

uCosminexus DocumentBroker Version 3 オブジェクト操作ツール

文法書

3000-3-F15-20

対象製品

R-1M95D-43 uCosminexus DocumentBroker Development Kit Version 3 03-60 (適用 OS : AIX 5L V5.1 , AIX 5L V5.2 , AIX 5L V5.3)
R-1595D-43 uCosminexus DocumentBroker Development Kit 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 , Windows XP , Windows Vista , Windows 7 , Windows 8)
R-1M95D-53 uCosminexus DocumentBroker Runtime Version 3 03-60 (適用 OS : AIX 5L V5.1 , AIX 5L V5.2 , AIX 5L V5.3)
R-1595D-53 uCosminexus DocumentBroker Runtime 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 , Windows XP , Windows Vista , Windows 7 , Windows 8)

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

Active Directory 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。
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。
Microsoft Office Word は、米国 Microsoft Corporation の商品名称です。
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。Microsoft Word は、米国 Microsoft Corporation の商品名称です。
OLE は、米国 Microsoft Corporation が開発したソフトウェア名称です。
Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
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。
VisualAge は、米国およびその他の国における International Business Machines Corporation の商標です。
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。
Windows Vista 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 (第 2 版) 3000-3-F15-20

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-F15-20) DocumentBroker Development Kit 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 , Windows XP , Windows Vista , Windows 7 , Windows 8) ,
DocumentBroker Runtime 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 , Windows XP , Windows Vista , Windows 7 , Windows 8)

追加・変更内容	変更箇所
次の前提 OS をサポートしました。 <ul style="list-style-type: none">• Windows Server 2012• Windows 8	1.2

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オブジェクト操作ツールの機能と使い方について説明したものです。

- R-1M95D-43 uCosminexus DocumentBroker Development Kit Version 3
- R-1595D-43 uCosminexus DocumentBroker Development Kit Version 3
- R-1M95D-53 uCosminexus DocumentBroker Runtime Version 3
- R-1595D-53 uCosminexus DocumentBroker Runtime Version 3

対象読者

このマニュアルは、uCosminexus DocumentBroker Development Kit Version 3 または uCosminexus DocumentBroker Runtime Version 3 で提供するオブジェクト操作ツールを利用して、クライアントアプリケーションで作成したオブジェクトを操作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。

- UNIX または Windows に関する知識
- SQL 言語に関する知識
- 分散オブジェクト技術に関する知識

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 オブジェクト操作ツールの概要

オブジェクト操作ツールの概要と機能について説明しています。

第 2 章 実行環境の設定

オブジェクト操作ツールを実行するための実行環境の設定方法について説明しています。

第 3 章 オブジェクト操作ツールで使用するファイル

オブジェクト操作ツールで使用するファイルについて説明しています。

第 4 章 コマンド実行ログと障害対策

コマンド実行ログと障害対策について説明しています。

第 5 章 コマンドリファレンス

オブジェクト操作ツールの実行コマンドの文法について説明しています。

付録 A OIID 指定時の URL の文法

OIID の形式および OIID を指定するときの URL の文法について説明しています。

付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲

文書空間で使用する文字コード種別が UTF-8 の場合にサポートする範囲について説明しています。

付録 C 用語解説

uCosminexus DocumentBroker で使用する用語について説明しています。

関連マニュアル

このマニュアルは次のマニュアルと関連がありますので、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

はじめに

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)

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)

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 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。

uCosminexus DocumentBroker Text Search Index Loader Version 3 (3020-3-U72)

uCosminexus DocumentBroker に格納されている文書からテキスト情報を抽出して、全文検索用のインデクスファイルを作成し、これを全文検索インデクスに登録するユティリティ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

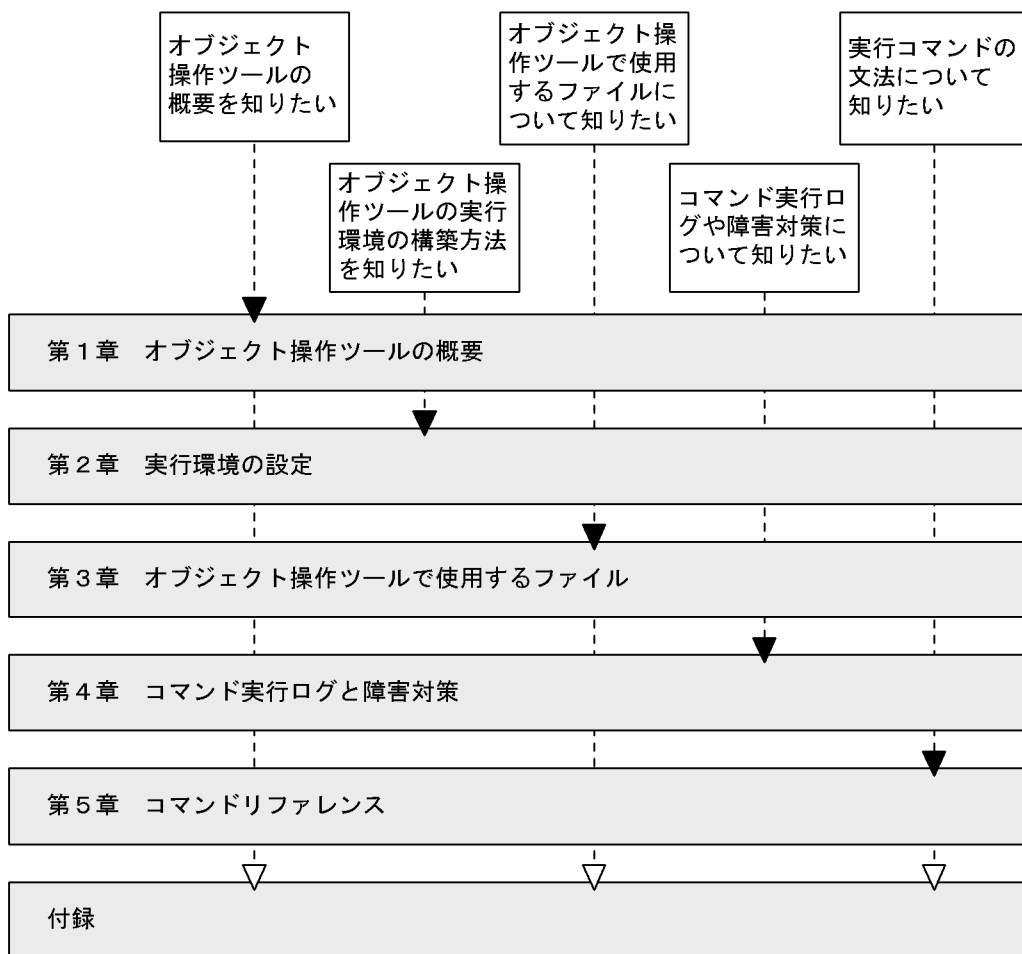
- HiRDB デジタルコンテンツアクセスプラグイン HiRDB File Link (UNIX(R) 用) (3000-6-253)
- HiRDB デジタルコンテンツアクセスプラグイン HiRDB File Link (Windows(R) 用) (3020-6-141)

注

このマニュアルでは、これらのマニュアルを「HiRDB File Link」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)



: 必ず読む項目



: 必要に応じて読む項目

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
uCosminexus DocumentBroker Server Version 3	DocumentBroker サーバ
HiRDB/Single Server Version 6	HiRDB

製品名称	略称
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	
IBM VisualAge C++ Professional for AIX V5	VisualAge C++ Professional for AIX
IBM VisualAge C++ Professional for AIX V6	
IBM XL C/C++ Enterprise Edition V7 for AIX	IBM XL C/C++ Enterprise Edition for AIX
IBM XL C/C++ Enterprise Edition V8 for AIX	
Microsoft(R) Active Directory(R)	Active Directory
Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版	Windows Server 2003
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	
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	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP
Microsoft(R) Windows Vista(R) Ultimate 日本語版	Windows Vista
Microsoft(R) Windows Vista(R) Business 日本語版	
Microsoft(R) Windows Vista(R) Enterprise 日本語版	
Microsoft(R) Windows(R) 7 Professional 日本語版 (32 ビット版)	Windows 7
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (32 ビット版)	
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (32 ビット版)	
Microsoft(R) Windows(R) 7 Professional 日本語版 (64 ビット版)	
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (64 ビット版)	
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (64 ビット版)	
Windows(R) 8 Pro 日本語版 (32 ビット版)	Windows 8
Windows(R) 8 Pro 日本語版 (64 ビット版)	
Windows(R) 8 Enterprise 日本語版 (32 ビット版)	
Windows(R) 8 Enterprise 日本語版 (64 ビット版)	
Microsoft(R) Office Word	Word

製品名称	略称
Microsoft(R) Windows Vista(R) Ultimate 日本語版	Windows Vista
Microsoft(R) Windows Vista(R) Business 日本語版	
Microsoft(R) Windows Vista(R) Enterprise 日本語版	
Microsoft(R) Windows(R) 7 Professional 日本語版 (32 ビット版)	Windows 7
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (32 ビット版)	
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (32 ビット版)	
Microsoft(R) Windows(R) 7 Professional 日本語版 (64 ビット版)	
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (64 ビット版)	
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (64 ビット版)	
Sun Java(TM) System Directory Server 5.1	Sun Java System Directory Server
Sun Java(TM) System Directory Server 5.2	
Sun Java(TM) System Directory Server 6.3	
TPBroker Developer for C++	TPBroker V3
TPBroker for C++	
TPBroker	TPBroker V5

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。

- DocumentBroker Development Kit の 01-21 以前のバージョンについては、Version 1 と表記します。
- uCosminexus DocumentBroker Development Kit Version 3 で提供する C++ クラスライブラリをクラスライブラリと表記します。
- AIX を UNIX と表記することがあります。
- TPBroker V3 および TPBroker V5 を合わせて TPBroker 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 , Windows Server 2012 , Windows XP , Windows Vista , Windows 7 , および Windows 8 を合わせて Windows と表記することがあります。

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 <u>I</u> nternational
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

英略語	英字での表記
BOM	<u>B</u> yte <u>O</u> rder <u>M</u> ark
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> atabase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
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
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 <u>F</u> ormat
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 <u>S</u> tandardization
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

英略語	英字での表記
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
SGML	<u>S</u> tandard <u>G</u> eneralized <u>M</u> arkup <u>L</u> anguage
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol/ <u>I</u> nternet <u>P</u> rotocol
TIFF	<u>T</u> ag <u>I</u> mage <u>F</u> ile <u>F</u> ormat
UCS-2	<u>U</u> niversal <u>C</u> haracter <u>S</u> et coded in <u>2</u> octets
UCS-4	<u>U</u> niversal <u>C</u> haracter <u>S</u> et coded in <u>4</u> octets
UNC	<u>U</u> niversal <u>N</u> aming <u>C</u> onvention
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator
UTC	<u>U</u> niversal <u>T</u> ime <u>C</u> oordinated
UTF-8	<u>8</u> -bit <u>U</u> CS <u>T</u> ransformation <u>F</u> ormat
W3C	<u>W</u> orld <u>W</u> ide <u>W</u> eb <u>C</u> onsortium
WWW	<u>W</u> orld <u>W</u> ide <u>W</u> eb
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage

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。

記号	意味
{ }	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

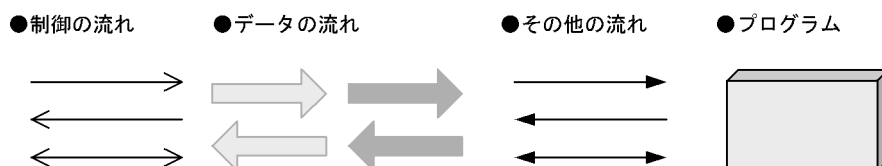
このマニュアルで使用する構文要素の種類を次に示します。

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 すべての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（かしょ）

KB（キロバイト）などの単位表記について

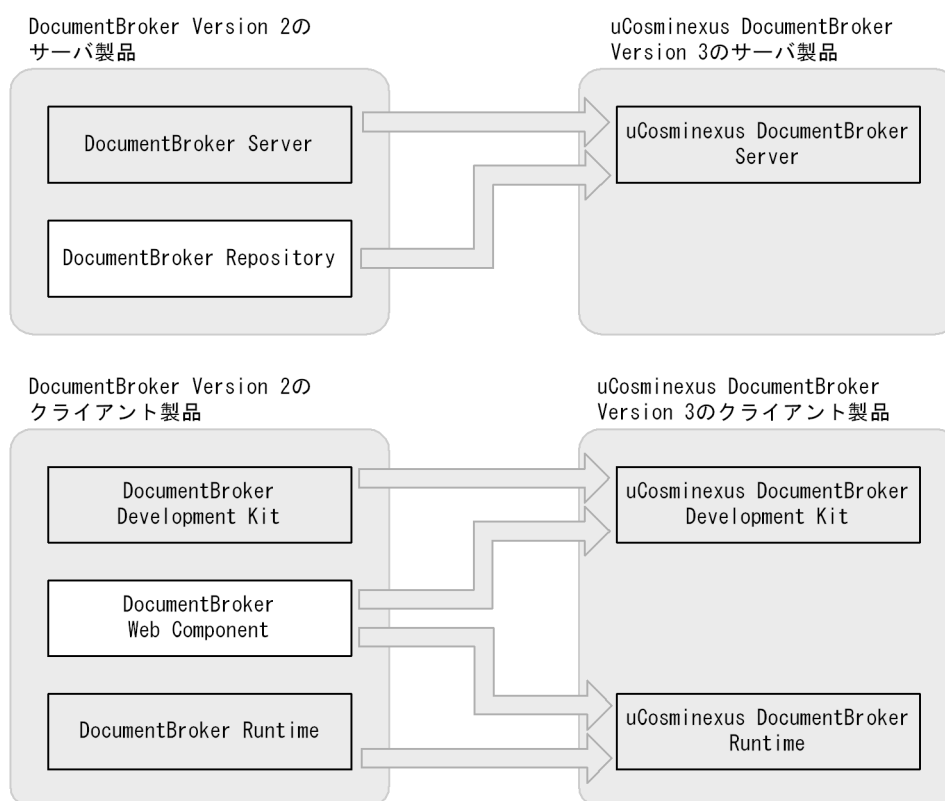
1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ $1,024$ バイト、 $1,024^2$ バイト、 $1,024^3$ バイト、 $1,024^4$ バイトです。

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
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	オブジェクト操作ツールの概要	1
1.1	オブジェクト操作ツールとは	2
1.1.1	オブジェクト操作ツールとは	2
1.1.2	オブジェクト操作ツールの使用例	2
1.2	オブジェクト操作ツールの機能	3
1.3	システム構成およびクラスライブラリとの関係	4
1.3.1	システム構成	4
1.3.2	クラスライブラリとの関係	4
1.4	オブジェクト操作ツールで利用できる文字コード種別	6
2	実行環境の設定	7
2.1	実行環境設定の流れ	8
2.2	実行環境設定の詳細	10
2.2.1	実行ユーザについて	10
2.2.2	環境変数の設定	10
2.2.3	実行環境のセットアップ	13
2.2.4	実行環境制御ファイルの編集	13
2.2.5	クラス定義情報ファイルの取得	13
2.2.6	実行環境の更新	14
2.3	ファイル転送機能の使用	15
3	オブジェクト操作ツールで使用するファイル	17
3.1	オブジェクト操作ツールで使用するファイル一覧	18
3.2	実行環境制御ファイル	19
3.2.1	記述規則	19
3.2.2	記述形式	19
3.2.3	実行環境制御ファイルの記述例	26
3.3	クラス定義情報ファイル	28
3.3.1	ファイル名の規則	28
3.3.2	記述規則	28
3.3.3	記述形式	28
3.3.4	クラス定義情報ファイルの記述例	31
3.4	ID ファイル	32
3.4.1	記述規則	32
3.4.2	記述形式	32
3.4.3	ID ファイルの記述例	32
3.5	プロパティ情報ファイル	34
3.5.1	記述規則	34

3.5.2	記述形式	34
3.5.3	実行するコマンドと指定するエントリの対応	45
3.5.4	プロパティ情報ファイルの記述例	49
3.5.5	複数オブジェクトを一括作成する場合のプロパティ情報ファイル	53
3.6	問い合わせファイル	55
3.6.1	記述規則	55
3.6.2	基本項目	55
3.6.3	記述形式	55
3.6.4	問い合わせファイルの記述例	55
3.7	アップロード情報ファイル	57
3.7.1	記述規則	57
3.7.2	記述形式	57
3.7.3	アップロード情報ファイルの記述例	59
3.8	リファレンス情報ファイル	60
3.8.1	記述規則	60
3.8.2	記述形式	60
3.8.3	リファレンス情報ファイルの記述例	62

4

コマンド実行ログと障害対策 63

4.1	コマンド実行ログ	64
4.1.1	コマンド実行ログの制御	64
4.1.2	出力レベル	64
4.1.3	出力先ディレクトリおよびファイル	64
4.1.4	ログの出力形式と出力ログ情報	65
4.1.5	出力例	69
4.2	障害対策	70
4.2.1	エラーが発生したときのオブジェクト操作ツールの動作	70
4.2.2	メッセージ	71
4.2.3	トレースファイル	71
4.2.4	詳細エラーログファイル	71

5

コマンドリファレンス 73

実行コマンド一覧	75
コマンド形式	76
コマンド実行結果	78
dbrbind (パブリック ACL のバインド)	79
dbrcrtacl (パブリック ACL の作成)	81
dbrcrtcntr (コンテナの作成)	83
dbrcrtdata (独立データの作成)	88
dbrcrtdoc (文書の作成)	90

dbrcrtidx (全文検索インデクスの作成)	101
dbrdelacl (パブリック ACL の削除)	103
dbrdelcntr (コンテナの削除)	104
dbrdeldata (独立データの削除)	111
dbrdeldoc (文書の削除)	112
dbrdelidx (全文検索インデクスの削除)	116
dbrdowndoc (文書のダウンロード)	117
dbrexquery (問い合わせの実行)	120
dbrgetprop (プロパティの取得)	122
dbrlink (オブジェクト間の関連づけ)	125
dbrlsacl (バインドしているパブリック ACL 一覧またはバインドされているオブジェクト一覧の取得)	128
dbrlschild (コンテナに含まれるオブジェクト一覧の取得)	131
dbrlscompo (マルチファイル文書のコンテンツ情報一覧の取得)	134
dbrlsparent (オブジェクトを包含するコンテナ一覧の取得)	136
dbrlsversion (バージョン一覧の取得)	139
dbrlsversionable (バージョン管理情報一覧の取得)	141
dbrmndrdoc (文書のレンディション操作)	143
dbrsetmode (構成管理モードの設定)	151
dbrsetprop (プロパティの設定)	155
dbrtoolsetup (実行環境のセットアップ)	158
dbrunbind (パブリック ACL のアンバインド)	160
dbrunlink (オブジェクト間の関連づけの解除)	162
dbrupcntr (コンテナのバージョンアップ)	165
dbrupdoc (文書のアップロード)	166
リファレンスファイル管理機能を使用する場合のオブジェクトの削除	173

付録

付録 A OIID 指定時の URL の文法	175
付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲	176
付録 C 用語解説	178
	180

索引

195

1

オブジェクト操作ツールの概要

この章では、オブジェクト操作ツールの概要と機能について、クラスライブラリとの関係と合わせて説明します。

-
- 1.1 オブジェクト操作ツールとは
 - 1.2 オブジェクト操作ツールの機能
 - 1.3 システム構成およびクラスライブラリとの関係
 - 1.4 オブジェクト操作ツールで利用できる文字コード種別
-

1.1 オブジェクト操作ツールとは

ここでは、オブジェクト操作ツールの概要、および使用例について説明します。

1.1.1 オブジェクト操作ツールとは

オブジェクト操作ツールは、DocumentBroker Development Kit が提供するクラスライブラリを使用したクライアントアプリケーションです。DocumentBroker では、クライアントアプリケーションを円滑に開発していただくために必要なクラスライブラリの機能を、コマンド群（オブジェクト操作ツール）として提供しています。

1.1.2 オブジェクト操作ツールの使用例

オブジェクト操作ツールは、クライアントアプリケーションの開発を円滑に進められるように支援することを目的としており、次のような使われ方を想定しています。

クライアントアプリケーションで作成したオブジェクトの設定内容を確認するための、設定確認用プログラムとして使用する。

クライアントアプリケーションの動作確認用に作成したオブジェクトが不要になった場合に、それらのオブジェクトを削除するための削除用プログラムとして使用する。

このように、オブジェクト操作ツールは、クライアントアプリケーションの開発支援を目的としているため、クラスライブラリが提供するすべての機能を備えてはいません。したがって、DocumentBroker システムを本格的に運用する場合には、クラスライブラリを使用してクライアントアプリケーションを開発してください。なお、オブジェクト操作ツールは、システム管理者（Administrator）の権限を持つユーザで実行してください。

1.2 オブジェクト操作ツールの機能

オブジェクト操作ツールは、クラスライブラリが提供する機能をコマンド形式で提供しています。提供するコマンドの種類と各コマンドの文法については、「5. コマンドリファレンス」を参照してください。

なお、クラスライブラリが提供する機能のうち、次に示す機能については現在サポートしていません。

バージョン付き文書またはバージョン付き構成管理コンテナのチェックアウト状態の取得

文書間リレーション管理に関する操作

XML 文書管理機能

File Link 連携機能

注

TPBroker V5 を使用する場合、Windows Server 2003 R2 x64 Edition、Windows Server 2008、Windows Server 2008 R2、Windows Server 2012、Windows XP、Windows Vista、および Windows 7 を使用する場合となります。

これらの機能についての動作確認には、クラスライブラリを使用して作成したクライアントアプリケーションを使用してください。

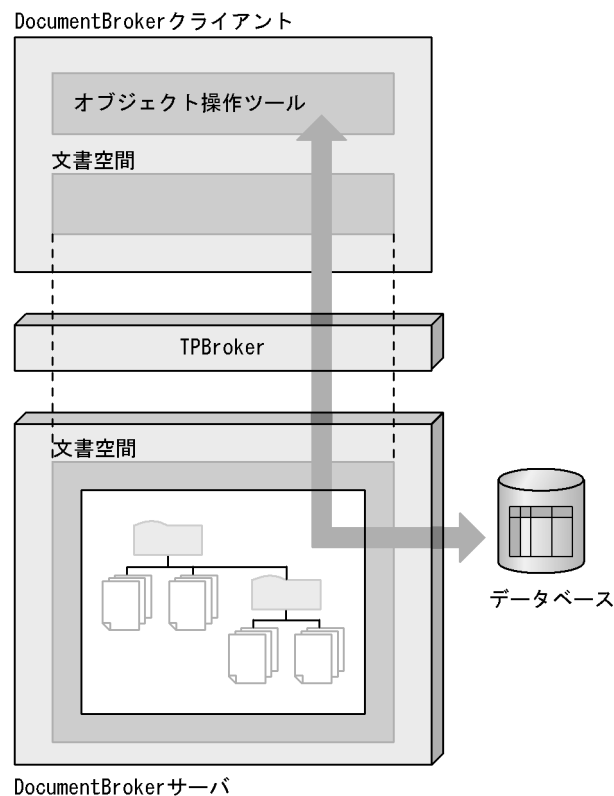
1.3 システム構成およびクラスライブラリとの関係

ここでは、オブジェクト操作ツールを使用する場合のシステム構成について説明します。また、オブジェクト操作ツールの対象となるオブジェクトと、クラスライブラリのクラスとの関係について説明します。

1.3.1 システム構成

オブジェクト操作ツールを使用するためのシステム構成を次の図に示します。

図 1-1 オブジェクト操作ツールのシステム構成



オブジェクト操作ツールは、DocumentBroker Development Kit で提供されているクラスライブラリを使用したクライアントアプリケーションです。したがって、データベースに格納されているオブジェクトを、DocumentBroker サーバの文書空間上で操作します。

1.3.2 クラスライブラリとの関係

オブジェクト操作ツールの操作対象となるオブジェクトの種別とクラスライブラリのクラスとの関係を、表 1-1 に示します。

表 1-1 クラスライブラリとの関係

オブジェクト種別	クラスライブラリのクラス
独立データ	CdbrIndependentPersistence
バージョンなし文書	CdbrDocument
バージョン付き文書	CdbrVersionableDocument

オブジェクト種別	クラスライブラリのクラス
文書	CdbrDocument CdbrVersionableDocument
バージョンなしコンテナ	CdbrReferentialContainer
バージョンなし構成管理コンテナ	CdbrVersionTraceableContainer
バージョン付き構成管理コンテナ	CdbrConfiguratedReferentialContainer
コンテナ	CdbrReferentialContainer CdbrVersionTraceableContainer CdbrConfiguratedReferentialContainer
構成管理コンテナ	CdbrVersionTraceableContainer CdbrConfiguratedReferentialContainer
パブリック ACL	CdbrPublicACL

また、表 1-1 に示したクラスライブラリのクラスは、オブジェクト操作ツールのオプション引数に指定する場合や、出力結果およびコマンド実行ログなどに出力される場合には、表 1-2 に示す記号で表されます。

表 1-2 クラスライブラリのクラスを表す記号

クラスライブラリのクラス	オブジェクト操作ツールで使用する記号
CdbrIndependentPersistence	IP
CdbrDocument	D
CdbrVersionableDocument	VD
CdbrReferentialContainer	RC
CdbrVersionTraceableContainer	VTC
CdbrConfiguratedReferentialContainer	CRC
CdbrPublicACL	ACL

1.4 オブジェクト操作ツールで使用できる文字コード種別

ここでは、オブジェクト操作ツールで使用できる文字コード種別について説明します。

オブジェクト操作ツールで使用する文字コード種別は、DocumentBroker サーバの文書空間で使用する文字コード種別に従います。そのため、オブジェクト操作ツールで文字コード種別を設定する必要はありません。

DocumentBroker サーバの文書空間で使用できる文字コード種別は次のどちらかです。

- Shift-JIS
- UTF-8 (使用できる文字コードの範囲は UCS-2 または UCS-4 です)

DocumentBroker サーバの文書空間で使用する文字コード種別の設定および運用方法については、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。

(1) 文書空間の文字コード種別に UTF-8 を設定した場合の注意事項

DocumentBroker サーバの文書空間の文字コード種別に UTF-8 を設定した場合の注意事項を次に示します。

- オブジェクト操作ツールで使用できるコマンドに制限があります。コマンドのサポート範囲の詳細については、「付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲」を参照してください。
- ユーザが記述する項目で使用できる文字コード種別に、次の表に示す制限があります。

表 1-3 ユーザが記述する項目で使用できる文字コード種別

ユーザが記述する項目	使用できる文字コード種別
<ul style="list-style-type: none"> • コマンド引数に指定する値 • コマンド引数に指定するファイルのファイルパス • コマンド引数に指定するファイルのファイル名 • コマンド引数に指定するファイルを記述する文字 	ASCII コードだけ使用できます。
<ul style="list-style-type: none"> • プロパティ情報ファイルに記述するプロパティ値 • 問い合わせファイルに記述する問い合わせ式 	UTF-8 および ASCII コードを使用できます。なお、ASCII コードだけ使用することもできます。

- オブジェクト操作ツールが出力する文字コード種別は、UTF-8 および ASCII コード、または ASCII コードだけのどちらかになります。そのため、オブジェクト操作ツールが出力したデータを参照する場合は、Shift-JIS および UTF-8 の両方に対応したエディタを使用してください。

2

実行環境の設定

この章では、オブジェクト操作ツールを使用する場合の、実行環境の設定について説明します。

2.1 実行環境設定の流れ

2.2 実行環境設定の詳細

2.3 ファイル転送機能の使用

2.1 実行環境設定の流れ

オブジェクト操作ツールの実行環境を設定する順序を、次の図に示します。なお、ここでは、オブジェクト操作ツールを使用するための環境設定についてだけ説明します。DocumentBroker Development Kit および DocumentBroker Runtime の環境設定については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

図 2-1 実行環境設定の流れ



説明 :

1. 環境変数を設定します。

オブジェクト操作ツールでは、次の環境変数の設定が必要です。

AIX の場合

- LANG
- TZ
- XDK_HOME
- LIBPATH
- PATH
- DBRTOOLS DIR
- PSALLOC
- NODISCLAIM

Windows の場合

- PATH
- DBRTOOLS DIR

なお、DBR_DETAIL_ERRORLOG は、必要に応じて設定してください。

2. 実行環境のセットアップコマンド (dbrtoolsetup -x CREATE コマンド) を実行します。

このコマンドを実行すると、次のファイルがデフォルトの内容でオブジェクト操作ツールの実行環境にコピーされます。

- 実行環境制御ファイル
- クラス定義情報ファイル

これらのファイルのデフォルトの内容に従って、実行環境が作成されます。

なお、デフォルトのファイルは、オブジェクト操作ツールが動作する最小限の内容になっています。ユーザの実行環境に合わせて、これらのファイルを手順 3. 以降で示す方法に従って編集してください。

3. 実行環境制御ファイルを編集します。
手順 2. でコピーされる実行環境制御ファイルの内容を、使用環境に応じて編集します。ユーザ名とパスワードは、必ず編集してください。
4. 接続先の DocumentBroker サーバで管理しているクラス定義情報ファイルを取得します。
手順 2. でコピーされるクラス定義情報ファイルには、オブジェクト操作ツールが動作するための最小限の情報だけが記述されています。オブジェクト操作ツールを運用する場合には、接続先の DocumentBroker サーバの情報が記述されているクラス定義情報ファイルを取得する必要があります。DocumentBroker サーバが提供しているクラス定義情報ファイル出力コマンド（EDMCrtSimMeta コマンド）によって出力したクラス定義情報ファイルを、オブジェクト操作ツールの実行環境にコピーしてください。EDMCrtSimMeta 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。
5. 実行環境のセットアップコマンド（dbrtoolsetup -x UPDATE コマンド）を実行します。
手順 3. および手順 4. で編集および取得した実行環境制御ファイルおよびクラス定義情報ファイルの内容で、実行環境を更新します。手順 3. または手順 4. でファイルを編集した場合は、必ず実行してください。

なお、DocumentBroker Development Kit または DocumentBroker Runtime をバージョンアップした場合は、手順 5. の実行環境のセットアップコマンドを実行して、実行環境を更新してください。

2.2 実行環境設定の詳細

ここでは、オブジェクト操作ツールを使用するために必要な環境設定の詳細について説明します。

2.2.1 実行ユーザについて

オブジェクト操作ツールでは、コマンドを実行するマシンの OS ユーザではなく、オブジェクト操作ツールの実行環境を定義するために使用する「実行環境制御ファイル」に記載されているユーザでセッションを確立（ログイン）します。ただし、実行環境制御ファイルに記述する接続ユーザが、DocumentBroker サーバで使用しているユーザ管理機能に登録されていない場合は、文書空間に接続するときにエラーが発生します。ユーザ管理機能とは、LDAP 対応のディレクトリサービスや UNIX のパスワードファイルと連携して、DocumentBroker を使用するユーザのユーザ情報を管理する機能です。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。

実行環境制御ファイルについては、「3.2 実行環境制御ファイル」を参照してください。

2.2.2 環境変数の設定

オブジェクト操作ツールを使用するためには、環境変数の設定が必要です。

なお、設定する環境変数は、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てください。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 の設定

環境変数「LANG」には、使用する文字コードセットを設定します。DocumentBroker の文書空間で使用する文字コード種別に合わせて値を設定してください。

文書空間で使用する文字コード種別が Shift-JIS の場合

設定する値は「Ja_JP」です。

文書空間で使用する文字コード種別が UTF-8 の場合

使用する言語に合わせて適切な値を設定してください。

ただし、次のときは「C」を設定してください。

- 文書空間で使用する文字コード種別に関係なく、ASCII コードのデータだけを扱う英語環境の場合
- 一つのアプリケーションから文字コード種別の異なる複数の文書空間に接続する場合に、接続先に文字コード種別が UTF-8 の文書空間が含まれるとき

なお、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。「Ja_JP」を設定した場合は、日本語のメッセージが出力されます。「Ja_JP」以外を設定した場合は、英語のメッセージが出力されます。

(b) TZ の設定

環境変数「TZ」には、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) XDK_HOME の設定

環境変数「XDK_HOME」には、SystemManager オブジェクトのレジストリファイルの格納場所を設定します。設定する値は、「/opt/HiEDMS/client/etc」です。

(d) LIBPATH の設定

関連プログラムのライブラリおよび DocumentBroker のクライアント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。

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

TPBroker V3 を使用する場合

```
:/opt/hitachi/common/lib
:/usr/vacpp/lib
:/opt/TPBroker/lib
```

TPBroker V5 を使用する場合

```
:/opt/hitachi/common/lib
:/usr/vacpp/lib
:TPBroker V5インストールディレクトリ/lib
```

注 VisualAge C++ Professional for AIX, または IBM XL C/C++ Enterprise Edition for AIX のライブラリの格納ディレクトリを設定します。これは、デフォルトのインストールディレクトリにインストールされている場合の値です。

DocumentBroker のクライアントライブラリのインストールディレクトリ

TPBroker V3 を使用する場合

```
:/opt/HiEDMS/client/lib
:/opt/HiEDMS/ACLibrary/lib
```

TPBroker V5 を使用する場合

```
:/opt/HiEDMS/client/lib_tp5
:/opt/HiEDMS/ACLibrary/lib_tp5
```

(e) PATH の設定

環境変数「PATH」には、オブジェクト操作ツールのインストールディレクトリを追加します。オブジェクト操作ツールは、「/opt/HiEDMS/client/tools」および「/opt/HiEDMS/client/tools_tp5」ディレクトリにインストールされます。したがって、追加する値は、TPBroker V3 を使用する場合は、「:/opt/HiEDMS/client/tools」、TPBroker V5 を使用する場合は、「:/opt/HiEDMS/client/tools_tp5」です。

(f) DBRTOOLS DIR の設定

環境変数「DBRTOOLS DIR」には、オブジェクト操作ツールの実行環境ディレクトリを設定してください。

(g) DBR_DETAIL_ERRORLOG の設定 (詳細メッセージを標準エラー出力に出力する場合)

環境変数「DBR_DETAIL_ERRORLOG」は、詳細メッセージを取得する場合に設定します。詳細メッセージを取得する場合に設定する値は「ON」です。

2. 実行環境の設定

環境変数「DBR_DETAIL_ERRORLOG」に「ON」が設定されている場合、オブジェクト操作ツール実行時の詳細メッセージが標準エラー出力および詳細エラーログファイルに出力されます。

出力される詳細メッセージは、オブジェクト操作ツールのコマンドが使用している、クラスライブラリのメソッドに関する詳細メッセージです。

この環境変数を省略した場合または値に「ON」以外を設定した場合は、詳細メッセージは標準エラー出力および詳細エラーログファイルのどちらにも出力されません。

出力される詳細メッセージの内容については、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

(h) PSALLOC の設定

環境変数「PSALLOC」には、メモリ領域の確保をメモリ使用時でなく、メモリ領域要求時に行う、早期ページスペース割り当てを実行するための設定をします。設定する値は「early」です。

(i) NODISCLAIM の設定

環境変数「NODISCLAIM」には、メモリ領域の解放を抑止するための設定をします。設定する値は「true」です。

(j) 注意事項

環境変数「EDMCLASSDEFPATH」が設定されている環境でオブジェクト操作ツールを使用する場合、オブジェクト操作ツールの実行時には、オブジェクト操作ツール実行環境ディレクトリ下に存在するクラス定義情報ファイルではなく、環境変数「EDMCLASSDEFPATH」に設定されたディレクトリに存在するクラス定義情報ファイルが参照されます。

なお、文書空間で使用する文字コード種別が UTF-8 の場合、環境変数「EDMCLASSDEFPATH」には、印刷可能な ASCII コードで値を設定してください。

ただし、この場合も、文書空間識別子は、オブジェクト操作ツール実行環境ディレクトリ下の実行環境制御ファイルに記述されているものが使用されます。

(2) 環境変数の設定 (Windows の場合)

環境変数を設定します。

全ユーザで共通の値にする場合は、「システム環境変数」に設定してください。ログインユーザごとに設定する場合は、「ユーザ環境変数」に設定してください。

環境変数は [コントロールパネル] の [システム] の [環境変数] ボタンをクリックして表示される [環境変数] ダイアログで設定します。なお、詳細については、各 OS のヘルプで確認してください。

(a) PATH の設定

環境変数「PATH」には、オブジェクト操作ツールのインストールディレクトリを追加します。オブジェクト操作ツールは、「<インストールディレクトリ>%tools」ディレクトリにインストールされます。したがって、インストールディレクトリが「C:\Program Files\HITACHI\DocBroker\Client」の場合、追加する値は、「;C:\Program Files\HITACHI\DocBroker\Client\tools」です。

(b) DBRTOOLS DIR の設定

環境変数「DBRTOOLS DIR」には、オブジェクト操作ツールの実行環境ディレクトリを設定してください。

文書空間で使用する文字コード種別が UTF-8 の場合、印刷可能な ASCII コードで値を設定してください。

(c) DBR_DETAIL_ERRORLOG の設定 (詳細メッセージを標準エラー出力に出力する場合)

環境変数「DBR_DETAIL_ERRORLOG」は、詳細メッセージを取得する場合に設定します。詳細メッセージを取得する場合に設定する値は「ON」です。

環境変数「DBR_DETAIL_ERRORLOG」に「ON」が設定されている場合、オブジェクト操作ツール実行時の詳細メッセージが標準エラー出力および詳細エラーログファイルに出力されます。

出力される詳細メッセージは、オブジェクト操作ツールのコマンドが使用している、クラスライブラリのメソッドに関する詳細メッセージです。

この環境変数を省略した場合または値に「ON」以外を設定した場合は、詳細メッセージは標準エラー出力および詳細エラーログファイルのどちらにも出力されません。

出力される詳細メッセージの内容については、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

(d) 注意事項

環境変数「EDMCLASSDEFPATH」が設定されている環境でオブジェクト操作ツールを使用する場合、オブジェクト操作ツールの実行時には、オブジェクト操作ツール実行環境ディレクトリ下に存在するクラス定義情報ファイルではなく、環境変数「EDMCLASSDEFPATH」に設定されたディレクトリに存在するクラス定義情報ファイルが参照されます。

なお、文書空間で使用する文字コード種別が UTF-8 の場合、環境変数「EDMCLASSDEFPATH」には、印刷可能な ASCII コードで値を設定してください。

ただし、この場合も、文書空間識別子は、オブジェクト操作ツール実行環境ディレクトリ下の実行環境制御ファイルに記述されているものが使用されます。

2.2.3 実行環境のセットアップ

オブジェクト操作ツールが提供している実行環境セットアップコマンド「dbrtoolsetup」を実行します。形式は次のとおりです。

```
dbrtoolsetup -x CREATE 実行環境ディレクトリ
```

実行環境ディレクトリは、環境変数「DBRTOOLS DIR」に設定したディレクトリを指定します。なお、dbrtoolsetup の詳細については、「5. コマンドリファレンス」の「dbrtoolsetup (実行環境のセットアップ)」を参照してください。

2.2.4 実行環境制御ファイルの編集

実行環境制御ファイルは、オブジェクト操作ツールの実行環境を定義するファイルです。実行環境セットアップコマンドを実行すると、UNIX の場合は「実行環境ディレクトリ/etc」に、Windows の場合は「実行環境ディレクトリ\etc」に、サンプルファイルがコピーされます。オブジェクト操作ツールの実行環境に合わせて、このサンプルファイルを編集してください。なお、実行環境制御ファイルの詳細については、「3.2 実行環境制御ファイル」を参照してください。

2.2.5 クラス定義情報ファイルの取得

クラス定義情報ファイルは、DocumentBroker で定義されている DMA オブジェクトのクラスもしくはそ

2. 実行環境の設定

のサブクラスのクラス名, またはプロパティ名から GUID, データ型, 基本単位などの情報を取得するために使用します。

実行環境セットアップコマンドを実行した時に, UNIX の場合は「実行環境ディレクトリ/etc」に, Windows の場合は「実行環境ディレクトリ ¥etc」にコピーされるクラス定義情報ファイルには, オブジェクト操作ツールが動作するための最低限の情報だけが記述されています。このため, オブジェクト操作ツールを運用する場合には, 接続先の DocumentBroker サーバの情報が記述されているクラス定義情報ファイルを取得する必要があります。

