下に各項目の内容が出力されます。分類項目とその内容を次に示します。

date

トレースログの出力日付と時間が 23 バイトで出力されます。

同じ tid のエントリが複数ある場合は、その中で date の昇順にソートされて出力されます。

tid

スレッド ID が 8 バイトで出力されます。ただし、サーバトレース解析 (-sv オプション指定) かクライアントトレース解析 (-cl オプション指定) によって、次に示す違いがあります。

サーバトレース解析の場合

サーバプロセスのスレッド ID

クライアントトレース解析の場合

ユーザプログラムのスレッド ID

tid は昇順にソートされて出力されます。

message

トレースログの message という項目の内容がそのまま出力されます。トレースログの message という項目については、「4.2.1(2)(b) トレースログの項目」を参照してください。

(3) エントリ抽出機能の出力形式

-x オプションを指定した場合、抽出エントリ定義ファイルに指定した処理の種別によって、トレースログエントリが抽出されて出力されます。

エントリ抽出機能の出力例を次の図に示します。この例では、抽出エントリ定義ファイルに処理の種別「msg」を指定して、msg エントリを抽出しています。

図 6-10 EDMRptTrace コマンドのエントリ抽出機能の出力例

```
# EDMRptAclog -x extract.def Input.log
# COUNT START = 2000/07/05 14:27:04.727
# COUNT END   = 2000/07/05 14:27:04.734
# LOG START   = 2000/07/05 14:27:04.727
# LOG END     = 2000/07/05 14:27:04.733

0003 2000/07/05 14:27:04.731    DocumentBroker    00006A41 00000001    msg msg_SELECT
```

6.3.4 EDMRptTrace コマンドの出力例

EDMRptTrace コマンド実行結果の出力例を次に示します。

(1) サーバトレースの集計結果の出力例 (-sv 指定)

サーバトレースを解析して、ユーザプログラムの処理内容の集計結果を出力した例を次に示します。

```
# EDMRptTrace -sv
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

program name  pid      message                                     times
-----
prog01        00005124 CdbSession::Commit()                       2
              00005124 CdbSession::Begin()                        3
              00005124 CdbSession::GetSearchResult()             5
-----
              00005132 CdbSession::Commit()                       21
              00005132 CdbSession::Begin()                        32
              00005132 CdbEqlStatement::Execute()                18
              00005132 CdbSession::GetSearchResult()             18
-----
prog02        00009588 CdbSession::Commit()                       7
              00009588 CdbSession::Begin()                        8
              00009588 CdbSession::GetSearchResult()             4
-----
```

(2) サーバトレースの集計結果の CSV 形式での出力例 (-sv -v 指定)

サーバトレースを解析して、ユーザプログラムの処理内容の集計結果を CSV 形式で出力した例を示します。なお、CSV 形式での出力では、分類項目の内容（この例ではユーザプログラムのプログラム名とユーザプログラムのプロセス ID）は、集計結果の行数分、繰り返し出力されます。

```
"#", "EDMRptTrace -sv -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"program name", "pid", "message", "times"
"prog01", "00005124", "CdbSession::Commit()", 2
"prog01", "00005124", "CdbSession::Begin()", 3
"prog01", "00005124", "CdbSession::GetSearchResult()", 5
"prog01", "00005132", "CdbSession::Commit()", 21
"prog01", "00005132", "CdbSession::Begin()", 32
"prog01", "00005132", "CdbEqlStatement::Execute()", 18
"prog01", "00005132", "CdbSession::GetSearchResult()", 18
"prog02", "00009588", "CdbSession::Commit()", 7
"prog02", "00009588", "CdbSession::Begin()", 8
"prog02", "00009588", "CdbSession::GetSearchResult()", 4
```

(3) クライアントトレースの集計結果の出力例 (-cl 指定)

クライアントトレースを解析して、ユーザプログラムの処理内容の集計結果を出力した例を次に示します。

```
# EDMRptTrace -cl
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000

pid      tid      message                                     times
-----
00002365 00005124 CdbSession::Commit()                      2
          CdbSession::Begin()                  3
          CdbSession::GetSearchResult()       5
-----
          00005132 CdbSession::Commit()                      21
          CdbSession::Begin()                  32
          CdbEqlStatement::Execute()         18
          CdbSession::GetSearchResult()       18
-----
00007845 00009588 CdbSession::Commit()                      7
          CdbSession::Begin()                  8
          CdbSession::GetSearchResult()       4
-----
```

(4) クライアントトレースの集計結果の CSV 形式での出力例 (-cl -v 指定)