クラス定義情報ファイルは, DocumentBroker サーバが提供しているクラス定義情報ファイル出力コマンド (EDMCrtSimMeta コマンド) によって取得してください。このコマンドで取得したクラス定義情報ファイルを, UNIX の場合は「実行環境ディレクトリ/etc」に, Windows の場合は「実行環境ディレクトリ ¥etc」にコピーしてください。

クラス定義情報ファイルの詳細については, 「3.3 クラス定義情報ファイル」を参照してください。

2.2.6 実行環境の更新

実行環境制御ファイルおよびクラス定義情報ファイルを編集した場合は, 実行環境セットアップコマンドを再実行して, オブジェクト操作ツールの実行環境を更新します。

また, DocumentBroker Development Kit または DocumentBroker Runtime をバージョンアップした場合も, 必ずセットアップコマンドを再実行して, オブジェクト操作ツールの実行環境を更新してください。

形式は次のとおりです。

```
dbrtoolsetup -x UPDATE 実行環境ディレクトリ
```

実行環境ディレクトリは, 環境変数「DBRTOOLS DIR」に設定したディレクトリを指定します。なお, dbrtoolsetup の詳細については, 「5. コマンドリファレンス」の「dbrtoolsetup (実行環境のセットアップ)」を参照してください。

2.3 ファイル転送機能の使用

オブジェクト操作ツールを実行するマシンと異なるマシンに構築された DocumentBroker サーバに接続する場合は、ファイル転送機能を使用してください。ファイル転送機能については、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。

3

オブジェクト操作ツールで使用するファイル

この章では、オブジェクト操作ツールで使用するファイルについて説明します。

3.1 オブジェクト操作ツールで使用するファイル一覧

3.2 実行環境制御ファイル

3.3 クラス定義情報ファイル

3.4 ID ファイル

3.5 プロパティ情報ファイル

3.6 問い合わせファイル

3.7 アップロード情報ファイル

3.8 リファレンス情報ファイル

3.1 オブジェクト操作ツールで使用するファイル一覧

オブジェクト操作ツールで使用するファイルの一覧を次の表に示します。

表 3-1 オブジェクト操作ツールで使用するファイル一覧

名称	説明	作成場所
実行環境制御ファイル	オブジェクト操作ツールの実行環境を定義する	UNIX の場合 実行環境ディレクトリ /etc Windows の場合 実行環境ディレクトリ \etc
クラス定義情報ファイル	オブジェクト操作ツールで使用するクラスおよびプロパティに関する情報を定義する	UNIX の場合 実行環境ディレクトリ /etc Windows の場合 実行環境ディレクトリ \etc
ID ファイル	複数のオブジェクトを一括で操作する場合に、操作対象となるオブジェクトの ID または識別子を記述する	任意
プロパティ情報ファイル	操作するオブジェクトのプロパティ情報を記述する	任意
問い合わせファイル	オブジェクトの検索に使用する問い合わせ式を記述する	任意
アップロード情報ファイル	アップロードするマルチファイル文書の情報を記述する	任意
リファレンス情報ファイル	操作するリファレンスファイル文書の情報を記述する	任意

注

環境変数「EDMCLASSDEFPATH」が設定されている環境では、環境変数「EDMCLASSDEFPATH」に設定されているディレクトリに存在するクラス定義情報ファイルが使用されます。

3.2 実行環境制御ファイル

実行環境制御ファイルは、オブジェクト操作ツールの実行環境を定義するファイルです。-x CREATE オプションを指定して実行環境セットアップコマンド「dbrtoolsetup」を実行すると、UNIX の場合は「実行環境ディレクトリ/etc」に、Windows の場合は「実行環境ディレクトリ\etc」に、サンプルファイルがコピーされます。オブジェクト操作ツールの実行環境に合わせて、編集して使用してください。

3.2.1 記述規則

実行環境制御ファイルの記述規則は次のとおりです。

ファイル名は「EDMOotCtrl.ini」で固定です。

印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

空白行は無視します。

行の終端は改行コードまたは EOF です。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR+NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

3.2.2 記述形式

実行環境制御ファイルは、五つのセクションと各セクションに指定するエントリによって構成されます。

セクションとエントリは、次のように記述します。

- セクション名は、角括弧 ([]) で囲んで指定します。一つのセクションは、セクション名を指定してから、次のセクション名を指定するまで、またはファイルの終端までの範囲です。
- エントリは、「エントリ名 = 指定値」の形式で指定します。

記述形式は次のとおりです。

```
[<セクション名>]
[<エントリ名>=<値> ]
```

注

エントリの値は、その値が数値か文字列かによって記述方法が異なります。

- 数値の場合

任意の 10 進数を -2,147,483,648 ~ 2,147,483,647 の間で記述します。

- 文字列の場合

指定できる文字列が限られている場合は、そのうちのどれかの文字列を記述します。任意の文字列を指定する場合は、1,039 バイト以内の文字列をシングルクォート (') で囲んで記述します。

各セクションとセクションごとに指定するエントリについて説明します。

なお、各セクション内でエントリが重複して記述されている場合は、最初に指定されたエントリが有効となります。

(1) Session セクション

このセクションには、DocumentBroker サーバの文書空間に接続するための情報を記述します。セクションを構成するエントリについて、次に説明します。なお、このセクションおよび各エントリの記述は省略できません。省略した場合は、オブジェクト操作ツールの実行時にエラーとなります。

DocSpaceId エントリ

接続先の DocumentBroker サーバの文書空間 ID をシングルクォート (') で囲んで記述します。

< 記述例 >

```
DocSpaceId = '673d2be0-d1fd-11d0-ab59-08002be29e1d'
```

UserName エントリ

オブジェクト操作ツールの実行ユーザとして、接続先の DocumentBroker サーバに登録されているユーザ名をシングルクォート (') で囲んで記述します。なお、DocumentBroker サーバで使用しているユーザ管理機能に登録されていないユーザ名を指定した場合は、文書空間に接続するときにエラーが発生します。ユーザ管理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。

< 記述例 >

```
UserName = 'user01'
```

Password エントリ

UserName エントリに指定したユーザが、接続先の DocumentBroker サーバにログインするために使用するパスワードをシングルクォート (') で囲んで記述します。

< 記述例 >

```
Password = 'password01'
```

(2) Log セクション

このセクションには、オブジェクト操作ツールを実行するときに出力されるコマンド実行ログを制御する情報を記述します。セクションを構成するエントリについて、次に説明します。なお、このセクションを省略した場合、各エントリの値はデフォルト値が仮定されます。

Level エントリ

コマンド実行ログの出力レベルを指定します。出力レベルには、「None」、「Write」、「Read」または「Error」を記述します。デフォルトは「None」です。省略した場合は、デフォルト値が仮定されます。不正な値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

指定する値についての詳細は、「4.1.2 出力レベル」を参照してください。

< 記述例 >

```
Level = Read
```

FileCount エントリ

コマンド実行ログを出力するファイルサイズの上限を超えた場合に、切り替えるファイルの数を、2 ~ 16 の間で記述します。デフォルトは 2 です。省略した場合は、デフォルト値が仮定されます。このエントリで指定できる範囲外の値が指定されている場合は、デフォルト値が仮定されます。なお、数値として指定できる範囲外の値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

< 記述例 >

```
FileCount = 4
```

FileSize エントリ

コマンド実行ログを出力するファイルサイズを 4,096 ~ 2,147,483,647 (単位はバイト) の間で記述

します。デフォルトは 1,048,576 です。省略した場合は、デフォルト値が仮定されます。このエントリで指定できる範囲外の値が指定されている場合は、デフォルト値が仮定されます。なお、数値として指定できる範囲外の値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

<記述例>

FileSize = 8192

Directory エントリ

コマンド実行ログの出力先ディレクトリを、シングルクォート (') で囲んで記述します。デフォルトは、UNIX の場合は「実行環境ディレクトリ /spool/aclog」、Windows の場合は「実行環境ディレクトリ %spool%aclog」です。省略した場合は、デフォルト値が仮定されます。不正な値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

<記述例>

Directory = 'c:%home%user01%aclog'

Prefix エントリ

コマンド実行ログを出力するファイル名として使用するプリフィックスを、シングルクォート (') で囲んで記述します。デフォルトは「EDMOotAccess」です。省略した場合は、デフォルト値が仮定されます。不正な値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

<記述例>

Prefix= 'Tool'

(3) Output セクション

このセクションには、コマンドの実行結果の出力を制御するための情報を記述します。セクションを構成するエントリについて、次に説明します。なお、このセクションを省略した場合、各エントリの値はデフォルト値が仮定されます。

LineSize エントリ

コマンドの実行結果を出力するときに、1 行当たりの出力サイズを記述します。1,022 ~ 2,147,483,645 (単位はバイト) で記述します。デフォルトは 8,190 です。省略した場合は、デフォルト値が仮定されます。このエントリで指定できる範囲外の値が指定されている場合は、デフォルト値が仮定されます。なお、数値として指定できる範囲外の値が指定されている場合は、オブジェクト操作ツールの実行時にエラーとなります。

<記述例>

LineSize = 1048576

Format エントリ

一覧取得コマンドの出力形式を選択します。「VR0200」または「VR0112」を指定します。「VR0200」を指定すると、一覧取得コマンドの出力情報として、リンク識別子なども出力されます。「VR0112」を指定すると、Version 1 の出力形式と同じ形式で出力します。この場合、リンク識別子は出力されません。デフォルトは、「VR0200」です。

指定する値についての詳細は、「5. コマンドリファレンス」の「コマンド実行結果」および各コマンドの実行例を参照してください。

<記述例>

Format = VR0200

(4) FileLink セクション

このセクションには、File Link 連携機能を使用する場合の情報を記述します。

3. オブジェクト操作ツールで使用するファイル

セクションを構成するエントリについて、次に説明します。なお、このセクションを省略した場合、各エントリの値はデフォルト値が仮定されます。

DefaultFAMName エントリ

コンテンツの登録先となるファイルサーバの FAM 名を、シングルクォート (') で囲んで記述します。このエントリの指定は次の場合に有効になります。

- プロパティ情報ファイルの filelink_famname エントリの指定を省略して、dbrertdoc コマンド (文書の作成) を実行した場合
- -j オプションを省略して、dbrnrdrdoc コマンド (文書のレンディション操作), または dbrupdoc コマンド (文書のアップロード) を実行した場合

なお、ファイルサーバの FAM 名の詳細については、マニュアル「HiRDB File Link」を参照してください。

<記述例>

DefaultFAMName = 'fam01'

DefaultContentPath エントリ

コンテンツの登録先となるファイルサーバ上のディレクトリのパスを、シングルクォート (') で囲んで記述します。ディレクトリのパスの末尾にはデリミタ「/」(UNIX の場合) または「¥」(Windows の場合) を記述します。

このエントリの指定は次の場合に有効になります。

- プロパティ情報ファイルの filelink_contentpath エントリの指定を省略して、dbrertdoc コマンド (文書の作成) を実行した場合
- -g オプションを省略して、dbrnrdrdoc コマンド (文書のレンディション操作), または dbrupdoc コマンド (文書のアップロード) を実行した場合

エントリに指定したディレクトリが存在しない場合、ディレクトリが作成され、コンテンツが登録されます。ただし、ディレクトリのパスの末尾にデリミタ「/」(UNIX の場合) または「¥」(Windows の場合) を記述しない場合、コンテンツ名称の自動生成を行わない (FileNamingMode エントリに「Specify」を指定している) 環境で、このエントリに指定したディレクトリがファイルサーバ上に存在するとエラーになり、存在しないとこのエントリに指定した名称でコンテンツが登録されます。なお、コンテンツの登録先ディレクトリの詳細については、マニュアル「HiRDB File Link」を参照してください。

<記述例>

UNIX の場合に、コンテンツの登録先ディレクトリが、「/dir01/data」の下位にある「area1」ディレクトリのとときの記述例を次に示します。

指定形式の種別	指定例
フルパス指定	DefaultContentPath = '/dir01/data/area1/'
エイリアス指定	DefaultContentPath = 'DATA/area1/' (「/dir01/data」のエイリアス名が「DATA」のとき)

Windows の場合に、コンテンツの登録先ディレクトリが、「c:¥dir01¥data」の下位にある「area1」ディレクトリのとときの記述例を次に示します。

指定形式の種別	指定例
フルパス指定	DefaultContentPath = 'c:¥dir01¥data¥area1¥'
エイリアス指定	DefaultContentPath = 'DATA¥area1¥' (「c:¥dir01¥data」のエイリアス名が「DATA」のとき)

FileNamingMode エントリ

ファイルサーバに登録されるコンテンツの名称付与モードを指定します。名称付与モードには、「Auto」または「Specify」を指定します。

「Auto」を指定した場合、HiRDB File Link のコンテンツの名称を自動生成する機能を使用して、登録するコンテンツの名称を自動生成します。

「Specify」を指定した場合、プロパティ情報ファイルの url エントリに指定されたファイルパスのファイル名が、ファイルサーバに登録されるコンテンツの名称となります。

デフォルトは「Specify」です。省略した場合は、デフォルト値が仮定されます。なお、HiRDB File Link のコンテンツの名称を自動生成する機能の詳細については、マニュアル「HiRDB File Link」を参照してください。

< 記述例 >

```
FileNamingMode = Specify
```

ClientContentDeleteMode エントリ

クライアントのコンテンツを指定して、文書を作成もしくはアップロードする場合、またはレンディションを追加する場合に設定される、文書、文書のバージョン、または文書のレンディションを削除したときの、ファイルサーバ上のコンテンツの削除モードを指定します。コンテンツの削除モードには、「Sync」または「Async」を指定します。

「Sync」を指定した場合、文書、文書のバージョン、またはレンディションの削除と同期してファイルサーバ上のコンテンツを削除します。

「Async」を指定した場合、文書、文書のバージョン、またはレンディションを削除してもファイルサーバ上のコンテンツは削除されません。

デフォルトは「Sync」です。省略した場合は、デフォルト値が仮定されます。

コンテンツの削除モードは、文書を作成またはアップロードした時点、もしくはレンディションを追加した時点の設定が有効になります。文書、文書のバージョン、またはレンディションを削除する時に設定を変更しても有効になりません。

< 記述例 >

```
ClientContentDeleteMode = Sync
```

ServerContentDeleteMode エントリ

ファイルサーバに登録されているコンテンツを指定して、文書を作成もしくはアップロードする場合、またはレンディションを追加する場合に設定される、文書、文書のバージョン、またはレンディションを削除したときの、ファイルサーバ上のコンテンツの削除モードを指定します。コンテンツの削除モードには、「Sync」または「Async」を指定します。

「Sync」を指定した場合、文書、文書のバージョン、またはレンディションの削除と同期してファイルサーバ上のコンテンツを削除します。

「Async」を指定した場合、文書、文書のバージョン、またはレンディションを削除してもファイルサーバ上のコンテンツは削除されません。

デフォルトは「Async」です。省略した場合は、デフォルト値が仮定されます。

コンテンツの削除モードは、文書を作成またはアップロードした時点、もしくはレンディションを追加した時点の設定が有効になります。文書、文書のバージョン、またはレンディションを削除する時に設定を変更しても有効になりません。

なお、コンテンツを多重登録する場合（すでに File Link 連携機能を使用してファイルサーバに登録されたコンテンツを指定して文書を作成もしくはアップロードする場合、またはレンディションを追加する場合は、このエントリを設定を、前回コンテンツを使用した時と同じ設定にする必要があります。

< 記述例 >

```
ServerContentDeleteMode = Async
```

(5) Reference セクション

このセクションには、リファレンスファイル管理機能を使用する場合の情報を記述します。セクションを構成するエントリについて、次に説明します。

DefaultReferenceType エントリ

デフォルトの、コンテンツの格納先とリファレンス種別を指定します。リファレンス種別には「Relative」を指定してください。

このエントリの指定は次の場合に有効になります。

- プロパティ情報ファイルの `reference_type` エントリの指定を省略して、`dbrcrtdoc` コマンド（文書の作成）を実行した場合
- `-a` オプションを省略して、`dbrdelcntr` コマンド（コンテナの削除）、`dbrdeldoc` コマンド（文書の削除）、`dbrdowndoc` コマンド（文書のダウンロード）、または `dbrupdoc` コマンド（文書のアップロード）を実行した場合

「Relative」を指定した場合、コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理します。

<記述例>

```
DefaultReferenceType = Relative
```

DefaultContentBasePath エントリ

リファレンスファイル文書のリファレンス種別が「Relative」の場合、デフォルトのコンテンツ格納先ベースパスをシングルクォート（'）で囲んで記述します。

このエントリの指定は次の場合に有効になります。

- プロパティ情報ファイルの `reference_contentbasepath` エントリの指定を省略して、`dbrcrtdoc` コマンド（文書の作成）を実行した場合
- リファレンス情報ファイルの `reference_contentbasepath` エントリの指定を省略して、`dbrdelcntr` コマンド（コンテナの削除）を実行した場合
- `-b` オプションを省略して、`dbrdelcntr` コマンド（コンテナの削除）、`dbrdeldoc` コマンド（文書の削除）、`dbrdowndoc` コマンド（文書のダウンロード）、または `dbrupdoc` コマンド（文書のアップロード）を実行した場合

このエントリには、リファレンスファイル文書の格納先となるディレクトリの基点となるパスを指定します。例えば、コンテンツを「/drv01/dir01/userdir」ディレクトリの下位にある「user01」ディレクトリに格納する場合、「user01」ディレクトリの基点となるパスとして、「/drv01/dir01/userdir」をシングルクォート（'）で囲んで指定します。このエントリの指定の末尾に、パスのデリミタ「/」（UNIX の場合）または「¥」（Windows の場合）を記述する必要はありません。

このエントリに指定したコンテンツ格納先ベースパスは、コンテンツ格納先パスと結合して絶対パスとして使用されます。ContentLocation プロパティには次に示すパスが格納されます。

```
aaaa/bbbb/cccc
```

aaaa : コンテンツ格納先パス

bbbb : DocumentBroker がコンテンツを管理するためのディレクトリ

cccc : ファイル名

注

次に示すどちらかのファイル名が設定されます。

- 登録ファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合）
- DocumentBroker が指定するファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合）

Windows の場合、デフォルトのコンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、「\\server01\share」をシングルクォート（'）で囲んで指定します。

< 記述例 >

```
DefaultContentBasePath = '/drv01/dir01/userdir'
```

DefaultDeleteRootPath エントリ

リファレンスファイル文書と、リファレンスファイル文書が格納されているディレクトリを同時に削除する場合、デフォルトの削除ディレクトリのルートパスをシングルクォート（'）で囲んで記述します。

このエントリの指定は次の場合に有効になります。

- リファレンス情報ファイルの reference_deleteerootpath エントリの指定を省略して、dbrdelcntr コマンド（コンテナの削除）を実行した場合
- -y オプションを省略して、dbrdelcntr コマンド（コンテナの削除）または dbrdeldoc コマンド（文書の削除）を実行した場合

このエントリには、削除するディレクトリのルートパスを指定します。例えば、「/drv01/dir01/userdir/user01/<DocumentBroker がコンテンツを管理するためのディレクトリ ><登録ファイル名 >」のディレクトリ中の「user01」ディレクトリ以下のディレクトリを削除する場合は、「user01」ディレクトリのルートパスとして、「/drv01/dir01/userdir」をシングルクォート（'）で囲んで指定します。ただし、コンテンツ格納先ベースパスの上位のディレクトリは指定できません。また、エントリの指定の末尾に、パスのデリミタ「/」（UNIX の場合）または「\」（Windows の場合）を記述する必要はありません。

DocumentBroker がコンテンツを管理するためのディレクトリは、このエントリまたはコマンドオプションで削除ディレクトリのルートパスの指定を省略した場合も、コンテンツと同時に削除されます。Windows の場合、デフォルトの削除ディレクトリのルートパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンがネットワーク上で共有しているフォルダ「share」を削除ディレクトリのルートパスに指定する場合は、「\\server01\share」をシングルクォート（'）で囲んで指定します。

< 記述例 >

```
DefaultDeleteRootPath = '/drv01/dir01/userdir'
```

ContentDeleteMode エントリ

リファレンスファイル文書を削除する場合の削除モードを指定します。削除モードには、「Auto」または「Specify」を指定します。デフォルト値は「Specify」です。

「Auto」を指定した場合、DocumentBroker が自動的にリファレンスファイル文書かどうかを判別して、リファレンスファイル文書を削除します。

「Specify」を指定した場合、dbrdelcntr コマンド（コンテナの削除）または dbrdeldoc コマンド（文書の削除）の、-k オプションの指定に従って文書を削除します。このとき、-k オプションを省略すると、エラーになります。ContentDeleteMode エントリの指定と dbrdelcntr コマンド（コンテナの削除）および dbrdeldoc コマンド（文書の削除）の実行結果の対応を次の表に示します。

表 3-2 ContentDeleteMode エントリの指定と dbrdelcntr コマンド、dbrdeldoc コマンドの実行結果

削除する文書の種類	ContentDeleteMode エントリの指定	-k オプションの指定	実行結果
リファレンスファイル文書	Auto	Reference	
		-	

3. オブジェクト操作ツールで使用するファイル

削除する文書の種類	ContentDeleteMode エントリの指定	-k オプションの指定	実行結果
	Specify	Reference	
		-	×
リファレンスファイル文書以外	Auto	Reference	
		-	
	Specify	Reference	
		-	

(凡例)

- : 指定しません。
- : 削除されます。
- × : エラーになるため、削除されません。

<記述例>

ContentDeleteMode = Auto

3.2.3 実行環境制御ファイルの記述例

-x CREATE オプションを指定して実行環境セットアップコマンドを実行した場合にコピーされる実行環境制御ファイルは次のとおりです。

```
;; Session
[Session]
DocSpaceId = '673d2be0-d1fd-11d0-ab59-08002be29e1d'
UserName = 'user'
Password = 'passwd'
```

これは、オブジェクト操作ツールが動作する最小限の内容です。

このファイルを、それぞれのオブジェクト操作ツールの実行環境に合わせて編集します。「UserName」と「Password」は必ず編集してください。

Windows の場合の記述例を次に示します。UNIX の場合は、[FileLink] セクションの Directory エントリを、「/home/user01/aclog」のように指定し、DefaultContentPath エントリを「/dir01/data/area1/」のように指定してください。また、[Reference] セクションの DefaultContentBasePath エントリを「/drv01/dir01/userdir」のように指定し、DefaultDeleteRootPath エントリを「/drv01/dir01/userdir」のように指定してください。

<記述例>

```
;; Session
[Session]
DocSpaceId = '673d2be0-d1fd-11d0-ab59-08002be29e1d'
UserName = 'user01'
Password = 'passwd01'

;; Log
[Log]
Level = Read
FileCount = 4
FileSize = 8192
Directory = 'c:¥home¥user01¥aclog'
Prefix = 'Tool'
```

```
;; Output
[Output]
LineSize = 1048576
Format = VR0200

;; FileLink
[FileLink]
DefaultFAMName = 'fam01'
DefaultContentPath = 'c:¥dir01¥data¥areal¥'

;; Reference
[Reference]
DefaultReferenceType = Relative
DefaultContentBasePath = 'c:¥dir01¥userdir'
DefaultDeleteRootPath = 'c:¥dir01¥userdir'
ContentDeleteMode = Auto
```

3.3 クラス定義情報ファイル

クラス定義情報ファイルは、DocumentBroker サーバで定義されている DMA オブジェクトのクラスもしくはそのサブクラスのクラス名、またはプロパティ名から GUID、データ型、基本単位などの情報を取得するために使用するファイルです。したがって、オブジェクト操作ツールで使用するクラス定義情報ファイルの内容は、接続する DocumentBroker サーバで実際に定義されている情報と一致している必要があります。

-x CREATE オプションを指定して実行環境セットアップコマンドを実行した時に、UNIX の場合は「実行環境ディレクトリ /etc」に、Windows の場合は「実行環境ディレクトリ \etc」にコピーされたサンプルファイルには、オブジェクト操作ツールが動作するための最小限の情報だけが記述されています。このため、オブジェクト操作ツールを運用する時には、DocumentBroker サーバが提供しているクラス定義情報ファイル出力コマンド「EDMCrtSimMeta」を実行して、接続先の DocumentBroker サーバの情報が記述されているクラス定義情報ファイルを取得してください。取得したクラス定義情報ファイルを、UNIX の場合は「実行環境ディレクトリ /etc」に、Windows の場合は「実行環境ディレクトリ \etc」にコピーして、再度実行環境セットアップコマンドを実行して、実行環境を更新してください。

クラス定義情報ファイル出力コマンドの文法については、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。

注意

環境変数「EDMCLASSDEFPATH」が設定されている環境では、実行環境ディレクトリ /etc または実行環境ディレクトリ \etc のクラス定義情報ファイルではなく、環境変数「EDMCLASSDEFPATH」に設定されているディレクトリに存在するクラス定義情報ファイルが使用されます。

3.3.1 ファイル名の規則

クラス定義情報ファイル名は、実行環境制御ファイルの DocSpaceId エントリに指定した接続文書空間 ID にサフィックス「.ini」を付与した名前にしてください。ファイル名のプリフィックスとして使用する GUID は、実行環境制御ファイルの DocSpaceId エントリに指定した接続先文書空間 ID と一致するようにしてください。

3.3.2 記述規則

クラス定義情報ファイルの記述規則は次のとおりです。

印刷可能な ASCII コードで記述してください。

セミコロン (;) で始まる行はコメント行として扱われます。なお、シャープ (#) で始まる行はコメント行として扱われません。

空白行は無視します。

行の終端は改行コードまたは EOF です。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

3.3.3 記述形式

クラス定義情報ファイルは、クラスの定義情報を記述するセクションとプロパティの定義情報を記述する

セクションで構成されます。

セクションとエントリは、次のように記述します。

- セクション名は、角括弧 ([]) で囲んで指定します。一つのセクションは、セクション名を指定してから、次のセクション名を指定するまで、またはファイルの終端までの範囲です。
- エントリは、「エントリ名 = 指定値」の形式で指定します。

記述形式は次のとおりです。

```
[<クラス名>
ClassId=guid=<クラス識別子>
SuperClass=text=<スーパークラス名>
(<プロパティ名>=guid=<プロパティ識別子>)...
(
<プロパティ識別子>
dmaProp_DataType=int= { DMA_DATATYPE_BOOLEAN |
                        DMA_DATATYPE_INTEGER32 |
                        DMA_DATATYPE_OBJECT |
                        DMA_DATATYPE_STRING }
dmaProp_Cardinality=int= { DMA_CARDINALITY_SINGLE |
                           DMA_CARDINALITY_LIST |
                           DMA_CARDINALITY_ENUM |
                           255 }
{ dmaProp_RequiredClass=obj=<クラス名> }
)...
```

次に、各セクションとエントリについて説明します。

(1) クラスの定義情報を記述するセクション

クラスの定義情報を記述します。

クラス名

DocumentBroker サーバのメタ情報ファイル (edms.ini) で dmaProp_DisplayName プロパティに指定する値で、スラント (/) の後ろに記述するクラスの名前をセクション名に記述してください。

<記述例>

```
[usrClass_DocVersion]
```

クラス識別子

DocumentBroker サーバのメタ情報ファイル (edms.ini) で dmaProp_Ids プロパティに指定するクラスのクラス識別子 (GUID) を、ClassId エントリに「=guid=」を区切り文字として記述してください。

GUID は、次の形式で指定してください。X は 0 ~ 9 および a ~ f (小文字) で表される 16 進数の値です。

「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX」(8けた-4けた-4けた-4けた-12けた)

<記述例>

```
ClassId=guid=01a3a891-7aec-11d1-a31b-0020af9fbb1c
```

スーパークラス名

DocumentBroker サーバのメタ情報ファイル (edms.ini) で dmaProp_DisplayName プロパティに指定する値のうち、スラント (/) の前に記述するスーパークラスの名前を、SuperClass エントリに「=text=」を区切り文字として記述してください。

<記述例>

3. オブジェクト操作ツールで使用するファイル

SuperClass=text=dmaClass_DocVersion

プロパティ名

DocumentBroker サーバのメタ情報ファイル (edms.ini) でプロパティの dmaProp_DisplayName プロパティに指定する値をエントリ名として指定してください。指定したエントリに「=guid=」を区切り文字として、プロパティの識別子 (GUID) を記述してください。オブジェクト操作ツールで使用するプロパティが複数ある場合は、すべて記述してください。

<記述例>

usrProp_Title=guid=01234567-0123-0123-0123-012345678901

プロパティ識別子

「(2) プロパティの定義情報を記述するセクション」の「プロパティ識別子」を参照してください。

(2) プロパティの定義情報を記述するセクション

プロパティの定義情報を記述します。クラス定義情報ファイル中に同じプロパティの情報を定義するセクションが記述されていてもチェックしません。ファイル中で先に指定されたプロパティの記述を有効とします。ただし、重複してプロパティの定義情報を記述すると、クラス定義情報ファイルのデータ量が増加して、プロパティ定義情報の探索処理性能が劣化することがあります。

プロパティ識別子

DocumentBroker サーバのメタ情報ファイル (edms.ini) でプロパティの dmaProp_Ids プロパティに指定するプロパティ識別子 (GUID) をセクション名に記述してください。

GUID は、次の形式で指定してください。X は 0 ~ 9 および a ~ f (小文字) で表される 16 進数の値です。

「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX」(8けた-4けた-4けた-4けた-12けた)

<記述例>

[01234567-0123-0123-0123-012345678901]

dmaProp_DataType (データ型)

DocumentBroker サーバのメタ情報ファイル (edms.ini) でプロパティの dmaProp_DataType プロパティに指定するプロパティのデータ型を dmaProp_DataType エントリに「=int=」を区切り文字として記述してください。指定できるデータ型は次のとおりです。() 内は、指定する文字列です。

- Boolean 型 (DMA_DATATYPE_BOOLEAN)
- Integer32 型 (DMA_DATATYPE_INTEGER32)
- Object 型 (DMA_DATATYPE_OBJECT)
- String 型 (DMA_DATATYPE_STRING)

<記述例>

dmaProp_DataType=int=DMA_DATATYPE_STRING

dmaProp_Cardinality (基本単位)

DocumentBroker サーバのメタ情報ファイル (edms.ini) でプロパティの dmaProp_Cardinality プロパティに指定するプロパティの基本単位を dmaProp_Cardinality エントリに「=int=」を区切り文字として記述してください。指定できる基本単位は次のとおりです。() 内は、指定する文字列です。

- Scalar 型 (DMA_CARDINALITY_SINGLE)
- List 型 (DMA_CARDINALITY_LIST)
- Enumeration 型 (DMA_CARDINALITY_ENUM)
- VariableArray 型 (255)

< 記述例 >

```
dmaProp_Cardinality=int=DMA_CARDINALITY_SINGLE
```

dmaProp_RequiredClass (配布オブジェクトクラス)

このエントリは、dmaProp_DataType エントリの値に「DMA_DATATYPE_OBJECT」を指定するときだけ記述してください。DocumentBroker サーバのメタ情報ファイル (edms.ini) で、dmaProp_DataType プロパティが Object 型で、dmaProp_Cardinality が VariableArray 型の場合には、dmaProp_RequiredClass エントリに「=obj=」を区切り文字として dmaProp_RequiredClass プロパティに指定する値のクラス名を記述してください。DocumentBroker サーバのメタ情報ファイルでファイルの指定がある場合は、単価記号 (@) の後ろに指定する値を記述してください。

< 記述例 >

```
dmaProp_RequiredClass=obj=usrClass_Author
```

3.3.4 クラス定義情報ファイルの記述例

クラス定義情報ファイルの記述例を次に示します。記述例は、次の条件に従って記述しています。

dmaClass_DocVersion クラスのサブクラスとして usrClass_DocVersion クラスを定義する

usrClass_DocVersion クラスには次のプロパティを追加する

- usrProp_Title プロパティ
データ型は String 型、基本単位は Scalar 型のプロパティ
- usrProp_Authors プロパティ
データ型は Object 型、基本単位は VariableArray 型のプロパティ

usrProp_Authors プロパティの構成要素を定義するクラスは usrClass_Author クラスである

```
;; usrClass_DocVersion
[usrClass_DocVersion]
ClassId=guid=01a3a891-7aec-11d1-a31b-0020af9fbb1c
SuperClass=text=dmaClass_DocVersion
usrProp_Title=guid=01234567-0123-0123-0123-012345678901
usrProp_Authors=guid=01234567-abcd-abcd-abcd-012345678901

;; usrProp_Title
[01234567-0123-0123-0123-012345678901]
dmaProp_DataType=int=DMA_DATATYPE_STRING
dmaProp_Cardinality=int=DMA_CARDINALITY_SINGLE

;; usrProp_Authors
[01234567-efab-efab-efab-012345678901]
dmaProp_DataType=int=DMA_DATATYPE_OBJECT
dmaProp_Cardinality=int=255
dmaProp_RequiredClass=obj=usrClass_Author

;; usrClass_Author
[usrClass_Author]
ClassId=guid=80ff3500-88b6-11d2-963f-00c04fbc3de9
SuperClass=text=edmClass_Struct
usrProp_Name=guid=01234567-8901-8901-8901-012345678901
usrProp_Age=guid=01234567-abcd-efab-cdef-012345678901
```

3.4 ID ファイル

ID ファイルは複数のオブジェクトを一括して操作する場合に、操作対象となるオブジェクトの OIID、バージョン識別子またはリンク識別子を記述するファイルです。

3.4.1 記述規則

ID ファイルの記述規則を次に示します。

ファイル名は任意です。

印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

空白行は無視します。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

3.4.2 記述形式

ID ファイルの記述形式を以下に示します。

ID または識別子は 1 行に一つずつ記述します。

ID または識別子の前後の空白およびタブは無視します。

ID または識別子が一つも記述されていない場合はエラーになります。

シングルクォート (') で囲んだ ID または識別子も指定できます。

記述可能な ID または識別子は次のとおりです。

- OIID
OIID の記述形式については、「付録 A OIID 指定時の URL の文法」を参照してください。
なお、OIID を指定した ID ファイルを OIID ファイルといいます。
- バージョン識別子
この識別子は、バージョン一覧取得コマンド (dbrlsversion コマンド) によって取得してください。
なお、バージョン識別子を指定した ID ファイルをバージョン識別子ファイルといいます。
- リンク識別子
この識別子は、コンテインメントを設定した時に出力されます。コンテインメントの設定後に取得する場合は、下位オブジェクト一覧取得コマンド (dbrlschild コマンド) または上位コンテナ一覧取得コマンド (dbrlsparent コマンド) によって取得してください。ただし、実行環境制御ファイルの Output セクションの Format エントリに「VR0112」を指定している場合は、リンク識別子を取得できません。
なお、リンク識別子を指定した ID ファイルをリンク識別子ファイルといいます。

3.4.3 ID ファイルの記述例

三つのオブジェクトを一括して操作する場合に指定する ID ファイルの記述例を次に示します。この例では、三つのオブジェクトの OIID を記述しています。なお、記述例中の「...」は、途中の省略を示します。

```
dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000001  
dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000002  
dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000003
```

また、次のようにシングルクォート（'）で囲んで指定することもできます。

```
'dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000001'  
'dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000002'  
'dma:///07a17522-a626-11d0-b11f-0020af27a837/...000000000003'
```

3.5 プロパティ情報ファイル

プロパティ情報ファイルは、次のようにプロパティを設定、参照または更新する場合に記述するファイルです。

- オブジェクトを作成する時にプロパティの初期値を設定する場合
- オブジェクトに設定されているプロパティを参照する場合
- オブジェクトに設定されているプロパティを更新する場合

3.5.1 記述規則

プロパティ情報ファイルの記述規則を次に示します。

ファイル名は任意です。

文書空間で使用する文字コード種別 `UTF-8`、または印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

空白行は無視します。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

注

文字コード種別に UTF-8 を使用する場合は、Windows のメモ帳を使わないでください。UTF-8 を前提とするテキストファイルを編集して保存する場合は、「BOM」制御コードが付加されないテキストエディタを使用してください。

Windows のメモ帳で編集して保存すると、ファイルの先頭に「BOM」と呼ばれる数バイトの制御コードが付加され、オブジェクト操作ツールが提供する機能やコマンドでエラーとなることがあります。

3.5.2 記述形式

プロパティ情報ファイルは複数のセクションから構成されます。各セクションは複数のエントリで構成されます。プロパティ情報ファイル中に同じエントリが指定されていてもチェックしません。

セクション内に同一のプロパティに対する記述が複数ある場合、プロパティの取得はできませんが、オブジェクトを作成するときおよびプロパティを設定するときには実行メソッドでエラーになります。プロパティ以外のエントリの場合は、最初に記述されているエントリを有効とします。

セクションとエントリは、次のように記述します。

- セクション名は、角括弧 ([]) で囲んで指定します。一つのセクションは、セクション名を指定してから、次のセクション名を指定するまで、またはファイルの終端までの範囲です。
- エントリは、「エントリ名 = 指定値」の形式で指定します。ただし、プロパティを取得する場合に使用するプロパティ情報ファイルでは、エントリ名だけを指定して、指定値を省略できます。この場合、指定値を指定しても無視されます。

記述形式は次のとおりです。

```
[<セクション名>]  
(
```

```

(<プロパティ>)...
{ link=<コンテンツの種別> }
{ mode=<構成管理モード種別> }
{ { url=<ファイルパス指定>
  | directory= <ディレクトリパス指定>
    [ files[<ファイル数>]={ <ファイル指定> } ] } }
{ mime='<MIME名>' }
{ filelink_famname= <コンテンツ登録先FAM名> }
{ filelink_contentpath= <コンテンツ登録先パス指定> }
{ reference_type= <リファレンス種別> }
{ reference_contentbasepath= <コンテンツ格納先ベースパス指定> }
{ reference_contentpath= <コンテンツ格納先パス指定> }
)

<プロパティ> ::= { <プロパティ設定> | <プロパティ取得> }
<プロパティ設定> ::= { <Scalar型プロパティ設定> |
  <VariableArray型プロパティ設定> }
<Scalar型プロパティ設定> ::= <プロパティ名>=<値>
<VariableArray型プロパティ設定> ::=
  <プロパティ名>[<配列数>][<構成要素プロパティ数>]=
  { <構成要素プロパティ設定> }
<構成要素プロパティ設定> ::=
  { <構成要素プロパティ名>=<値> [ , <構成要素プロパティ名>=<値> ] }
<プロパティ取得> ::= <プロパティ名>
<値> ::= { <Boolean型プロパティ値> |
  <Integer32型プロパティ値> |
  <String型プロパティ値> }
<Boolean型プロパティ値> ::= { TRUE | FALSE | UNKNOWN }
<Integer32型プロパティ値> ::= <任意の10進数>
<String型プロパティ値> ::= '<任意の文字列>'
<コンテンツの種別> ::= { DCR | RCR | VCR }
<構成管理モード種別> ::= { FLOATING | FIX }

<サブジェクト種別用プロパティ> ::=
  <サブジェクト種別用プロパティ名> = { USR | GRP | SYS }
<サブジェクト種別用プロパティ名> ::= dbrProp_SubjectType
<パーミッション種別用プロパティ設定> ::=
  <パーミッション種別用プロパティ名>=<パーミッション>
<パーミッション> ::= { NONE | CP | <パーミッション論理和> }
<パーミッション論理和> ::=
  { <パーミッション種別> | <パーミッション論理和> |
  <パーミッション種別> }
<パーミッション種別用プロパティ名> ::= { dbrProp_EveryonePermission |
  dbrProp_OwnerPermission |
  dbrProp_PrimaryGroupPermission |
  dbrProp_UserPermission |
  dbrProp_Permission }
<パーミッション種別> ::= { PRP | PWP | PRC | PWC | PL | PV | PD }

<ファイルパス指定> ::= { NONE | DBRSVCONTENT | '<ファイルURL>' }
<ファイルURL> ::= <"file:"で始まるURL>
<ディレクトリパス指定> ::= { NONE | '<ディレクトリURL>' }
<ディレクトリURL> ::= <"file:"で始まるURL>

<ファイル数> ::= <任意の10進数>
<ファイル指定> ::= { edmProp_RetrievalName = '<ファイル名>',
  [ edmProp_ComponentType = '<コンポーネント種別>' ] }, ...
<ファイル名> ::= <任意の文字列>
<コンポーネント種別> ::= <任意の文字列>
<コンテンツ登録先FAM名> ::= '<任意の文字列>'
<コンテンツ登録先パス指定> ::= '<任意の文字列>'
<コンテンツ格納先ベースパス指定> ::= '<任意の文字列>'
<コンテンツ格納先パス指定> ::= '<任意の文字列>'
<リファレンス種別> ::= { NONE | RELATIVE }