クライアントトレースを解析して、ユーザプログラムの処理内容の集計結果を CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、分類項目の内容（この例ではユーザプログラムのプロセス ID とユーザプログラムのスレッド ID）は、集計結果の行数分、繰り返し出力されます。

```
"#", "EDMRptTrace -cl -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"pid", "tid", "message", "times"
"00002365", "00005124", "CdbSession::Commit()", 2
"00002365", "00005124", "CdbSession::Begin()", 3
"00002365", "00005124", "CdbSession::GetSearchResult()", 5
"00002365", "00005132", "CdbSession::Commit()", 21
"00002365", "00005132", "CdbSession::Begin()", 32
"00002365", "00005132", "CdbEqlStatement::Execute()", 18
"00002365", "00005132", "CdbSession::GetSearchResult()", 18
"00007845", "00009588", "CdbSession::Commit()", 7
"00007845", "00009588", "CdbSession::Begin()", 8
"00007845", "00009588", "CdbSession::GetSearchResult()", 4
```

(5) エントリの抽出例 (-sv -x 抽出エントリ定義ファイル 指定)

この例では、-x オプションで、次の抽出エントリ定義ファイルを指定したと仮定します。

```
[Extract Trace]
msg
```

また、この例では、コマンド引数で、次のサーバトレースを指定したと仮定します。

```
Input_trace.log:
0000 2000/07/05 14:27:04.727 DocumentBroker 00006A41 00000001 enter abc_enter()
0001 2000/07/05 14:27:04.728 DocumentBroker 00006A41 00000001 exit abc_exit()
0002 2000/07/05 14:27:04.730 DocumentBroker 00006A41 00000001 arg argc_9
0003 2000/07/05 14:27:04.731 DocumentBroker 00006A41 00000001 msg msg_SELECT
0004 2000/07/05 14:27:04.732 DocumentBroker 00006A41 00000001 call abc_call()
0005 2000/07/05 14:27:04.733 DocumentBroker 00006A41 00000001 return abc_return()
```

サーバトレースから msg エントリを抽出して出力した例を次に示します。

6. コマンドリファレンス

```
# EDMRptTrace -sv -x extract.def Input_trace.log
# COUNT START = 2000/07/05 14:27:04.727
# COUNT END   = 2000/07/05 14:27:04.734
# LOG START   = 2000/07/05 14:27:04.727
# LOG END     = 2000/07/05 14:27:04.733

0003 2000/07/05 14:27:04.731 DocumentBroker 00006A41 00000001 msg msg_SELECT
```

(6) トレースファイル分割の出力例 (-cl -w 指定)

クライアントトレースを分割して出力した例を次に示します。

```
# EDMRptTrace -cl -w
# COUNT START = 2000/07/24 19:00:00.000
# COUNT END   = 2000/07/24 19:00:00.028
# LOG START   = 2000/07/24 19:00:00.000
# LOG END     = 2000/07/24 19:00:00.027

date            tid            message
-----
2000/07/24 19:00:00.000 00000001 enter CdbrEqlStatement::CdbrEqlStatement()
2000/07/24 19:00:00.002 00000001 enter CdbrEqlStatement::Initialize()
2000/07/24 19:00:00.004 00000001 enter CdbrCoreImp::SetObjectID()
2000/07/24 19:00:00.006 00000001 enter CdbrEqlStatement::Set()
2000/07/24 19:00:00.008 00000001 enter CdbrEqlStatement::ChangeGetObjCount()
2000/07/24 19:00:00.010 00000001 enter CdbrEqlStatement::Execute()
2000/07/24 19:00:00.012 00000001 enter CdbrEqlStatement::GetResult()
2000/07/24 19:00:00.014 00000001 enter dbrDelete(SDBR_QUERYRESULT*)()
2000/07/24 19:00:00.015 00000001 enter CdbrEqlStatement::GetACLMode()
2000/07/24 19:00:00.017 00000001 enter CdbrEqlStatement::DeleteResult()
2000/07/24 19:00:00.019 00000001 enter CdbrEqlStatement::Terminate()
2000/07/24 19:00:00.021 00000001 enter CdbrCoreImp::ResetObjectID()
2000/07/24 19:00:00.023 00000001 enter CdbrEqlStatement::~CdbrEqlStatement()
2000/07/24 19:00:00.025 00000001 enter CdbrCoreImp::ResetObjectID()
2000/07/24 19:00:00.027 00000001 enter CdbrSession::Commit()
```

(7) トレースファイル分割の CSV 形式での出力例 (-sv -w -v 指定)

サーバトレースを分割して CSV 形式で出力した例を次に示します。

```
"#", "EDMRptTrace -sv -w -v"
"#", "COUNT START", "2000/06/05", "12:27:48.506"
"#", "COUNT END", "2000/06/05", "12:27:48.516"
"#", "LOG START", "2000/06/05", "12:27:48.506"
"#", "LOG END", "2000/06/05", "12:27:48.515"
"date", "time", "tid", "message"
"2000/06/05", "12:27:48.506", "00000001", "enter", "CedmRasMsg::cs_setMsgFormatter()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
"2000/06/05", "12:27:48.506", "00000001", "enter", "CedmRasMsg::cs_setMsgFormatter()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
"2000/06/05", "12:27:48.507", "00000002", "enter", "CedmCmdClassDescriptionForMulti::Initialize()"
"2000/06/05", "12:27:48.507", "00000002", "msg", "SELECT
P9FFOR7100OFC9972P521A2HD56,P9FFOR7100GC99727V7140CF56,P9FFOR7100OFC9972VM01ESQF56,P9FFOR7100GC9972E061CD0LM5 FROM T9FFOR7100OFC9972O521IUPU96 WHERE
PI29HEIOK08PBO872DLUUMRGB4='00000000ABC42239'"
"2000/06/05", "12:27:48.507", "00000002", "enter", "Edm_sys_open_directory()"
"2000/06/05", "12:27:48.514", "00000002", "enter", "Edm_sys_open_directory()"
"2000/06/05", "12:27:48.514", "00000003", "enter", "CdbrSession::Commit()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
"2000/06/05", "12:27:48.514", "00000003", "enter", "CdbrSession::Begin()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
"2000/06/05", "12:27:48.514", "00000003", "enter", "CdbrEqlStatement::Execute()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
"2000/06/05", "12:27:48.515", "00000003", "enter", "CdbrSession::GetSearchResult()", "sis01", "WETG2J0", "12491", "000052823906310000040a0d(userdoc)", "EDMS000021122", "()"
```


(8) SUBTOTAL, TOTAL の出力例 (-sv -t 指定)

SUBTOTAL および TOTAL を出力した例を次に示します。

```
# EDMRptTrace -sv -t
# COUNT START = 2000/06/19 15:00:00.000
# COUNT END   = 2000/06/19 18:00:00.001
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 18:00:00.000
```

program name	pid	message	times
prog01	00005124	CdbrSession::Commit()	2
		CdbrSession::Begin()	3
		CdbrSession::GetSearchResult()	5
SUBTOTAL		CdbrSession::Commit()	23
		CdbrSession::Begin()	35
		CdbrEqlStatement::Execute()	18
		CdbrSession::GetSearchResult()	23
prog02	00009588	CdbrSession::Commit()	7
		CdbrSession::Begin()	8
		CdbrSession::GetSearchResult()	4
SUBTOTAL		CdbrSession::Commit()	7
		CdbrSession::Begin()	8
		CdbrSession::GetSearchResult()	4
TOTAL		CdbrSession::Commit()	30
		CdbrSession::Begin()	43
		CdbrEqlStatement::Execute()	18
		CdbrSession::GetSearchResult()	27

(9) SUBTOTAL, TOTAL の CSV 形式での出力例 (-sv -t -v 指定)

SUBTOTAL および TOTAL を CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、分類項目の内容（この例ではユーザプログラムのプログラム名とユーザプログラムのプロセス ID）と -t オプションの SUBTOTAL および TOTAL は、集計結果の行数分、繰り返し出力されます。

```
"#", "EDMRptTrace -sv -t -v"
"#", "COUNT START", "2000/06/19", "15:00:00.000"
"#", "COUNT END", "2000/06/19", "18:00:00.001"
"#", "LOG START", "2000/06/19", "15:00:00.000"
"#", "LOG END", "2000/06/19", "18:00:00.000"
"program name", "pid", "message", "times"
"prog01", "00005124", "CdbrSession::Commit()", 2
"prog01", "00005124", "CdbrSession::Begin()", 3
"prog01", "00005124", "CdbrSession::GetSearchResult()", 5
"prog01", "00005132", "CdbrSession::Commit()", 21
"prog01", "00005132", "CdbrSession::Begin()", 32
"prog01", "00005132", "CdbrEqlStatement::Execute()", 18
"prog01", "00005132", "CdbrSession::GetSearchResult()", 18
"SUBTOTAL", "", "CdbrSession::Commit()", 23
"SUBTOTAL", "", "CdbrSession::Begin()", 35
"SUBTOTAL", "", "CdbrEqlStatement::Execute()", 18
"SUBTOTAL", "", "CdbrSession::GetSearchResult()", 23
"prog02", "00009588", "CdbrSession::Commit()", 7
"prog02", "00009588", "CdbrSession::Begin()", 8
"prog02", "00009588", "CdbrSession::GetSearchResult()", 4
"SUBTOTAL", "", "CdbrSession::Commit()", 7
"SUBTOTAL", "", "CdbrSession::Begin()", 8
"SUBTOTAL", "", "CdbrSession::GetSearchResult()", 4
"TOTAL", "", "CdbrSession::Commit()", 30
"TOTAL", "", "CdbrSession::Begin()", 43
"TOTAL", "", "CdbrEqlStatement::Execute()", 18
"TOTAL", "", "CdbrSession::GetSearchResult()", 27
```

(10) トレース表の出力例 (-sv -vt 指定)