```

3. オブジェクト操作ツールで使用するファイル

注 dbrProp_ACLプロパティまたはdbrProp_SACLプロパティの場合だけ指定します。

記述形式に関する注意事項

files エントリ, <VariableArray 型プロパティ設定>, <構成要素プロパティ設定> および <ファイル指定> で示した波括弧 ({ }) は文字として記述してください。

また, <パーミッション論理和> で示したストローク (|) のうち, 2 番目のストローク (「<パーミッション論理和> | <パーミッション種別>」の「 | 」) は, 文字として記述してください。

そのほかの波括弧 ({ }) は, ストローク (|) で区切られた項目から一つを選択することを意味します。

次に, セクションと各エントリについて説明します。

(1) セクション名

プロパティ情報ファイルには, プロパティ値を設定または取得するオブジェクトを構成する DMA オブジェクトの, クラスまたはそのサブクラスのクラス名をセクション名として, 128 バイト以内の半角文字列で記述します。セクション名として記述するクラス名は, クラス定義情報ファイルに記述されている必要があります。セクションは, 一つ以上のエントリで構成されます。セクションを構成するエントリについて次に示します。

(2) プロパティエントリ

エントリ名には, セクション名に指定したクラスに定義されているプロパティのうち, 値を設定または取得するプロパティの名前を 128 バイト以内の半角文字列で記述します。プロパティの基本単位および種類 (用途) によって記述形式が異なります。

(a) Scalar 型プロパティ

プロパティの値を設定する場合と取得する場合とで記述形式が異なります。

Scalar 型プロパティの設定

値を設定するプロパティ名と設定する値を, 等号 (=) を区切り文字として記述します。プロパティのデータ型に対応した値の記述方法については, 「(16) 値の記述方法」の項目を参照してください。

<記述例>

```
usrProp_IsReadOnly=TRUE ( Boolean 型のプロパティの場合 )
usrProp_State=1 ( Integer32 型のプロパティの場合 )
usrProp_Title='Document' ( String 型のプロパティの場合 )
```

Scalar 型プロパティの取得

値を取得するプロパティ名を記述します。

<記述例>

```
usrProp_IsReadOnly
usrProp_State
usrProp_Title
```

(b) VariableArray 型プロパティ

プロパティの基本単位が VariableArray 型でデータ型が Object 型のプロパティについても, プロパティの値を設定する場合と取得する場合とで記述形式が異なります。

VariableArray 型プロパティの設定

値を設定するプロパティ名に続けて, 配列数と構成要素プロパティ数をそれぞれ角括弧 ([]) で囲ん

で記述します。その後ろに等号 (=) を区切り文字として、設定する値を記述します。

例えば、著者の情報を管理するために使用する VariableArray 型のプロパティ「usrProp_Authors」は、著者名を管理する「usrProp_Name」と著者の年齢を管理する「usrProp_Age」という二つのプロパティから構成されているとします。この「usrProp_Authors」というプロパティに対して、3人の著者情報を値として設定する場合は、配列数は3、構成要素プロパティ数は2となります。設定する値には、構成要素プロパティ名と値を等号 (=) を区切り文字として記述します。なお、指定する構成要素プロパティ名は、クラス定義情報ファイルに記述されている必要があります。記述例を次に示します。

< 記述例 >

```
usrProp_Authors[3][2] =
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',  usrProp_Age=32}
}
```

構成要素プロパティに対する値は、それぞれのプロパティのデータ型によって異なります。プロパティのデータ型に対応した値の記述方法については、「(16) 値の記述方法」の項目を参照してください。

VariableArray 型プロパティの取得

値を取得するプロパティ名を記述します。

< 記述例 >

```
usrProp_Authors
```

(c) パーミッション種別設定用プロパティ

パーミッション種別用プロパティ名とパーミッションを指定します。

パーミッション種別用プロパティ名

次のどれかを指定してください。

- dbrProp_EveryonePermission
- dbrProp_OwnerPermission
- dbrProp_PrimaryGroupPermission
- dbrProp_UserPermission
- dbrProp_Permission

ただし、「dbrProp_Permission」を指定するのは、dbrProp_ACL プロパティまたは dbrProp_SACL プロパティの構成要素のプロパティの場合だけです。また、「dmaProp_UserPermission」は、プロパティを取得するときだけに指定できます。

それぞれのプロパティに指定できるパーミッションの値については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

パーミッション

NONE (アクセス権を設定しない場合) またはパーミッション論理和を指定します。パーミッション論理和とは、パーミッション種別をストローク (|) で組み合わせた値です。

パーミッション種別には、PRP (基本プロパティ参照権), PWP (基本プロパティ更新権), PRC (基本コンテンツ参照権), PWC (基本コンテンツ更新権), PL (基本リンク権), PV (基本バージョン管理権), PD (基本オブジェクト削除権), または CP (アクセス制御情報変更権) のどれかを指定してください。

3. オブジェクト操作ツールで使用するファイル

PRP, PWP, PRC, PWC, PL, PV, PD は、組み合わせて指定できます。組み合わせて指定する場合は、これらのパーミッション種別を、ストローク (|) でつなげて指定してください。

なお、NONE および CP を、ほかのパーミッション種別と組み合わせて指定することはできません。

(d) サブジェクト種別用プロパティ

アクセス制御機能を使用する場合の、dbrProp_ACL プロパティまたは dbrProp_SACL プロパティを設定するときに記述します。なお、dbrProp_ACL プロパティおよび dbrProp_SACL プロパティの構成要素として指定する dbrProp_Subject プロパティ、dbrProp_SubjectType プロパティおよび dbrProp_Permission プロパティは、必ず三つ一組で指定してください。

サブジェクト種別用プロパティ名
dbrProp_SubjectType を指定してください。

サブジェクト種別
アクセス制御エレメントおよびアクセス制御リストに設定するサブジェクト種別を記述します。USR (ユーザ識別子を指定する場合)、GRP (グループ識別子を指定する場合)、または SYS (システムサブジェクトを指定する場合) のどれかを指定してください。

記述例を次に示します。

< 記述例 >
dbrProp_SubjectType=USR

(3) link エントリ

オブジェクトを作成するコマンドの実行時に -T オプションを指定する場合は、設定するコンテナメントの種別を記述します。セクション (クラス) ごとに、DCR (直接型のコンテナメント)、RCR (参照型のコンテナメント) または VCR (構成管理型のコンテナメント) のどれかを記述してください。

< 記述例 >
link=RCR

(4) mode エントリ

link エントリにコンテナメントの種別として VCR を指定した場合に、構成管理モード種別を指定します。FLOATING (FLOATING モードの場合) または FIX (FIX モードの場合) のどちらかを記述してください。

< 記述例 >
mode=FLOATING

(5) url エントリ

シングルファイル文書またはリファレンスファイル文書を作成する場合は、登録するファイルのファイルパス、または「NONE」を指定します。なお、「NONE」を指定した場合、登録ファイルなしの文書が作成されます。

リファレンスファイル管理機能を使用する場合、このエントリに「NONE」を指定すると、reference_type エントリは、「NONE」を指定したものと仮定されます。

File Link 連携機能を使用する場合、ファイルサーバにファイルまたはディレクトリを登録するときは、登録するファイルまたはディレクトリのパスを指定します。ファイルサーバ上に登録されているコンテンツを指定して文書を作成するときは、このエントリに「DBRSVCONTENT」を指定して、filelink_contentpath エントリに、文書の作成に使用するファイルサーバ上のコンテンツのパスを指定し

ます。登録ファイルなしの文書を作成するときは、「NONE」を指定します。なお、このエントリに「DBRSVCONTENT」を指定して、filelink_contentpath エントリの指定を省略するとエラーになります。

ファイルパスは、URL 形式で記述します。値はシングルクォート（'）で囲んで記述します。例えば、UNIX の場合に「/tmp/create.doc」を文書に登録したいときは、「file:///tmp/create.doc」をシングルクォート（'）で囲んで記述します。Windows の場合に「c:¥temp¥create.doc」を文書に登録したいときは、「file:///c:¥temp¥create.doc」をシングルクォート（'）で囲んで記述します。

< 記述例 >

シングルファイル文書およびリファレンスファイル文書を作成する場合の記述例を示します。

(UNIX の場合)

url='file:///tmp/create.doc'

(Windows の場合)

url='file:///c:¥temp¥create.doc'

File Link 連携機能でクライアントのコンテンツ（ファイル）を指定して文書を作成する場合の記述例を示します。

(UNIX の場合)

url='file:///tmp/create.doc'

(Windows の場合)

url='file:///c:¥temp¥create.doc'

File Link 連携機能でクライアントのコンテンツ（ディレクトリ）を指定して文書を作成する場合の記述例を示します。

(UNIX の場合)

url='file:///tmp'

(Windows の場合)

url='file:///c:¥temp'

File Link 連携機能で、ファイルサーバ上に登録されているコンテンツを指定して文書を作成する場合の記述例を示します。

url='DBRSVCONTENT'

(6) directory エントリ

マルチファイル文書を作成する場合に、登録するファイルが格納されているディレクトリパス、または「NONE」を指定します。なお、「NONE」を指定した場合、登録ファイルなしのマルチファイル文書が作成され、files エントリの指定は無効になります。

ディレクトリパスは、URL 形式で記述してください。値はシングルクォート（'）で囲んで記述します。例えば、UNIX の場合に「/tmp」にマルチファイル文書として登録するファイルが格納されているときは、「file:///tmp」をシングルクォート（'）で囲んで記述します。Windows の場合に「c:¥temp」にマルチファイル文書として登録するファイルが格納されているときは、「file:///c:¥temp」をシングルクォート（'）で囲んで記述します。

< 記述例 >

UNIX の場合の記述例を示します。

directory='file:///tmp'

Windows の場合の記述例を示します。

3. オブジェクト操作ツールで使用するファイル

```
directory='file:///c:/temp'
```

なお、directory エントリに値を指定して、files エントリの指定を省略した場合、directory エントリに指定したディレクトリ下にあるすべてのファイルが、マルチファイル文書の登録対象になります。ただし、この場合は、コンポーネント種別を指定できません。登録対象になるのは、directory エントリに指定したディレクトリの直下に格納されているファイル、シンボリックリンクおよび隠しファイルです。directory エントリに指定したディレクトリの下位ディレクトリに格納されたファイルは、登録対象になりません。

(7) files エントリ

マルチファイル文書を作成する場合に、マルチファイル文書として登録するファイル数、ファイル名およびコンポーネント種別を指定します。

それぞれの値の記述形式を次に示します。

ファイル数

マルチファイル文書として登録するファイルの数を、1 ~ 2,147,483,647 の間で記述します。値はエントリ名に続けて角括弧 ([]) で囲んで記述します。ファイル数の後ろに、等号 (=) を区切り文字として、ファイル名およびコンポーネント種別を記述します。

ファイル名

マルチファイル文書として登録するファイルの名前を記述します。マルチファイル文書として登録できるのは、directory エントリで指定したディレクトリ下にあるファイル、シンボリックリンク、および隠しファイルです。値はシングルクォート (') で囲み、edmProp_RetrievalName エントリに等号 (=) を区切り文字として記述します。

コンポーネント種別

edmProp_RetrievalName エントリに指定したファイルのコンポーネント種別を、シングルクォート (') で囲み、edmProp_ComponentType エントリに等号 (=) を区切り文字として記述します。なお、文書空間で使用する文字コード種別が UTF-8 の場合は、印刷可能な ASCII コードで記述してください。

登録するファイルが複数ある場合は、一つのファイルに対するファイル名およびコンポーネント種別の指定ごとにコンマ (,) で区切って記述します。ファイル数に指定した値と、ファイル名およびコンポーネント種別の数が一致しない場合はエラーになります。

記述例を次に示します。

< 記述例 >

記述例では、次に示す二つのファイルを登録することが前提となっています。

ファイル 1

- ファイル名：foo
- コンポーネント種別：body

ファイル 2

- ファイル名：bar
- コンポーネント種別：fig-1

```
files[2]=  
{  
{ edmProp_RetrievalName = 'foo' , edmProp_ComponentType = 'body' },  
{ edmProp_RetrievalName = 'bar' , edmProp_ComponentType = 'fig-1' }  
}
```

(8) mime エントリ

文書を作成する場合に、登録するファイルの表現形式を「クラス/サブクラス」形式の MIME 名で記述してください。値はシングルクォート (') で囲んで記述します。例えば、HTML 文書の場合、MIME 名には「text/html」をシングルクォート (') で囲んで指定します。

< 記述例 >

```
mime='text/html'
```

(9) filelink_famname エントリ

File Link 連携機能を使用する場合に、コンテンツの登録先となるファイルサーバの FAM 名を、シングルクォート (') で囲んで記述します。

このエントリは次のどちらかの場合に指定してください。

実行環境制御ファイルの FileLink セクションの、DefaultFAMName エントリを省略して、dbrcrtdoc コマンド (文書の作成) を実行する場合

実行環境制御ファイルの FileLink セクションの、DefaultFAMName エントリの指定とは異なる FAM 名のファイルサーバを使用して、dbrcrtdoc コマンド (文書の作成) を実行する場合

なお、このエントリの指定を省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultFAMName エントリに指定された FAM 名のファイルサーバを使用します。

< 記述例 >

```
filelink_famname='fam01'
```

(10) filelink_contentpath エントリ

File Link 連携機能を使用する場合に、クライアントのコンテンツを指定して文書を作成するときは、ファイルサーバ上の登録先ディレクトリのパスを指定します。また、ファイルサーバ上に登録されているコンテンツを指定して文書を作成するときは、使用するファイルサーバ上のコンテンツのパスを指定します。

このエントリは次のどちらかの場合に指定してください。

実行環境制御ファイルの FileLink セクションの、DefaultContentPath エントリを省略して、dbrcrtdoc コマンド (文書の作成) を実行する場合

実行環境制御ファイルの FileLink セクションの、DefaultContentPath エントリの指定とは異なるファイルサーバ上の登録先ディレクトリを使用して、dbrcrtdoc コマンド (文書の作成) を実行する場合

クライアントのコンテンツを指定して文書を作成するとき、およびファイルサーバ上に登録されているコンテンツを指定して文書を作成するときのエントリの指定方法を次に示します。

クライアントのコンテンツを指定して文書を作成するとき

ファイルサーバ上の登録先ディレクトリのパスを、シングルクォート (') で囲んで記述します。ディレクトリの末尾にはデリミタ「/」(UNIX の場合) または「¥」(Windows の場合) を記述します。登録するコンテンツは、url エントリで指定します。

エントリに指定した登録先ディレクトリが存在しない場合、ディレクトリが作成され、コンテンツが登録されます。ただし、デリミタ「/」(UNIX の場合) または「¥」(Windows の場合) を記述しない場合、コンテンツ名称の自動生成を行わない (実行環境制御ファイルで、FileLink セクションの FileNamingMode エントリに「Specify」を指定している) 環境で、このエントリに指定したディレクトリがファイルサーバ上に存在するとエラーになり、存在しないとディレクトリは作成しないで、このエントリに指定した名称でコンテンツが登録されます。

3. オブジェクト操作ツールで使用するファイル

なお、このエントリの指定を省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultContentPath エントリに指定されたディレクトリにコンテンツが登録されます。

<記述例>

UNIX の場合に、コンテンツの登録先ディレクトリが、「/dir01/data」の下位にある「area1」ディレクトリのときの記述例を次に示します。

指定形式の種類別	指定例
フルパス指定	filelink_contentpath = '/dir01/data/area1/'
エイリアス指定	filelink_contentpath = 'DATA/area1/' (「/dir01/data」のエイリアス名が「DATA」のとき)

Windows の場合に、コンテンツの登録先ディレクトリが、「c:\dir01\data」の下位にある「area1」ディレクトリのときの記述例を次に示します。

指定形式の種類別	指定例
フルパス指定	filelink_contentpath = 'c:\dir01\data\area1\'
エイリアス指定	filelink_contentpath = 'DATA\area1\' (「c:\dir01\data」のエイリアス名が「DATA」のとき)

ファイルサーバ上に登録されているコンテンツを指定して文書を作成するとき

ファイルサーバ上のコンテンツのパスを指定します。値はシングルクォート (') で囲んで記述します。

なお、ファイルサーバ上に登録されているコンテンツを指定して文書を作成するとき、このエントリの指定は省略できません。また、このとき、url エントリに、「DBRSVCONTENT」を指定してください。

<記述例>

UNIX の場合に、文書の作成に使用するコンテンツが、「/dir01/data/area1」の下位にある「file.jpg」の場合の記述例を次に示します。

指定形式	指定例
フルパス指定	filelink_contentpath = '/dir01/data/area1/file.jpg'
エイリアス指定	filelink_contentpath = 'DATA/area1/file.jpg' のとき (「/dir01/data」のエイリアス名が「DATA」のとき)

Windows の場合に、文書の作成に使用するコンテンツが、「c:\dir01\data\area1」の下位にある「file.jpg」の場合の記述例を次に示します。

指定形式	指定例
フルパス指定	filelink_contentpath = 'c:\dir01\data\area1\file.jpg'
エイリアス指定	filelink_contentpath = 'DATA\area1\file.jpg' (「c:\dir01\data」のエイリアス名が「DATA」のとき)

(11)reference_type エントリ

リファレンスファイル管理機能を使用する場合の、コンテンツの格納先とリファレンス種別を指定します。リファレンス種別には、「NONE」または「RELATIVE」を指定してください。ただし、url エントリの指定が「NONE」の場合、このエントリは「NONE」を指定したものと仮定されます。

このエントリの指定は次の場合に有効になります。

- 実行環境制御ファイルの Reference セクションで DefaultReferenceType エントリの指定を省略して、dbrcrtdoc コマンド（文書の作成）を実行する場合
- 実行環境制御ファイルの Reference セクションの DefaultReferenceType エントリと異なるリファレンス種別を指定して、dbrcrtdoc コマンド（文書の作成）を実行する場合

「NONE」を指定した場合、コンテンツがなく、コンテンツロケーションを管理しないオブジェクトが作成されます。このエントリの指定が「NONE」で、url エントリの指定が「NONE」以外のとき、url エントリの指定は無効となります。

「RELATIVE」を指定した場合、コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理するオブジェクトが作成されます。

<記述例>

```
reference_type = RELATIVE
```

(12) reference_contentbasepath エントリ

リファレンスファイル管理機能を使用する場合に、コンテンツ格納先ベースパスをシングルクォート（'）で囲んで記述します。

このエントリの指定は次の場合に有効になります。

- 実行環境制御ファイルの Reference セクションで DefaultContentBasePath エントリの指定を省略して、dbrcrtdoc コマンド（文書の作成）を実行する場合
- 実行環境制御ファイルの Reference セクションの DefaultContentBasePath エントリと異なるコンテンツ格納先ベースパスを指定して、dbrcrtdoc コマンド（文書の作成）を実行する場合

このエントリには、リファレンスファイル文書の格納先となるディレクトリの基点となるパスを指定します。例えば、url エントリで指定したコンテンツを、「/drv01/dir01/userdir」ディレクトリの下位にある「user01」ディレクトリに格納する場合、「user01」ディレクトリの基点となるパスとして、「/drv01/dir01/userdir」をシングルクォート（'）で囲んで指定します。このエントリの指定の末尾に、パスのデリミタ「/」（UNIX の場合）または「\」（Windows の場合）を記述する必要はありません。

このエントリに指定したコンテンツ格納先ベースパスは、コンテンツ格納先パスと結合して絶対パスとして使用されます。ContentLocation プロパティには次に示すパスが格納されます。

```
aaaa/bbbb/cccc
```

aaaa : コンテンツ格納先パス

bbbb : DocumentBroker がコンテンツを管理するためのディレクトリ

cccc : ファイル名

注

次に示すどちらかのファイル名が設定されます。

- 登録ファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合）
- DocumentBroker が指定するファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合）

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、

3. オブジェクト操作ツールで使用するファイル

「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、「¥¥server01¥share」をシングルクォート（'）で囲んで指定します。

<記述例>

```
reference_contentbasepath = '/drv01/dir01/userdir'
```

(13) reference_contentpath エントリ

リファレンスファイル管理機能を使用する場合に、コンテンツ格納先ベースパスからの相対パスで、コンテンツ格納先パスをシングルクォート（'）で囲んで記述します。

リファレンス種別が「NONE」の場合、このエントリの指定は無効となります。また、このエントリを省略すると、コンテンツ格納先ベースパスディレクトリの下位にコンテンツが格納されます。

このエントリには、reference_contentbasepath で指定したコンテンツ格納先ベースパスからの相対パスで、コンテンツの格納先とするディレクトリを指定します。例えば、url エントリで指定したコンテンツを、reference_contentbasepath で指定した「/drv01/dir01/userdir」ディレクトリの下位にある「user01」ディレクトリに格納する場合、コンテンツの格納先とするディレクトリとして、「user01」をシングルクォート（'）で囲んで指定します。このエントリの指定の先頭に、パスのデリミタ「/」（UNIX の場合）または「¥」（Windows の場合）を記述する必要はありません。

このエントリに指定したコンテンツ格納先パスは、コンテンツ格納先ベースパスと結合して絶対パスとして使用されます。ContentLocation プロパティには次に示すパスが格納されます。

```
aaaa/bbbb/cccc
```

aaaa : コンテンツ格納先パス

bbbb : DocumentBroker がコンテンツを管理するためのディレクトリ

cccc : ファイル名

注

次に示すどちらかのファイル名が設定されます。

- 登録ファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合）
- DocumentBroker が指定するファイル名（DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合）

<記述例>

```
reference_contentpath = 'user01'
```

(14) 値の記述方法

値を設定するプロパティのデータ型によって記述形式が異なります。

(a) Boolean 型プロパティの値

「TRUE」、「FALSE」または「UNKNOWN」のどれかを記述します。

<記述例>

```
usrProp_IsReadOnly=TRUE
```

(b) Integer32 型プロパティの値

任意の 10 進数を -2,147,483,648 ~ 2,147,483,647 の間で記述します。

< 記述例 >

```
usrProp_State=1
```

(c) String 型プロパティの値

1,039 バイト以内の任意の文字列をシングルクォート (') で囲んで記述します。

< 記述例 >

```
usrProp_Title='Document'
```

空文字列を指定する場合は、シングルクォート (') を二つ続けて記述します。

< 記述例 >

```
usrProp_Title=""
```

シングルクォート (') を文字列として指定する場合は、文字列を囲むシングルクォート (') とは別にシングルクォート (') を二つ続けて記述します。

< 記述例 >

```
usrProp_Title='user"s container'
```

3.5.3 実行するコマンドと指定するエントリの対応

実行するコマンド (オプションの指定を含む) ごとに、プロパティ情報ファイルに指定するエントリを示します。なお、各コマンドの詳細については、「5. コマンドリファレンス」を参照してください。

(1) dbrcrtacl (パブリック ACL の作成)

作成するプロパティ情報ファイル

edmClass_PublicACL 用プロパティ情報ファイルを作成します。

セクション名

パブリック ACL を作成するために使用する edmClass_PublicACL クラスのクラス名を指定します。

指定するエントリ

プロパティエントリを指定します。

(2) dbrcrtcntr (コンテナの作成)

作成するプロパティ情報ファイル

次のプロパティ情報ファイルを作成します。

- dmaClass_Container 用プロパティ情報ファイル
バージョンなしコンテナを作成する場合 (-c RC オプションを指定する場合) に必要です。
- edmClass_ContainerVersion 用プロパティ情報ファイル
構成管理コンテナを作成する場合 (-c VTC オプションまたは -c CRC オプションを指定する場合) に必要です。
- dmaClass_ConfigurationHistory 用プロパティ情報ファイル
バージョン付き構成管理コンテナを作成する場合 (-c CRC オプションを指定する場合) に必要です。

セクション名

プロパティ情報ファイルごとに、次のセクション名を指定します。

- dmaClass_Container 用プロパティ情報ファイル
バージョンなしコンテナを作成するために使用する dmaClass_Container クラスまたはそのサブク

3. オブジェクト操作ツールで使用するファイル

ラスのクラス名を指定します。

- edmClass_ContainerVersion 用プロパティ情報ファイル
構成管理コンテナを作成するために使用する edmClass_ContainerVersion クラスまたはそのサブクラスのクラス名を指定します。
- dmaClass_ConfigurationHistory 用プロパティ情報ファイル
構成管理コンテナのバージョンを管理するために使用する dmaClass_ConfigurationHistory クラスまたはそのサブクラスのクラス名を指定します。

指定するエントリ

バージョンなしコンテナを作成する場合

- -c RC オプションだけ指定する場合
dmaClass_Container 用プロパティ情報ファイルにプロパティエントリを指定します。
- -c RC オプションおよび -T オプションを指定する場合
dmaClass_Container 用プロパティ情報ファイルにプロパティエントリおよび link エントリを指定します。

バージョンなし構成管理コンテナを作成する場合

- -c VTC オプションだけ指定する場合
edmClass_ContainerVersion 用プロパティ情報ファイルにプロパティエントリを指定します。
- -c VTC オプションおよび -T オプションを指定する場合
edmClass_ContainerVersion 用プロパティ情報ファイルにプロパティエントリおよび link エントリを指定します。

バージョン付き構成管理コンテナを作成する場合

- -c CRC オプションだけ指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび edmClass_ContainerVersion 用プロパティ情報ファイルにプロパティエントリを指定します。
- -c CRC オプションおよび -T オプションを指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび edmClass_ContainerVersion 用プロパティ情報ファイルにプロパティエントリを指定します。link エントリ, mode エントリ (link エントリに VCR を指定した場合) は, どちらかのプロパティ情報ファイルに指定します。どちらにも指定されていない場合はエラーになります。両方に指定されている場合は, dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。

(3) dbrcrtdata (独立データの作成)

作成するプロパティ情報ファイル

edmClass_IndependentPersistence 用プロパティ情報ファイルを作成します。

セクション名

独立データを作成するために使用する edmClass_IndependentPersistence クラスまたはそのサブクラスのクラス名を指定します。

指定するエントリ

プロパティエントリを指定します。

(4) dbrcrtdoc (文書の作成)

作成するプロパティ情報ファイル

次のプロパティ情報ファイルを作成します。

- dmaClass_DocVersion 用プロパティ情報ファイル
- dmaClass_ConfigurationHistory 用プロパティ情報ファイル
dmaClass_ConfigurationHistory 用プロパティ情報ファイルは、バージョン付き文書を作成する場合 (-c VD オプションを指定する場合) に必要です。

セクション名

プロパティ情報ファイルごとに、次のセクション名を指定します。

- dmaClass_DocVersion 用プロパティ情報ファイル
文書を作成するために使用する dmaClass_DocVersion クラスまたはそのサブクラスのクラス名を指定します。
- dmaClass_ConfigurationHistory 用プロパティ情報ファイル
文書のバージョンを管理するために使用する dmaClass_ConfigurationHistory クラスまたはそのサブクラスのクラス名を指定します。

指定するエントリ

次のエントリを指定します。

マルチファイル文書であるバージョンなし文書を作成する場合

- -c D オプションだけ指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, directory エントリ, files エントリおよび mime エントリを指定します。
- -c D オプションおよび -T オプションを指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, link エントリ, directory エントリ, files エントリおよび mime エントリを指定します。

シングルファイル文書であるバージョンなし文書を作成する場合

- -c D オプションだけ指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, url エントリおよび mime エントリを指定します。
- -c D オプションおよび -T オプションを指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, link エントリ, url エントリおよび mime エントリを指定します。

マルチファイル文書であるバージョン付き文書を作成する場合

- -c VD オプションだけ指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに、プロパティエントリを指定します。
directory エントリ, files エントリおよび mime エントリは、どちらかのプロパティ情報ファイルに指定します。どちらにも指定されていない場合はエラーとなります。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。
- -c VD オプションおよび -T オプションを指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに、プロパティエントリを指定します。
link エントリ, mode エントリ (link エントリに VCR を指定した場合), directory エントリ, files エントリおよび mime エントリは、どちらかのプロパティ情報ファイルに指定しおよび mime エントリは、どちらかのプロパティ情報ファイルに指定します。どちらにも指定されていない場合はエラーとなります。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。

シングルファイル文書であるバージョン付き文書を作成する場合

- -c VD オプションだけ指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに、プロパティエントリを指定します。
url エントリおよび mime エントリは、どちらかのプロパティ情報ファイルに指定します。どちらにも指定されていない場合はエラーとなります。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。
- -c VD オプションおよび -T オプションを指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに、プロパティエントリを指定します。
link エントリ、mode エントリ (link エントリに VCR を指定した場合)、url エントリおよび mime エントリは、どちらかのプロパティ情報ファイルに指定します。どちらにも指定されていない場合はエラーとなります。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。

File Link 連携機能を利用したバージョンなし文書を作成する場合

- -c D オプションだけ指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ、url エントリ、mime エントリ、filelink_famname エントリおよび filelink_contentpath エントリを指定します。
- -c D オプションおよび -T オプションを指定する場合
dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ、link エントリ、url エントリ、mime エントリ、filelink_famname エントリおよび filelink_contentpath エントリを指定します。

File Link 連携機能を利用したバージョン付き文書を作成する場合

- -c VD オプションだけ指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに、プロパティエントリを指定します。
url エントリ、mime エントリ、filelink_famname エントリおよび filelink_contentpath エントリは、どちらかのプロパティ情報ファイルに指定します。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。url エントリおよび mime エントリがどちらにも指定されていない場合はエラーとなります。
- -c VD オプションおよび -T オプションを指定する場合
dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリを指定します。
link エントリ、mode エントリ (link エントリに VCR を指定した場合)、url エントリ、mime エントリ、filelink_famname エントリおよび filelink_contentpath エントリは、どちらかのプロパティ情報ファイルに指定します。両方に指定されている場合は、dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。
link エントリ、mode エントリ (link エントリに VCR を指定した場合)、url エントリおよび mime エントリがどちらにも指定されていない場合はエラーとなります。

リファレンスファイル管理機能を利用したバージョンなし文書を作成する場合

- -c D オプションだけ指定する場合

dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, url エントリ, mime エントリ, reference_type エントリ, reference_contentbasepath エントリ (reference_type エントリに RELATIVE を指定した場合) および reference_contentpath エントリ (reference_type エントリに RELATIVE を指定した場合) を指定します。

- -c D オプションおよび -T オプションを指定する場合

dmaClass_DocVersion 用プロパティ情報ファイルにプロパティエントリ, link エントリ, url エントリ, mime エントリ, reference_type エントリ, reference_contentbasepath エントリ (reference_type エントリに RELATIVE を指定した場合) および reference_contentpath エントリ (reference_type エントリに RELATIVE を指定した場合) を指定します。

リファレンスファイル管理機能を利用したバージョン付き文書を作成する場合

- -c VD オプションだけ指定する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに, プロパティエントリおよび reference_type エントリを指定します。

url エントリ, mime エントリ, reference_contentbasepath エントリ (reference_type エントリに RELATIVE を指定した場合) および reference_contentpath エントリ (reference_type エントリに RELATIVE を指定した場合) は, どちらかのプロパティ情報ファイルに指定します。両方に指定されている場合は, dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。url エントリおよび mime エントリがどちらにも指定されていない場合はエラーとなります。

- -c VD オプションおよび -T オプションを指定する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルに, プロパティエントリおよび reference_type エントリを指定します。

link エントリ, mode エントリ (link エントリに VCR を指定した場合), url エントリ, mime エントリ, reference_contentbasepath エントリ (reference_type エントリに RELATIVE を指定した場合) および reference_contentpath エントリ (reference_type エントリに RELATIVE を指定した場合) は, どちらかのプロパティ情報ファイルに指定します。両方に指定されている場合は, dmaClass_ConfigurationHistory 用プロパティ情報ファイルに指定されているエントリが有効となります。link エントリ, mode エントリ (link エントリに VCR を指定した場合), url エントリおよび mime エントリがどちらにも指定されていない場合はエラーとなります。

(5) dbrgetprop (プロパティの取得)

プロパティを取得するオブジェクトに対応するプロパティ情報ファイルを作成します。プロパティ情報ファイルには, セクション名と取得するプロパティ名だけを指定します。

(6) dbrsetprop (プロパティの設定)

プロパティを設定するオブジェクトに対応するプロパティ情報ファイルを作成します。プロパティ情報ファイルには, 値を設定するプロパティエントリを指定します。

3.5.4 プロパティ情報ファイルの記述例

マルチファイル文書であるバージョンなし文書を作成する場合のプロパティ情報ファイルの記述例

バージョンなし文書を作成する場合に指定する dmaClass_DocVersion のサブクラスである

usrClass_DocVersion 用プロパティ情報ファイルの記述例を示します。記述していないプロパティは

3. オブジェクト操作ツールで使用するファイル

デフォルトの値が設定されます。なお、この例は Windows の場合です。UNIX の場合は、directory エントリを、「directory='file:///tmp」のように指定してください。

usrClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
;; usrProp_IsReadOnly
;;   DataType : Boolean
;;   Cardinality : Scalar
usrProp_IsReadOnly=TRUE
;; usrProp_State
;;   DataType : Integer32
;;   Cardinality : Scalar
usrProp_State=1
;; usrProp_Title
;;   DataType : String
;;   Cardinality : Scalar
usrProp_Title='Document'
;; Link Type
link=RCR
;; Directory Path
directory='file:///c:¥temp'
;; File Specification
files[2]=
{
  {edmProp_RetrievalName='foo',edmProp_ComponentType='body'},
  {edmProp_RetrievalName='bar',edmProp_ComponentType='fig-1'}
}
;; MIME Name
mime='application/x-edm-undefined'
```

シングルファイル文書であるバージョン付き文書を作成する場合のプロパティ情報ファイルの記述例

バージョン付き文書を作成する場合に指定する dmaClass_ConfigurationHistory のサブクラスである usrClass_ConfigurationHistory 用プロパティ情報ファイルと dmaClass_DocVersion のサブクラスである usrClass_DocVersion 用プロパティ情報ファイルの記述例を示します。記述していないプロパティはデフォルトの値が設定されます。なお、この例は Windows の場合です。UNIX の場合は、url エントリを、「url='file:///tmp/create.doc」のように指定してください。

usrClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
;; usrProp_IsReadOnly
;;   DataType : Boolean
;;   Cardinality : Scalar
usrProp_IsReadOnly=TRUE
;; usrProp_State
;;   DataType : Integer32
;;   Cardinality : Scalar
usrProp_State=1
;; usrProp_Title
;;   DataType : String
;;   Cardinality : Scalar
usrProp_Title='Document'
;; usrProp_Authors
;;   DataType : Object
;;   Cardinality : VariableArray
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',usrProp_Age=27},
  {usrProp_Name='yamada',usrProp_Age=30},
  {usrProp_Name='sato',usrProp_Age=32}
}
```

```

;; Link Type
link=RCR
;; File Path
url='file:///c:¥temp¥create.doc'
;; MIME Name
mime='text/plain'

```

usrClass_DocVersion 用プロパティ情報ファイルの記述例

```

;; usrClass_DocVersion
[usrClass_DocVersion]
;; usrProp_IsReadOnly
;;   DataType   : Boolean
;;   Cardinality : Scalar
usrProp_IsReadOnly=TRUE
;; usrProp_State
;;   DataType   : Integer32
;;   Cardinality : Scalar
usrProp_State=1
;; usrProp_Title
;;   DataType   : String
;;   Cardinality : Scalar
usrProp_Title='Version'

```

独立データのプロパティを取得する場合のプロパティ情報ファイルの記述例

独立データのプロパティを取得する場合に指定する edmClass_IndependentPersistence のサブクラスである usrClass_IndependentPersistence 用プロパティ情報ファイルの記述例を示します。記述していないプロパティは取得しません。

```

;; usrClass_IndependentPersistence
[usrClass_IndependentPersistence]
;; usrProp_IsReadOnly
;;   DataType   : Boolean
;;   Cardinality : Scalar
usrProp_IsReadOnly
;; usrProp_State
;;   DataType   : Integer32
;;   Cardinality : Scalar
usrProp_State
;; usrProp_Title
;;   DataType   : String
;;   Cardinality : Scalar
usrProp_Title
;; usrProp_Authors
;;   DataType   : Object
;;   Cardinality : VariableArray
usrProp_Authors

```

アクセス制御情報を設定、取得する場合のプロパティ情報ファイルの記述例

アクセス制御情報を設定または取得する場合の、プロパティ情報ファイルに記述するアクセス制御情報を表すプロパティの記述例を示します。

アクセス制御情報を設定する場合のプロパティ情報ファイルの記述例

この情報は、アクセス制御情報を設定するオブジェクトを作成する場合に指定する、プロパティ情報ファイル内に記述します。

```

dbrProp_OwnerId='user01'
dbrProp_OwnerPermission=PRP|PWP|PRC|PWC|PV|PL|PD
dbrProp_PrimaryGroupId='group01'
dbrProp_PrimaryGroupPermission=PRP|PWP|PRC|PWC|PV|PL
dbrProp_EveryonePermission=PRP|PRC|PV

```

3. オブジェクト操作ツールで使用するファイル

```
dbrProp_ACL[2][3]=
{
  {dbrProp_Subject='user02',
   dbrProp_SubjectType=USR,
   dbrProp_Permission=PRP|PWP|PRC|PWC|PD},
  {dbrProp_Subject='group02',
   dbrProp_SubjectType=GRP,
   dbrProp_Permission=PRP|PRC}
}
dbrProp_SACL[1][3]=
{
  {dbrProp_Subject='user01',
   dbrProp_SubjectType=USR,
   dbrProp_Permission=CP}
}
```

アクセス制御情報を取得する場合のプロパティ情報ファイルの記述例

```
dbrProp_OwnerId
dbrProp_OwnerPermission
dbrProp_UserPermission
dbrProp_PrimaryGroupId
dbrProp_PrimaryGroupPermission
dbrProp_EveryonePermission
dbrProp_ACL
dbrProp_SACL
```

File Link 連携機能を使用したバージョンなし文書を作成する場合のプロパティ情報ファイルの記述例

バージョンなし文書を作成する場合に指定する dmaClass_DocVersion のサブクラスである usrClass_DocVersion 用プロパティ情報ファイルの記述例を示します。記述していないプロパティはデフォルトの値が設定されます。なお、この例は Windows の場合です。UNIX の場合は、url エントリを、「url='file:///tmp/create.doc'」のように指定し、filelink_contentpath エントリを、「filelink_contentpath='dir01/data/area1/'」のように指定してください。

usrClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
;; usrProp_IsReadOnly
;;   DataType : Boolean
;;   Cardinality : Scalar
usrProp_IsReadOnly=TRUE
;; usrProp_State
;;   DataType : Integer32
;;   Cardinality : Scalar
usrProp_State=1
;; usrProp_Title
;;   DataType : String
;;   Cardinality : Scalar
usrProp_Title='Version'
;; Link Type
link=RCR
;; File Path
url='file:///c:¥temp¥create.doc'
;; MIME Name
mime='text/plain'
;; FAM Name
filelink_famname='fam01'
;; Content Path
filelink_contentpath='c:¥dir01¥data¥area1¥'
```

リファレンスファイル管理機能を使用したバージョン付き文書を作成する場合のプロパティ情報ファイル

の記述例

バージョン付き文書を作成する場合に指定する dmaClass_ConfigurationHistory のサブクラスである usrClass_ConfigurationHistory 用プロパティ情報ファイルの記述例と、dmaClass_DocVersion のサブクラスである usrClass_DocVersion 用プロパティ情報ファイルの記述例を示します。記述していないプロパティはデフォルトの値が設定されます。なお、この例は Windows の場合です。UNIX の場合は、url エントリを「url='file:///tmp/create.doc'」のように指定し、reference_contentbasepath エントリを「reference_contentbasepath='/drv01/dir01/userdir'」のように指定してください。

usrClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',   usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
url='file:///c:¥temp¥create.doc'
mime='MIME::application/x-edm-undefined'
reference_type=RELATIVE
reference_contentbasepath='c:¥dir01¥data'
reference_contentpath='user01'
```

usrClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

3.5.5 複数オブジェクトを一括作成する場合のプロパティ情報ファイル

オブジェクトを作成するために使用するプロパティ情報ファイルに、複数のセクションを記述することで、複数のオブジェクトを一括して作成できます。ただし、次の場合はエラーになります。