サーバトレースのトレース表を出力した例を次に示します。トレース表は、-v オプションを指定しなくても、CSV 形式で出力されます。

```

"#", "EDMRptTrace -sv -vt"
"#", "COUNT START", " 2000/06/19", " 02:18:19.388"
"#", "COUNT END", " 2000/06/19", " 02:19:16.421"
"#", "LOG START", " 2000/06/19", " 02:18:19.388"
"#", "LOG END", " 2000/06/19", " 02:19:16.420"
"date", "time", "process ID", "thread ID", "classification", "host name", "program name", "process
ID", "session ID", "program ID", "message"
"2000/06/19", "02:18:19.388", "00004FDD", "0000186E", "msg", "", "", "", "", "", "UPDATE
"edmClass_VTCRelationship" SET "edmProp_VTMode"=2, "dmaProp_Head"=? WHERE
"dmaProp_OIID"='0000000000000F10' "
"2000/06/
19", "02:18:52.261", "00004FDD", "0000186E", "msg", "", "", "", "", "", "CdabSqlStringValue::DoSetPara
m, SetParam:0bc46760-366c-11d2-917c-0000e213036700000000000000F08"
"2000/06/
19", "02:18:59.392", "00005FDD", "0000186F", "enter", "maruko", "dbrsetmode", "11519", "00004fdd397e
5ef20009c3f7 (ACUser1)", "EDMS000020445", "CdbrDMA::ReleaseObject()"
"2000/06/
19", "02:19:08.015", "00004FDD", "0000186E", "enter", "maruko", "dbrsetmode", "11519", "00004fdd397e
5ef20009c3f7 (ACUser1)", "EDMS000020445", "CdbrSession::Commit()"
"2000/06/19", "02:19:16.420", "00004FDD", "0000186E", "msg", "", "", "", "", "", "[FPT02]
Session_Commit
( ObjectName=00004fdd397e5ef20009c3f7 )"

```

(11) 集計間隔を指定した出力例 (-sv -s '開始時刻' -e '終了時刻' -d 集計間隔 指定)

集計の開始時刻 (2000/06/19 14:30:00.000) と終了時刻 (2000/06/19 16:30:00.000) に加えて、集計間隔 (20) を指定してユーザプログラムの処理内容を集計して出力した例を次に示します。

```

# EDMRptTrace -sv -s '2000/06/19 14:30:00.000' -e '2000/06/19 16:30:00.000' -d 20
# COUNT START = 2000/06/19 14:30:00.000
# COUNT END   = 2000/06/19 16:30:00.000
# LOG START   = 2000/06/19 15:00:00.000
# LOG END     = 2000/06/19 16:00:00.000
# INTERVAL    = 20 minutes

# COUNT START = 2000/06/19 14:30:00.000

# No entry found.

# COUNT START = 2000/06/19 14:50:00.000

program name  pid      message                                     times
-----
prog01       00005124  CdbrSession::Commit()                      1
              CdbrSession::Begin()                       2
              CdbrSession::GetSearchResult()         3
-----

# COUNT START = 2000/06/19 15:10:00.000

program name  pid      message                                     times
-----
prog01       00005124  CdbrSession::Commit()                      4
              CdbrSession::Begin()                       5
              CdbrSession::GetSearchResult()         6
-----

# COUNT START = 2000/06/19 15:30:00.000

program name  pid      message                                     times
-----
prog01       00005124  CdbrSession::Commit()                      7
              CdbrSession::Begin()                       8
              CdbrSession::GetSearchResult()         9
-----

# COUNT START = 2000/06/19 15:50:00.000

program name  pid      message                                     times
-----
prog01       00005124  CdbrSession::Commit()                     10
              CdbrSession::Begin()                      11
              CdbrSession::GetSearchResult()   12
-----

# COUNT START = 2000/06/19 16:10:00.000

# No entry found.

```

この例では、集計対象となるのは、2000/06/19 14:30:00.000 から 2000/06/19 16:29:59.999 までのトレースログエントリです。2000/06/19 14:30:00.000 から 20 分間隔で集計しています。

ログ終了時刻 2000/06/19 16:00:00.000 を超えて 2000/06/19 16:10:00.000 に集計したときには、該当するエントリがない、つまりユーザプログラムによる処理がなかったため「No entry found.」という結果が出力されています。

(12) 集計間隔を指定した CSV 形式での出力例 (-sv -s '開始時刻' -e '終了時刻' -d 集計間隔 -v 指定)

集計の開始時刻 (2000/06/19 14:30:00.000) と終了時刻 (2000/06/19 16:30:00.000) に加えて、集計間隔 (20) を指定してユーザプログラムの処理内容を集計して CSV 形式で出力した例を次に示します。なお、CSV 形式での出力では、分類項目の内容 (この例ではユーザプログラムのプログラム名とユーザプログラムのプロセス ID) は、集計結果の行数分、繰り返し出力されます。

6. コマンドリファレンス

```
"#", "EDMRptTrace -sv -s '2000/06/19 14:30:00.000' -e '2000/06/19 16:30:00.000' -d 20 -v"  
"#", "COUNT START", "2000/06/19", "14:30:00.000"  
"#", "COUNT END", "2000/06/19", "16:30:00.000"  
"#", "LOG START", "2000/06/19", "15:00:00.000"  
"#", "LOG END", "2000/06/19", "16:00:00.000"  
"#", "INTERVAL", "20", "minutes"  
"#", "COUNT START", "2000/06/19", "14:30:00.000"  
"#", "No entry found."  
"#", "COUNT START", "2000/06/19", "14:50:00.000"  
"program name", "pid", "message", "times"  
"prog01", "00005124", "CdbrSession::Commit()", 1  
"CdbrSession::Begin()", 2  
"CdbrSession::GetSearchResult()", 3  
"#", "COUNT START", "2000/06/19", "15:10:00.000"  
"program name", "pid", "message", "times"  
"prog01", "00005124", "CdbrSession::Commit()", 4  
"CdbrSession::Begin()", 5  
"CdbrSession::GetSearchResult()", 6  
"#", "COUNT START", "2000/06/19", "15:30:00.000"  
"program name", "pid", "message", "times"  
"prog01", "00005124", "CdbrSession::Commit()", 7  
"CdbrSession::Begin()", 8  
"CdbrSession::GetSearchResult()", 9  
"#", "COUNT START", "2000/06/19", "15:50:00.000"  
"program name", "pid", "message", "times"  
"prog01", "00005124", "CdbrSession::Commit()", 10  
"CdbrSession::Begin()", 11  
"CdbrSession::GetSearchResult()", 12  
"#", "COUNT START", "2000/06/19", "16:10:00.000"  
"#", "No entry found."
```

付録

付録 A 用語解説

付録 A 用語解説

このマニュアルで使用している、統計解析ツールに関する用語について説明します。ここで説明していない DocumentBroker の用語については、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

(英字)

ClassId エントリ

プロパティ拡張情報ファイルのクラス名セクションを構成する要素の一つで、このエントリに OIID 変換、操作対象識別子変換をするプロパティのクラス識別子を指定します。

connect (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの CONNECT エントリが、この項目に分類されて集計されます。

create (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの CREATE OBJECT エントリが、この項目に分類されて集計されます。

CSV 形式

データをテキストとしてファイルに記録するときの形式の一つで、1 レコード内のデータをコンマで区切って並べる形式のことです。統計解析ツールでは、コマンドによる処理結果をこの形式で出力できるようにしています。

delete (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの DELETE OBJECT エントリが、この項目に分類されて集計されます。

DocumentSpace 構成定義ファイル (docspace.ini)

文書空間の構成を定義するためのファイルで、EDMRptAclog コマンドのデータベース (HiRDB) への接続に使用します。HiRDB のホスト名とポート番号、ユーザのユーザ名とパスワードを指定しておきます。

EDMRptAclog コマンドは、データベースに接続する際に、DocumentSpace 構成定義ファイルを参照します。

EDMRptAclog コマンド

統計解析ツールが提供するコマンドの一つで、アクセスログファイルの解析・分類・集計をする際にユーザが使用しません。

EDMRptTrace コマンド

統計解析ツールが提供するコマンドの一つで、トレースファイルの解析・分類・集計をする際にユーザが使用します。

enter エントリ

EDMRptTrace コマンドが処理の対象としているトレースログの一つです。

error (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの ERROR エントリが、この項目に分類されて集計されます。

GUID

Globally Unique Identifier の略です。DMA のクラス、プロパティ、検索オペレータなどに与えるユニークな識別子です。GUID は、「X」を 0 ~ 9 および a ~ f (小文字) で表される 16 進数とした

「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8 けた -4 けた -4 けた -4 けた -12 けた)」の形式で表されます。

msg エントリ

EDMRptTrace コマンドが処理の対象としているトレースログの一つです。

OIID

Object Instance Identifier の略です。文書空間での永続オブジェクトの存在や格納位置などを明確にするために使用する識別子です。OIID は DMA URL として定義されます。

OIID 変換機能

EDMRptAclog コマンドが提供する機能の一つで、アクセスログ中の OIID を、ユーザが見てわかるように、アクセス対象オブジェクトのプロパティの値に置き換えます。

PdHost エントリ

DocumentSpace 構成定義ファイルの [Entry0001] セクションを構成する要素の一つで、このエントリに HiRDB サーバのホスト名を指定します。

PdNamePort エントリ

DocumentSpace 構成定義ファイルの [Entry0001] セクションを構成する要素の一つで、このエントリに HiRDB サーバのポート番号を指定します。

PdUser エントリ

DocumentSpace 構成定義ファイルの [Entry0001] セクションを構成する要素の一つで、このエントリにデータベース (HiRDB) に接続するユーザのユーザ名とパスワードを指定します。