バージョン付き文書を作成するときに使用する dmaClass_DocVersion 用プロパティ情報ファイルと dmaClass_ConfigurationHistory 用プロパティ情報ファイルに記述しているセクションの数が一致しない場合

バージョン付き構成管理コンテナを作成するときに使用する edmClass_ContainerVersion 用プロパティ情報ファイルと dmaClass_ConfigurationHistory 用プロパティ情報ファイルに記述しているセクションの数が一致しない場合

次に、二つのコンテナを一括して作成する場合に指定する dmaClass_Container クラスのサブクラスである usrClass_Container 用プロパティ情報ファイルの記述例を示します。記述していないプロパティはデフォルトの値が設定されます。

```
[usrClass_Container]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='container1'

[usrClass_Container]
usrProp_IsReadOnly=FALSE
```

3. オブジェクト操作ツールで使用するファイル

```
usrProp_State=10  
usrProp_Title='container2'
```


3.6 問い合わせファイル

問い合わせファイルは、問い合わせの実行コマンド (dbrexquery) で実行する問い合わせ式を記述するファイルです。

3.6.1 記述規則

問い合わせファイルの記述規則を次に示します。

ファイル名は任意です。

文書空間で使用する文字コード種別 `UTF-8`、または印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

記述できる問い合わせ式は一つだけです。

注

文字コード種別に UTF-8 を使用する場合は、Windows のメモ帳を使わないでください。UTF-8 を前提とするテキストファイルを編集して保存する場合は、「BOM」制御コードが付加されないテキストエディタを使用してください。

Windows のメモ帳で編集して保存すると、ファイルの先頭に「BOM」と呼ばれる数バイトの制御コードが付加され、オブジェクト操作ツールが提供する機能やコマンドでエラーとなることがあります。

3.6.2 基本項目

問い合わせファイルには、edmSQL 文を記述します。edmSQL によって実現できる検索については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

! 注意事項

オブジェクト操作ツールでは、? パラメタを使用した問い合わせ式を指定した検索は実行できません。

3.6.3 記述形式

edmSQL の文法については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

3.6.4 問い合わせファイルの記述例

問い合わせファイルの記述例を次に示します。

```
SELECT ALL
    usrClass_Container.usrProp_IsReadOnly,
    usrClass_Container.usrProp_State,
    usrClass_Container.usrProp_Title,
```

3. オブジェクト操作ツールで使用するファイル

```
        usrClass_Container.usrProp_Authors
FROM  usrClass_Container
WHERE  usrClass_Container.usrProp_Title
        LIKE 'container_' ESCAPE '¥'
        AND  usrClass_Container.usrProp_Authors[ANY].usrProp_Age <= 30
ORDER BY  usrClass_Container.usrProp_State
```

3.7 アップロード情報ファイル

アップロード情報ファイルは、アップロードするマルチファイル文書の情報を記述するファイルです。

3.7.1 記述規則

アップロード情報ファイルの記述規則を次に示します。

ファイル名は任意です。

印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

空白行は無視します。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

記述できるマルチファイル文書の情報は一つだけです。

3.7.2 記述形式

アップロード情報ファイルは、マルチファイル文書に登録されているファイルの格納場所を記述するエンタリと、登録されているファイルの情報を記述するエンタリで構成されます。同一のエンタリが複数指定されている場合は、最初に記述されているエンタリを有効とします。エンタリは、「エンタリ名 = 指定値」の形式で指定します。

記述形式は次のとおりです。

```
(
  ( directory=<ディレクトリパス指定> )
  [ files[<ファイル数>]={ <ファイル指定> } ]
)
<ディレクトリパス指定> ::= '<ディレクトリURL>'
<ディレクトリURL> ::= <"file:"で始まるURL>
<ファイル数> ::= <任意の10進数>
<ファイル指定> ::= { edmProp_RetrievalName='<ファイル名>',
  [ edmProp_ComponentType='<コンポーネント種別>' ] } , ...
<ファイル名> ::= <任意の文字列>
<コンポーネント種別> ::= <任意の文字列>
```

記述形式に関する注意事項

この記述形式で示したすべての波括弧 ({ }) は文字として記述してください。

次に、各エンタリについて説明します。

(1) directory エンタリ

アップロードするマルチファイル文書として登録するファイルが格納されているディレクトリパスを、URL 形式で記述してください。値はシングルクォート (') で囲んで記述します。例えば、UNIX の場合に「/tmp」にマルチファイル文書として登録するファイルが格納されているときは、「file:///tmp」をシングルクォート (') で囲んで記述します。Windows の場合に「c:¥temp」にマルチファイル文書として登録するファイルが格納されているときは、「file:///c:¥temp」をシングルクォート (') で囲んで記述します。

<記述例>

3. オブジェクト操作ツールで使用するファイル

UNIX の場合の記述例を示します。

```
directory='file:///tmp'
```

Windows の場合の記述例を示します。

```
directory='file:///c:¥temp'
```

directory エントリに値を指定して、files エントリの指定を省略した場合、directory エントリに指定したディレクトリ下にあるすべてのファイルが、マルチファイル文書の登録対象になります。ただし、この場合は、コンポーネント種別を指定できません。登録対象になるのは、directory エントリに指定したディレクトリの直下に格納されているファイル、シンボリックリンクおよび隠しファイルです。directory エントリに指定したディレクトリの下位ディレクトリに格納されたファイルは、登録対象になりません。なお、directory エントリに指定したディレクトリ下にファイルがないとエラーになります。

(2) files エントリ

アップロードするマルチファイル文書として登録するファイル数、ファイル名およびコンポーネント種別を指定します。

それぞれの値の記述形式を次に示します。

ファイル数

マルチファイル文書として登録するファイルの数を、1 ~ 2,147,483,647 の間で記述します。値はエントリ名に続けて角括弧 ([]) で囲んで記述します。ファイル数の後ろに、等号 (=) を区切り文字として、ファイル名およびコンポーネント種別を記述します。

ファイル名

マルチファイル文書として登録するファイルの名前を記述します。マルチファイル文書として登録できるのは、directory エントリで指定したディレクトリ下にあるファイル、シンボリックリンク、および隠しファイルです。値はシングルクォート (') で囲み、edmProp_RetrievalName エントリに等号 (=) を区切り文字として記述します。

コンポーネント種別

edmProp_RetrievalName エントリに指定したファイルのコンポーネント種別を、シングルクォート (') で囲み、edmProp_ComponentType エントリに等号 (=) を区切り文字として記述します。

登録するファイルが複数ある場合は、一つのファイルに対するファイル名およびコンポーネント種別の指定ごとにコンマ (,) で区切って記述します。ファイル数に指定した値と、ファイル名およびコンポーネント種別の数が一致しない場合はエラーになります。

記述例を次に示します。

< 記述例 >

記述例では、次に示す二つのファイルを登録することが前提となっています。

ファイル 1

- ファイル名 : foo
- コンポーネント種別 : body

ファイル 2

- ファイル名 : bar
- コンポーネント種別 : fig-1

```
files[2]=
{
{ edmProp_RetrievalName = 'foo', edmProp_ComponentType = 'body' },
{ edmProp_RetrievalName = 'bar', edmProp_ComponentType = 'fig-1' }
}
```

3.7.3 アップロード情報ファイルの記述例

directory エントリに指定したディレクトリ下にある、すべてのファイルをアップロードする場合のアップロード情報ファイル

directory エントリに指定したディレクトリ下にある、すべてのファイルをアップロードする場合の、アップロード情報ファイルの記述例を示します。なお、この例は Windows の場合です。UNIX の場合は、directory エントリを、「directory='file:///tmp」のように指定してください。

アップロード情報ファイルの記述例

```
directory='file:///c:¥temp'
```

directory エントリに指定したディレクトリ下にある、特定のファイルをアップロードする場合のアップロード情報ファイル

directory エントリに指定したディレクトリ下にある、特定のファイルをアップロードする場合の、アップロード情報ファイルの記述例を示します。なお、この例は Windows の場合です。UNIX の場合は、directory エントリを、「directory='file:///tmp」のように指定してください。

アップロード情報ファイルの記述例

```
directory='file:///c:¥temp'
files[2]=
{
{edmProp_RetrievalName='foo',edmProp_ComponentType='body'},
{edmProp_RetrievalName='bar',edmProp_ComponentType='fig-1'}
}
```

3.8 リファレンス情報ファイル

リファレンス情報ファイルは、リファレンスファイル管理機能を使用する場合に、リファレンス種別やコンテンツ格納先ベースパスなどのリファレンス情報を記述するファイルです。

3.8.1 記述規則

リファレンス情報ファイルの記述規則を次に示します。

ファイル名は任意です。

印刷可能な ASCII コードで記述してください。

セミコロン (;) およびシャープ (#) で始まる行はコメント行として扱います。

空白行は無視します。

改行コードは、UNIX の場合、NL (0x0a) です。Windows の場合、CR + NL (0x0d + 0x0a) です。

ファイルの終端は EOF です。

3.8.2 記述形式

リファレンス情報ファイルは、DMA オブジェクトのクラス名をセクション名とする複数のセクションから構成されます。各セクションは複数のエントリで構成されます。リファレンス情報ファイル中に同一のエントリが複数指定されている場合は、最初に記述されているエントリを有効とします。

記述形式は次のとおりです。

```
[ <クラス名> ]
(
  [ reference_type=<リファレンス種別> ]
  [ reference_contentbasepath=<コンテンツ格納先ベースパス指定> ]
  [ reference_deleterootpath=<削除ディレクトリのルートパス指定> ]
)
<リファレンス種別> ::= {NONE | RELATIVE}
<コンテンツ格納先ベースパス> ::= '<任意の文字列>'
<削除ディレクトリのルートパス指定> ::= '<任意の文字列>'
```

(1) <クラス名> セクション

リファレンス情報ファイルには、プロパティ値を設定または取得するオブジェクトを構成する DMA オブジェクトの、クラスまたはそのサブクラスのクラス名をセクション名として、128 バイト以内の半角文字列で記述します。セクション名として記述するクラス名は、クラス定義情報ファイルに記述されている必要があります。セクションは、一つ以上のエントリで構成されます。セクションを構成するエントリについて次に示します。

(2) reference_type エントリ

コンテンツの格納先と管理するリファレンス情報を示すリファレンス種別を指定します。リファレンス種別には、「NONE」または「RELATIVE」を指定してください。

このエントリの指定は、-i オプションを指定して dbrdelentr コマンド (コンテナの削除) を実行する場合に有効になります。

「NONE」を指定した場合、コンテンツがなく、コンテンツロケーションを管理しないオブジェクトとし

て扱います。

「RELATIVE」を指定した場合、コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理するオブジェクトとして扱います。

<記述例>

```
reference_type = RELATIVE
```

(3) reference_contentbasepath エントリ

コンテンツ格納先ベースパスをシングルクォート（'）で囲んで記述します。

このエントリの指定は、`-i` オプションを指定し、かつリファレンス種別に RELATIVE を指定して、`dbrdelcntr` コマンド（コンテナの削除）を実行する場合に有効になります。

このエントリには、コンテンツが格納されているディレクトリの基点となるパスを指定します。例えば、コンテナに含まれるリファレンスファイル文書の基点となるパスが「/drv01/dir01/userdir」ディレクトリの場合、このエントリに「/drv01/dir01/userdir」をシングルクォート（'）で囲んで指定します。このエントリの指定の末尾に、パスのデリミタ「/」（UNIX の場合）または「¥」（Windows の場合）を記述する必要はありません。

このエントリに指定したコンテンツ格納先ベースパスは、ContentLocation プロパティに格納されたコンテンツ格納先を示す相対パスと結合して、絶対パスとして使用されます。

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、「¥¥server01¥share」をシングルクォート（'）で囲んで指定します。

<記述例>

```
reference_contentbasepath = 'drv01/dir01/userdir'
```

(4) reference_deleterootpath エントリ

リファレンスファイル文書と、リファレンスファイル文書が格納されているディレクトリを同時に削除する場合、削除ディレクトリのルートパスをシングルクォート（'）で囲んで記述します。

このエントリの指定は、`-i` オプションを指定し、かつリファレンス種別に RELATIVE を指定して、`dbrdelcntr` コマンド（コンテナの削除）を実行する場合に有効になります。

このエントリには、削除するディレクトリのルートパスを指定します。例えば、「/drv01/dir01/userdir/user01/<DocumentBroker がコンテンツを管理するためのディレクトリ>/<登録ファイル名>」のディレクトリ中の「user01」ディレクトリ以下のディレクトリを削除する場合は、「user01」ディレクトリのルートパスとして、「/drv01/dir01/userdir」をシングルクォート（'）で囲んで指定します。ただし、コンテンツ格納先ベースパスの上位のディレクトリは指定できません。また、エントリの指定の末尾に、パスのデリミタ「/」（UNIX の場合）または「¥」（Windows の場合）を記述する必要はありません。

DocumentBroker がコンテンツを管理するためのディレクトリは、このエントリまたは実行環境制御ファイルで削除ディレクトリのルートパスの指定を省略した場合も、コンテンツと同時に削除されます。

Windows の場合、削除ディレクトリのルートパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンがネットワーク上で共有しているフォルダ「share」を削除ディレクトリのルートパスに指定する場合は、「¥¥server01¥share」をシングルクォート（'）で囲んで指定します。

<記述例>

```
reference_deleterootpath = '/drv01/dir01/userdir'
```

3.8.3 リファレンス情報ファイルの記述例

リファレンス情報ファイルの記述例を次に示します。

記述例は、三つのオブジェクトを一括して操作する場合を示しています。なお、この例は Windows の場合です。UNIX の場合は、reference_contentbasepath エントリを「reference_contentbasepath='/drv01/dir01/data1'」のように指定し、reference_deleterootpath エントリを「reference_deleterootpath='/drv01/dir01/data1'」のように指定してください。

```
[usrClass_ConfigurationHistory]
reference_type=RELATIVE
reference_contentbasepath='c:¥dir01¥userdir'
reference_deleterootpath='c:¥dir01¥userdir'
```

```
[usrClass_ConfigurationHistory]
reference_type=RELATIVE
reference_contentbasepath='C:¥drv01¥dir01¥data2'
reference_deleterootpath='C:¥drv01¥dir01¥data2'
```

```
[usrClass_ConfigurationHistory]
reference_type=RELATIVE
reference_contentbasepath='C:¥drv01¥dir01¥data3'
reference_deleterootpath='C:¥drv01¥dir01¥data3'
```


4

コマンド実行ログと障害対策

この章では、オブジェクト操作ツールで出力するコマンド実行ログと、オブジェクト操作ツールの障害対策について説明します。

4.1 コマンド実行ログ

4.2 障害対策

4.1 コマンド実行ログ

オブジェクト操作ツールでは、セッションの確立・切断、およびオブジェクト操作情報をコマンド実行ログとして出力できます。

4.1.1 コマンド実行ログの制御

コマンド実行ログがどのように出力されるかはオブジェクト操作ツールの実行環境に依存します。オブジェクト操作ツール全体でコマンド実行ログの出力を制御することはできません。コマンド実行ログの出力はオブジェクト操作ツール実行下の実行環境制御ファイルの記述によって制御されます。

4.1.2 出力レベル

実行環境制御ファイルで、Log セクションの Level エントリに指定した出力レベルによってコマンド実行ログに出力する情報量を調節します。出力レベルによる出力ログ情報を次の表に示します。

表 4-1 出力レベルによる出力ログ情報

出力レベル	出力ログ情報
None	出力なし
Write	<ul style="list-style-type: none"> • セッションの確立・切断 • オブジェクトの作成・削除 • バージョンの削除 • 全文検索インデックスの作成・削除 • プロパティの設定 • コンテナメントによるオブジェクト間の関連づけと関連づけの解除 • 構成管理モードの設定 • コンテナのバージョンアップ • ファイルのアップロード • パブリック ACL のバインド・アンバインド • レンディションの追加・削除 • マスタレンディションの変更
Read	<ul style="list-style-type: none"> • 出力レベルが「Write」のときに出力される情報 • プロパティの取得 • ファイルのダウンロードまたは取得 • 一覧の取得 • 問い合わせの実行および結果の取得 • レンディション一覧の取得
Error	<ul style="list-style-type: none"> • 出力レベルが「Read」のときに出力される情報 • エラー

4.1.3 出力先ディレクトリおよびファイル

コマンド実行ログは、実行環境制御ファイルの Log セクションに定義されている内容に従って出力されません。

(1) 出力先ディレクトリ

Directory エントリに指定したディレクトリです。コマンド実行ログを出力しようとしたディレクトリが存在しない場合はエラーメッセージが出力されてコマンド実行ログは出力されません。オブジェクト操作ツール実行前にディレクトリが存在することを確認してください。

(2) 出力先ファイル

Prefix エントリに指定したプリフィックスを持つ次のファイルに出力します。

< 出力ファイル名プリフィックス >_NO.log

「NO」は出力ファイル通番を示します。出力ファイル通番とは、1 ~ FileCount エントリに指定した数（出力ファイル数）です。

コマンド実行ログは、出力ファイル通番が「1」のファイルから順番に出力されます。あるファイルにコマンド実行ログを出力しようとしたときに、ファイルサイズが Log セクションの FileSize エントリに指定した出力ファイルサイズより大きくなってしまふ場合、出力ファイル通番が一つ大きいファイルに出力します。例えば、出力ファイル通番が「1」であるファイルに出力する場合、このファイルのサイズの最大量を超えてしまう場合は、ファイル出力通番が「2」のファイルにログを出力します。

FileCount エントリに指定したファイル数と等しいファイルにログを出力しようとして、そのファイルのファイルサイズも FileSize エントリに指定した出力ファイルサイズより大きくなってしまふ場合、出力ファイル通番が「1」のファイルを初期化してログの出力を続けます。

4.1.4 ログの出力形式と出力ログ情報

(1) 実行ログの出力形式

コマンド実行ログは、次の形式で出力します。

コマンドの実行日時 コマンド名 プロセス ID ログ情報

- コマンドの実行日時
コマンドの実行日時を「YYYY/MM/DD hh/mm/ss」の形式で出力します。
- コマンド名
実行したコマンド名を出力します。
- プロセス ID
実行したコマンドのプロセス ID を 16 進数で出力します。
- ログ情報
実行した操作に対応するログ情報を出力します。詳細は、「(2) 出力ログ情報」を参照してください。

(2) 出力ログ情報

操作に対応する出力ログ情報を次の表に示します。

表 4-2 操作に対応する出力ログ情報

操作	出力ログ情報	出力レベル
エラー	ERROR	E
オブジェクトの作成	CREATE OBJECT <オブジェクト種別> <OIID>	W, R, E
オブジェクトの削除	DELETE OBJECT <オブジェクト種別> <OIID>	W, R, E
オブジェクトを包含するコンテナ一覧の取得	LIST PARENTS <コンテナメントの種別> <オブジェクト種別> <OIID>	R, E

4. コマンド実行ログと障害対策

操作	出力ログ情報	出力レベル
オブジェクト間の関連づけの解除 (コンテンツの解除)	次のどちらかが出力されます。 <ul style="list-style-type: none"> • UNLINK <コンテンツの種別><包含するオブジェクト種別><包含するオブジェクトの OIID><包含するオブジェクトのバージョン識別子><包含されるオブジェクト種別><包含されるオブジェクトの OIID> • UNLINK <リンク識別子><包含するオブジェクトの OIID> 	W, R, E
オブジェクト間の関連づけの設定 (コンテンツの設定)	LINK <コンテンツの種別><構成管理モード種別><包含するオブジェクト種別><包含するオブジェクトの OIID><包含するオブジェクトのバージョン識別子><包含されるオブジェクト種別><包含されるオブジェクトの OIID><包含されるオブジェクトのバージョン識別子>	W, R, E
構成管理モードの設定	次のどれかが出力されます。 <ul style="list-style-type: none"> • SET MODE <構成管理モード種別><包含するオブジェクト種別><包含するオブジェクトの OIID><包含するオブジェクトのバージョン識別子><包含されるオブジェクト種別><包含されるオブジェクトの OIID><包含されるオブジェクトのバージョン識別子> • SET MODE <構成管理モード種別><リンク識別子><包含するオブジェクト種別><包含するオブジェクトの OIID><包含されるオブジェクトのバージョン識別子> 	W, R, E
コンテナに包含されるオブジェクト一覧の取得	LIST CHILDREN <コンテンツの種別><オブジェクト種別><OIID><バージョン識別子>	R, E
コンテナのバージョンアップ	VERSION UP <オブジェクト種別><OIID>	W, R, E
コンテンツ一覧の取得	LIST CONTENTS <オブジェクト種別><OIID><バージョン識別子><レンディションタイプ>	R, E
セッションの確立	CONNECT <文書空間 ID><ユーザ名>	W, R, E
セッションの切断	DISCONNECT	W, R, E
全文検索インデクスの作成	CREATE INDEX <オブジェクト種別><OIID><バージョン識別子>	W, R, E
全文検索インデクスの削除	DELETE INDEX <オブジェクト種別><OIID><バージョン識別子>	W, R, E
バージョンの削除	DELETE VERSION <オブジェクト種別><OIID><バージョン識別子>	W, R, E
問い合わせの実行	EXECUTE SEARCH	R, E
問い合わせ結果の取得	GET RESULT	R, E
バージョン一覧の取得	LIST VERSIONS <オブジェクト種別><OIID>	R, E
バージョン管理情報一覧の取得	LIST VERSIONABLES <オブジェクト種別><OIID>	R, E
バインドされているオブジェクト一覧の取得	LIST OBJECTS <オブジェクト種別><OIID>	R, E
バインドしているパブリック ACL 一覧の取得	LIST ACLS <オブジェクト種別><OIID><バージョン識別子>	R, E
パブリック ACL のアンバインド	UNBIND ACLS <オブジェクト種別><OIID><バージョン識別子>	W, R, E
パブリック ACL のバインド	BIND ACLS <オブジェクト種別><OIID><バージョン識別子>	W, R, E
マスタレンディションの変更	CHANGE MASTER RENDITION <オブジェクト種別><OIID><バージョン識別子><レンディションタイプ>	W, R, E

操作	出力ログ情報	出力レベル
ファイルのアップロード	UPLOAD CONTENT <オブジェクト種別> <OIID> <バージョン識別子> <レンディションタイプ>	W, R, E
ファイルのダウンロード	DOWNLOAD CONTENT <オブジェクト種別> <OIID> <バージョン識別子> <レンディションタイプ>	R, E
プロパティの取得	次のどちらかが出力されます。 <ul style="list-style-type: none"> • GET PROPERTIES <オブジェクト種別> <OIID> <バージョン識別子> • GET PROPERTIES <リンク識別子> <オブジェクト種別> <OIID> 	R, E
プロパティの設定	次のどちらかが出力されます。 <ul style="list-style-type: none"> • SET PROPERTIES <オブジェクト種別> <OIID> <バージョン識別子> • SET PROPERTIES <リンク識別子> <オブジェクト種別> <OIID> 	W, R, E
レンディション一覧の取得	LIST RENDITIONS <オブジェクト種別> <OIID> <バージョン識別子>	R, E
レンディションの削除	DELETE RENDITION <オブジェクト種別> <OIID> <バージョン識別子> <レンディションタイプ>	W, R, E
レンディションの追加	ADD RENDITION <オブジェクト種別> <OIID> <バージョン識別子> <レンディションタイプ>	W, R, E

(凡例) W: Write R: Read E: Error

出力される情報について説明します。

オブジェクト種別

操作したオブジェクトのオブジェクト種別です。オブジェクト種別と出力文字の対応については、「1.3.2 クラスライブラリとの関係」を参照してください。

OIID

操作したオブジェクトの OIID です。

バージョン識別子

操作対象となったオブジェクトのバージョンに対する情報です。操作対象となったオブジェクトのバージョンと出力文字の対応を次の表に示します。

表 4-3 操作対象となったオブジェクトのバージョンと出力文字の対応

操作内容	出力文字
最新バージョンに対する操作	latest
バージョンアップを伴う操作	versionup
バージョントレースしているオブジェクトに対する操作	traced
バージョンなしオブジェクトに対する操作	出力なし
バージョン全体に対する操作	出力なし

コンテインメントの種別

設定、解除したコンテインメントの種別です。コンテインメントの種別と出力文字の対応を次の表に示します。

表 4-4 コンテナメントの種類と出力文字の対応

コンテナメントの種類	出力文字
直接型のコンテナメント	DCR
参照型のコンテナメント	RCR
構成管理型のコンテナメント	VCR
すべてのコンテナメント	ALL

包含するオブジェクトの OIID

オブジェクト間の関連づけ（コンテナメント）を設定、解除した場合の、オブジェクトを包含するオブジェクト（コンテナ）の OIID です。

包含されるオブジェクトの OIID

オブジェクト間の関連づけ（コンテナメント）を設定、解除した場合の、オブジェクトに包含されるオブジェクトの OIID です。

構成管理モード種別

構成管理型のコンテナメントの設定時、または構成管理モードの設定時の構成管理モード種別です。構成管理モードと出力文字の対応を次の表に示します。

表 4-5 構成管理モードの種別と出力文字の対応

構成管理モード種別	出力文字
FLOATING モード	FLOATING
FIX モード	FIX
コンテナメント種別が直接型または参照型の場合	出力なし

リンク識別子

コンテナメントを解除した、または構成管理モードを設定したコンテナメントのリンク識別子です。

包含するオブジェクトのバージョン識別子

コンテナメントを設定した対象である包含するオブジェクト（コンテナ）のバージョン識別子です。操作に対応する出力文字については、表 4-3 を参照してください。

包含されるオブジェクトのバージョン識別子

コンテナメントを設定したり、構成管理モードの FIX モードを設定したりした、包含されるオブジェクトのバージョン識別子です。構成管理モードの FLOATING モードを設定した時には出力されません。操作に対応する出力文字については、表 4-3 を参照してください。

文書空間 ID

実行環境制御ファイルで、Session セクションの DocSpaceId エントリに指定した値です。

ユーザ名

実行環境制御ファイルで、Session セクションの UserName エントリに指定した値です。

レンディションタイプ

操作したレンディションのレンディションタイプです。レンディションタイプを指定しないで操作した場合は出力されません。

4.1.5 出力例

出力レベルに Error を指定した場合のコマンド実行ログの出力例を次に示します。セッションを確立するまでにエラーが発生した場合、セッションの確立・切断の情報は出力されません。また、オブジェクトを操作するときの情報は、操作が正常に終了した場合だけ出力されます。エラーが発生すると、オブジェクトを操作したときの情報は、エラー情報に置き換えられて出力されます。

```

1999/05/18 17:00:00 dbrcrtdoc 0000021D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:00:00 dbrcrtdoc 0000021D CREATE VD dma:///...00001
1999/05/18 17:00:00 dbrcrtdoc 0000021D LINK RCR RC dma:///...00003 VD dma:///...00001
1999/05/18 17:00:00 dbrcrtdoc 0000021D DISCONNECT
1999/05/18 17:01:00 dbrdowndoc 0000022D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:01:00 dbrdowndoc 0000022D DOWNLOAD CONTENT VD dma:///...00001 lastest
1999/05/18 17:01:00 dbrdowndoc 0000022D DISCONNECT
1999/05/18 17:02:00 dbrupdoc 0000023D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:02:00 dbrupdoc 0000023D UPLOAD CONTENT VD dma:///...00001 versionup
1999/05/18 17:02:00 dbrupdoc 0000023D DISCONNECT
1999/05/18 17:03:00 dbrgetprop 0000024D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:03:00 dbrgetprop 0000024D GET PROPERTIES VD dma:///...00001
1999/05/18 17:03:00 dbrgetprop 0000024D DISCONNECT
1999/05/18 17:04:00 dbrsetprop 0000025D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:04:00 dbrsetprop 0000025D SET PROPERTIES VD dma:///...00001
1999/05/18 17:04:00 dbrsetprop 0000025D DISCONNECT
1999/05/18 17:05:00 dbrdelldoc 0000026D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:05:00 dbrdelldoc 0000026D DELETE VD dma:///...00001
1999/05/18 17:05:00 dbrdelldoc 0000026D DISCONNECT
1999/05/18 17:06:00 dbrexquery 0000027D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:06:00 dbrexquery 0000027D EXECUTE SEARCH
1999/05/18 17:06:01 dbrexquery 0000027D GET RESULT
1999/05/18 17:06:02 dbrexquery 0000027D DISCONNECT
1999/05/18 17:07:00 dbrcrtdoc 0000028D CONNECT 673d2be0-d1fd-11d0-ab59-08002be29e1d user
1999/05/18 17:07:00 dbrcrtdoc 0000028D ERROR
1999/05/18 17:07:00 dbrcrtdoc 0000028D DISCONNECT

```

4.2 障害対策

ここでは、オブジェクト操作ツールの障害対策について説明します。

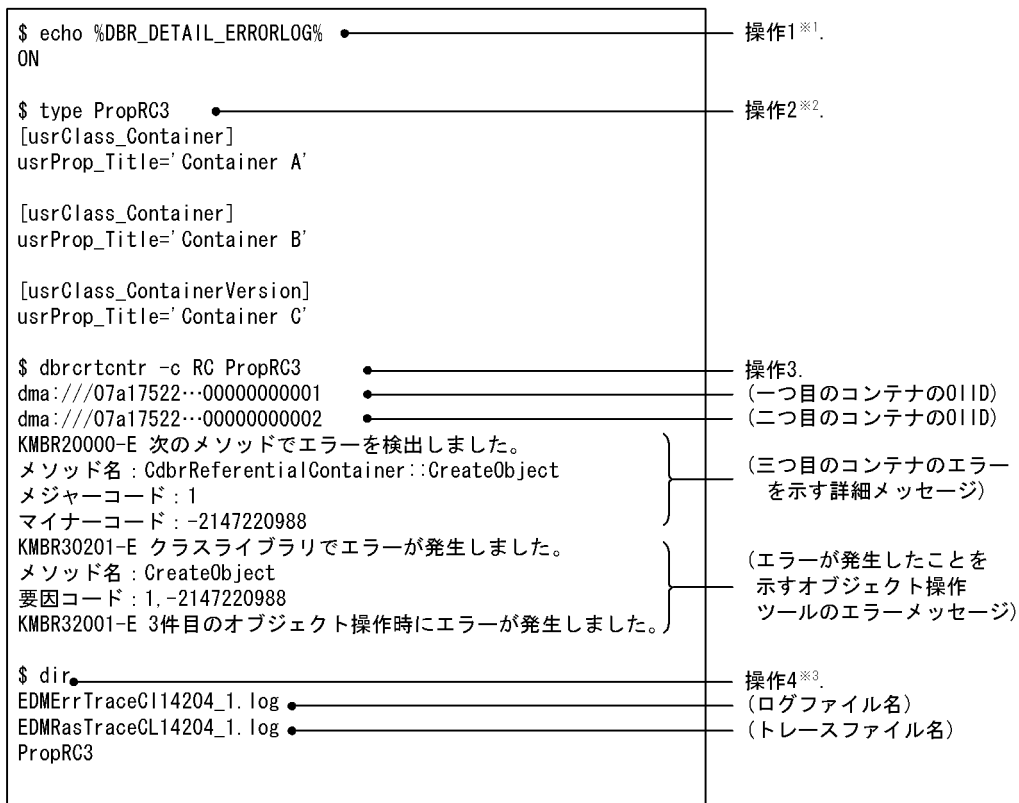
4.2.1 エラーが発生したときのオブジェクト操作ツールの動作

コマンド実行中にエラーが発生すると、操作が終了していないオブジェクトに対する処理は無効になります。

例として、A、B、C という三つのバージョンなしコンテナを作成する場合について説明します。三つのバージョンなしコンテナを作成するためには、dmaClass_Container 用プロパティ情報ファイルを用意して、dbrcrtrcntr を実行します。このとき、C のコンテナ作成中にエラーが発生すると A、B のコンテナは作成されますが C のコンテナは作成されません。

三つのバージョンなしコンテナを作成するときのエラーの発生時期と出力情報の例を次の図に示します。なお、この図は Windows の場合の例です。UNIX の場合の出力情報の例については、図中の注を参照してください。

図 4-1 バージョンなしコンテナを作成するときの、エラーの発生時期と出力情報の例 (Windows の場合)



注※1 UNIXの場合の操作1は、「echo \$DBR_DETAIL_ERRORLOG」になります。

注※2 UNIXの場合の操作2は、「cat PropRC3」になります。

注※3 UNIXの場合の操作4は、「ls」になります。

1. 環境変数を表示します。

環境変数「DBR_DETAIL_ERRORLOG」の設定を確認します。この例の場合、「ON」が設定されているため、詳細メッセージが標準エラー出力および詳細エラーログファイルに出力されることがわかり

ます。

2. プロパティ情報ファイルを表示します。
コンテナを作成するときに指定するプロパティ情報ファイルの内容を表示します。この例で指定するのは、三つのバージョンなしコンテナを作成するための dmaClass_Container 用プロパティ情報ファイルです。ただし、三つ目のコンテナ用のプロパティ情報は、セクション名が正しくありません。
3. dbrctcncr コマンドを実行して、コンテナを作成します。
 - 一つ目のコンテナが作成され、コンテナの OIID が出力されます。
 - 二つ目のコンテナが作成され、コンテナの OIID が出力されます。
 - 三つ目のコンテナ作成時に、エラーが発生したため、詳細メッセージが出力されます。
 - 三つ目のコンテナ作成時にエラーが発生したことを示す、オブジェクト操作ツールのエラーメッセージが出力されます。
4. ファイル内容を確認します。
 - 詳細メッセージが出力された詳細エラーログファイル名が出力されます。この例は、詳細エラーログファイル出力ディレクトリを示す環境変数「DBR_DETAIL_ERRORLOG_DIR」を指定していない場合です。
 - トレースファイル名が出力されます。この例は、トレースファイルをカレントディレクトリに出力した場合です。トレースファイルの出力先の指定については、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。

ただし、コマンド実行ログを出力するときにエラーが発生した場合、コマンド実行ログの出力は行われませんが、オブジェクトの操作は完了しているのでコマンドの処理自体は無効になりません。

4.2.2 メッセージ

エラーが発生すると、メッセージを標準エラー出力に出力します。また、環境変数「DBR_DETAIL_ERRORLOG」に「ON」が設定されている場合には、詳細メッセージも標準エラー出力に出力します。発生したエラーには、これらのメッセージの内容に従って対処してください。メッセージの詳細については、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

4.2.3 トレースファイル

コマンドの実行ごとにトレースファイルが作成されます。不要なトレースファイルはディスク容量を圧迫するので削除してください。トレースファイルの詳細については、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。

4.2.4 詳細エラーログファイル

環境変数「DBR_DETAIL_ERRORLOG」に「ON」が設定されている場合、詳細メッセージは詳細エラーログファイルにも出力されます。詳細エラーログファイルについては、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

5

コマンドリファレンス

この章では、オブジェクト操作ツールのコマンドについて説明します。

実行コマンド一覧

コマンド形式

コマンド実行結果

dbrbind (パブリック ACL のバインド)

dbrcrtacl (パブリック ACL の作成)

dbrcrtcncr (コンテナの作成)

dbrcrtdata (独立データの作成)

dbrcrtdoc (文書の作成)

dbrcrtidx (全文検索インデクスの作成)

dbrdelacl (パブリック ACL の削除)

dbrdelcncr (コンテナの削除)

dbrdeldata (独立データの削除)

dbrdeldoc (文書の削除)

dbrdelidx (全文検索インデクスの削除)

dbrdowndoc (文書のダウンロード)

dbrexquery (問い合わせの実行)

dbrgetprop (プロパティの取得)

dbrlink (オブジェクト間の関連づけ)

dbrlsacl (バインドしているパブリック ACL 一覧またはバインドされているオブジェクト一覧の取得)

dbrlschild (コンテナに含まれるオブジェクト一覧の取得)

dbrlscompo (マルチファイル文書のコンテンツ情報一覧の取得)

dbrlsparent (オブジェクトを包含するコンテナ一覧の取得)

dbrlsversion (バージョン一覧の取得)

dbrlsversionable (バージョン管理情報一覧の取得)

dbrnrdrdoc (文書のレンディション操作)

dbrsetmode (構成管理モードの設定)

dbrsetprop (プロパティの設定)

dbrtoolsetup (実行環境のセットアップ)

dbrunbind (パブリック ACL のアンバインド)

dbrunlink (オブジェクト間の関連づけの解除)

dbrupcntr (コンテナのバージョンアップ)

dbrupdoc (文書のアップロード)

リファレンスファイル管理機能を使用する場合のオブジェクトの削除

実行コマンド一覧

ここでは、オブジェクト操作ツールが提供する実行コマンドの一覧を示します。

実行コマンド一覧

オブジェクト操作ツールが提供する実行コマンドの一覧を次の表に示します。

なお、文書空間の文字コード種別に UTF-8 を指定している場合、使用できるコマンドに制限があります。詳細については、「付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲」を参照してください。

表 5-1 オブジェクト操作ツールが提供する実行コマンドの一覧

コマンド名	機能
dbrbind	パブリック ACL のバインド
dbrcrtacl	パブリック ACL の作成
dbrcrtentr	コンテナの作成
dbrcrtdata	独立データの作成
dbrcrtdoc	文書の作成
dbrcrtidx	全文検索インデックスの作成
dbrdelacl	パブリック ACL の削除
dbrdelentr	コンテナの削除
dbrdeldata	独立データの削除
dbrdeldoc	文書の削除
dbrdelidx	全文検索インデックスの削除
dbrdowndoc	文書のダウンロード
dbrexquery	問い合わせの実行
dbrgetprop	プロパティの取得
dbrlink	オブジェクト間の関連づけ
dbrlsacl	バインドしているパブリック ACL 一覧またはバインドされているオブジェクト一覧の取得
dbrlschild	コンテナに包含されるオブジェクト一覧の取得
dbrlscompo	マルチファイル文書のコンテンツ情報一覧の取得
dbrlsparent	オブジェクトを包含するコンテナ一覧の取得
dbrlsversion	バージョン一覧の取得
dbrlsversionable	バージョン管理情報一覧の取得
dbrrndrdoc	文書のレンディション操作
dbrsetmode	構成管理モードの設定
dbrsetprop	プロパティの設定
dbrtoolsetup	実行環境のセットアップ
dbrunbind	パブリック ACL のアンバインド
dbrunlink	オブジェクト間の関連づけの解除
dbrupentr	コンテナのバージョンアップ
dbrupdoc	文書のアップロード

コマンド形式

ここでは、実行コマンドの入力形式について説明します。

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

コマンド名

コマンド名には、実行するコマンドのファイル名を指定します。

オプション

オプションの記述形式を次に示します。説明で使用する「\$」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ています。また、「cmd」はコマンド名を表しています。