PrepresentProperties エントリ

プロパティ拡張情報ファイルのクラス名セクションを構成する要素の一つで、このエントリに OIID 変換、操作対象識別子変換をするプロパティのプロパティ名を指定します。ここで指定したプロパティの値が OIID、操作対象識別子の変換対象になります。

refer (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの GET RESULT エントリ以外の GET XXXX エントリ (XXXX は処理対象を表す引数) などの、オブジェクトの参照に関連するアクセスログが、この項目に分類されて集計されます。

search (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの EXECUTE SEARCH エントリ、GET RESULT エントリなどの、検索の実行に関連するアクセスログが、この項目に分類されて集計されます。

update (集計項目)

EDMRptAclog コマンドの集計項目の一つです。アクセスログファイルの CHANGE XXXX エントリ、SET XXXX エントリ (XXXX は処理対象を表す引数) などの、オブジェクトの更新を伴ったアクセスログが、この項目に分類されて集計されます。

(ア行)

アクセスログ

アクセスログファイルに残されている一つ一つの記録のことです。DocumentBroker は、DocumentBroker で構築された文書管理システムへのアクセス状況を、アクセスログとして記録しています。

アクセスログ表

集計機能で利用するために、EDMRptAclog コマンドが作成する一時ファイルです。

アクセスログファイル (EDMAccess_"NO".log)

アクセスログが格納されているファイルのことです。なお, "NO" は 1 ~ 16 の通し番号です。

エントリ

ソフトウェアで扱う一つの単位となる情報です。EDMRptAclog コマンドでは、アクセスログファイルに記録されている一つ一つのアクセスログがエントリとして扱われます。EDMRptTrace コマンドでは、トレースファイルに記録されている一つ一つのトレースログがエントリとして扱われます。また、EDMRptAclog コマンド、EDMRptTrace コマンドが出力する結果の 1 行 1 行も、エントリということになります。

エントリ抽出機能

EDMRptAclog コマンド、EDMRptTrace コマンドが提供する機能の一つです。EDMRptAclog コマンドでは、抽出エントリ定義ファイルに指定されたメソッド名と引数によって、アクセスログエントリが抽出されて出力されます。EDMRptTrace コマンドでは、抽出エントリ定義ファイルに指定された処理の種別によって、トレースログエントリが抽出されて出力されます。

(カ行)

クライアントトレース (EDMRasTraceCL"PID"_"NO".log)

DocumentBroker クライアントでのユーザプログラムによる処理内容を記録したトレースファイルです。なお, "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。

クラス名セクション

プロパティ拡張情報ファイルを構成するセクションで、ClassId エントリと PrepresentProperties エントリから構成されます。

(サ行)

サーバトレース (EDMRasTrace"PID"_"NO".log)

DocumentBroker サーバでのユーザプログラムによる処理内容を記録したトレースファイルです。なお, "PID" はプロセス ID, "NO" は 1 ~ 16 の通し番号です。

集計キー定義ファイル

EDMRptAclog コマンドの集計機能で、-k オプションを指定してキーとなる項目によって集計する場合に、キーとなる項目を指定しておくファイルです。EDMRptAclog コマンドの -k オプションで、集計キー定義ファイルのファイル名を指定します。

集計機能

EDMRptAclog コマンド、EDMRptTrace コマンドが提供する機能の一つです。EDMRptAclog コマンドでは、アクセスログファイルに残された記録 (アクセスログ) が、ユーザ名およびオブジェクト名で分類・集計されて結果が出力されます。EDMRptTrace コマンドでは、トレースファイルに残された記録 (トレース) が、ユーザプログラムのプログラム名、プロセス ID およびスレッド ID で分類・集計されて結果が出力されます。

操作対象識別子

アクセスログにアクセス対象オブジェクトの情報として出力される、バージョン識別子、リレーション識別子およびリンク識別子の総称です。

操作対象識別子変換機能

EDMRptAclog コマンドが提供する機能の一つで、アクセスログ中の操作対象識別子を、ユーザが見てわかるように、アクセス対象オブジェクトのプロパティの値に置き換えます。

(夕行)

抽出エントリ定義ファイル

EDMRptAclog コマンド、EDMRptTrace コマンドのエントリ抽出機能で、-x オプションを指定してエントリを抽出する場合に、抽出するエントリのメソッド名、引数および処理の種別を指定しておくファイルです。EDMRptAclog コマンド、EDMRptTrace コマンドの -x オプションで、抽出エントリ定義ファイルのファイル名を指定します。

動作環境定義ファイル (edmrpt.ini)

統計解析ツールが動作する時に参照する情報を定義するファイルです。ユーザ ID の集計方法に関する情報、および OIID 変換、操作対象識別子変換の実行に関する情報が定義されています。

トレース

トレースファイルに残されている一つ一つの記録のことで、トレースログということもあります。DocumentBroker は、サーバおよびクライアントのプログラムの処理内容を、トレースとして記録しています。

トレース表

集計機能で利用するために、EDMRptTrace コマンドが作成する一時ファイルです。

トレースファイル (EDMRasTrace"PID"_"NO".log または EDMRasTraceCL"PID"_"NO".log)

DocumentBroker サーバおよびクライアントのプログラムの処理内容の履歴 (トレース) を記録したファイルです。トレースファイルには、サーバトレース (EDMRasTrace"PID"_"NO".log) と、クライアントトレース (EDMRasTraceCL"PID"_"NO".log) があります。なお、"PID" はプロセス ID、"NO" は 1 ~ 16 の通し番号です。

トレースファイル分割機能

EDMRptTrace コマンドが提供する機能の一つです。サーバトレース解析の場合はユーザプログラムのプログラム名とユーザプログラムのプロセス ID ごとに、クライアントトレース解析の場合はユーザプログラムのスレッド ID ごとに、トレースファイルの内容が分割されて別々のファイルに出力されます。

(八行)

標準エラー出力 (stderr)

コマンド実行時などにエラーが発生した場合、そのことをユーザに知らせるためにシステムが使用する標準の出力先のことです。

標準出力 (stdout)

コマンドの実行結果などをユーザに知らせるために、システムが使用する標準の出力先のことです。

プロパティ拡張情報ファイル (propex.ini)

DocumentBroker の文書モデルを構成するクラスに定義されている複数のプロパティの中から、OIID 変換、操作対象識別子変換の対象となるプロパティを特定するために使用するファイルです。OIID 変換、操作対象識別子変換の対象となるプロパティを指定しておきます。

EDMRptAclog コマンドは、データベース (HiRDB) からアクセス対象のオブジェクトのプロパティの値を取得して、アクセスログ中の OIID、操作対象識別子をプロパティの値 (文書名など) に変換する際に、プロパティ拡張情報ファイルを参照します。

(マ行)

メッセージトレース (aclog_msg.log または trace_msg.log)

EDMRptAclog コマンド、EDMRptTrace コマンドのメッセージを出力するファイルです。メッセージトレースは、環境変数「EDMARUMSG」で設定したディレクトリに出力されます。

索引

記号

[Entry0001] セクション 24
[Extract Aclog] セクション 28
[Extract Trace] セクション 29
[Key Object] セクション 26
[Key User] セクション 25
[ObjectConvertingMode] セクション 32
[UserCountingMode] セクション 32

A

arg 30

C

call 30
ClassId エントリ 23, 92
connect 19, 92
create 19, 92
CSV 形式 92

D

delete 19, 92
DOCBROKERDIR 13, 14
docspace.ini 24, 92
DocumentBroker の実行環境作成コマンド 12
DocumentSpace 構成定義ファイル 24, 92

E

EDMARUMSG 13, 14, 42, 44
edmrpt.ini 95
EDMRptAclog コマンド 92
EDMRptAclog コマンド (アクセスログの解析) 48
EDMRptAclog コマンドに必要なファイル 18
EDMRptAclog コマンドの機能 3
EDMRptAclog コマンドの機能概要 3
EDMRptAclog コマンドの機能の処理方式 52
EDMRptAclog コマンドの集計方法 53
EDMRptAclog コマンドの出力形式 59
EDMRptAclog コマンドの出力例 62
EDMRptAclog コマンドの詳細 48
EDMRptAclog コマンドの処理概要 8
EDMRptAclog コマンドのトラブルシューティング機能 42
EDMRptAclog コマンドの目的 3
EDMRptTrace コマンド 92
EDMRptTrace コマンド (トレースの解析) 70

EDMRptTrace コマンドに必要なファイル 35
EDMRptTrace コマンドの機能 5
EDMRptTrace コマンドの機能概要 5
EDMRptTrace コマンドの機能の処理方式 75
EDMRptTrace コマンドの集計方法 76
EDMRptTrace コマンドの出力形式 79
EDMRptTrace コマンドの出力例 84
EDMRptTrace コマンドの詳細 70
EDMRptTrace コマンドの処理概要 9
EDMRptTrace コマンドのトラブルシューティング機能 44
EDMRptTrace コマンドの目的 5
EDMSSetup 12
enter 30, 37
enter エントリ 92
error 19, 30, 92
exit 30

G

GUID 92
GUID 文字列 23

H

HiRDB サーバのポート番号 93
HiRDB サーバのホスト名 93

L

LANG 13
LD 33
LIBPATH 13
LR 33
LV 33

M

mark 30
msg 30, 37
msg エントリ 93

O

OIID 3, 33, 93
OIID 変換 33
OIID 変換, 操作対象識別子変換時のエラー処理 42
OIID 変換機能 3, 93
OIID 変換機能, 操作対象識別子変換機能 (-c オプション指定) の出力形式 61