オプションの形式

オプションはマイナス (-) 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> オプションの引数

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

オプションの指定規則

- オプションの引数がないオプションフラグは、一つのマイナス (-) でまとめて指定できます。次の二つは同じ指定です。
\$ cmd -a -b
\$ cmd -ab
- オプションの引数が必要なオプションフラグのオプションの引数は省略できません。
- オプションフラグとオプションの引数の間には空白およびタブを指定しても指定しなくてもどちらでもかまいません。次の二つは同じ指定です。
\$ cmd -a file
\$ cmd -afile
- オプションの引数に空白およびタブを含めて指定する場合は、前後をダブルクォート (") で囲ってください。次に示す例は、オプションの引数として「1 2」を指定しています。
\$ cmd -a "1 2"
- 同じオプションフラグを 2 回以上指定することはできません。次に示すような指定はエラーとなります。
\$ cmd -a file1 -a file2
- 二つのマイナス (--) はオプションの最後を示す区切り記号とみなします。したがって、次に示す例では「-b」をコマンド引数とみなします。
\$ cmd -a -- -b
- マイナス (-) だけのオプションは入力できません。例えば、「\$ cmd -」と入力すると、マイナス (-) をコマンド引数とみなします。
- オプションはコマンド引数より前に指定してください。次に示す例で、-a がオプションの引数を必要としない場合、「file」および「-b」をコマンド引数とみなします。

```
$ cmd -a file -b
```

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。指定するコマンド引数は、各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。

オプションおよびコマンド引数に指定する内容について

オプションおよびコマンド引数に、パス、識別子、ファイル、レンディションタイプなどを指定する場合の記述形式を次に示します。必要に応じてマニュアルまたは記述箇所を参照して、指定してください。

- オプションフラグの引数に OIID ファイルやダウンロードファイル URL などのパスを指定する場合、パスは印刷可能な ASCII コードで記述してください。
- OIID を指定する場合の記述形式については、「付録 A OIID 指定時の URL の文法」を参照してください。
- OIID ファイル、バージョン識別子ファイル、リンク識別子ファイルなど、ID ファイルの記述形式については、「3.4 ID ファイル」を参照してください。
- プロパティ情報ファイルを指定する場合の記述形式については、「3.5 プロパティ情報ファイル」を参照してください。
- 問い合わせファイルを指定する場合の記述形式については、「3.6 問い合わせファイル」を参照してください。
- レンディションタイプを指定する場合の記述形式については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

コマンド実行結果

ここでは、戻り値と実行結果について説明します。

戻り値

戻り値は、オブジェクト操作ツールのすべてのコマンドで共通です。コマンドの終了状態によって、次の表に示す値を返却します。

表 5-2 戻り値一覧

戻り値	意味
0	正常終了
1	警告つき正常終了
2	引数不正による異常終了
上記以外	そのほかの異常終了

実行結果

実行結果は 1 オブジェクトにつき 1 行で標準出力に出力されます。

各項目はコンマ(,)で区切って出力されます。

エラーメッセージは標準エラー出力に出力されます。

実行環境制御ファイルで、Output セクションの Format エントリに指定した値によって、出力情報が異なります。Format エントリに「VR0112」を指定した場合は、Version 1 のオブジェクト操作ツールでの出力情報と同じ形式で出力されます。Format エントリの指定を省略した場合、または「VR0200」を指定した場合は、リンク識別子などの情報が新たに追加された形式で出力されます。それぞれの場合の出力形式の詳細は、各コマンドの実行例を参照してください。

dbrbind (パブリック ACL のバインド)

機能

オブジェクトにパブリック ACL をバインドします。

操作クラス

バインドするオブジェクト

CdbrIndependentPersistence クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

CdbrReferentialContainer クラス

CdbrVersionTraceableContainer クラス

CdbrConfiguratedReferentialContainer クラス

バインドされるオブジェクト

CdbrPublicACL クラス

形式

```
dbrbind [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           { OIID | OIID ファイル }
           {パブリック ACL OIID | パブリック ACL OIID ファイル}
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書またはバージョン付き構成管理コンテナの場合、バインドするバージョンのバージョン識別子を指定します。このオプションを省略した場合は、オブジェクト自身にバインドされます。

バージョン付き文書およびバージョン付き構成管理コンテナ以外の場合にこのオプションを指定するとエラーになります。

バージョン識別子

バインドするバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

バインドするバージョン識別子を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

OIID

バインドするオブジェクトの OIID を指定します。

OIID ファイル

5. コマンドリファレンス

バインドするオブジェクトの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

パブリック ACL OIID

バインドされるパブリック ACL の OIID を指定します。

パブリック ACL ファイル

バインドされるパブリック ACL の OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

すでにバインドされているパブリック ACL をバインドしようとすると、実行メソッドでエラーになります。

dbrcrtacl (パブリック ACL の作成)

機能

プロパティ情報ファイルに記述された内容に従ってパブリック ACL を作成します。プロパティ情報ファイルに複数のセクションを記述することで、複数のパブリック ACL を一括して作成できます。

パブリック ACL が作成されると、作成したパブリック ACL の OIID が標準出力に出力されます。

操作クラス

CdbrPublicACL クラス

形式

dbrcrtacl edmClass_PublicACL 用プロパティ情報ファイル

オプション

なし

コマンド引数

edmClass_PublicACL 用プロパティ情報ファイル

作成するパブリック ACL の edmClass_PublicACL クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

プロパティ情報ファイルの記述例

パブリック ACL を作成する場合の、edmClass_PublicACL 用プロパティ情報ファイルの記述例を示します。

```
edmClass_PublicACL
[edmClass_PublicACL]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='PublicACL'
```

実行例

実行例を次に示します。なお、この例は Windows の場合です。UNIX の場合は、「type PropACL」を「cat PropACL」に読み替えてください。

```
$ type PropACL
;; edmClass_PublicACL
[edmClass_PublicACL]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='PublicACL'

$ dbrcrtacl PropACL
dma:///07a17522...00000000039
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

5. コマンドリファレンス

注意事項

なし

dbrcrtrcncr (コンテナの作成)

機能

プロパティ情報ファイルに記述された内容に従ってコンテナを作成します。プロパティ情報ファイルに複数のセクションを記述することで、複数のコンテナを一括して作成できます。

コンテナが作成されると、作成されたコンテナの OIID が標準出力に出力されます。また、コンテインメントを設定した場合には、リンク識別子も標準出力に出力されます。

操作クラス

作成するオブジェクト

- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

作成したオブジェクトを包含するオブジェクト

- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

形式

バージョンなしコンテナの場合

```
dbrcrtrcncr -c RC [ -T { OIID | OIID ファイル } ]
                    dmaClass_Container 用プロパティ情報ファイル
```

バージョンなし構成管理コンテナの場合

```
dbrcrtrcncr -c VTC [ -T { OIID | OIID ファイル } ]
                    edmClass_ContainerVersion 用プロパティ情報ファイル
```

バージョン付き構成管理コンテナの場合

```
dbrcrtrcncr -c CRC [ -T { OIID | OIID ファイル } ]
                    dmaClass_ConfigurationHistory 用プロパティ情報ファイル
                    edmClass_ContainerVersion 用プロパティ情報ファイル
```

オプション

`-c {RC | VTC | CRC}`

作成するコンテナの種類を指定します。このオプションは省略できません。

RC

バージョンなしコンテナを作成します。

VTC

バージョンなし構成管理コンテナを作成します。

CRC

バージョン付き構成管理コンテナを作成します。

-T { OIID | OIID ファイル }

既存のコンテナと作成したコンテナ間の包含関係を設定します。作成したコンテナが既存のコンテナに包含されます。OIID を記述した OIID ファイルを指定することもできます。このオプションを省略した場合は、どのコンテナとも包含関係を設定しません。

なお、このオプションを指定した場合は、プロパティ情報ファイルに link エントリを必ず指定してください。

OIID

作成したコンテナと包含関係を設定するコンテナの OIID を指定します。

OIID ファイル

作成したコンテナと包含関係を設定するコンテナの OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

コマンド引数

dmaClass_Container 用プロパティ情報ファイル

作成するコンテナの dmaClass_Container クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

edmClass_ContainerVersion 用プロパティ情報ファイル

作成するコンテナの edmClass_ContainerVersion クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

dmaClass_ConfigurationHistory 用プロパティ情報ファイル

作成するコンテナの dmaClass_ConfigurationHistory クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

バージョンなしコンテナを作成する場合のプロパティ情報ファイルの記述例

バージョンなしコンテナを作成する場合の dmaClass_Container 用プロパティ情報ファイルの記述例を示します。

-T オプションを指定しないでバージョンなしコンテナを作成する場合

```
[usrClass_Container]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
```

-T オプションを指定してバージョンなしコンテナを作成する場合

```
[usrClass_Container]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
link=DCR
```

注 link エントリは必ず指定します。

バージョンなし構成管理コンテナを作成する場合のプロパティ情報ファイルの記述例

バージョンなし構成管理コンテナを作成する場合の edmClass_ContainerVersion 用プロパティ情報ファイルの記述例を示します。

-T オプションを指定しないでバージョンなし構成管理コンテナを作成する場合

```
[usrClass_ContainerVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
```

-T オプションを指定してバージョンなし構成管理コンテナを作成する場合

```
[usrClass_ContainerVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
link=DCR
```

注 link エントリは必ず指定します。

バージョン付き構成管理コンテナを作成する場合のプロパティ情報ファイルの記述例

バージョン付き構成管理コンテナを作成する場合の、dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび edmClass_ContainerVersion 用プロパティ情報ファイルの記述例を示します。

-T オプションを指定しないでバージョン付き構成管理コンテナを作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
```

5. コマンドリファレンス

```
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
```

edmClass_ContainerVersion 用プロパティ情報ファイルの記述例

```
[usrClass_ContainerVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定してバージョン付き構成管理コンテナを作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
link=VCR 1
mode=FLOATING 2
```

注 1 link エントリは必ず指定します。

注 2 link エントリに VCR を指定した場合、mode エントリは必ず指定します。

edmClass_ContainerVersion 用プロパティ情報ファイルの記述例

```
[usrClass_ContainerVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

実行例

実行例を次に示します。なお、実行環境制御ファイルの Output セクションの Format エントリの指定、および指定したオプションによって、出力形式が異なります。

また、この例は Windows の場合です。UNIX の場合は、「type PropRC」を「cat PropRC」に読み替えてください。

```
$ type PropRC
;; usrClass_Container
[usrClass_Container]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Container'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
```



```

    {usrProp_Name='yamada', usrProp_Age=30},
    {usrProp_Name='sato',   usrProp_Age=32}
}
link=RCR

```

(Formatエントリが「VR0112」の場合)

```

$ dbrcrtcntr -c RC PropRC
dma:///07a17522...00000002
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

```

```

$ dbrcrtcntr -c RC -T dma:///07a1...0001 PropRC
dma:///07a17522...00000003
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

```

(Formatエントリが「VR0200」の場合)

```

$ dbrcrtcntr -c RC PropRC
dma:///07a17522...00000004
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

```

```

$ dbrcrtcntr -c RC -T dma:///07a1...0001 PropRC
ac1://LR...0005,dma:///07a17522...00000005
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

```

注意事項

-T オプションを指定する場合は、次のプロパティ情報ファイルの各セクションに、link エントリを必ず記述してください。

- バージョンなしコンテナを作成するとき
dmaClass_Container 用プロパティ情報ファイル
- バージョンなし構成管理コンテナを作成するとき
edmClass_ContainerVersion 用プロパティ情報ファイル
- バージョン付き構成管理コンテナを作成するとき
dmaClass_ConfigurationHistory 用プロパティ情報ファイルまたは edmClass_ContainerVersion 用プロパティ情報ファイル

エントリの記述がない場合はエラーになります。また、バージョン付き構成管理コンテナを作成する場合、両方のプロパティ情報ファイルに link エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効になります。

構成管理モードのコンテインメントを設定できるのは、バージョン付き構成管理コンテナ作成時に、-T オプションのオプションの引数に構成管理コンテナの OIID または OIID ファイルを指定した場合だけです。

link エントリに VCR を指定した場合は、コマンド引数に指定する edmClass_ContainerVersion 用プロパティ情報ファイルまたは dmaClass_ConfigurationHistory 用プロパティ情報ファイルの各セクションに mode エントリを必ず記述してください。エントリの記述がない場合には、エラーになります。また、両方のプロパティ情報ファイルに mode エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効になります。

-T オプションのオプションの引数にバージョン付き構成管理コンテナの OIID または OIID ファイルを指定した場合は、最新バージョンのバージョン付き構成管理コンテナに包含されます。

バージョン付き構成管理コンテナを作成する時に指定した dmaClass_ConfigurationHistory 用プロパティ情報ファイルと edmClass_ContainerVersion 用プロパティ情報ファイルに記述したセクション数が同じでない場合、エラーになります。

dbrcrtdata (独立データの作成)

機能

プロパティ情報ファイルに記述された内容に従って独立データを作成します。プロパティ情報ファイルに複数のセクションを記述することで、複数の独立データを一括して作成できます。

独立データが作成されると、作成した独立データの OIID が標準出力に出力されます。

操作クラス

CdbrIndependentPersistence クラス

形式

dbrcrtdata edmClass_IndependentPersistence 用プロパティ情報ファイル

オプション

なし

コマンド引数

edmClass_IndependentPersistence 用プロパティ情報ファイル

作成する文書の edmClass_IndependentPersistence クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

プロパティ情報ファイルの記述例

独立データを作成する場合の edmClass_IndependentPersistence 用プロパティ情報ファイルの記述例を示します。

```
;; usrClass_IndependentPersistence
[usrClass_IndependentPersistence]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Independent'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
```

実行例

実行例を次に示します。なお、この例は Windows の場合です。UNIX の場合は、「type PropIP」を「cat PropIP」に読み替えてください。

```
$ type PropIP
;; usrClass_IndependentPersistence
[usrClass_IndependentPersistence]
usrProp_IsReadOnly=TRUE
usrProp_State=1
```

```
usrProp_Title='Independent'  
usrProp_Authors[3][2]=  
{  
  {usrProp_Name='suzuki', usrProp_Age=27},  
  {usrProp_Name='yamada', usrProp_Age=30},  
  {usrProp_Name='sato',   usrProp_Age=32}  
}  
  
$ dbrcrtdata PropIP  
dma:///07a17522...00000006  
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

なし

dbrcrtdoc (文書の作成)

機能

プロパティ情報ファイルに記述された内容に従って文書を作成します。プロパティ情報ファイルに複数のセクションを記述することで、複数の文書を一括して作成できます。

文書が作成されると、作成された文書の OIID が標準出力に出力されます。コンテインメントを設定した場合には、リンク識別子も標準出力に出力されます。

操作クラス

作成するオブジェクト

CdbrDocument クラス

CdbrVersionableDocument クラス

作成したオブジェクトを包含するオブジェクト

CdbrReferentialContainer クラス

CdbrVersionTraceableContainer クラス

CdbrConfiguratedReferentialContainer クラス

形式

バージョンなし文書の場合

```
dbrcrtdoc -c D [-I]
                [-T {リンク先 OIID | リンク先 OIID ファイル}]
                [-k {Content | MultiContent | FileLink | Reference}]
                dmaClass_DocVersion 用プロパティ情報ファイル
```

バージョン付き文書の場合

```
dbrcrtdoc -c VD [-I]
                [-T {リンク先 OIID | リンク先 OIID ファイル}]
                [-k {Content | MultiContent | FileLink | Reference}]
                dmaClass_ConfigurationHistory 用プロパティ情報ファイル
                dmaClass_DocVersion 用プロパティ情報ファイル
```

オプション

-c {D | VD}

作成する文書の種類を指定します。このオプションは省略できません。

D

バージョンなし文書を作成します。

VD

バージョン付き文書を作成します。

-l

全文検索インデックスを作成しない場合に指定します。シングルファイル文書を作成する場合、このオプションを省略していて、かつ作成する文書が次の条件を満たしているとき、プロパティ情報ファイルの url エントリに指定したファイルから、全文検索インデックスが作成されます。

- dmaClass_DocVersion 用プロパティ情報ファイルのセクション名に、全文検索機能付き文書クラスを記述している。
- 指定した MIME 名の先頭が「text/」である。

なお、マルチファイル文書、File Link 文書、またはリファレンスファイル文書の場合は、このオプションを省略しても全文検索インデックスは作成されません。

-T { リンク先 OIID | リンク先 OIID ファイル }

作成した文書とコンテナ間の包含関係を設定します。リンク先 OIID を記述したリンク先 OIID ファイルを指定することもできます。このオプションを省略した場合は、作成した文書はどのコンテナにも含まれません。

なお、このオプションを指定した場合は、プロパティ情報ファイルに link エントリを必ず指定してください。

リンク先 OIID

作成した文書を包含するコンテナのリンク先 OIID を指定します。

リンク先 OIID ファイル

作成した文書を包含するコンテナのリンク先 OIID を記述したリンク先 OIID ファイルのファイルパスを指定します。

リンク先 OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、リンク先 OIID ファイルに記述できる OIID の数は一つだけです。

-k { Content | MultiContent | FileLink | Reference }

作成する文書のコンテンツ種別を指定します。このオプションを指定した場合、プロパティ情報ファイルの指定が、このオプションの指定と一致していないとエラーになります。このオプションを省略した場合、プロパティ情報ファイルの指定に従ってコンテンツを登録します。

Content

シングルファイル文書を作成します。プロパティ情報ファイルの url エントリに指定されたファイルを、データベースに登録します。

MultiContent

マルチファイル文書を作成します。プロパティ情報ファイルの directory エントリに指定したディレクトリ下の全ファイルか、files エントリに指定されたファイルをデータベースに登録します。

FileLink

File Link 文書を作成します。

Reference

リファレンスファイル文書を作成します。コンテンツをファイルシステム上の領域に格納し、コンテンツの格納先を示す情報をデータベースで管理します。

オプション省略時

プロパティ情報ファイルの url エントリ、または directory エントリの指定に従ってコンテンツを登録します。この場合、コンテンツはデータベースに登録します。

- url エントリが指定されている場合
シングルファイル文書を作成します。

- directory エントリが指定されている場合
マルチファイル文書を作成します。

コマンド引数

dmaClass_ConfigurationHistory 用プロパティ情報ファイル

作成する文書の dmaClass_ConfigurationHistory クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

dmaClass_DocVersion 用プロパティ情報ファイル

作成する文書の dmaClass_DocVersion クラスのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

バージョンなし文書を作成する場合のプロパティ情報ファイルの記述例

バージョンなし文書を作成する場合は、コマンド引数として dmaClass_DocVersion 用プロパティ情報ファイルを指定します。次に、dmaClass_DocVersion 用プロパティ情報ファイルの記述例を示します。なお、この例は Windows の場合です。UNIX の場合は、url エントリを「url=file:///tmp/create.doc」のように指定してください。

-T オプションを指定しないで、マルチファイル文書を作成する場合

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
directory='file:///c:¥temp'
files[2]=
{
  {edmProp_RetrievalName='foo',edmProp_ComponentType='body'},
  {edmProp_RetrievalName='bar',edmProp_ComponentType='fig-1'}
}
mime='application/x-edm-undefined'
```

注 directory エントリおよび mime エントリは必ず指定します。

-T オプションを指定しないで、シングルファイル文書を作成する場合

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
url='file:///c:¥temp¥create.doc'
mime='text/plain'
```

注 url エントリおよび mime エントリは必ず指定します。

-T オプションを指定して、マルチファイル文書を作成する場合

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
```

```

usrProp_Title='Document'
link=RCR
directory='file:///c:¥temp'
files[2]=
{
  {edmProp_RetrievalName='foo',edmProp_ComponentType='body'},
  {edmProp_RetrievalName='bar',edmProp_ComponentType='fig-1'}
}
mime='application/x-edm-undefined'

```

注 link エントリ, directory エントリおよび mime エントリは必ず指定します。

-T オプションを指定して, シングルファイル文書を作成する場合

```

;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
link=RCR
url='file:///c:¥temp¥create.doc'
mime='text/plain'

```

注 link エントリ, url エントリおよび mime エントリは必ず指定します。

バージョン付き文書を作成する場合のプロパティ情報ファイルの記述例

バージョン付き文書を作成する場合は, コマンド引数として dmaClass_ConfigurationHistory 用プロパティ情報ファイルおよび dmaClass_DocVersion 用プロパティ情報ファイルを指定します。次に, 各プロパティ情報ファイルの記述例を示します。なお, この例は, Windows の場合です。UNIX の場合は, url エントリを「url='file:///tmp/create.doc'」または「url='file:///tmp/create.txt'」, filelink_contentpath エントリを「filelink_contentpath='DATA/area1/」または「filelink_contentpath='DATA/area1/file.jpg'」, reference_contentbasepath エントリを「reference_contentbasepath='/drv01/dir01/userdir'」のように指定してください。

-T オプションを指定しないで, マルチファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```

;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato', usrProp_Age=32}
}
directory='file:///c:¥temp'
files[2]=
{
  {edmProp_RetrievalName='foo',edmProp_ComponentType='body'},
  {edmProp_RetrievalName='bar',edmProp_ComponentType='fig-1'}
}
mime='application/x-edm-undefined'

```

注 directory エントリおよび mime エントリは必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定しないで、シングルファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
url='file:///c:¥temp¥create.doc'
mime='text/plain'
```

注 url エントリおよび mime エントリは必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定して、マルチファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
link=RCR
directory='file:///c:¥temp'
files[2]=
{
  {edmProp_RetrievalName='foo', edmProp_ComponentType='body'},
  {edmProp_RetrievalName='bar', edmProp_ComponentType='fig-1'}
}
mime='application/x-edm-undefined'
```

注 link エントリ, directory エントリおよび mime エントリは必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例


```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定して、シングルファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
link=RCR
url='file:///c:¥temp¥create.doc'
mime='text/plain'
```

注 link エントリ, url エントリおよび mime エントリは必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定しないで、File Link 文書をクライアントのコンテンツを使用して作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32}
}
url='file:///c:¥temp¥create.txt' 1
mime='text/plain' 1
filelink_famname='fam01' 2
filelink_contentpath='DATA¥areal¥' 3
```

注 1 url エントリおよび mime エントリは必ず指定します。

注 2 実行環境制御ファイルの FileLink セクションに DefaultFAMName エントリの指定がない場合は必ず指定します。

注 3 実行環境制御ファイルの FileLink セクションに DefaultContentPath エントリの指定がない場合は、必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定して、File Link 文書をファイルサーバ上のコンテンツを使用して作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
link=RCR 1
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',usrProp_Age=27},
  {usrProp_Name='yamada',usrProp_Age=30},
  {usrProp_Name='sato',usrProp_Age=32}
}
url=DBRSVCONTENT 1
mime='text/plain' 1
filelink_famname='fam01' 2
filelink_contentpath='DATA¥area1¥file.jpg' 1
```

注 1 link エントリ、url エントリ、mime エントリおよび filelink_contentpath エントリは必ず指定します。url エントリには、「DBRSVCONTENT」を指定します。filelink_contentpath エントリには、登録するファイルサーバ上のコンテンツのパスを指定します。

注 2 実行環境制御ファイルの FileLink セクションに DefaultFAMName エントリの指定がない場合は、必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定しないでリファレンスファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
url='file:///c:¥temp¥create.doc' 1
mime=' MIME::application/x-edm-undefined' 1
reference_type=RELATIVE 2
reference_contentbasepath='c:¥dir01¥data' 3
reference_contentpath='user01'
```

注 1 url エントリおよび mime エントリは必ず指定します。

注 2 実行環境制御ファイルの Reference セクションに DefaultReferenceType エントリの指定がない場合は、必ず指定します。

注 3 reference_type の指定が「RELATIVE」で、実行環境制御ファイルの Reference セクションに DefaultContentBasePath エントリの指定がない場合は、必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

-T オプションを指定してリファレンスファイル文書を作成する場合

dmaClass_ConfigurationHistory 用プロパティ情報ファイルの記述例

```
;; usrClass_ConfigurationHistory
[usrClass_ConfigurationHistory]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki', usrProp_Age=27},
  {usrProp_Name='yamada', usrProp_Age=30},
  {usrProp_Name='sato',   usrProp_Age=32}
}
link = RCR 1
url='file:///c:¥temp¥create.doc' 1
mime='MIME::application/x-edm-undefined' 1
reference_type=RELATIVE 2
reference_contentbasepath='c:¥dir01¥data' 3
reference_contentpath='user01'
```

注 1 link エントリ、url エントリおよび mime エントリは必ず指定します。

注 2 実行環境制御ファイルの Reference セクションに DefaultReferenceType エントリの指定がない場合は、必ず指定します。

注 3 reference_type の指定が「RELATIVE」で、実行環境制御ファイルの Reference セクションに DefaultContentBasePath エントリの指定がない場合は、必ず指定します。

dmaClass_DocVersion 用プロパティ情報ファイルの記述例

```
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Version'
```

実行例

実行例を次に示します。なお、実行環境制御ファイルの Output セクションの Format エントリの指定、および指定したオプションによって、出力形式が異なります。

また、この例は Windows の場合です。UNIX の場合は、「type PropDV」を「cat PropDV」に、「url=file:///c:¥temp¥create.txt」を「url=file:///tmp/create.txt」に読み替えてください。

5. コマンドリファレンス

```
$ type PropDV
;; usrClass_DocVersion
[usrClass_DocVersion]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Document'
link=RCR
url='file:///c:¥temp¥create.txt'
mime='text/plain'
```

(Formatエントリが「VR0112」の場合)

```
$ dbrcrtdoc -c D PropDV
dma:///07a17522...00000008
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

```
$ dbrcrtdoc -c D -T dma:///07a1...0007 PropDV
dma:///07a17522...00000009
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

(Formatエントリが「VR0200」の場合)

```
$ dbrcrtdoc -c D PropDV
dma:///07a17522...00000010
```

```
$ dbrcrtdoc -c D -T dma:///07a1...0007 PropDV
acl://LR...0010,dma:///07a17522...00000011
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

マルチファイル文書の場合、コマンド引数に指定する次のプロパティ情報ファイルの各セクションに、directory エントリおよび mime エントリを必ず記述してください。

- バージョンなし文書を作成するとき
dmaClass_DocVersion 用プロパティ情報ファイル
- バージョン付き文書を作成するとき
dmaClass_ConfigurationHistory 用プロパティ情報ファイルまたは dmaClass_DocVersion クラス用プロパティ情報ファイル

エントリの記述がない場合はエラーになります。また、バージョン付き文書の場合、両方のプロパティ情報ファイルに directory エントリおよび mime エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効となります。

シングルファイル文書の場合、コマンド引数に指定する次のプロパティ情報ファイルの各セクションに、url エントリおよび mime エントリを必ず記述してください。

- バージョンなし文書を作成するとき
dmaClass_DocVersion 用プロパティ情報ファイル
- バージョン付き文書を作成するとき
dmaClass_ConfigurationHistory 用プロパティ情報ファイルまたは dmaClass_DocVersion クラス用プロパティ情報ファイル

エントリの記述がない場合はエラーになります。また、バージョン付き文書の場合、両方のプロパティ情報ファイルに url エントリおよび mime エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効となります。

-T オプションを指定する場合は、コマンド引数に指定する次のプロパティ情報ファイルの各セクションに link エントリを必ず記述してください。

- バージョンなし文書を作成するとき
dmaClass_DocVersion 用プロパティ情報ファイル
- バージョン付き文書を作成するとき

dmaClass_ConfigurationHistory 用プロパティ情報ファイルまたは dmaClass_DocVersion クラス用プロパティ情報ファイル

エントリの記述がない場合はエラーになります。また、バージョン付き文書の場合、両方のプロパティ情報ファイルに link エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効となります。

構成管理モードのコンテインメントを設定できるのは、次の場合だけです。

- バージョン付き文書作成時に、-T オプションのオプションの引数に構成管理コンテナの OIID または OIID ファイルを指定して作成した場合
- dmaClass_DocVersion 用プロパティ情報ファイルのセクション名に、edmClass_VersionTracedDocVersion クラスもしくはそのサブクラス、または edmClass_VersionTracedComponentDocVersion クラスもしくはそのサブクラスのクラス名を記述した場合

link エントリに VCR を指定した場合は、必ず mode エントリを指定してください。エントリの記述がない場合は、エラーになります。また、両方のプロパティ情報ファイルに mode エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効になります。

-T オプションのオプションの引数にバージョン付き構成管理コンテナの OIID または OIID ファイルを指定した場合は、最新バージョンのバージョン付き構成管理コンテナに包含されます。

バージョン付き文書を作成する時に指定した dmaClass_ConfigurationHistory 用プロパティ情報ファイルと dmaClass_DocVersion 用プロパティ情報ファイルに記述したセクション数が同じでない場合、エラーになります。

コマンド引数に指定するどちらかのプロパティ情報ファイルの各セクションに、次のエントリを必ず指定してください。

- url エントリ
文書として管理するファイルのファイルパスを指定します。
- mime エントリ
文書として管理するファイルの MIME 名を指定します。

エントリの記述がない場合はエラーになります。また、バージョン付き文書の場合、両方のプロパティ情報ファイルに url エントリまたは mime エントリを指定すると、dmaClass_ConfigurationHistory 用プロパティ情報ファイルの値が有効となります。

マルチファイル文書の場合、dbrProp_RetrievalName プロパティに値を設定するには、プロパティ情報ファイルに、プロパティとして dbrProp_RetrievalName を指定します。

コンテンツの登録先を、ファイルサーバとする場合、-k オプションで FileLink を指定します。この場合、-k オプションは省略できません。

File Link 文書の、ファイルサーバ上のコンテンツの削除モードは、クライアントのコンテンツを指定して文書を作成した場合と、ファイルサーバ上に登録されているコンテンツを指定して文書を作成した場合で異なります。それぞれの場合のコンテンツの削除モードを次に示します。

- クライアントのコンテンツを指定して文書を作成した場合
クライアントのコンテンツを指定して作成した文書を削除すると、ファイルサーバ上のコンテンツも同期して削除されます。なお、コンテンツとしてディレクトリを登録したときは、ディレクトリごと削除されます。
- ファイルサーバ上に登録されているコンテンツを指定して文書を作成した場合
ファイルサーバ上に登録されているコンテンツを指定して作成した文書を削除しても、ファイルサーバ上のコンテンツは削除されません。なお、コンテンツにディレクトリを使用しているときでも、ディレクトリは削除されません。

5. コマンドリファレンス

コンテンツの削除モードを変更する場合は、実行環境制御ファイルの FileLink セクションの、ClientContentDeleteMode エントリ、または ServerContentDeleteMode エントリの設定を変更してください。

リファレンスファイル文書の場合、コンテンツの格納先には、DocumentBroker サーバから接続可能であり、かつ DocumentBroker サーバから参照可能なディレクトリを設定してください。

リファレンスファイル文書の場合、コンテンツ格納先ベースパスのディレクトリおよびコンテンツ格納先パスのディレクトリの下位に、DocumentBroker がコンテンツを管理するためのディレクトリが作成され、その下位にコンテンツが格納されます。なお、リファレンスファイル文書のコンテンツと、コンテンツが格納されているディレクトリを同時に削除する場合は、dbrdeldoc コマンド（文書の削除）で、削除ディレクトリのルートパスを指定してください。削除ディレクトリのルートパスの指定方法については、コマンド「dbrdeldoc（文書の削除）」を参照してください。

-T オプションを指定する場合に、コンテナへの包含に失敗すると、ファイルシステム上のコンテンツは削除されません。そのため、必要に応じて手動で削除してください。

dbrctidx (全文検索インデクスの作成)

機能

文書の全文検索インデクスを作成します。全文検索インデクスが作成できるのは、作成時に、`dmaClass_DocVersion` クラスまたはそのサブクラスとして、全文検索機能付き文書クラスを指定した文書です。

操作クラス

`CdbrDocument` クラス

`CdbrVersionableDocument` クラス

形式

```
dbrctidx  [-v {バージョン識別子 | バージョン識別子ファイル}]
          {OIDD | OIDD ファイル} {全文検索インデクスファイル URL}
```

オプション

`-v` {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、全文検索インデクスを作成するバージョンのバージョン識別子を指定します。このオプションを省略した場合は最新バージョンに対して全文検索インデクスが作成されます。バージョンなし文書の場合に、このオプションを指定するとエラーになります。

バージョン識別子

全文検索インデクスを作成するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

全文検索インデクスを作成するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

OIDD

全文検索インデクスを作成する文書のOIDDを指定します。

OIDD ファイル

全文検索インデクスを作成する文書のOIDDを記述したOIDDファイルのファイルパスを指定します。OIDDファイルのファイル名はオブジェクトのOIDD形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIDDファイルに記述できるOIDDの数は一つだけです。

全文検索インデクスファイル URL

全文検索インデクスファイルのファイルパスをURL形式で指定します。例えば、UNIXの場合、「/tmp/index.txt」を使用して全文検索インデクスを作成したいときには、「file:///tmp/index.txt」と指定します。Windowsの場合、「c:\temp\index.txt」を使用して全文検索インデクスを作成したいときには、「file:///c:\temp\index.txt」と指定します。この引数を省略した場合は、すでに登録されているファイルから全文検索インデクスが作成されます。

全文検索インデクスファイルには、登録されている文書の内容に対応するプレーンテキスト形式の

ファイルを指定してください。例えば、Word で作成したファイルを登録した文書に対しては、ユーザプログラムなどでその Word ファイルから抽出したテキストファイルを指定してください。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

すでに全文検索インデックスが作成されている文書に対してこのコマンドを実行した場合、全文検索インデックスを再作成します。

プレーンテキスト用全文検索インデックスが作成できるのは、dmaClass_DocVersion クラスまたはそのサブクラスとして、プレーンテキスト用全文検索機能付き文書クラスを指定して作成した文書です。全文検索機能付き文書クラスについては、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」を参照してください。

全文検索機能付き文書クラス以外の dmaClass_DocVersion クラスまたはそのサブクラスを指定して作成された文書に対して、このコマンドを実行して全文検索インデックスを作成しようとすると、実行メソッドでエラーになります。

dbrdelacl (パブリック ACL の削除)

機能

パブリック ACL を削除します。OIDD ファイルに複数の OIDD を指定すると、複数のパブリック ACL を一括して削除できます。

操作クラス

CdbrPublicACL クラス

形式

dbrdelacl {OIDD | OIDD ファイル}

オプション

なし

コマンド引数

OIDD

削除するパブリック ACL の OIDD を指定します。

OIDD ファイル

削除するパブリック ACL の OIDD を記述した OIDD ファイルのファイルパスを指定します。OIDD ファイルには、複数の OIDD が指定できます。

OIDD ファイルのファイル名はオブジェクトの OIDD 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

注意事項

なし

dbrdelcntr (コンテナの削除)

機能

コンテナを削除します。OIDD ファイルに複数の OIDD を指定すると、複数のコンテナを一括して削除できます。

操作クラス

CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス

形式