OIID 変換機能, 操作対象識別子変換機能の処理方式
55

OIID 変換機能の処理の流れ 55

P

PATH 14

PdHost エントリ 24, 93

PdNamePort エントリ 24, 93

PdUser エントリ 24, 93

PrepresentProperties エントリ 23, 27, 93
propex.ini 22, 95

R

refer 19, 93

return 30

RL 33

S

search 19, 93

stderr 95

stdout 95

SUBTOTAL 61, 82

syslog ファイル 42, 44

T

TOTAL 61, 82

TZ 13, 14

U

update 19, 93

V

VD 33

W

warn 30

あ

アクセス種別 19

アクセス種別とその意味 60

アクセスの内容 20

アクセスログ 3, 93

アクセスログの項目 19

アクセスログの出力形式 19

アクセスログ表 50, 53, 93

アクセスログ表作成時の処理方式 53

アクセスログ表の項目 53

アクセスログ表の例 53

アクセスログファイル 2, 18, 94

アクセスログファイル格納ディレクトリ 18, 19,
52

アクセスログファイルのコピー 18

アクセスログファイルの出力例 21

アクセスログファイル名 51

い

イベントログ 42, 44

え

エントリ 94

エントリ抽出機能 4, 5, 94

エントリ抽出機能の出力形式 61, 83

エントリ抽出機能の処理方式 58, 79

お

オブジェクト変換モード 33

オブジェクト名 26

オプションの指定の組み合わせ (EDMRptAclog コマ
ンド) 52

オプションの指定の組み合わせ (EDMRptTrace コマ
ンド) 74

か

環境変数の設定 (AIX の場合) 13

環境変数の設定 (Windows の場合) 13

き

共通情報 59, 80, 82

く

クライアントトレース 2, 94

クライアントトレース解析 70

クライアントトレース格納ディレクトリ 35, 36,
74

クライアントトレースの出力例 38

クラス定義情報ファイル 23

クラス名セクション 94

こ

コマンドの記述形式 46

コマンドの実行に必要なファイル 17

コマンド引数 51, 73

コマンドリファレンス 45

さ

サーバトレース 2, 94
 サーバトレース解析 70
 サーバトレース格納ディレクトリ 35, 36, 74
 サーバトレースの出力例 38

し

実行環境設定の流れ 12
 実行環境の設定 11
 集計キー定義ファイル 3, 25, 94
 集計キー定義ファイルで指定された項目別の集計対象
 54
 集計キー定義ファイルと抽出エントリ定義ファイルを
 一つのファイルに記述する場合の記述規則 30
 集計キー定義ファイルの記述規則 25
 集計キー定義ファイルの記述形式 25
 集計キー定義ファイルの記述例 27
 集計キー定義ファイル名 48
 集計機能 3, 5, 94
 集計機能で分類されるアクセスの内容と集計項目 20
 集計機能の出力形式 59, 79
 集計機能の処理方式 53, 76
 集計結果 60, 81
 集計項目 20, 92, 93
 終了コード一覧 47
 出力ファイル名 50, 73

そ

操作対象識別子 3, 94
 操作対象識別子変換機能 3, 94
 操作対象識別子変換機能の処理の流れ 56

ち

抽出エントリ定義ファイル 4, 5, 28, 95
 抽出エントリ定義ファイルの記述規則 28
 抽出エントリ定義ファイルの記述形式 28
 抽出エントリ定義ファイルの記述例 30
 抽出エントリ定義ファイル名 51, 73

て

ディレクトリ名 51, 73

と

統計解析ツールが提供するコマンドの一覧 2
 統計解析ツールとは 2

統計解析ツールの機能概要 1
 統計解析ツールの実行環境 12
 統計解析ツールの処理概要 7
 統計解析ツールのディレクトリ構成 (UNIX の場合)
 15
 統計解析ツールのディレクトリ構成 (Windows の場
 合) 16
 動作環境定義ファイル 4, 31, 95
 動作環境定義ファイルの記述規則 31
 動作環境定義ファイルの記述形式 32
 動作環境定義ファイルの記述例 34
 トラブルシュート機能 41
 トレース 95
 トレース表 73, 76, 95
 トレース表の項目 76
 トレース表の例 76
 トレースファイル 2, 35, 95
 トレースファイルのコピー 35
 トレースファイルの出力例 38
 トレースファイル分割機能 5, 95
 トレースファイル分割機能の出力形式 82
 トレースファイル分割機能の処理方式 77
 トレースファイル名 73
 トレースログ 5
 トレースログの項目 36
 トレースログの出力形式 36

は

バージョン識別子 3, 33
 パスワード 93

ひ

標準エラー出力 95
 標準出力 95

ふ

プロパティ拡張情報ファイル 4, 21, 27, 95
 プロパティ拡張情報ファイルの記述規則 22
 プロパティ拡張情報ファイルの記述形式 22
 プロパティ拡張情報ファイルの記述例 24
 分類項目 60, 81, 83

め

メソッド名と引数 29
 メッセージトレース 42, 44, 95
 メッセージトレースを出力する 13, 14
 メッセージの出力先 42, 44

ゆ

ユーザ名 26, 93

ユーザ名集計モード 33

り

リレーション識別子 3, 33

リンク識別子 3, 33