```
dbrdelcntr [ -l { DCR [ -R ] | RCR | VCR | ALL } |
            -v { バージョン識別子 | バージョン識別子ファイル } ]
            [ -k Reference
              {{{ [ -a { RELATIVE | NONE } ]
                [ -b コンテント格納先ベースパス ]
                [ -y 削除ディレクトリのルートパス ] } |
              [ -i リファレンス情報ファイル ]
              [ -z ] ]
              { OIDD | OIDD ファイル }
```

オプション

-l { DCR | RCR | VCR | ALL }

包含するオブジェクトとの間に設定されているコンテインメントの種別を指定します。このオプションを省略した場合は、コンテナに包含されているオブジェクトを削除しないで、操作対象に指定したコンテナだけを削除します。

DCR

直接型のコンテインメントでコンテナに包含されているオブジェクトを削除します。

RCR

参照型のコンテインメントでコンテナに包含されているオブジェクトを削除します。

VCR

構成管理型のコンテインメントでコンテナに包含されているオブジェクトを削除します。

ALL

コンテインメントの種別に関係なくコンテナに包含されているオブジェクトを削除します。

-R

操作対象となるコンテナに包含されているコンテナに対して、直接型のコンテインメントでコンテナに包含されているオブジェクトを再帰的に削除します。このオプションは -l DCR オプションを指定した場合だけ有効です。-l DCR オプションを指定していない場合にこのオプションを指定するとエ

ラーになります。

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き構成管理コンテナの場合、削除するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、すべてのバージョンおよびバージョン付き構成管理コンテナ自身が削除されます。

バージョンなしコンテナおよびバージョンなし構成管理コンテナを削除するときにこのオプションを指定するとエラーになります。

バージョン識別子

削除するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

削除するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-k Reference

コンテナに包含されているリファレンスファイル文書を削除します。このオプションは、-l オプションを指定した場合だけ有効です。

-l オプションを省略してこのオプションを指定すると、エラーになります。また、実行環境制御ファイルの Reference セクションで ContentDeleteMode エントリの指定が Specify であり、かつリファレンスファイル文書がコンテナに包含されている場合、このオプションを省略するとエラーになります。

コンテナ内で、リファレンスファイル文書を、シングルファイル文書、マルチファイル文書や FileLink 文書と混在して使用する場合、このオプションの指定を省略してリファレンスファイル文書を削除するには、実行環境制御ファイルの Reference セクションで、ContentDeleteMode エントリに Auto を指定してください。ContentDeleteMode エントリに Auto を指定することで、自動的にリファレンスファイル文書かどうかを判別して、リファレンスファイル文書を削除します。ただし、ContentDeleteMode エントリの指定が Auto の場合でも、実行環境制御ファイルに、リファレンス情報としてリファレンス種別、コンテンツ格納先ベースパスおよび削除ディレクトリのルートパスを指定してください。

削除するコンテナとコンテナに包含されるリファレンスファイル文書の、リファレンス情報が同じかどうかで、リファレンスファイル文書の削除方法が異なります。削除方法を次に示します。

- 削除するコンテナごとのリファレンス情報が同一の場合
 - a オプション、-b オプションおよび -y オプションを指定してください。
- 削除するコンテナごとにリファレンス情報が異なる場合
 - コンテナごとのリファレンス情報をリファレンス情報ファイルに指定し、-i オプションを指定してください。
- 削除するコンテナ内でリファレンス情報が異なる場合
 - このコマンドで、コンテナ内のリファレンスファイル文書のすべては削除できません。

コンテナに包含されるリファレンスファイル文書の削除方法の詳細は、「表 5-8 リファレンスファイル文書を扱う場合のオブジェクトの削除方法」を参照してください。

このオプションを指定した場合も、シングルファイル文書、マルチファイル文書、または FileLink 文書を削除できます。

-a {RELATIVE | NONE}

コンテナに包含されるリファレンスファイル文書を削除する場合に、リファレンスファイル文書のリファレンス種別を指定します。

-k オプションを省略してこのオプションを指定した場合、またはこのオプションと同時に -i オプシ

ンを指定した場合、エラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultReferenceType エントリの指定を使用します。

RELATIVE

コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理するオブジェクトです。

NONE

コンテンツがなく、コンテンツロケーションを管理しないオブジェクトです。

RELATIVE を指定すると、リファレンス種別が NONE のリファレンスファイル文書も削除できません。

-b コンテンツ格納先ベースパス

コンテナに包含されるリファレンスファイル文書を削除する場合に、コンテンツ格納先ベースパスを指定します。このオプションは、実行環境制御ファイルまたは -a オプションで、リファレンス種別の指定が RELATIVE の場合に有効です。

オプション引数には、コンテンツが格納されているディレクトリの基点となるパスを指定します。例えば、コンテナに包含されるリファレンスファイル文書の基点となるパスが「/drv01/dir01/userdir」ディレクトリの場合、このオプションに「/drv01/dir01/userdir」を指定します。このオプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

-k オプションの指定を省略してこのオプションを指定した場合、またはこのオプションと -i オプションを同時に指定した場合、エラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultContentBasePath エントリの指定を使用します。

コンテナとコンテナに包含されるリファレンスファイル文書を削除する場合、リファレンスファイル文書のコンテンツ格納先ベースパスが、すべて一致している必要があります。また、-R オプションを指定した場合も、コンテナの下位のコンテナに包含されるリファレンスファイル文書のコンテンツ格納先ベースパスが、すべて一致している必要があります。コンテナ内でコンテンツ格納先ベースパスが異なる場合の、リファレンスファイル文書の削除方法については、「表 5-8 リファレンスファイル文書を扱う場合のオブジェクトの削除方法」を参照してください。

このオプションに指定したコンテンツ格納先ベースパスは、ContentLocation プロパティに格納されたコンテンツ格納先を示す相対パスと結合して、絶対パスとして使用されます。

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

-y 削除ディレクトリのルートパス

コンテナに包含されるリファレンスファイル文書を削除する場合、リファレンスファイル文書とリファレンスファイル文書が格納されているディレクトリを同時に削除するときは、削除ディレクトリのルートパスを指定します。ただし、コンテンツ格納先ベースパスの上位のディレクトリは指定できません。

このオプションは、実行環境制御ファイルまたは -a オプションで、リファレンス種別の指定が RELATIVE の場合に有効です。

オプション引数には、削除するディレクトリのルートパスを指定します。例えば、「/drv01/dir01/userdir/user01/<DocumentBroker がコンテンツを管理するためのディレクトリ ><登録ファイル名 >」のディレクトリ中の「user01」ディレクトリ以下を削除する場合は、「user01」ディレクトリのルートパスとして「/drv01/dir01/userdir」を指定します。また、コンテンツ格納先ベースパスに「/drv01/dir01/userdir」ディレクトリを指定している場合、コンテンツ格納先ベースパスの上位のディレクトリとなる「/drv01/dir01」は、削除ディレクトリのルートパスに指定できません。なお、この

オプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

-k オプションの指定を省略してこのオプションを指定した場合、またはこのオプションと -i オプションを同時に指定した場合、エラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultDeleteRootPath エントリの指定を使用します。

コンテナに包含されるリファレンスファイル文書を削除する場合、削除するすべてのリファレンスファイル文書の、コンテンツの絶対パスの基点となるパスを指定します。-R オプションを指定する場合も、すべてのリファレンスファイル文書のコンテンツの絶対パスの基点となるパスを指定します。コンテンツの絶対パスには、コンテンツ格納先ベースパスと ContentLocation プロパティに格納されたコンテンツ格納先を示す相対パスを結合したパスが使用されます。

DocumentBroker がコンテンツを管理するためのディレクトリは、このオプションおよび実行環境制御ファイルの Reference セクションの DefaultDeleteRootPath エントリで、削除ディレクトリのルートパスの指定を省略した場合も、コンテンツと同時に削除されます。

Windows の場合、削除ディレクトリのルートパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」を削除ディレクトリのルートパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

-i リファレンス情報ファイル

コンテナごとにコンテンツ格納先ベースパスが異なる運用で、複数のコンテナを一括して、コンテナに包含されているリファレンスファイル文書を削除する場合に、リファレンス情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

-k オプションの指定を省略してこのオプションを指定した場合、またはこのオプションを、-a オプション、-b オプションおよび -y オプションと同時に指定した場合、エラーになります。

リファレンス情報ファイルには、コンテナごとに、リファレンスファイル文書のリファレンス情報として、リファレンス種別、コンテンツ格納先ベースパス、および削除ディレクトリのルートパスを記述します。リファレンス情報ファイルに記述する指定順序は、OIID ファイルで指定する OIID の順序と一致させてください。また、リファレンス情報ファイルに指定するセクション数は、OIID ファイルに指定する OIID の数と一致させてください。リファレンス情報ファイルの記述形式については、「3.8 リファレンス情報ファイル」を参照してください。

-z

コンテナに包含されているリファレンスファイル文書を削除する場合に、トランザクションをコンテナに包含されているオブジェクト単位に分割して実行します。

このオプションを指定すると、コマンドの実行時にエラーが発生しても、リファレンスファイル文書のコンテンツとオブジェクトの不整合が起きません。このオプションを指定しない場合、トランザクションは、コマンド引数で指定するコンテナ単位に分割して実行します。-k オプションの指定を省略してこのオプションを指定した場合、エラーになります。

コマンド引数

OIID

削除するコンテナの OIID を指定します。

OIID ファイル

削除するコンテナの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、-v オプションを指定している場合、記述できる OIID の数は一つだけです。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

ん。

実行例

図 5-1 に示すオブジェクト構成を前提として OBJ_11 を削除した場合の、削除対象となる下位オブジェクトとオプション指定との関係を表 5-3 に示します。

図 5-1 オブジェクト構成

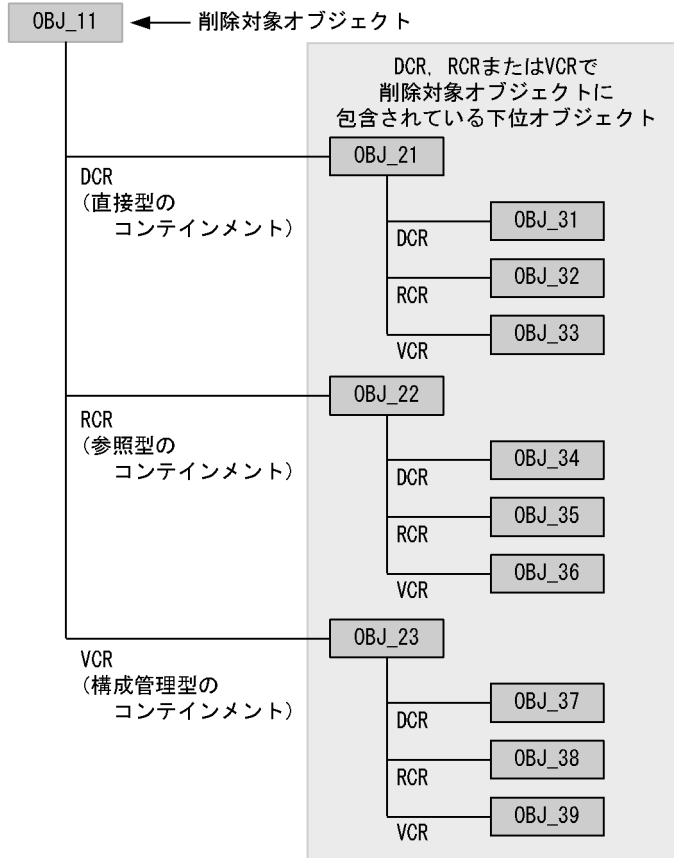


表 5-3 -l オプションの指定と削除対象になる下位オブジェクト

-l オプションの指定	削除されるオブジェクト	コンテナメントによる包含関係が解除されるオブジェクト
指定なし	OBJ_11	OBJ_21, OBJ_22, OBJ_23
-l DCR	OBJ_11, OBJ_21	OBJ_22, OBJ_23, OBJ_31, OBJ_32, OBJ_33
-l DCR -R	OBJ_11, OBJ_21, OBJ_31	OBJ_22, OBJ_23, OBJ_32, OBJ_33
-l RCR	OBJ_11, OBJ_22	OBJ_21, OBJ_23, OBJ_34, OBJ_35, OBJ_36
-l VCR	OBJ_11, OBJ_23	OBJ_21, OBJ_22, OBJ_37, OBJ_38, OBJ_39
-l ALL	OBJ_11, OBJ_21, OBJ_22, OBJ_23	OBJ_31, OBJ_32, OBJ_33, OBJ_34, OBJ_35, OBJ_36, OBJ_37, OBJ_38, OBJ_39

注意事項

バージョン付き構成管理コンテナまたはバージョンなし構成管理コンテナを包含しているバージョンなしコンテナに対して、-l オプションを指定すると、エラーになります。

-l オプションの指定によって削除されたコンテナと、そのコンテナに含まれるオブジェクトの間に設定されていた包含関係はすべて解除されます。

コンテナの種別に関係なく、-l オプションを指定した場合は、ほかのオブジェクトを包含している場合でも、包含されるオブジェクトは削除されます。

-l オプションの指定によって削除される下位オブジェクトが文書の場合に、文書に文書間リレーションが設定されているとき、リレーションは次のように扱われます。

- 削除される下位オブジェクトがリレーション元文書の場合、リレーションも削除されます。
- 削除される下位オブジェクトがリレーション先文書の場合、リレーションは削除されません。

-v オプション指定時には、複数のオブジェクトを一括して削除できません。指定したバージョンを削除できるコンテナは一つだけです。OID ファイルに複数の OID を指定した場合はエラーになります。

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

バージョンなしコンテナの場合に、-IVCR オプションを指定すると、下位オブジェクトは削除されず、バージョンなしコンテナだけが削除されます。

バージョン付き構成管理コンテナと同時に削除できるのは、最新バージョンに含まれているオブジェクトです。

リファレンスファイル文書の場合、コンテンツの格納先には、DocumentBroker サーバから接続可能であり、かつ DocumentBroker サーバから参照可能なディレクトリを設定してください。

リファレンスファイル文書を扱うコンテンツの操作でエラーになると、オブジェクトとコンテンツの状態が不整合となり、メッセージ KMBR30201-E (要因コード 3, 86) が出力される場合があります。この場合は、再度、dbrdelcntr コマンド (コンテナの削除) を実行してオブジェクトを削除してください。また、メッセージの詳細については、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

リファレンスファイル文書を削除する場合に、誤って異なるリファレンスファイル文書のコンテンツ格納先ベースパスを指定すると、オブジェクトは削除され、コマンドは正常終了します。ただし、コンテンツは削除されないため、必要に応じて手動で削除してください。

リファレンスファイル文書を削除する場合に、リファレンスファイル文書かどうかを自動的に判別して削除するときは、実行環境制御ファイルの Reference セクションで ContentDeleteMode エントリに Auto を指定します。

また、ContentDeleteMode エントリの指定が Specify の場合は、-k オプションの指定に従ってリファレンスファイル文書を削除します。そのため、-k オプションの指定を省略するとエラーになります。ContentDeleteMode エントリの指定と dbrdelcntr コマンド (コンテナの削除) の実行結果については、「表 3-2 ContentDeleteMode エントリの指定と dbrdelcntr コマンド、dbrdeldoc コマンドの実行結果」を参照してください。

-l オプションを指定して、コンテナに含まれるリファレンスファイル文書を削除する場合、コンテナごとにリファレンスファイル文書のコンテンツ格納先ベースパスが同一である必要があります。コンテナ内に、異なるコンテンツ格納先ベースパスのリファレンスファイル文書が混在している場合は、コンテンツ格納先ベースパスが同一のリファレンスファイル文書に絞り込んで、同一のコンテンツ格納先ベースパスごとに dbrdeldoc コマンド (文書の削除) でリファレンスファイル文書を削除します。そのあと、dbrdelcntr コマンド (コンテナの削除) でコンテナを削除してください。リファレンスファイル文書の削除の詳細については、「表 5-8 リファレンスファイル文書を扱う場合のオブジェクトの削除方法」を参照してください。

-i オプションと OID ファイルを指定して、複数のコンテナを一括して削除する場合、-i オプションで指定したリファレンス情報ファイルのセクション数と OID ファイルの OID の数が異なると、エラー

5. コマンドリファレンス

になります。

コンテナに包含されるリファレンスファイル文書を削除する場合、`-k Reference` オプションおよび `-z` オプションを指定します。これらのオプションを指定することで、トランザクションがコンテナに包含されるオブジェクト単位に分割して実行され、エラー発生時のオブジェクトとコンテンツの不整合を防ぐことができます。`-z` オプションを指定しない場合、トランザクションは、コマンド引数で指定するコンテナ単位に分割して実行されます。

dbrdeldata (独立データの削除)

機能

独立データを削除します。OIDD ファイルに複数の OIDD を指定すると、複数の独立データを一括して削除できます。

操作クラス

CdbrIndependentPersistence クラス

形式

dbrdeldata {OIDD | OIDD ファイル}

オプション

なし

コマンド引数

OIDD

削除する独立データの OIDD を指定します。

OIDD ファイル

削除する独立データの OIDD を記述した OIDD ファイルのファイルパスを指定します。OIDD ファイルには、複数の OIDD が指定できます。

OIDD ファイルのファイル名はオブジェクトの OIDD 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

注意事項

なし

dbrdeldoc (文書の削除)

機能

文書を削除します。OIDD ファイルに複数の OIDD を指定すると、複数の文書を一括して削除できます。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

```
dbrdeldoc [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           [ -k Reference
           [ -a {RELATIVE | NONE} ]
           [ -b コンテンツ格納先ベースパス ]
           [ -y 削除ディレクトリのルートパス ]]
           {OIDD | OIDD ファイル}
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、削除するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、すべてのバージョンおよびバージョン付き文書自身が削除されます。バージョンなし文書の場合にこのオプションを指定するとエラーになります。

バージョン識別子

削除するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

削除するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-k Reference

リファレンスファイル文書を削除します。

実行環境制御ファイルの Reference セクションで ContentDeleteMode エントリの指定が Specify の場合、このオプションを省略するとエラーになります。

リファレンスファイル文書を、シングルファイル文書、マルチファイル文書や FileLink 文書と混在して使用する場合、このオプションの指定を省略してリファレンスファイル文書を削除するには、実行環境制御ファイルの Reference セクションで、ContentDeleteMode エントリに Auto を指定してください。ContentDeleteMode エントリに Auto を指定することで、自動的にリファレンスファイル文書かどうかを判別して、リファレンスファイル文書を削除します。ただし、ContentDeleteMode エントリの指定が Auto の場合でも、実行環境制御ファイルにリファレンス情報として、リファレンス種別、コンテンツ格納先ベースパスおよび削除ディレクトリのルートパスを指定してください。

このオプションを指定した場合も、シングルファイル文書、マルチファイル文書、または File Link 文書を削除できます。

-a { RELATIVE | NONE }

削除するリファレンスファイル文書のリファレンス種別を指定します。

-k オプションの指定を省略して、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultReferenceType エントリの指定を使用します。

RELATIVE

コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理するオブジェクトです。

NONE

コンテンツがなく、コンテンツロケーションを管理しないオブジェクトです。

RELATIVE を指定すると、リファレンス種別が NONE のリファレンスファイル文書も削除できません。

-b コンテンツ格納先ベースパス

リファレンスファイル文書を削除する場合に、コンテンツ格納先ベースパスを指定します。このオプションは、実行環境制御ファイルまたは -a オプションで、リファレンス種別の指定が RELATIVE の場合に有効です。

オプション引数には、コンテンツが格納されているディレクトリの基点となるパスを指定します。例えば、削除するリファレンスファイル文書の基点となるパスが「/drv01/dir01/userdir」ディレクトリの場合、このオプションに「/drv01/dir01/userdir」を指定します。このオプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

-k オプションの指定を省略して、このオプションを指定するとエラーになります。このオプションの指定を省略した場合、実行環境制御ファイルの、Reference セクションの DefaultContentBasePath エントリの指定を使用します。

複数のリファレンスファイル文書を一括して削除する場合は、同一のコンテンツ格納先ベースパスのリファレンスファイル文書に絞り込んだあと、このコマンドを実行してください。

このオプションに指定したコンテンツ格納先ベースパスは、ContentLocation プロパティに格納されたコンテンツ格納先を示す相対パスと結合して、絶対パスとして使用されます。

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

-y 削除ディレクトリのルートパス

リファレンスファイル文書を削除する場合、リファレンスファイル文書とリファレンスファイル文書が格納されているディレクトリを同時に削除するときは、削除ディレクトリのルートパスを指定します。ただし、コンテンツ格納先ベースパスの上位のディレクトリは指定できません。

このオプションは、実行環境制御ファイルまたは -a オプションで、リファレンス種別の指定が RELATIVE の場合に有効です。

オプション引数には、削除するディレクトリのルートパスを指定します。例えば、「/drv01/dir01/userdir/user01/<DocumentBroker がコンテンツを管理するためのディレクトリ >/<登録ファイル名 >」のディレクトリ中の「user01」ディレクトリ以下を削除する場合は、「user01」ディレクトリのルートパスとして「/drv01/dir01/userdir」を指定します。また、コンテンツ格納先ベースパスに「/drv01/dir01/userdir」ディレクトリを指定している場合、コンテンツ格納先ベースパスの上位のディレクトリとなる「/drv01/dir01」は、削除ディレクトリのルートパスに指定できません。なお、このオプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

-k オプションの指定を省略して、このオプションを指定するとエラーになります。このオプションを

省略した場合、実行環境制御ファイルの、Reference セクションの DefaultDeleteRootPath エントリの指定を使用します。

複数の文書を一括して削除する場合、削除ディレクトリのルートパスには、削除するすべてのリファレンスファイル文書の、コンテンツの絶対パスの基点となるパスを指定します。

DocumentBroker がコンテンツを管理するためのディレクトリは、このオプションおよび実行環境制御ファイルの Reference セクションの DefaultDeleteRootPath エントリで、削除ディレクトリのルートパスの指定を省略した場合も、コンテンツと同時に削除されます。

Windows の場合、削除ディレクトリのルートパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」を削除ディレクトリのルートパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

コマンド引数

OIID

削除する文書の OIID を指定します。

OIID ファイル

削除する文書の OIID を記述した ID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、-v オプションを指定している場合、記述できる OIID の数は一つだけです。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

注意事項

-v オプション指定時には、複数の文書を一括して削除できません。指定したバージョンを削除できる文書は一つだけです。OIID ファイルに複数の OIID を指定した場合はエラーになります。

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

文書に文書間リレーションが設定されている場合、リレーションは次のように扱われます。

- 削除する文書がリレーション元文書の場合、リレーションも削除されます。
- 削除する文書がリレーション先文書の場合、リレーションは削除されません。

File Link 文書の、ファイルサーバ上のコンテンツの削除モードは、クライアントのコンテンツを指定して、文書を作成もしくはアップロードされている場合、またはレンディションを追加されている場合と、ファイルサーバ上に登録されているコンテンツを指定して、文書を作成もしくはアップロードされている場合、またはレンディションを追加されている場合で異なります。それぞれの場合のコンテンツの削除モードを次に示します。

- クライアントのコンテンツを指定して、文書を作成もしくはアップロードされている場合、またはレンディションを追加されている場合
文書、文書のレンディション、またはバージョンを削除すると、ファイルサーバ上のコンテンツも同期して削除されます。なお、コンテンツとしてディレクトリを登録したときは、ディレクトリごと削除されます。
- ファイルサーバ上に登録されているコンテンツを指定して、文書を作成もしくはアップロードされている場合、またはレンディションを追加されている場合
文書、文書のレンディション、またはバージョンを削除しても、ファイルサーバ上のコンテンツは削除されません。なお、コンテンツにディレクトリを使用しているときでも、ディレクトリは削除されません。

コンテンツの削除モードを変更する場合は、実行環境制御ファイルの FileLink セクションの、

ClientContentDeleteMode エントリ，または ServerContentDeleteMode エントリの設定を変更してください。

リファレンスファイル文書の場合，コンテンツの格納先には，DocumentBroker サーバから接続可能であり，かつ DocumentBroker サーバから参照可能なディレクトリを設定してください。

リファレンスファイル文書を扱うコンテンツの操作でエラーになると，オブジェクトとコンテンツの状態が不整合となり，メッセージ KMBR30201-E（要因コード 3，86）が出力される場合があります。この場合は，再度，dbrdeldoc コマンド（文書の削除）を実行してオブジェクトを削除してください。メッセージの詳細については，マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

リファレンスファイル文書を削除する場合に，誤って異なるリファレンスファイル文書のコンテンツ格納先ベースパスを指定すると，オブジェクトは削除され，コマンドは正常終了します。ただし，コンテンツは削除されないため，必要に応じて手動で削除してください。

リファレンスファイル文書を削除する場合に，リファレンスファイル文書かどうかを自動的に判別して削除するときは，実行環境制御ファイルの Reference セクションで ContentDeleteMode エントリに Auto を指定します。

また，ContentDeleteMode エントリの指定が Specify の場合は，-k オプションの指定に従ってリファレンスファイル文書を削除します。このとき，-k オプションの指定を省略するとエラーになります。ContentDeleteMode エントリの指定と dbrdeldoc コマンド（文書の削除）の実行結果については，「表 3-2 ContentDeleteMode エントリの指定と dbrdelcntr コマンド，dbrdeldoc コマンドの実行結果」を参照してください。

リファレンスファイル文書を削除する場合に，指定できるコンテンツ格納先ベースパスは一つだけです。複数のコンテンツ格納先ベースパスを管理する運用で，一括して複数の文書を削除したい場合は，同一のコンテンツ格納先ベースパスのリファレンスファイル文書に絞り込んで，dbrdeldoc コマンド（文書の削除）を実行してください。リファレンスファイル文書の削除の詳細については，「表 5-8 リファレンスファイル文書を扱う場合のオブジェクトの削除方法」を参照してください。

dbrdelidx (全文検索インデクスの削除)

機能

文書の全文検索インデクスを削除します。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

```
dbrdelidx [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           { OIID | OIID ファイル }
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、全文検索インデクスを削除するバージョンのバージョン識別子を指定します。このオプションを省略した場合は最新バージョンの全文検索インデクスが削除されます。バージョンなし文書の場合、このオプションを指定するとエラーになります。

バージョン識別子

全文検索インデクスを削除するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

全文検索インデクスを削除するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

OIID

全文検索インデクスを削除する文書の OIID を指定します。

OIID ファイル

全文検索インデクスを削除する文書の OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

全文検索インデクスがない場合は、何もしないで正常終了します。

全文検索機能付き文書クラス以外の dmaClass_DocVersion クラスまたはそのサブクラスを指定して作成した文書に対して、このコマンドを実行して全文検索インデクスを削除しようとする、実行メソッドでエラーになります。

dbrdowndoc (文書のダウンロード)

機能

文書に登録されているファイルをダウンロードします。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

```
dbrdowndoc [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           [ -M レンディションタイプ ]
           [ -k { Content | MultiContent | FileLink | Reference
               [ -a {RELATIVE} ]
               [ -b コンテント格納先ベースパス ] } ]
           { OIID | OIID ファイル }
           { ダウンロードファイル URL | ダウンロードディレクトリ URL }
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、ダウンロードするバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンが仮定されます。

バージョンなし文書をダウンロードするときにこのオプションを指定するとエラーになります。

バージョン識別子

ダウンロードするバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

ダウンロードするバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-M レンディションタイプ

ダウンロードするレンディションタイプを指定します。このオプションを省略した場合は、マスターレンディションが仮定されます。

-k { Content | MultiContent | FileLink | Reference }

ダウンロードする文書のコンテンツ種別を指定します。このオプションを指定した場合、ダウンロードファイル URL、またはダウンロードディレクトリ URL の指定が、このオプションの指定と一致していないとエラーになります。このオプションを省略した場合、ダウンロードファイル URL、またはダウンロードディレクトリ URL の指定に従ってコンテンツをダウンロードします。

Content

シングルファイル文書のコンテンツを、データベースからダウンロードします。ダウンロード

URL に指定されたファイルに、コンテンツをダウンロードします。

MultiContent

マルチファイル文書のコンテンツを、データベースからダウンロードします。ダウンロードディレクトリ URL に指定されたディレクトリに、コンテンツをダウンロードします。

FileLink

File Link 文書のコンテンツを、ファイルサーバからダウンロードします。ダウンロード URL に指定されたファイル、またはダウンロードディレクトリ URL に指定されたディレクトリに、コンテンツをダウンロードします。

Reference

リファレンスファイル文書のコンテンツを、ファイルシステム上の領域からダウンロードします。ダウンロード URL に指定されたファイルに、コンテンツをダウンロードします。

オプション省略時

ダウンロードファイル URL、またはダウンロードディレクトリ URL の指定に従ってコンテンツをダウンロードします。

- ダウンロードファイル URL が指定されている場合
シングルファイル文書のコンテンツをダウンロードします。
- ダウンロードディレクトリ URL が指定されている場合
マルチファイル文書のコンテンツをダウンロードします。

-a { RELATIVE }

ダウンロードするリファレンスファイル文書のリファレンス種別を指定します。

-k オプションに Reference を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultReferenceType エントリの指定を使用します。

RELATIVE

リファレンス種別が RELATIVE のリファレンスファイル文書のコンテンツをダウンロードします。このオプションを指定した場合、-b オプションにコンテンツ格納先ベースパスを指定します。

-b コンテンツ格納先ベースパス

リファレンスファイル文書をダウンロードする場合に、コンテンツ格納先ベースパスを指定します。オプション引数には、ダウンロードするコンテンツが格納されているディレクトリの基点となるパスを指定します。例えば、リファレンスファイル文書のコンテンツの基点となるパスが「/drv01/dir01/userdir」ディレクトリの場合、このオプションに「/drv01/dir01/userdir」を指定します。このオプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

-k オプションに Reference を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultContentBasePath エントリの指定を使用します。

このオプションに指定したコンテンツ格納先ベースパスは、ContentLocation プロパティに格納されたコンテンツ格納先を示す相対パスと結合して、絶対パスとして使用されます。

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

コマンド引数

OIID

ダウンロードする文書の OIID を指定します。

OIID ファイル

ダウンロードする文書の OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

ダウンロードファイル URL

シングルファイル文書、File Link 文書（コンテンツとしてファイルを登録している）またはリファレンスファイル文書の場合、ダウンロードファイルのファイルパスを URL 形式で指定します。例えば、UNIX の場合、「/tmp/download.doc」にダウンロードするときには、「file:///tmp/download.doc」と指定します。Windows の場合、「c:¥temp¥download.doc」にダウンロードするときには、「file:///c:¥temp¥download.doc」と指定します。

ダウンロードディレクトリ URL

マルチファイル文書、または File Link 文書（コンテンツとしてディレクトリまたはファイルを登録している）の場合、ダウンロードファイルまたはダウンロードディレクトリを格納するディレクトリパスを、URL 形式で指定します。例えば、UNIX の場合、「/tmp」にダウンロードファイルを格納するときには、「file:///tmp」と指定します。Windows の場合、「c:¥temp」にダウンロードファイルを格納するときには、「file:///c:¥temp」と指定します。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

-k オプションを省略した場合、シングルファイル文書（レンディション）またはマルチファイル文書（レンディション）をダウンロードの対象にします。コマンド引数に、ダウンロードファイル URL を指定すると、ダウンロードする文書はシングルファイル文書として扱われます。また、コマンド引数に、ダウンロードディレクトリ URL を指定すると、ダウンロードする文書はマルチファイル文書として扱われます。コマンド引数の指定と、ダウンロードする文書の種類が一致しないとエラーになります。

File Link 文書で、ファイルサーバにコンテンツを登録した文書（レンディション）のコンテンツをダウンロードする場合、-k オプションで FileLink を指定します。この場合、-k オプションは省略できません。ダウンロードファイル URL またはダウンロードディレクトリ URL に指定したディレクトリが存在しないと、ディレクトリが作成され、コンテンツがダウンロードされます。なお、ダウンロードファイル URL に指定したファイルが存在するとエラーになります。

リファレンスファイル文書の場合、コンテンツの格納先には、DocumentBroker サーバから接続可能であり、かつ DocumentBroker サーバから参照可能なディレクトリを設定してください。

リファレンスファイル文書のコンテンツをダウンロードする場合に、誤って異なるリファレンスファイル文書のコンテンツ格納先ベースパスを指定すると、エラーになります。

dbrexquery (問い合わせの実行)

機能

問い合わせファイルに記述された問い合わせ式を実行します。実行結果は標準出力に出力されます。

実行結果の出力形式について、次に示します。

1 オブジェクトにつき 1 行で標準出力に出力されます。

指定したプロパティがコンマ (,) で区切って出力されます。

操作クラス

検索可能な DMA のクラス

形式

dbrexquery 問い合わせファイル

オプション

なし

コマンド引数

問い合わせファイル

実行する問い合わせ式を記述したファイルのファイルパスを指定します。問い合わせファイルのファイル名は任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

実行例

実行例を次に示します。なお、実行環境制御ファイルでの Output セクションの Format エントリの指定によって、出力形式が異なります。

また、この例は Windows の場合です。UNIX の場合は、「type QueryFile」を「cat QueryFile」に読み替えてください。

```
$ type QueryFile
SELECT ALL
    usrClass_Container.usrProp_IsReadOnly,
    usrClass_Container.usrProp_State,
    usrClass_Container.usrProp_Title,
    usrClass_Container.usrProp_Authors
FROM  usrClass_Container
WHERE  usrClass_Container.usrProp_Title
        LIKE 'container_' ESCAPE '¥'
    AND  usrClass_Container.usrProp_Authors[ANY] .usrProp_Age <= 30
ORDER BY  usrClass_Container.usrProp_State
```

(Format エントリが「VR0112」の場合)

```
$ dbrexquery QueryFile
TRUE,1,container1,{{suzuki,27},{yamada,30},{sato,32}}
FALSE,2,container2,{tanaka,22}
KMBR32115-I 検索を実行しました。
```

(Format エントリが「VR0200」の場合)

```
$ dbrexquery QueryFile
```

```
TRUE, 1, 'container1', {{ 'suzuki', 27 }, { 'yamada', 30 }, { 'sato', 32 }}
FALSE, 2, 'container2', { 'tanaka', 22 }
KMBR32115-I 検索を実行しました。
```

注意事項

出力可能なプロパティは基本単位が Scalar 型でデータ型が String 型, Integer32 型, Boolean 型のプロパティ, および基本単位が VariableArray 型でデータ型が Object 型のプロパティです。

出力可能な構成要素プロパティのデータ型は String 型, Integer32 型, Boolean 型です。

基本単位が VariableArray 型でデータ型が Object 型プロパティは, プロパティおよび構成要素プロパティが, 波括弧 ({ }) で囲まれて出力されます。各構成要素プロパティはコマンドで区切られて出力されます。この時, 出力できるすべての構成要素プロパティが出力されます。

1 行に出力できるサイズは, UNIX の場合, NL (0x0a) を除いてデフォルトでは 8,190 バイト以内です。Windows の場合, CR + NL (0x0d + 0x0a) を除いてデフォルトでは 8,190 バイト以内です。8,190 バイトを超える場合は可能な範囲で出力されて, メッセージが標準エラー出力に出力されます。

1 行に出力するサイズを変更するときは, UNIX の場合, 実行環境ファイルで Output セクションの LineSize エントリに NL (0x0a) を除いた出力サイズを指定してください。Windows の場合, 実行環境ファイルで Output セクションの LineSize エントリに CR + NL (0x0d + 0x0a) を除いた出力サイズを指定してください。

選択リストに基本単位が VariableArray 型でデータ型が Object 型のプロパティを指定した場合 DISTINCT を指定しても重複排除されません。

選択リストとして指定できるプロパティのデータ型は, String 型, Integer32 型, Boolean 型, および基本単位が VariableArray 型の Object 型です。ただし, 基本単位が VariableArray 型の Object 型のプロパティの添え字と, その構成要素となるクラスのプロパティは指定できません。

指定できる問い合わせ文字列のサイズはコメント行および空白行を除いて 8,190 バイトです。

値が NULL 値の場合は, 何も出力されません。

プロパティ出力時にエラーが発生した場合, 「ERROR」という文字列と, 付加情報が出力されます。

実行環境制御ファイルで Output セクションの Format エントリに「VR0200」を指定した場合, 文字としてのシングルクォート (') を含む String 型プロパティが出力されるときは, 文字列を囲むシングルクォート (') とは別にシングルクォート (') が二つ続けて出力されます。

dbbrgetprop (プロパティの取得)

機能

プロパティ情報ファイルに記述した内容に従って、オブジェクトのプロパティを取得します。OIDD ファイルに複数の OIDD を指定すると、複数のオブジェクトのプロパティを一括して取得できます。

操作クラス

CdbrIndependentPersistence クラス
 CdbrDocument クラス
 CdbrVersionableDocument クラス
 CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス
 CdbrPublicACL クラス

形式

```
dbbrgetprop [ {-v {バージョン識別子 | バージョン識別子ファイル} |
              -L {リンク識別子 | リンク識別子ファイル}} ]
              {OIDD | OIDD ファイル} プロパティ情報ファイル
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書またはバージョン付き構成管理コンテナの場合、プロパティを取得するバージョンのバージョン識別子を指定します。このオプションおよび -L オプションを省略した場合は、すべてのバージョンで共有するプロパティを取得します。

バージョン付き文書およびバージョン付き構成管理コンテナ以外のプロパティを取得する場合にこのオプションを指定するとエラーになります。

バージョン識別子

プロパティを取得するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

プロパティを取得するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-L {リンク識別子 | リンク識別子ファイル}

コンテナに設定したコンテナメントのプロパティを取得する場合に、プロパティを取得するコンテナメントのリンク識別子を指定します。このオプションおよび -v オプションを省略した場合は、すべてのバージョンで共有するプロパティを取得します。

コンテナ以外のプロパティを取得するときにこのオプションを指定するとエラーになります。

リンク識別子

プロパティを取得するコンテナメントのリンク識別子を指定します。

リンク識別子ファイル

プロパティを取得するコンテナメントのリンク識別子を記述したリンク識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「acl://」で始まるパスは指定できません。なお、リンク識別子ファイルに記述できるリンク識別子の数は一つだけです。

コマンド引数

OIID

プロパティを取得するオブジェクトの OIID を指定します。

OIID ファイル

プロパティを取得するオブジェクトの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、`-v` オプションを指定している場合または `-L` オプションを指定している場合、記述できる OIID の数は一つだけです。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

プロパティ情報ファイル

取得するオブジェクトのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

プロパティ情報ファイルの記述例

独立データのプロパティを取得する場合の `edmClass_IndependentPersistence` 用プロパティ情報ファイルの記述例を示します。このファイルには、取得するプロパティの名前だけ指定します。

```
;; usrClass_IndependentPersistence
[usrClass_IndependentPersistence]
usrProp_IsReadOnly
usrProp_State
usrProp_Title
usrProp_Authors
```

実行例

実行例を次に示します。なお、この例は Windows の場合です。UNIX の場合は、「type PropFile」を「cat PropFile」に読み替えてください。

```
$ type PropFile
[usrClass_IndependentPersistence]
usrProp_IsReadOnly
usrProp_State
usrProp_Title
usrProp_Authors
$ dbrgetprop dma:///07a17522-...-0000000001 PropFile
# Properties by dbrgetprop 1999/08/01 15:00:00
# dma:///07a17522-a626-11d0-b11f-0020af27a837/673d2be0-...-0000000001
[usrClass_IndependentPersistence]
usrProp_IsReadOnly=TRUE
usrProp_State=1
usrProp_Title='Independent'
usrProp_Authors[3][2]=
{
  {usrProp_Name='suzuki',
```

```

    usrProp_Age=27},
  {usrProp_Name='yamada',
    usrProp_Age=30},
  {usrProp_Name='sato',
    usrProp_Age=32}
}

```

注意事項

文字としてのシングルクォート (') を含む String 型プロパティを取得する場合、文字列を囲むシングルクォート (') とは別にシングルクォート (') が二つ続けて出力されます。

OIID ファイルに記述した OIID 数とプロパティ情報ファイルに記述したセクション数が同じでない場合、エラーになります。オブジェクトの OIID を指定した場合、プロパティ情報ファイルに記述するセクション数は一つでなければなりません。

取得するプロパティを記述したプロパティ情報ファイルには値を指定する必要はありません。また、指定していてもエラーにはなりません。

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

-v オプションまたは -L オプション指定時には、複数のオブジェクトのプロパティを一括して取得することはできません。指定したバージョンまたはコンテナメントのプロパティを取得できるオブジェクトは一つだけです。OIID ファイルに複数の OIID を指定した場合はエラーになります。

バージョン付き構成管理コンテナに対して、-v オプションと -L オプションを同時には指定できません。

基本単位が VariableArray 型のプロパティとしては、データ型が Object 型以外のプロパティを出力することはできません。

クラスライブラリ固有のプロパティは、クラス定義情報ファイルに記述していなくても、セクション名に指定したクラスのプロパティとして取得できます。ただし、指定したオブジェクトの操作クラスに導入されていないクラスライブラリ固有のプロパティを指定した場合は、実行メソッドでエラーになります。

プロパティ情報ファイルに記述したクラス名は、クラス定義情報ファイルにアクセスするときだけ使用します。プロパティの取得には影響しません。

値が NULL 値の場合は、「NULL」という文字列が出力されます。

値取得時にエラーが発生した場合、「ERROR」という文字列と付加情報が出力されます。

dbmlink (オブジェクト間の関連づけ)

機能

コンテインメントによる二つのオブジェクト間の関連づけを設定します。コンテインメントが設定されると、設定されたコンテインメントのリンク識別子が標準出力に出力されます。

操作クラス

包含するオブジェクト

CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス

包含されるオブジェクト

CdbrDocument クラス
 CdbrVersionableDocument クラス
 CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス

形式

```
dbmlink -l { DCR | RCR | VCR } -m { FLOATING | FIX }
      [-w { 包含されるオブジェクトのバージョン識別子 |
            包含されるオブジェクトのバージョン識別子ファイル } ] ] ]
      [-v { 包含するオブジェクトのバージョン識別子 |
            包含するオブジェクトのバージョン識別子ファイル } ]
      { 包含するオブジェクトの OIID |
        包含するオブジェクトの OIID ファイル }
      { 包含されるオブジェクトの OIID |
        包含されるオブジェクトの OIID ファイル }
```

オプション

-l { DCR | RCR | VCR }
 設定するコンテインメントの種別を指定します。このオプションは省略できません。

DCR
 直接型のコンテインメントを設定します。

RCR
 参照型のコンテインメントを設定します。

VCR

構成管理型のコンテインメントを設定します。VCR を設定したときは、必ず `-m` オプションを指定してください。

`-m { FLOATING | FIX }`

設定する構成管理モード種別を指定します。このオプションは、`-l VCR` オプションを指定した場合だけ指定できます。また、`-l VCR` オプションを指定した場合は、省略できません。

FLOATING

FLOATING モードを設定します。

FIX

FIX モードを設定します。

このモードを指定したとき、`-w` オプションを指定できます。

`-w { バージョン識別子 | バージョン識別子ファイル }`

含まれるオブジェクトがバージョン付き文書またはバージョン付き構成管理コンテナの場合、コンテインメントを設定するバージョンのバージョン識別子を指定します。このオプションは、`-m FIX` オプションを指定した場合だけ指定できます。このオプションを省略した場合は、最新バージョンにコンテインメントが設定されます。

含まれるオブジェクトがバージョン付き文書およびバージョン付き構成管理コンテナ以外の場合、このオプションを指定するとエラーになります。

バージョン識別子

含まれるバージョン付き文書またはバージョン付き構成管理コンテナのバージョン識別子を指定します。

バージョン識別子ファイル

含まれるバージョン付き文書またはバージョン付き構成管理コンテナのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「`DBRVTVERSION`」という文字列および「`acl://`」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

`-v { バージョン識別子 | バージョン識別子ファイル }`

包含するオブジェクトがバージョン付き構成管理コンテナの場合、コンテインメントを設定するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンにコンテインメントが設定されます。

包含するオブジェクトがバージョンなしコンテナまたはバージョンなし構成管理コンテナの場合、このオプションを指定するとエラーになります。

バージョン識別子

包含するオブジェクトであるバージョン付き構成管理コンテナのバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

包含するオブジェクトであるバージョン付き構成管理コンテナのバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「`DBRVTVERSION`」という文字列および「`acl://`」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

包含するオブジェクトの OIID

オブジェクトを包含するコンテナの OIID を指定します。

包含するオブジェクトの OIID を記述した OIID ファイル

オブジェクトを包含するコンテナの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

包含されるオブジェクトの OIID

コンテナに包含されるオブジェクトの OIID を指定します。

包含されるオブジェクトの OIID を記述した OIID ファイル

コンテナに包含されるオブジェクトの OIID を記述した ID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

実行例

実行例を次に示します。なお、実行環境制御ファイルの Output セクションの Format エントリの指定、および指定したオプションによって、出力形式が異なります。

```
(Formatエントリが「VR0112」の場合)
$ dbrlink -l RCR dma:///07a1...0018 dma:///07a1...0019
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

```
(Formatエントリが「VR0200」の場合)
$ dbrlink -l RCR dma:///07a1...0018 dma:///07a1...0019
acl:///LR...00000019
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

すでに直接型のコンテナメントが設定されている二つのオブジェクト間に、さらに直接型のコンテナメントを設定しようとするると実行メソッドでエラーになります。ただし、すでに参照型のコンテナメントまたは構成管理型のコンテナメントが設定されている二つのオブジェクト間に、直接型のコンテナメント、参照型のコンテナメントまたは構成管理型のコンテナメントをさらに設定することはできません。

-l VCR オプション指定時に包含するオブジェクトとして指定できるのは、バージョン付き文書またはバージョン付き構成管理コンテナです。

-v オプションおよび -w オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

構成管理の対象になる文書は、次のクラスを指定して作成したバージョン付き文書だけです。

- edmClass_VersionTracedDocVersion クラスまたはそのサブクラス
- edmClass_VersionTracedComponentDocVersion クラスまたはそのサブクラス

-l VCR オプション指定時に -m オプションを指定しないとエラーになります。また、-l DCR オプションおよび -l RCR オプション指定時に -m オプションを指定した場合はエラーになります。

dbrlsACL (バインドしているパブリック ACL 一覧または バインドされているオブジェクト一覧の取得)

機能

バインドしているパブリック ACL またはバインドされているオブジェクトの一覧を取得します。取得した情報は標準出力に出力されます。

指定したオブジェクトがパブリック ACL 以外の場合は、バインドしているパブリック ACL の情報が出力されます。

指定したオブジェクトがパブリック ACL の場合は、バインドされているオブジェクトの情報が出力されます。

OIID ファイルに複数の OIID を指定すると、バインドしているパブリック ACL の一覧を複数のオブジェクト分出力したり、またはバインドされているオブジェクト一覧を複数のパブリック ACL 分一括して出力したりできます。

一覧として出力できる情報は、オブジェクト種別および OIID です。

操作クラス

CdbrIndependentPersistence クラス
 CdbrDocument クラス
 CdbrVersionableDocument クラス
 CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス
 CdbrPublicACL クラス

形式

```
dbrlsACL [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           { OIID | OIID ファイル }
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

操作対象オブジェクトがバージョン付き文書またはバージョン付き構成管理コンテナの場合、バインドしているパブリック ACL の一覧を取得するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、オブジェクト自身にバインドしているパブリック ACL 一覧を取得します。

操作対象オブジェクトがバージョン付き文書およびバージョン付き構成管理コンテナ以外の場合にこのオプションを指定するとエラーになります。

バージョン識別子

バインドしているパブリック ACL 一覧を取得するバージョン識別子を指定します。

バージョン識別子ファイル

バインドしているパブリック ACL 一覧を取得するバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

OIID

バインドしているパブリック ACL 一覧を取得するオブジェクトの OIID、またはバインドされているオブジェクト一覧を取得するパブリック ACL の OIID を指定します。

OIID ファイル

バインドしているパブリック ACL 一覧を取得するオブジェクトの OIID、またはバインドされているオブジェクト一覧を取得するパブリック ACL の OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、`-v` オプションを指定している場合、記述できる OIID の数は一つだけです。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

実行例

実行例を次に示します。なお、この例は Windows の場合です。UNIX の場合は、「type OIIDs」を「cat OIIDs」に読み替えてください。

```
$ type OIIDs
;; CdbIndependentPersistence
dma:///07a17522...00000040
;; CdbPublicACL
dma:///07a17522...00000041

$ dbri sacl OIIDs
# dma:///07a17522...00000040
ACL,dma:///07a17522...00000041

# dma:///07a17522...00000041
IP,dma:///07a17522...00000040
```

注意事項

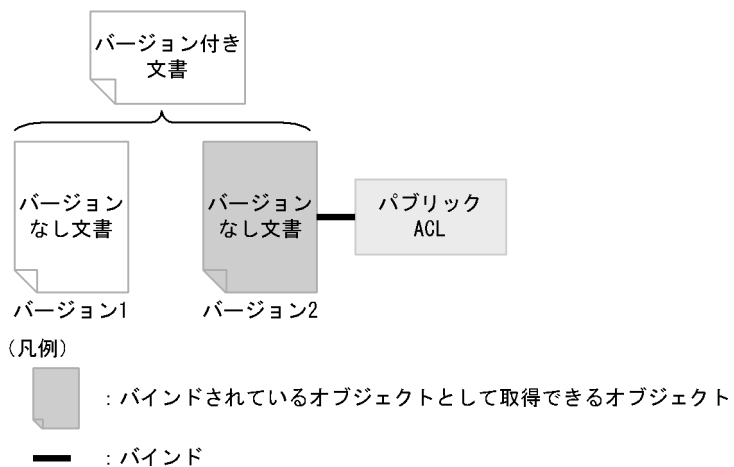
`-v` オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

`-v` オプション指定時には、複数オブジェクトにバインドされているパブリック ACL の一覧、および複数のパブリック ACL にバインドしているオブジェクト一覧を一括して取得できません。バージョンを指定してバインドされているパブリック ACL 一覧を取得できるオブジェクトは一つだけです。OIID ファイルに複数の OIID を指定した場合はエラーになります。

バインドしているオブジェクトの一覧を取得する場合、バージョン付きオブジェクトの 1 バージョンはバージョンなしオブジェクトとして表示されます。

例えば、図 5-2 のようにバージョン付き文書の 1 バージョンにパブリック ACL がバインドしている場合、このコマンドを実行すると、バージョンなし文書バージョン 2 の情報が出力されます。

図 5-2 バインドしているパブリック ACL から一覧として取得できる文書



dbrlschild (コンテナに包含されるオブジェクト一覧の取得)

機能

コンテナに包含されるオブジェクトの情報を取得します。取得した情報は、標準出力に出力されます。OID ファイルに複数の OID を指定すると、複数のコンテナに包含されるオブジェクトの一覧を一括して出力できます。

出力される情報はコンテインメントの種類、リンク識別子、オブジェクト種別および OID です。構成管理型のコンテインメントが設定されている場合は、構成管理モードおよびバージョントレースしているオブジェクトの OID も出力できます。各情報はコンマ(,)で区切って出力されます。

出力される情報について、次に説明します。

コンテインメントの種類

次のどれかが出力されます。

- DCR
直接型のコンテインメントが設定されている場合
- RCR
参照型のコンテインメントが設定されている場合
- VCR
構成管理型のコンテインメントが設定されている場合

リンク識別子

コンテインメントを識別するためのリンク識別子が出力されます。

オブジェクト種別

オブジェクトの種別を表す文字列が出力されます。「1.3.2 クラスライブラリとの関係」を参照してください。

OID

オブジェクトの OID が出力されます。

構成管理モード

次のどちらかが出力されます。

- FIX
FIX モードが設定されている場合
- FLOATING
FLOATING モードが設定されている場合

バージョントレースしているオブジェクトの OID

構成管理型モードによって追跡しているオブジェクト (DocVersion オブジェクトまたは ContainerVersion オブジェクト) の OID が出力されます。

操作クラス

- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

形式

```
dbrlschild [-l {DCR | RCR | VCR | ALL}]  
           [-v {バージョン識別子 | バージョン識別子ファイル}]  
           {OIID | OIID ファイル}
```

オプション

-l {DCR | RCR | VCR | ALL}

設定されているコンテナの種別を指定します。このオプションを省略した場合は ALL が仮定されます。

DCR

直接型のコンテナに関連づけられているオブジェクトの情報を出力します。

RCR

参照型のコンテナに関連づけられているオブジェクトの情報を出力します。

VCR

構成管理型のコンテナに関連づけられているオブジェクトの情報を出力します。

ALL

コンテナの種別に関係なく、含まれているオブジェクトの情報を出力します。

コマンド引数

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き構成管理コンテナの場合、含まれるオブジェクトの一覧を取得するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンのコンテナに含まれるオブジェクトの一覧を取得します。

バージョンなしコンテナおよびバージョンなし構成管理コンテナに含まれるオブジェクトの一覧を取得するときにこのオプションを指定するとエラーになります。

バージョン識別子

含まれるオブジェクトの一覧を取得するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

含まれるオブジェクトの一覧を取得するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

OIID

含まれるオブジェクトの一覧を取得するコンテナの OIID を指定します。

OIID ファイル

含まれるオブジェクトの一覧を取得するコンテナの OIID を記述した ID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、-v オプションを指定している場合、記述できる OIID の数は一つだけです。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

実行例

実行例を次に示します。なお、実行環境制御ファイルの Output セクションの Format エントリの指定、および指定したオプションによって、出力形式が異なります。また、例で改行されている場合も、実際には 1 オブジェクトに対応する情報は 1 行で出力されます。

(Format エントリが「VR0112」の場合)

```
$ dbrlschild dma:///07a17522...00000020
# dma:///07a17522...00000020
DCR,RC,dma:///07a17522...00000021
RCR,VD,dma:///07a17522...00000022
VCR,CRC,dma:///07a17522...00000023
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

(Format エントリが「VR0200」の場合)

```
$ dbrlschild dma:///07a17522...00000020
# dma:///07a17522...00000020
DCR,ac1:///LD...0021,RC,dma:///07a17522...00000021
RCR,ac1:///LR...0022,VD,dma:///07a17522...00000022
VCR,ac1:///LV...0023,CRC,dma:///07a17522...00000023
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

-l VCR オプションを指定した場合には、構成管理モードおよびバージョントレース OIID も出力されません。

```
$ dbrlschild -l VCR dma:///07a17522...00000020
# dma:///07a17522...00000020
VCR,FLOATING,ac1:///LV...0023,CRC,dma:///07a1...0023,dma:///07a1...0024
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

実行環境制御ファイルで Output セクションの Format エントリに「VR0112」を指定した場合、重複は排除されません。

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

-v オプション指定時には、複数のコンテナに包含されるオブジェクトの一覧は一括して取得できません。指定したバージョンの包含されるオブジェクトの一覧が取得できるコンテナは一つだけです。OIID ファイルに複数の OIID を指定した場合はエラーになります。

バージョンなしコンテナに対して -l VCR オプションを指定した場合、何もしないで正常終了します。

dbrlscompo (マルチファイル文書のコンテンツ情報一覧の取得)

機能

マルチファイル文書のコンテンツ情報を取得します。取得した情報は、標準出力に出力されます。OIID ファイルに複数の OIID を指定すると、マルチファイル文書のコンテンツ情報一覧を一括して取得できません。

出力される情報は、コンテンツ情報の有効モード、コンポーネント種別、ファイル名、およびコンテンツサイズです。なお、1 行目にコンテンツ情報の有効モード、2 行目以降にコンポーネント種別、ファイル名およびコンテンツサイズが出力されます。各情報はコンマ(,)で区切って出力されます。出力される情報を次に示します。

コンテンツ情報の有効モード

コンテンツ情報の有効モードが出力されます。出力される情報は、([SYSTEM]) 固定です。

コンポーネント種別

ファイルを識別する文字列が出力されます。

ファイル名

ファイル名が出力されます。

コンテンツサイズ

ファイルのサイズが出力されます。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

```
dbrlscompo [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           [ -M レンディションタイプ ]
           { OIID | OIID ファイル }
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、コンテンツ情報を取得するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンのコンテンツ情報を取得します。

なお、バージョンなし文書のコンテンツ情報を取得するときに、このオプションを指定するとエラーになります。

バージョン識別子

コンテンツ情報を取得するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

コンテンツ情報を取得するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。

ん。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-M レンディションタイプ

マルチファイル文書のコンテンツ情報を取得する場合に、コンテンツ情報を取得するレンディションのタイプを指定します。このオプションを省略した場合は、マスタレンディションを仮定します。

コマンド引数

OIID

コンテンツ情報を取得するマルチファイル文書の OIID を指定します。

OIID ファイル

コンテンツ情報を取得するマルチファイル文書の OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。ただし、`-v` オプションを指定している場合、記述できる OIID の数は一つだけです。

OIID ファイルのファイル名は、オブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「`dma://`」で始まるパスは指定できません。

実行例

実行例を次に示します。

```
$ dbrlscompo dma:///07a17522...00000036
# dma:///07a17522...00000036
[SYSTEM]
body,foo,128
fig-1,bar,64
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

`-v` オプションのオプションの引数に「`DBRVTVERSION`」を指定するとエラーになります。

`-v` オプション指定時は、複数のマルチファイル文書のコンテンツ情報一覧を一括して取得できません。

`-v` オプションに指定したバージョン識別子のマルチファイル文書のコンテンツ情報一覧を取得できる文書は一つだけです。OIID ファイルに複数の OIID を指定した場合はエラーになります。

dbrlsparent (オブジェクトを包含するコンテナ一覧の取得)

機能

対象オブジェクトを包含しているコンテナの情報を取得します。取得した情報は、標準出力に出力されます。OIDD ファイルに複数の OIDD を指定すると、複数のオブジェクトを包含しているコンテナの一覧を一括して取得できます。

出力される情報はコンテインメントの種別、リンク識別子、オブジェクト種別および OIDD です。各情報はコンマ(,)で区切って出力されます。出力される情報について、次に説明します。

コンテインメントの種別

次のどれかが出力されます。

- DCR
直接型のコンテインメントが設定されている場合
- RCR
参照型のコンテインメントが設定されている場合
- VCR
構成管理型のコンテインメントが設定されている場合

リンク識別子

コンテインメントを識別するためのリンク識別子が出力されます。

オブジェクト種別

オブジェクトの種別を表す文字列が出力されます。「1.3.2 クラスライブラリとの関係」を参照してください。

OIDD

オブジェクトの OIDD が出力されます。

操作クラス

- CdbrDocument クラス
- CdbrVersionableDocument クラス
- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

形式

```
dbrlsparent [-i {DCR | RCR | VCR | ALL}] {OIDD | OIDD ファイル}
```

オプション

`-i {DCR | RCR | VCR | ALL}`

出力するコンテナとの間に設定されているコンテインメントの種別を指定します。このオプションを省略した場合は ALL が仮定されます。

DCR

直接型のコンテインメントが設定されているコンテナを出力します。

RCR

参照型のコンテインメントが設定されているコンテナを出力します。

VCR

構成管理型のコンテインメントが設定されているコンテナを出力します。

ALL

バージョン付き文書またはバージョン付き構成管理コンテナの場合、直接型のコンテインメント、参照型のコンテインメントおよび構成管理型のコンテインメントが設定されているコンテナの情報を出力します。

バージョン付き文書およびバージョン付き構成管理コンテナ以外の場合、直接型のコンテインメントおよび参照型のコンテインメントが設定されている情報を出力します。

コマンド引数

OIID

包含するコンテナの一覧を取得するオブジェクトの OIID を指定します。

OIID ファイル

包含するコンテナの一覧を取得するオブジェクトの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルには、複数の OIID が指定できます。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

実行例

実行例を次に示します。なお、実行環境制御ファイルでの Output セクションの Format エントリの指定によって、出力形式が異なります。

```
(Format エントリが「VR0112」の場合)
$ dbrlsparent dma:///07a17522...00000030
# dma:///07a17522...00000030
DCR,RC,dma:///07a17522...00000031
RCR,VTC,dma:///07a17522...00000032
VCR,CRC,dma:///07a17522...00000033
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

```
(Format エントリが「VR0200」の場合)
$ dbrlsparent dma:///07a17522...00000030
# dma:///07a17522...00000030
RCR,ac1:///LD...0031,RC,dma:///07a17522...00000031
DCR,ac1:///LR...0032,VTC,dma:///07a17522...00000032
VCR,ac1:///LV...0033,CRC,dma:///07a17522...00000033
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

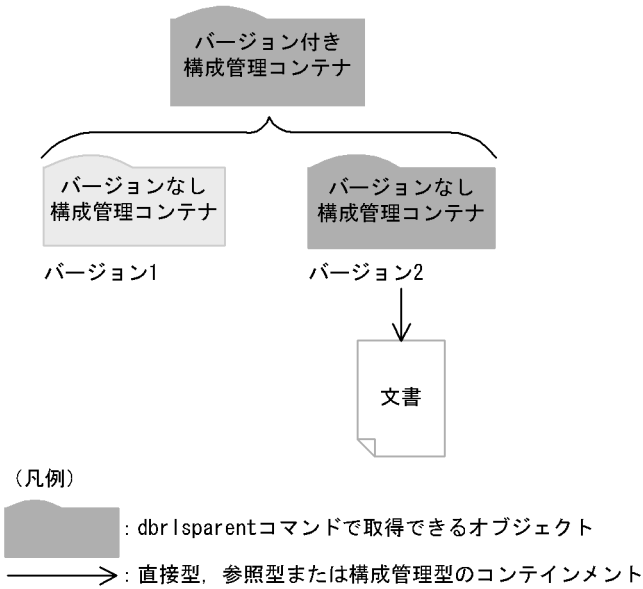
実行環境制御ファイルの Output セクションの Format エントリに「VR0112」を指定している場合、重複は排除されません。

バージョン付き構成管理コンテナの 1 バージョンは、バージョンなし構成管理コンテナとしても出力されます。

例えば、図 5-3 のようにバージョン付き構成管理コンテナの 1 バージョンが直接型、参照型または構成管理型のコンテインメントで文書を包含している場合、このコマンドを実行するとバージョン付きコン

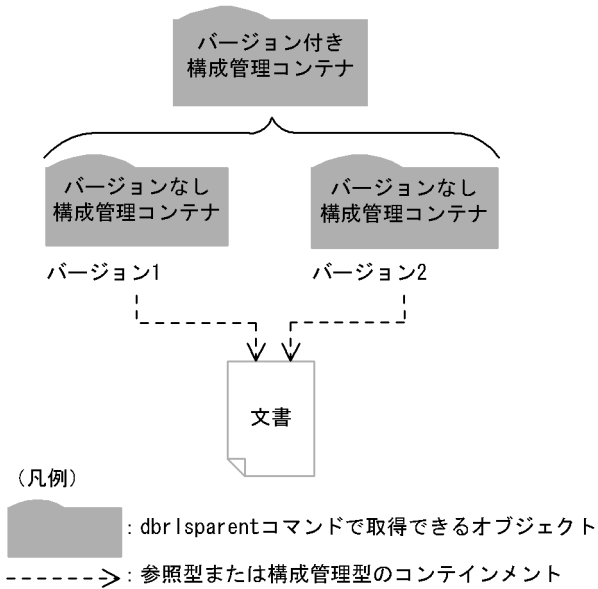
テナとバージョンなしコンテナバージョン 2 の情報が出力されます。

図 5-3 1バージョンが直接型、参照型または構成管理型のコンテインメントで文書を包含している場合に文書から取得できるコンテナ



また、図 5-4 のようにバージョン付き構成管理コンテナの複数のバージョンが参照型または構成管理型のコンテインメントで同じ文書を包含している場合、このコマンドを実行するとバージョン付きコンテナとバージョンなしコンテナバージョン 1 およびバージョン 2 の情報が出力されます。

図 5-4 複数のバージョンが参照型または構成管理型のコンテインメントで同じ文書を包含している場合に文書から取得できるコンテナ



構成管理型のコンテインメントを設定できないオブジェクトの場合に -1 VCR オプションを指定した場合、何もしないで正常終了します。

dbrlsversion (バージョン一覧の取得)

機能

オブジェクトに対してすでに確定しているバージョンの情報を取得します。取得した情報は、標準出力に出力されます。OIDD ファイルに複数の OIDD を指定すると、複数のオブジェクトのバージョン一覧が一括して出力されます。

出力される情報はバージョン識別子です。

操作クラス

CdbrVersionableDocument クラス

CdbrConfiguratedReferentialContainer クラス

形式

```
dbrlsversion [-o {ASC | DESC | NONE}] {OIDD | OIDD ファイル}
```

オプション

-o {ASC | DESC | NONE}

出力する順序を指定します。このオプションを省略した場合は NONE が仮定されます。

ASC

古いバージョンから出力します。

DESC

新しいバージョンから出力します。

NONE

出力する順序を指定しません。

コマンド引数

OIDD

バージョン一覧を取得するオブジェクトの OIDD を指定します。

OIDD ファイル

バージョン一覧を取得するオブジェクトの OIDD を記述した OIDD ファイルのファイルパスを指定します。OIDD ファイルには、複数の OIDD が指定できます。

OIDD ファイルのファイル名はオブジェクトの OIDD 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

実行例

実行例を次に示します。

```
$ dbrlsversion dma:///07a17522...00000034
# dma:///07a17522...00000034
acl:///VD...0035
acl:///VD...0036
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

5. コマンドリファレンス

注意事項

なし

dbrlsversionable (バージョン管理情報一覧の取得)

機能

管理オブジェクトの情報を取得します。取得した情報は、標準出力に出力されます。OID ファイルに複数の OID を指定すると、複数のオブジェクトのバージョン管理情報一覧を一括して取得できます。

出力される情報はオブジェクト種別、OID およびバージョン識別子です。各情報はコンマ (,) で区切って出力されます。出力される情報について、次に説明します。

オブジェクト種別

オブジェクトの種別を表す文字列が出力されます。「1.3.2 クラスライブラリとの関係」を参照してください。

OID

オブジェクトの OID が出力されます。

バージョン識別子

オブジェクトのバージョン識別子が出力されます。

操作クラス

CdbrDocument クラス

CdbrVersionTraceableContainer クラス

形式

dbrlsversionable {OID | OID ファイル}

オプション

なし

コマンド引数

OID

管理オブジェクトの一覧を取得するオブジェクトの OID を指定します。

OID ファイル

管理オブジェクトの一覧を取得するオブジェクトの OID を記述した OID ファイルのファイルパスを指定します。OID ファイルには、複数の OID が指定できます。

OID ファイルのファイル名はオブジェクトの OID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

実行例

実行例を次に示します。なお、実行環境制御ファイルでの Output セクションの Format エントリの指定によって、出力形式が異なります。

```
(Format エントリが「VR0112」の場合)
$ dbrlsversionable dma:///07a17522...0000037
```

5. コマンドリファレンス

```
# dma:///07a17522-a626-11d0-b11f-0020af27a837/...0000037
VD,dma:///07a17522...0000038
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

(Formatエントリが「VR0200」の場合)

```
$ dbrlsversionable dma:///07a17522...0000037
# dma:///07a17522-a626-11d0-b11f-0020af27a837/...0000037
VD,dma:///07a17522...0000038,ac1:///VD...00000037
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

なし

dbrnrdoc (文書のレンディション操作)

機能

文書のレンディションを操作します。次の操作ができます。

- レンディションの追加
 - マスタレンディションの変更
 - レンディションの削除
 - レンディション一覧の取得
- なお、レンディション一覧の取得で出力される情報は、レンディションタイプです。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

レンディションを追加する場合

```
dbrnrdoc -x ADD -M レンディションタイプ
        [ -v {バージョン識別子 | バージョン識別子ファイル} ]
        [ -t {USER | BATCH} ]
        [ -k {Content | FileLink
            [ -j コンテンツ登録先 FAM 名 ]
            [ -g コンテンツ登録先パス } ] ]
        { OIID | OIID ファイル } [ 登録ファイル URL ]
```

マスタレンディションを変更する場合

```
dbrnrdoc -x CHANGE -M レンディションタイプ
        [ -v {バージョン識別子 | バージョン識別子ファイル} ]
        { OIID | OIID ファイル }
```

レンディションを削除する場合

```
dbrnrdoc -x DELETE -M レンディションタイプ
        [ -v {バージョン識別子 | バージョン識別子ファイル} ]
        { OIID | OIID ファイル }
```

レンディション一覧を取得する場合

```
dbrnrdoc -x LIST
        [ -v {バージョン識別子 | バージョン識別子ファイル} ]
        [ -p プロパティ情報ファイル ]
```

{ OIID | OIID ファイル }

オプション

-x { ADD | CHANGE | DELETE | LIST }

操作の種類を指定します。このオプションを省略した場合は、LIST が仮定されます。

ADD

レンディションを追加します。

CHANGE

マスタレンディションを変更します。

DELETE

レンディションを削除します。

LIST

レンディション一覧を取得します。

-M レン디션タイプ

操作するレンディションのレン디션タイプを指定します。

-x オプションのオプションの引数に LIST 以外を指定した場合、このオプションは省略できません。

-x オプションのオプションの引数に LIST を指定した場合、このオプションを指定するとエラーになります。

-v { バージョン識別子 | バージョン識別子ファイル }

バージョン付き文書の場合、操作するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンが仮定されます。

バージョンなし文書の場合にこのオプションを指定するとエラーになります。

バージョン識別子

操作するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

操作するバージョンのバージョン識別子を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-t { USER | BATCH }

ファイル指定の種類を指定します。

-x オプションのオプションの引数に ADD 以外を指定した場合、このオプションを指定するとエラーになります。また、-k オプションに FileLink を指定した場合、このオプションに USER 以外を指定するとエラーになります。

-x オプションのオプションの引数に ADD を指定した場合、このオプションを省略すると USER が仮定されます。

USER

コマンド引数の登録ファイル URL に指定したファイルを、追加したレンディションに登録します。

BATCH

ファイルが未登録のレンディションを追加します。

追加したレンディションは、DocumentBroker Rendering Option によってマスタレンディションに登録されたファイルを変換する場合に、変換後のファイルを登録するために使用します。

-k { Content | FileLink [-j コンテンツ登録先 FAM 名] [-g コンテンツ登録先パス] }

レンディションを追加して登録する文書のコンテンツ種別を指定します。

-x ADD オプション以外を指定した場合、このオプションを指定するとエラーになります。また、このオプションに FileLink を指定した場合に、-t オプションに USER 以外を指定するとエラーになります。

なお、このオプションを省略した場合、Content が仮定されます。

Content

シングルファイル文書のレンディションを追加します。コンテンツは、データベースに登録します。

FileLink

File Link 文書のレンディションを追加します。コンテンツは、ファイルサーバに登録します。

-j コンテンツ登録先 FAM 名

File Link 文書のレンディションを追加する場合に、コンテンツの登録先となるファイルサーバの FAM 名を指定します。-k オプションに FileLink を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultFAMName エントリに指定された FAM 名のファイルサーバを使用します。

-g コンテンツ登録先パス

File Link 文書としてレンディションを追加する場合に、ファイルサーバ上の登録先パスを指定します。

クライアントのコンテンツを指定してレンディションを追加するとき、およびファイルサーバ上に登録されているコンテンツを指定してレンディションを追加するときの、オプションの引数の指定方法を次に示します。

- クライアントのコンテンツを指定してレンディションを追加するとき
ファイルサーバ上の登録先ディレクトリのパスを指定します。ディレクトリのパスの末尾にはデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述します。ファイルサーバに登録するコンテンツは、登録ファイル URL で指定します。

このオプションに指定した登録先ディレクトリが存在しない場合、ディレクトリが作成され、コンテンツが登録されます。ただし、ディレクトリの末尾にデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述しない場合、コンテンツ名称の自動生成を行わない(実行環境制御ファイルで、FileLink セクションの FileNamingMode エントリに Specify を指定している)環境で、このオプションに指定したディレクトリがファイルサーバ上に存在するとエラーになり、存在しないとこのオプションで指定した名称でコンテンツが登録されます。

UNIX の場合に、コンテンツの登録先ディレクトリが、「/dir01/data」の下位にある「area1」ディレクトリのとき、次のように指定します。

指定形式の種別	指定例
フルパス指定	/dir01/data/area1/
エイリアス指定	DATA/area1/ (「/dir01/data」のエイリアス名が「DATA」のとき)

また、Windows の場合に、コンテンツの登録先ディレクトリが、「c:¥dir01¥data」の下位にある「area1」ディレクトリのとき、次のように指定します。

指定形式の種別	指定例
フルパス指定	c:¥dir01¥data¥area1¥

指定形式の種類別	指定例
エイリアス指定	DATA¥area1¥ (「c:¥dir01¥data」のエイリアス名が「DATA」のとき)

- ファイルサーバ上に登録されているコンテンツを指定してレンディションを追加するとき
ファイルサーバ上のコンテンツのパスを指定します。登録ファイル URL には、DBRSVCONTENT
を指定します。

なお、-k オプションに FileLink を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultContentPath エントリの指定を使用します。ファイルサーバ上に登録されているコンテンツを指定してレンディションを追加する場合には、このオプションは省略できません。UNIX の場合に、レンディションの追加に使用するコンテンツが、「/dir01/data/area1」の下位にある「file.jpg」のとき、次のように指定します。

指定形式の種類別	指定例
フルパス指定	/dir01/data/area1/file.jpg
エイリアス指定	DATA/area1/file.jpg (「/dir01/data」のエイリアス名が「DATA」のとき)

Windows の場合に、レンディションの追加に使用するコンテンツが、「c:¥dir01¥data¥area1」の下位にある「file.jpg」のとき、次のように指定します。

指定形式の種類別	指定例
フルパス指定	c:¥dir01¥data¥area1¥file.jpg
エイリアス指定	DATA¥area1¥file.jpg (「/dir01/data」のエイリアス名が「DATA」のとき)

-p プロパティ情報ファイル

-x オプションに LIST を指定してレンディションの一覧を取得するときに、同時にプロパティ値を取得する場合、レンディションの一覧を取得するオブジェクトのクラス名、およびプロパティ値を取得するプロパティの名称を記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスの指定は、絶対パスまたは相対パスのどちらでも指定できます。

取得したプロパティ値は、レンディションタイプに続いてコンマ(,)で区切って出力されます。

指定できるプロパティを次に示します。次のプロパティ以外を指定した場合は、エラーとなります。

- dbrProp_RetrievalName

レンディションの dbrProp_RetrievalName プロパティの値を取得する場合に指定します。

- dbrProp_RenditionStatus

レンディションの dbrProp_RenditionStatus プロパティの値を取得する場合に指定します。

dbrProp_RenditionStatus プロパティの値は、マスタレンディションとサブレンディションの状態を表します。10 進数で表示されます。

dbrProp_RenditionStatus プロパティに設定される値とレンディションの状態の関係を次の表に示します。なお、dbrProp_RenditionStatus プロパティには、16 進数の値が設定されます。また、指定したレンディションがマスタレンディションの場合は、DBR_RENDSTATUS_MASTERD (00000000: 4 バイトの 16 進数) が設定されます。

表 5-4 dbrProp_RenditionStatus プロパティの値とレンディションの状態の関係（状態フラグ：下位 2 バイト）

dbrProp_RenditionStatus プロパティの値	対応する値（16 進数）	レンディションの状態
DBR_RENDSTATUS_NO_SUBREND	0001	指定したサブレンディションに対応するマスタレンディションのコンテンツは登録済みですが、サブレンディションのコンテンツが未登録です。
DBR_RENDSTATUS_SUBREND_EXIST	0002	指定したサブレンディションに対応するマスタレンディションおよびサブレンディションのコンテンツが登録されています。
DBR_RENDSTATUS_MASTERREND_UPDATE	0004	指定したサブレンディションに対応するマスタレンディションが更新されていますが、サブレンディションが更新されていないため、マスタレンディションとサブレンディションの内容が不一致です。

表 5-5 dbrProp_RenditionStatus プロパティの値とレンディションの状態の関係（変換フラグ：上位 2 バイト）

dbrProp_RenditionStatus プロパティの値	対応する値（16 進数）	レンディションの状態
DBR_RENDSTATUS_CONVERT_NOTREQUIRED	0000	レンディション変換が不要な状態です。この状態の場合、DocumentBroker Rendering Option によるレンディション変換の対象になりません。
DBR_RENDSTATUS_CONVERT_REQUIRED	0001	レンディション変換が必要な状態です。この状態の場合、DocumentBroker Rendering Option によるレンディション変換の対象になります。
DBR_RENDSTATUS_CONVERT_ERROR	0002	DocumentBroker Rendering Option によるレンディション変換時にエラーが発生した状態です。この状態になると、DocumentBroker Rendering Option によるレンディション変換はできなくなります。

- dbrProp_ContentType

レンディションの dbrProp_ContentType プロパティの値（コンテンツの登録先）を取得する場合に指定します。

dbrProp_ContentType プロパティは、各レンディションに対応したコンテンツ種別を示します。コンテンツ種別から、レンディションのコンテンツがマルチファイル文書か、シングルファイル文書か、File Link 文書か、またはリファレンスファイル文書かどうか知ることができます。このプロパティは、Integer32 型の 4 バイトの値として表示されます。

dbrProp_ContentType プロパティの値とコンテンツ種別の関係を次の表に示します。

表 5-6 dbrProp_ContentType プロパティの値とコンテンツ種別の関係

dbrProp_ContentType プロパティの値	対応する値	コンテンツ種別
DBR_CONTENTTYPE_CONTENT	0	ContentTransfer オブジェクト (シングルファイル文書であるコンテンツ) です。
DBR_CONTENTTYPE_FILELINK	1	ContentFileLink オブジェクト (File Link 文書であるコンテンツ) です。
DBR_CONTENTTYPE_MULTIFILE	2	ContentTransfers オブジェクト (マルチファイル文書であるコンテンツ) です。
DBR_CONTENTTYPE_REFERENCE	3	ContentReference オブジェクト (リファレンスファイル文書であるコンテンツ) です。
DBR_CONTENTTYPE_OTHER	-1	上記以外のコンテンツを持つオブジェクトです。

- dbrProp_ReferenceType

リファレンスファイル文書のレンディションの ReferenceType プロパティの値を取得する場合に指定します。

ReferenceType プロパティの値は、コンテンツのリファレンス種別を表します。ReferenceType プロパティの値とリファレンス種別の関係を次の表に示します。

表 5-7 dbrProp_ReferenceType プロパティの値とリファレンス種別の関係

dbrProp_ReferenceType プロパティの値	対応する値	リファレンス種別
DBR_REFERENCETYPE_NONE	0	コンテンツはありません。
DBR_REFERENCETYPE_USER_RELATIVE_CONTENT	1001	ユーザ管理領域に格納したコンテンツ (相対パス) です。

- dbrProp_ContentLocation

リファレンスファイル文書のレンディションの ContentLocation (コンテンツロケーション) を取得する場合に指定します。

コマンド引数

OID

操作する文書の OID を指定します。

OID ファイル

操作する文書の OID を記述した OID ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OID ファイルに記述できる OID の数は一つだけです。

登録ファイル URL

シングルファイル文書の場合、追加するレンディションに登録するファイルのファイルパスを URL 形式で指定します。

File Link 文書の場合、クライアントのコンテンツを指定してレンディションを追加するときは、ファイルサーバに登録するコンテンツのパスを URL 形式で指定します。また、ファイルサーバ上に登録

されているコンテンツを指定してレンディションを追加するときは、DBRSVCONTENT を指定して、
 -g オプションに利用するファイルサーバ上のコンテンツのパスを指定します。なお、ファイルサーバ上に登録されているコンテンツを使用するとき、-g オプションを省略するとエラーになります。
 例えば、UNIX の場合、「/tmp/add.doc」を登録するときには、「file:///tmp/add.doc」と指定します。
 Windows の場合、「c:¥temp¥add.doc」を登録するときには、「file:///c:¥temp¥add.doc」と指定します。
 なお、-x ADD オプション以外を指定した場合、このコマンド引数を指定するとエラーになります。

実行例

実行例を次に示します。なお、この例は Windows の場合です。UNIX の場合は、「file:///c:¥temp¥add.txt」を「file:///tmp/add.txt」に、「type PropFile」を「cat PropFile」に読み替えてください。

```
$ dbrnrdrdoc -x LIST dma:///07a17522...00000051
# dma:///07a17522...00000051
MIME::application/ms-word
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ dbrnrdrdoc -x ADD -M MIME::text/plain dma:///07a17522...00000051 file:///
c:¥temp¥add.txt
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ dbrnrdrdoc -x LIST dma:///07a17522...00000051
# dma:///07a17522...00000051
MIME::application/ms-word
MIME::text/plain
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ dbrnrdrdoc -x CHANGE -M MIME::text/plain dma:///07a17522...00000051
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ dbrnrdrdoc -x DELETE -M MIME::application/ms-word dma:///07a17522...00000051
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ dbrnrdrdoc -x LIST dma:///07a17522...00000051
# dma:///07a17522...00000051
MIME::text/plain
KMBR32002-I 1件のオブジェクトに対する操作を行いました。

$ type PropFile
[usrClass_Document]
dbrProp_ContentType
dbrProp_RetrievalName
dbrProp_RenditionStatus

$ dbrnrdrdoc -x LIST -p PropFile dma:///07a17522...00000051
# dma:///07a17522...00000051
MIME::application/ms-word,0,資料1.doc,0
MIME::text/plain,0,資料1.txt,2
KMBR32002-I 1件のオブジェクトに対する操作を行いました。
```

注意事項

- v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。
- x オプションのオプションの引数に ADD を指定した場合、すでに存在するレンディションタイプを
 -M オプションで指定すると、実行メソッドでエラーになります。
- x オプションのオプションの引数に ADD を指定して、-t オプションを省略または -t オプションのオブ

ションの引数に USER を指定する場合は、必ず登録ファイル URL を指定してください。登録ファイル URL を省略すると、実行メソッドでエラーになります。

-x オプションのオプションの引数に ADD を指定して、-t オプションのオプションの引数に BATCH を指定する場合は、登録ファイル URL を指定しないでください。登録ファイル URL を指定すると、実行メソッドでエラーになります。

-x オプションのオプションの引数に CHANGE を指定した場合、マスタレンディションのレンディションタイプまたは存在しないレンディションタイプを -M オプションで指定すると、実行メソッドでエラーになります。

-x オプションのオプションの引数に DELETE を指定した場合、マスタレンディションのレンディションタイプまたは存在しないレンディションタイプを -M オプションで指定すると、実行メソッドでエラーになります。

File Link 文書で、コンテンツの登録先を、ファイルサーバとする場合、-k オプションで File Link を指定します。この場合、-k オプションは省略できません。

File Link 文書の、ファイルサーバ上のコンテンツの削除モードは、クライアントのコンテンツを指定してレンディションを追加した場合と、ファイルサーバ上に登録されているコンテンツを指定してレンディションを追加した場合で異なります。それぞれの場合のコンテンツの削除モードを次に示します。

- クライアントのコンテンツを指定してレンディションを追加した場合
クライアントのコンテンツを指定して追加したレンディションを削除すると、ファイルサーバ上のコンテンツも同期して削除されます。なお、コンテンツとしてディレクトリを登録したときは、ディレクトリごと削除されます。
- ファイルサーバ上に登録されているコンテンツを指定してレンディションを追加した場合
ファイルサーバ上に登録されているコンテンツを指定して追加したレンディションを削除しても、ファイルサーバ上のコンテンツは削除されません。なお、コンテンツにディレクトリを使用しているときでも、ディレクトリは削除されません。

コンテンツの削除モードを変更する場合は、実行環境制御ファイルの FileLink セクションの、ClientContentDeleteMode エントリ、または ServerContentDeleteMode エントリの設定を変更してください。

リファレンスファイル文書の場合、レンディションは追加できません。

dbrsetmode (構成管理モードの設定)

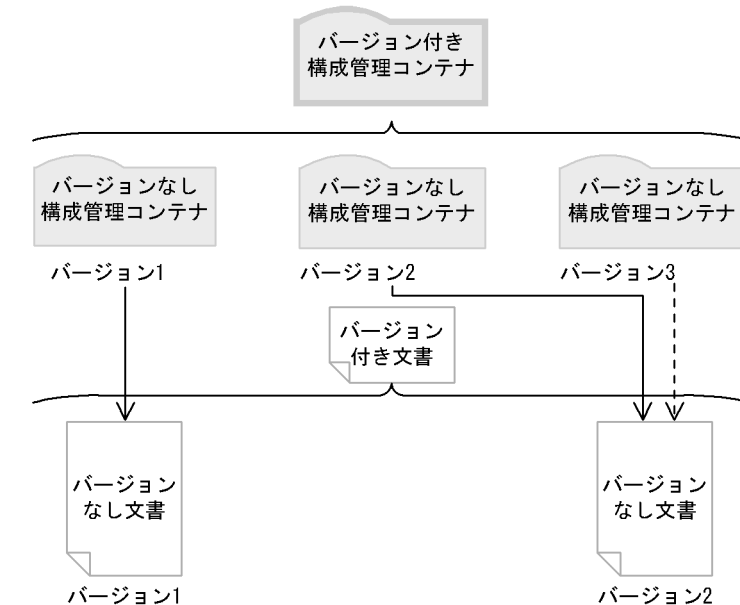
機能

構成管理モードを設定します。

構成管理モードは、バージョン付き構成管理コンテナによって包含されるオブジェクトのバージョンを構成管理するためのモードです。

バージョン付き構成管理コンテナによってバージョン付き文書を構成管理している例を、図 5-5 に示します。

図 5-5 バージョン付き構成管理コンテナによってバージョン付き文書を構成管理している例



(凡例)

——> : FIXモードによる管理

----> : FLOATINGモードによる管理

図 5-5 では、三つのバージョンを持つバージョン付き構成管理コンテナ (コンテナ) によって、二つのバージョンを持つバージョン付き文書 (文書) を、次のように構成管理しています。

- コンテナのバージョン 1 では、文書のバージョン 1 を FIX モードで管理しています。
- コンテナのバージョン 2 では、文書のバージョン 2 を FIX モードで管理しています。
- コンテナのバージョン 3 では、文書のバージョン 2 を FLOATING モードで管理しています。

構成管理モードの設定とは、コンテナの構成管理モードを、FIX モードまたは FLOATING モードに設定 (変更) することです。例えば、コンテナのバージョン 3 が文書を管理する構成管理モードとして FIX モードを設定すると、文書がバージョン 3 にバージョンアップした場合も、バージョンなし構成管理コンテナは文書のバージョン 2 を管理したままになります。これを FLOATING モードに変更すると、コンテナのバージョン 3 が管理する文書は、バージョン 3 の文書になります。

図 5-5 のコンテナのバージョン 3 に、FIX モードを設定した場合と FLOATING モードを設定した場合に管理する文書のバージョンを、図 5-6 に示します。

図 5-6 FIX モードを設定した場合と FLOATING モードを設定した場合に管理する文書のバージョン

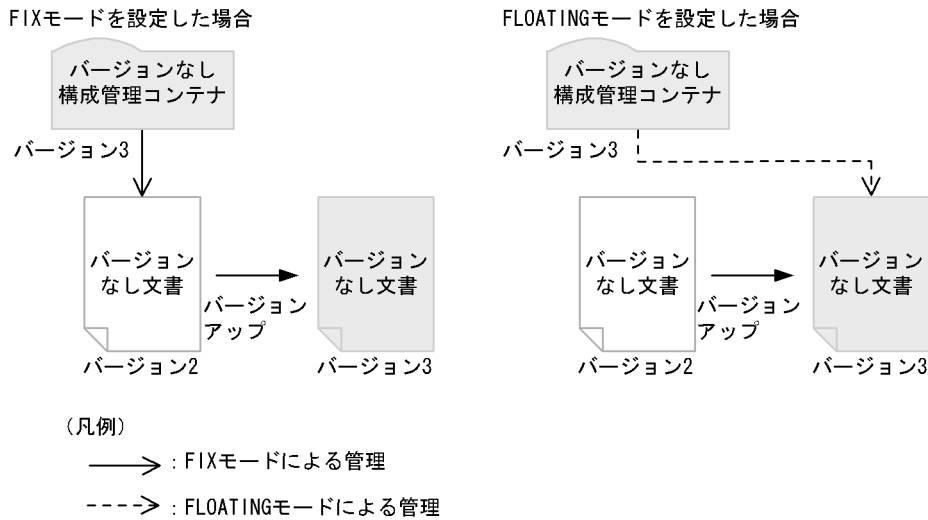
**!** 注意事項

図 5-5 の場合、コンテナのバージョン 3 が FLOATING モードで管理している文書のバージョン 2 の内容を更新すると、コンテナのバージョン 2 が FIX モードで管理している文書の内容も更新されます。

操作クラス

CdbrVersionTraceableContainer クラス

CdbrConfiguratedReferentialContainer クラス

形式

構成管理コンテナでリンク識別子を指定しない場合

```
dbrsetmode -m { FLOATING | FIX }
[ -v { バージョン識別子 | バージョン識別子ファイル } ]
{ 包含するオブジェクトの OIID
| 包含するオブジェクトの OIID ファイル }
{ 包含されるオブジェクトの OIID
| 包含されるオブジェクトの OIID ファイル }
```

構成管理コンテナでリンク識別子を指定する場合

```
dbrsetmode -m { FLOATING | FIX }
[ -w { DBRVTVERSION |
固定バージョン識別子 | 固定バージョン識別子ファイル } ] }
-L { リンク識別子 | リンク識別子ファイル }
{ 包含するオブジェクトの OIID }
```

| 包含するオブジェクトの OIID ファイル }

オプション

-m { FLOATING | FIX }

設定する構成管理モード種別を指定します。このオプションは省略できません。

FLOATING

FLOATING モードを設定します。

FIX

FIX モードを設定します。

-v { バージョン識別子 | バージョン識別子ファイル }

バージョン付き構成管理コンテナの場合、構成管理モードを設定するバージョンのバージョン識別子を指定します。このオプションを省略した場合は最新バージョンの構成管理モードが設定されます。バージョンなし構成管理コンテナの場合にこのオプションを指定するとエラーになります。

バージョン識別子

構成管理モードを設定するバージョン付き構成管理コンテナのバージョン識別子を指定します。

バージョン識別子ファイル

構成管理モードを設定するバージョン付き構成管理コンテナのバージョン識別子を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-w { 固定バージョン識別子 | 固定バージョン識別子ファイル }

FIX モードを設定する場合に固定するバージョン識別子を指定します。このオプションは、-L オプションおよび -m FIX オプションを指定した場合だけ指定できます。このオプションを省略した場合は最新バージョンが仮定されます。

DBRVTVERSION

現在構成管理されているバージョンで固定します。

固定バージョン識別子

FIX モードで固定するバージョン識別子を指定します。

固定バージョン識別子ファイル

FIX モードで固定するバージョンのバージョン識別子を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「acl://」で始まるパスは指定できません。なお、このバージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-L { リンク識別子 | リンク識別子ファイル }

構成管理モードを設定するリンク識別子を指定します。

リンク識別子

構成管理モードを設定するリンク識別子を指定します。

リンク識別子ファイル

構成管理モードを設定するリンク識別子を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「acl://」で始まるパスは指定できません。なお、リンク識別子ファイルに記述できるリンク識別子の数は一つだけです。

コマンド引数

包含するオブジェクトの OIID

構成管理モードを変更する包含するオブジェクトの OIID を指定します。

包含するオブジェクトの OIID ファイル

構成管理モードを変更する包含するオブジェクトの OIID を記述したファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ただし、「dma://」で始まるパスは指定できません。なお、包含するオブジェクトの OIID ファイルに記述できる OIID の数は一つだけです。

包含されるオブジェクトの OIID

構成管理モードを変更する包含されるオブジェクトの OIID を指定します。

包含されるオブジェクトの OIID ファイル

構成管理モードを変更する包含されるオブジェクトの OIID を記述したファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスの指定は、絶対パスまたは相対パスのどちらでもかまいません。ただし、「dma://」で始まるパスは指定できません。なお、包含されるオブジェクトの OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

バージョン付き構成管理コンテナで -v オプションと -L オプションを同時に指定することはできません。

指定したリンク識別子のコンテナが存在しない場合は、実行メソッドでエラーになります。

構成管理コンテナでリンク識別子を指定しないで FIX モードを設定する場合、二つのオブジェクト間に設定されているすべての構成管理型のコンテナに対して、FIX モードを設定して、現在構成管理しているバージョンで固定します。

すでに設定されている構成管理モードを指定した場合、構成管理モードを再設定します。

指定した二つのオブジェクト間に構成管理型のコンテナが設定されていない場合は、何もしないで正常終了します。

dbrsetprop (プロパティの設定)

機能

プロパティ情報ファイルに記述した内容に従ってオブジェクトのプロパティを設定します。OID ファイルに複数の OID を指定することで、複数のオブジェクトのプロパティを一括して設定できます。

操作クラス

CdbrIndependentPersistence クラス
 CdbrDocument クラス
 CdbrVersionableDocument クラス
 CdbrReferentialContainer クラス
 CdbrVersionTraceableContainer クラス
 CdbrConfiguratedReferentialContainer クラス
 CdbrPublicACL クラス

形式

```
dbrsetprop [ {-v {バージョン識別子 | バージョン識別子ファイル} |
             -L {リンク識別子 | リンク識別子ファイル}} ]
           {OID | OID ファイル} プロパティ情報ファイル
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}
 バージョン付き文書またはバージョン付き構成管理コンテナの場合、プロパティを設定するバージョンのバージョン識別子を指定します。このオプションおよび **-L** オプションを省略した場合は、すべてのバージョンで共有するプロパティが設定されます。
 バージョン付き文書およびバージョン付き構成管理コンテナ以外の場合に、このオプションを指定するとエラーになります。

バージョン識別子

プロパティを設定するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

プロパティを設定するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-L {リンク識別子 | リンク識別子ファイル}

コンテナの場合、プロパティを設定するコンテナメントのリンク識別子を指定します。このオプションおよび **-v** オプションを省略した場合は、すべてのバージョンで共有するプロパティが設定されます。

コンテナ以外のプロパティを設定するときこのオプションを指定するとエラーになります。

リンク識別子

プロパティを設定するコンテナメントのリンク識別子を指定します。

リンク識別子ファイル

プロパティを設定するコンテナのリンク識別子を記述したリンク識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「acl://」で始まるパスは指定できません。なお、リンク識別子ファイルに記述できるリンク識別子の数は一つだけです。

コマンド引数

OID

プロパティを設定するオブジェクトの OID を指定します。

OID ファイル

プロパティを設定するオブジェクトの OID を記述した OID ファイルのファイルパスを指定します。OID ファイルには、複数の OID が指定できます。ただし、`-v` オプションまたは `-L` オプションを指定している場合、記述できる OID の数は一つだけです。

OID ファイルのファイル名はオブジェクトの OID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。

プロパティ情報ファイル

設定するオブジェクトのクラス名および設定するプロパティを記述したプロパティ情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

プロパティ情報ファイルの記述例

独立データのプロパティを変更する場合の `edmClass_IndependentPersistence` 用プロパティ情報ファイルの記述例を示します。変更するプロパティだけ指定します。

```
;; usrClass_IndependentPersistence
[usrClass_IndependentPersistence]
usrProp_IsReadOnly=FALSE
usrProp_State=999
usrProp_Title='Independent2'
usrProp_Authors[4][2]= '
{
  {usrProp_Name='suzuki',  usrProp_Age=27},
  {usrProp_Name='yamada',  usrProp_Age=30},
  {usrProp_Name='sato',    usrProp_Age=32},
  {usrProp_Name='kimura',  usrProp_Age=25}
}
```

注意事項

OID ファイルに記述した OID 数とプロパティ情報ファイルに記述したセクション数が同じでない場合、エラーになります。オブジェクトの OID を指定した場合、プロパティ情報ファイルに記述するセクション数は一つでなければなりません。

`-v` オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

`-v` オプションまたは `-L` オプション指定時には、複数のオブジェクトに対してプロパティを一括して設定することはできません。指定したバージョンおよびコンテナのプロパティを設定できるオブジェクトは一つだけです。OID ファイルに複数の OID を指定した場合はエラーになります。

バージョン付き構成管理コンテナに対して、`-v` オプションと `-L` オプションを同時に指定することはできません。

プロパティエントリが一つもないセクションを記述したプロパティ情報ファイルを指定すると、実行メ

ソッドでエラーになります。

基本単位が `VariableArray` 型の `Object` 型のプロパティを設定する場合、一部の構成要素プロパティだけを設定することはできません。すべての配列および構成要素プロパティを設定してください。

クラスライブラリ固有のプロパティは、クラス定義情報ファイルに記述していなくても、セクション名に指定したクラスのプロパティとして設定できます。ただし、指定したオブジェクトの操作クラスに導入されていないクラスライブラリ固有のプロパティを指定した場合は、実行メソッドでエラーになります。

プロパティ情報ファイルに記述したクラス名はクラス定義情報ファイルにアクセスする時だけ参照します。プロパティの設定には影響しません。

dbrtoolsetup (実行環境のセットアップ)

機能

オブジェクト操作ツールの実行環境をセットアップします。

形式

dbrtoolsetup [-x {CREATE | UPDATE}] 実行環境ディレクトリ

オプション

-x {CREATE | UPDATE}

セットアップの処理種別を指定します。このオプションを省略した場合は、UPDATE が仮定されます。

CREATE

UNIX の場合は「実行環境ディレクトリ /etc」、Windows の場合は「実行環境ディレクトリ \etc」に、デフォルトの実行環境制御ファイルおよびクラス情報定義ファイルをコピーしてオブジェクト操作ツール実行環境を作成します。このオプションは最初にオブジェクト操作ツールを使用する場合に指定します。

UPDATE

オブジェクト操作ツールの実行環境を更新する場合に指定します。実行環境は、UNIX の場合は「実行環境ディレクトリ /etc」、Windows の場合は「実行環境ディレクトリ \etc」です。次の場合は必ずこのオプションを指定してコマンドを実行してください。

- 実行環境制御ファイルまたはクラス定義情報ファイルを変更した場合
- DocumentBroker Development Kit または DocumentBroker Runtime をバージョンアップした場合

コマンド引数

実行環境ディレクトリ

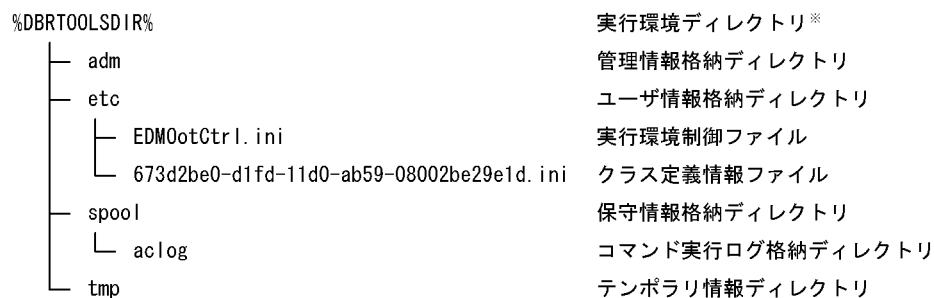
オブジェクト操作ツールの実行環境ディレクトリのディレクトリパスを指定します。ディレクトリパスは絶対パスで指定してください。

注意事項

実行環境ディレクトリに指定したディレクトリが存在しない場合、エラーとなります。コマンド実行前にディレクトリが存在することを確認してください。

-x CREATE オプションを指定してコマンドを実行すると、図 5-7 に示すディレクトリおよびファイルが作成されます。なお、図 5-7 は Windows の場合のディレクトリ構成です。UNIX の場合のディレクトリ構成については、図中の注を参照してください。

図 5-7 実行環境作成時のディレクトリ構成 (Windows の場合)



注※ UNIX の場合の実行環境ディレクトリは、「\$DBRTOOLS DIR」になります。

すでに `-x CREATE` オプションを指定したコマンドの実行によって実行環境が作成されている状態で、再度 `-x CREATE` オプションを指定して実行すると、コマンドは処理をスキップして正常終了します。

UNIX の場合、`-x UPDATE` オプションを指定してコマンドを実行できるのは、`-x CREATE` オプションを指定して実行環境を作成したユーザだけです。Windows の場合、`-x UPDATE` オプションを指定してコマンドを実行できるユーザは、実行環境に設定されているアクセス権に依存します。

実行環境制御ファイルまたはクラス定義情報ファイルを変更後、`-x UPDATE` オプションを指定してコマンドを実行しなかった場合、オブジェクト操作ツールの実行が遅くなったり、エラーとなったりすることがあります。

このコマンドの実行に対するコマンド実行ログは出力されません。

実行環境を使用しなくなった場合は、実行環境ディレクトリ下のディレクトリおよびファイルを削除してください。

dbrunbind (パブリック ACL のアンバインド)

機能

オブジェクトにバインドされているパブリック ACL をアンバインドします。

操作クラス

アンバインドするオブジェクト

CdbrIndependentPersistence クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

CdbrReferentialContainer クラス

CdbrVersionTraceableContainer クラス

CdbrConfiguratedReferentialContainer クラス

アンバインドされるオブジェクト

CdbrPublicACL クラス

形式

```
dbrunbind [ -v {バージョン識別子 | バージョン識別子ファイル} ]
           {OID | OID ファイル}
           {パブリック ACL OID | パブリック ACL OID ファイル}
```

オプション

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書またはバージョン付き構成管理コンテナの場合、アンバインドするバージョンのバージョン識別子を指定します。このオプションを省略した場合は、オブジェクト自身からアンバインドされます。

バージョン付き文書およびバージョン付き構成管理コンテナ以外の場合に、このオプションを指定するとエラーになります。

バージョン識別子

アンバインドするバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

アンバインドするバージョンのバージョン識別子を記述したファイルのファイルパスを指定します。ファイルパスの指定は、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列、および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

コマンド引数

OID

アンバインドするオブジェクトの OID を指定します。

OID ファイル

アンバインドするオブジェクトの OIID を記述した OIID ファイルのファイルパスを指定します。OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスの指定は、絶対パスまたは相対パスのどちらでもかまいません。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

パブリック ACL OIID

アンバインドされるパブリック ACL の OIID を指定します。

パブリック ACL OIID ファイル

アンバインドされるパブリック ACL の OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

バインドされていないパブリック ACL をアンバインドしようとすると、実行メソッドでエラーになります。

dbrunlink (オブジェクト間の関連づけの解除)

機能

コンテインメントによる二つのオブジェクト間の関連づけを解除します。

操作クラス

包含するオブジェクト

- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

包含されるオブジェクト

- CdbrDocument クラス
- CdbrVersionableDocument クラス
- CdbrReferentialContainer クラス
- CdbrVersionTraceableContainer クラス
- CdbrConfiguratedReferentialContainer クラス

形式

リンク識別子を指定しない場合

```
dbrunlink [ -l { DCR | RCR | VCR | ALL } ]
           [ -v { 包含するオブジェクトのバージョン識別子
                 | 包含するオブジェクトのバージョン識別子ファイル } ]
           { 包含するオブジェクトの OIID
             | 包含するオブジェクトの OIID ファイル }
           { 包含されるオブジェクトの OIID
             | 包含されるオブジェクトの OIID ファイル }
```

リンク識別子を指定する場合

```
dbrunlink -L { リンク識別子 | リンク識別子ファイル }
           { 包含するオブジェクトの OIID
             | 包含するオブジェクトの OIID ファイル }
```

オプション

-l { DCR | RCR | VCR | ALL }

解除するコンテインメントの種別を指定します。このオプションを省略した場合は ALL が仮定されます。

DCR

直接型のコンテナメントを解除します。

RCR

参照型のコンテナメントを解除します。

VCR

構成管理型のコンテナメントを解除します。

ALL

コンテナメントの種類に関係なく、すべての関連づけを解除します。

- v { 包含するオブジェクトのバージョン識別子 | 包含するオブジェクトのバージョン識別子ファイル }
包含するオブジェクトがバージョン付き構成管理コンテナの場合、コンテナメントを解除するバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンに設定されたコンテナメントが解除されます。
包含するオブジェクトがバージョンなしコンテナおよびバージョンなし構成管理コンテナの場合にこのオプションを指定するとエラーになります。

バージョン識別子

コンテナメントを解除するバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

コンテナメントを解除するバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

- L { リンク識別子 | リンク識別子ファイル }

関連づけを解除するコンテナメントのリンク識別子を指定します。

リンク識別子

解除するコンテナメントのリンク識別子を指定します。

リンク識別子ファイル

解除するコンテナメントのリンク識別子を記述したリンク識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「acl:/」で始まるパスは指定できません。なお、リンク識別子ファイルに記述できるリンク識別子の数は一つだけです。

コマンド引数

包含するオブジェクトの OIID

オブジェクトを包含しているコンテナの OIID を指定します。

包含するオブジェクトの OIID ファイル

オブジェクトを包含しているコンテナの OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

包含されるオブジェクトの OIID

コンテナに包含されているオブジェクト（関連づけを解除するオブジェクト）の OIID を指定します。

包含されるオブジェクトの OIID ファイル

コンテナに包含されているオブジェクト（関連づけを解除するオブジェクト）の OIID を記述した

OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

二つのオブジェクト間に複数の参照型コンテナメントが設定されている場合、`-l` オプションのオプションの引数に `RCR` または `ALL` を指定して実行すると、参照型コンテナメントによる関連づけがすべて解除されます。

二つのオブジェクト間に複数の構成管理型のコンテナメントが設定されている場合、`-l` オプションのオプションの引数に `VCR` または `ALL` を指定して実行すると、構成管理型コンテナメントによる関連づけがすべて解除されます。

バージョンなしコンテナに対して `-l` オプションのオプションの引数に `VCR` を指定した場合、何もしないで終了します。

`-v` オプションのオプションの引数に「`DBRVTVERSION`」を指定するとエラーになります。

バージョン付き構成管理コンテナに対して `-v` オプションと `-l` オプションは同時に指定できません。

二つのオブジェクト間にコンテナメントによる関連づけがない場合（包含関係がない場合）、何もしないで正常終了します。

指定したリンク識別子のコンテナメントが存在しない場合は、エラーになります。

dbrupcntr (コンテナのバージョンアップ)

機能

バージョン付き構成管理コンテナをバージョンアップします。

操作クラス

CdbrConfiguratedReferentialContainer クラス

形式

dbrupcntr { OIID | OIID ファイル }

オプション

なし

コマンド引数

OIID

バージョンアップするバージョン付き構成管理コンテナの OIID を指定します。

OIID ファイル

バージョンアップするバージョン付き構成管理コンテナの OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

注意事項

バージョンアップ前のバージョンに設定されていた直接型のコンテインメントは、バージョンアップ後のバージョンにはコピーされません。

バージョンアップ前のバージョンに設定されていた参照型のコンテインメントおよび構成管理型のコンテインメントは、バージョンアップ後のバージョンにコピーされます。

dbrupdoc (文書のアップロード)

機能

指定したファイルを文書にアップロードします。

操作クラス

CdbrDocument クラス

CdbrVersionableDocument クラス

形式

```
dbrupdoc [-I] [-v {バージョン識別子 | バージョン識別子ファイル}]
           [-M レンディションタイプ]
           [-N 変更レンディションタイプ]
           [-k {Content | MultiContent | FileLink
                [-j コンテント登録先 FAM 名]
                [-g コンテント登録先パス]
                | Reference [-a RELATIVE]
                [-b コンテント格納先ベースパス]
                [-h コンテント格納先パス]}]
           {OHD | OHD ファイル}
           {アップロードファイル URL | アップロード情報ファイル}
```

オプション

-I

全文検索インデクスを作成しない場合に指定します。シングルファイル文書をアップロードする場合、このオプションを省略して、かつアップロードする文書が次の条件を満たしているとき、アップロードファイル URL に指定したファイルから、全文検索インデクスが作成されます。

- 作成時に指定された dmaClass_DocVersion クラスまたはそのサブクラスが、全文検索機能付き文書クラスである。
- 設定されているレンディションタイプの先頭が「MIME::text/」である。

なお、マルチファイル文書または File Link 文書の場合は、このオプションを省略しても全文検索インデクスは作成されません。

-v {バージョン識別子 | バージョン識別子ファイル}

バージョン付き文書の場合、アップロードするバージョンのバージョン識別子を指定します。このオプションを省略した場合は、最新バージョンがチェックアウトされて、ファイルがアップロードされたあと、チェックインされます。

バージョンなし文書をアップロードするときにこのオプションを指定するとエラーになります。

バージョン識別子

アップロードするバージョンのバージョン識別子を指定します。

バージョン識別子ファイル

アップロードするバージョンのバージョン識別子を記述したバージョン識別子ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「DBRVTVERSION」という文字列および「acl://」で始まるパスは指定できません。なお、バージョン識別子ファイルに記述できるバージョン識別子の数は一つだけです。

-M レンディションタイプ

アップロードするレンディションタイプを指定します。このオプションを省略した場合は、マスタレンディションが仮定されます。

バージョン付き文書に対してこのオプションを指定する場合は、必ず `-v` オプションも指定してください。

-N 変更レンディションタイプ

アップロード時にレンディションタイプを変更する場合に、変更後のレンディションタイプを指定します。このオプションを省略した場合は、レンディションタイプは変更されません。なお、マルチファイル文書の場合、このオプションは指定できません。

`-k { Content | MultiContent | FileLink [-j コンテンツ登録先 FAM 名] [-g コンテンツ登録先パス] }`

`Reference [-a RELATIVE] [-b コンテンツ格納先ベースパス] [-h コンテンツ格納先パス] }`

アップロードする文書のコンテンツの種別を指定します。

このオプションを指定した場合、アップロードファイル URL、またはアップロード情報ファイルに指定されたコンテンツの登録先が、このオプションの指定と一致していないとエラーになります。また、アップロードしようとするレンディションのコンテンツの登録先と一致しないとエラーになります。

コンテンツの登録先は、`dbrrndrdoc` コマンドでレンディションの一覧を取得する (`-x` オプションに `LIST` を指定する) 場合に、`dbrProp_ContentType` プロパティの値を取得する (`-p` オプションで指定するプロパティ情報ファイルに「`dbrProp_ContentType`」を指定する) ことで確認できます。なお、このオプションを省略した場合は、アップロードファイル URL、またはアップロード情報ファイルの指定に従って文書をアップロードします。

Content

シングルファイル文書のコンテンツとして、データベースにアップロードします。アップロード URL に指定されたファイルに、コンテンツをアップロードします。

MultiContent

マルチファイル文書のコンテンツとして、データベースにアップロードします。アップロード情報ファイルの指定に従って、コンテンツをアップロードします。

FileLink

File Link 文書のコンテンツとして、ファイルサーバにアップロードします。

Reference

リファレンスファイル文書のコンテンツとして、ファイルシステム上の領域にアップロードします。

オプション省略時

アップロードファイル URL、またはアップロード情報ファイルの指定に従って文書のコンテンツをデータベースにアップロードします。

- アップロードファイル URL が指定されている場合
シングルファイル文書のコンテンツをデータベースにアップロードします。
- アップロード情報ファイルが指定されている場合
マルチファイル文書のコンテンツをデータベースにアップロードします。

`-j コンテンツ登録先 FAM 名`

File Link 文書の場合に、コンテンツの登録先となるファイルサーバの FAM 名を指定します。-k オプションに FileLink を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultFAMName エントリに指定された FAM 名のファイルサーバを使用します。

-g コンテンツ登録先パス

File Link 文書のコンテンツをアップロードする場合に、ファイルサーバ上の登録先パスを指定します。

クライアントのコンテンツを指定して文書をアップロードするとき、およびファイルサーバ上に登録されているコンテンツを指定して文書をアップロードするときの、オプションの引数の指定方法を次に示します。

- クライアントのコンテンツを指定して文書をアップロードするとき
ファイルサーバ上の登録先ディレクトリのパスを指定します。ディレクトリの末尾にはデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述します。ファイルサーバに登録するコンテンツは、アップロードファイル URL で指定します。
このオプションに指定した登録先ディレクトリが存在しない場合、ディレクトリが作成され、コンテンツが登録されます。ただし、ディレクトリの末尾にデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述しない場合、コンテンツ名称の自動生成を行わない(実行環境制御ファイルで、FileLink セクションの FileNamingMode エントリに Specify を指定している)環境で、このオプションに指定したディレクトリがファイルサーバ上に存在するとエラーになり、存在しないとディレクトリは作成しないで、このオプションで指定した名称でコンテンツが登録されます。UNIX の場合に、コンテンツの登録先ディレクトリが、「/dir01/data」の下位にある「area1」ディレクトリのとき、次のように指定します。

指定形式の種類	指定例
フルパス指定	/dir01/data/area1/
エイリアス指定	DATA/area1/ (「/dir01/data」のエイリアス名が「DATA」のとき)

また、Windows の場合に、コンテンツの登録先ディレクトリが、「c:¥dir01¥data」の下位にある「area1」ディレクトリのとき、次のように指定します。

指定形式の種類	指定例
フルパス指定	c:¥dir01¥data¥area1¥
エイリアス指定	DATA¥area1¥ (「c:¥dir01¥data」のエイリアス名が「DATA」のとき)

- ファイルサーバ上に登録されているコンテンツを指定して文書をアップロードするとき
ファイルサーバ上のコンテンツのパスを指定します。アップロードファイル URL には、DBRSVCONTENT を指定します。

なお、-k オプションに FileLink を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの FileLink セクションの、DefaultContentPath エントリの指定を使用します。ファイルサーバに登録されているコンテンツを指定する場合には、このオプションは省略できません。

UNIX の場合に、文書のアップロードに使用するコンテンツが、「/dir01/data/area1」の下位にある「file.jpg」のとき、次のように指定します。

指定形式の種別	指定例
フルパス指定	/dir01/data/area1/file.jpg
エイリアス指定	DATA/area1/file.jpg (「/dir01/data」のエイリアス名が「DATA」のとき)

Windows の場合に、文書のアップロードに使用するコンテンツが、「c:¥dir01¥data¥area1」の下位にある「file.jpg」のとき、次のように指定します。

指定形式の種別	指定例
フルパス指定	c:¥dir01¥data¥area1¥file.jpg
エイリアス指定	DATA¥area1¥file.jpg (「/dir01/data」のエイリアス名が「DATA」のとき)

-a RELATIVE

アップロードするリファレンスファイル文書のリファレンス種別を指定します。

-k オプションに Reference を指定していない場合に、このオプションを指定するとエラーになります。

このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultReferenceType エントリの指定を使用します。

リファレンス種別が NONE の、コンテンツが登録されていないリファレンスファイル文書をアップロードする場合も、このオプションまたは実行環境制御ファイルで、リファレンス種別を指定します。

RELATIVE

コンテンツの格納先をファイルシステム上の領域とし、コンテンツロケーションを相対パスで管理します。

-b コンテンツ格納先ベースパス

リファレンスファイル文書をアップロードする場合に、コンテンツ格納先ベースパスを指定します。アップロードするリファレンスファイル文書の、リファレンス種別が RELATIVE のとき、およびリファレンス種別が NONE のときの、オプション引数の指定内容を次に示します。

- リファレンス種別が RELATIVE のとき

オプション引数には、更新元のリファレンスファイル文書のコンテンツ格納先ベースパスを指定します。例えば、更新元となるリファレンスファイル文書のコンテンツ格納先ベースパスが「/drv01/dir01/userdir」の場合、このオプションに同じ値「/drv01/dir01/userdir」を指定します。

- リファレンス種別が NONE のとき

オプション引数には、更新するリファレンスファイル文書のコンテンツを格納するディレクトリの基点となるパスを指定します。例えば、アップロードファイル URL で指定したコンテンツを、「/drv01/dir01/userdir」ディレクトリの下位にある「user01」ディレクトリに格納する場合、「user01」ディレクトリの基点となるパスとして、このオプションに「/drv01/dir01/userdir」を指定します。また、コンテンツを格納するディレクトリとして、-h オプションで「user01」を指定します。このオプションの指定の末尾に、パスのデリミタ「/」(UNIX の場合)または「¥」(Windows の場合)を記述する必要はありません。

このオプションで指定したコンテンツ格納先ベースパスは、-h オプションで指定したコンテンツ格納先パスと結合して、絶対パスとして使用されます。ContentLocation プロパティには次に示すパスが格納されます。

aaaa/bbbb/cccc

aaaa : コンテンツ格納先パス

bbbb : DocumentBroker がコンテンツを管理するためのディレクトリ

cccc : ファイル名

注

次に示すどちらかのファイル名が設定されます。

- 登録ファイル名 (DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合)
- DocumentBroker が指定するファイル名 (DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合)

-k オプションに Reference を指定していない場合に、このオプションを指定するとエラーになります。このオプションを省略した場合、実行環境制御ファイルの、Reference セクションの DefaultContentBasePath エントリの指定を使用します。

Windows の場合、コンテンツ格納先ベースパスに UNC 形式のパスを使用することもできます。例えば、「server01」というマシンのネットワーク上で共有しているフォルダ「share」をコンテンツ格納先ベースパスに指定する場合は、このオプションに「¥¥server01¥share」を指定します。

-h コンテンツ格納先パス

リファレンスファイル文書をアップロードする場合に、コンテンツ格納先パスを指定します。このオプションは、リファレンス種別が NONE の、コンテンツが登録されていないリファレンスファイル文書をアップロードする場合だけ有効です。

-k オプションに Reference を指定していない場合に、このオプションを指定するとエラーになります。

オプション引数には、-b オプションで指定したコンテンツ格納先ベースパスからの相対パスで、コンテンツの格納先とするディレクトリを指定します。例えば、アップロードファイル URL で指定したコンテンツを、-b オプションで指定した「/drv01/dir01/userdir」ディレクトリの下位にある「user01」ディレクトリに格納する場合、このオプションに「user01」を指定します。このオプションの指定の先頭に、パスのデリミタ「/」(UNIX の場合) または「¥」(Windows の場合) を記述する必要はありません。

このオプションの指定を省略すると、-b オプションで指定したコンテンツ格納先ベースパスのディレクトリの下位に、DocumentBroker がコンテンツを管理するためのディレクトリが作成され、その下位にコンテンツが格納されます。

このオプションで指定したコンテンツ格納先パスは、-b オプションで指定したコンテンツ格納先ベースパスと結合して、絶対パスとして使用されます。ContentLocation プロパティには次に示すパスが格納されます。

aaaa/bbbb/cccc

aaaa : コンテンツ格納先パス

bbbb : DocumentBroker がコンテンツを管理するためのディレクトリ

cccc : ファイル名

注

次に示すどちらかのファイル名が設定されます。

- アップロードファイル URL で指定したコンテンツのファイル名 (DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合)
- DocumentBroker が指定するファイル名 (DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合)

コマンド引数

OIID

アップロードする文書の OIID を指定します。

OIID ファイル

アップロードする文書の OIID を記述した OIID ファイルのファイルパスを指定します。

OIID ファイルのファイル名はオブジェクトの OIID 形式以外であれば任意です。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。ただし、「dma://」で始まるパスは指定できません。なお、OIID ファイルに記述できる OIID の数は一つだけです。

アップロードファイル URL

シングルファイル文書の場合、アップロードするファイルのパスを URL 形式で指定します。

File Link 文書の場合、クライアントのコンテンツを指定して文書をアップロードするとき、ファイルサーバに登録するコンテンツのパスを URL 形式で指定します。また、ファイルサーバに登録されているコンテンツを指定して文書をアップロードするときは、DBRSVCONTENT を指定して、-g オプションにファイルサーバ上のコンテンツのパスを指定します。なお、ファイルサーバに登録されているコンテンツを使用するとき、-g オプションを省略するとエラーになります。

例えば、UNIX の場合、「/tmp/upload.doc」のファイルをアップロードするときは、「file:///tmp/upload.doc」と指定します。Windows の場合、「c:\temp\upload.doc」のファイルをアップロードするときは、「file:///c:\temp\upload.doc」と指定します。

アップロード情報ファイル

マルチファイルを管理する文書の場合、アップロードするマルチファイル文書の情報記述した、アップロード情報ファイルのファイルパスを指定します。ファイルパスは、絶対パスまたは相対パスのどちらでも指定できます。

ただし、「DBRSVCONTENT」という文字列のファイル、および「file://」で始まるパスは指定できません。

注意事項

-v オプションを指定した場合、チェックアウトおよびチェックインは実行されません。指定したバージョンのアップロードだけが実行されます。

-v オプションのオプションの引数に「DBRVTVERSION」を指定するとエラーになります。

-I オプションおよび -v オプションを指定しないでシングルファイル文書であるバージョン付き文書をアップロードした場合、チェックイン後のバージョンに対応する全文検索インデクスが作成されます。なお、マルチファイル文書の場合、File Link 文書、またはリファレンスファイル文書のレンディションの場合は、-I オプションの指定を省略しても全文検索インデクスは作成されません。

複数のレンディションを持つ、マルチファイルを管理しないバージョン付き文書に対してチェックアウトおよびチェックインをした場合、最新のバージョンの文書はマスタレンディションだけの文書になります。チェックアウト前の文書のサブレンディションは、最新バージョンには引き継がれません。

シングルファイル文書であるバージョン付き文書に対して -M オプションを指定する場合、-v オプションを指定しないとエラーになります。

アップロードファイルが格納されている場所として、ファイルパス（アップロードファイル URL）を指定した場合、アップロードファイルは、マルチファイルを管理する文書として扱われます。また、アップロードファイルが格納されている場所として、ディレクトリパス（アップロードディレクトリ URL）を指定した場合、アップロードファイルは、マルチファイルを管理する文書として扱われます。アップロードファイルが格納されている場所の指定と、アップロードファイルの種類が一致しない場合

はエラーとなります。

マルチファイル文書の場合、dbrProp_RetrievalName プロパティの値を更新するには、プロパティ情報ファイルの dbrProp_RetrievalName プロパティの値を更新して、dbrsetprop コマンドを実行します。

マルチファイル文書の場合、アップロード情報ファイルの directory エントリに指定したディレクトリ下にファイルがないとエラーになります。

シングルファイル文書のレンディションの場合、-v オプションおよび -I オプションを指定しないでバージョン付き文書をアップロードすると、チェックイン後のバージョンに対する全文検索インデックスが作成されます。

-k オプションを省略した場合、アップロードファイル URL を指定すると、シングルファイル文書として扱われ、アップロード情報ファイルを指定すると、マルチファイル文書として扱われます。

File Link 文書で、コンテンツの登録先を、ファイルサーバとする場合、-k オプションで FileLink を指定します。この場合、-k オプションは省略できません。

File Link 文書の、ファイルサーバ上のコンテンツの削除モードは、クライアントのコンテンツを指定してアップロードした場合と、ファイルサーバ上のコンテンツを指定してアップロードした場合で異なります。それぞれの場合のコンテンツの削除モードを次に示します。

- クライアントのコンテンツを指定してアップロードした場合
クライアントのコンテンツを指定してアップロードした文書を削除すると、ファイルサーバ上のコンテンツも同期して削除されます。なお、クライアントのコンテンツをディレクトリ単位で登録したときは、ディレクトリごと削除されます。
- ファイルサーバ上のコンテンツを指定してアップロードした場合
ファイルサーバ上のコンテンツを指定してアップロードした文書を削除しても、ファイルサーバ上のコンテンツは削除されません。なお、ファイルサーバ上のコンテンツをディレクトリ単位で指定したときも、ディレクトリは削除されません。

コンテンツの削除モードを変更する場合は、実行環境制御ファイルの FileLink セクションの、ClientContentDeleteMode エントリ、または ServerContentDeleteMode エントリの設定を変更してください。

リファレンスファイル文書の場合、コンテンツの格納先には、DocumentBroker サーバから接続可能であり、かつ DocumentBroker サーバから参照可能なディレクトリを設定してください。

リファレンスファイル文書をアップロードする場合に、誤って異なるリファレンスファイル文書のコンテンツ格納先ベースパスを指定すると、エラーになります。

リファレンスファイル文書を扱うコンテンツの操作でエラーになると、オブジェクトとコンテンツの状態が不整合となり、メッセージ KMBR30201-E (要因コード 3, 86) が出力される場合があります。この場合は、エラーになったオブジェクトを削除したあと、再度、dbrupdoc コマンド (文書のアップロード) を実行して、リファレンスファイル文書をアップロードしてください。また、メッセージの詳細については、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

リファレンスファイル文書の場合、コンテンツ格納先ベースパスのディレクトリおよびコンテンツ格納先パスのディレクトリの下に、DocumentBroker がコンテンツを管理するためのディレクトリが作成され、その下にコンテンツが格納されます。なお、リファレンスファイル文書のコンテンツと、コンテンツが格納されているディレクトリを同時に削除する場合は、dbrdeldoc コマンド (文書の削除) で、削除ディレクトリのルートパスを指定してください。削除ディレクトリのルートパスの指定方法については、コマンド「dbrdeldoc (文書の削除)」を参照してください。

リファレンスファイル管理機能を使用する場合のオブジェクトの削除

ここでは、リファレンスファイル管理機能を使用する場合のオブジェクトの削除方法について説明します。

リファレンスファイル管理機能を使用する場合、ユーザがコンテンツ格納先ベースパスを管理してください。特に、複数のコンテンツ格納先ベースパスを管理する場合、dbrdelcntr コマンド（コンテナの削除）および dbrdeldoc コマンド（文書の削除）では文書ごとにコンテンツ格納先ベースパスを指定できないため、オブジェクトの削除には注意が必要です。ただし、dbrdelcntr コマンド（コンテナの削除）は、コンテナごとにコンテンツ格納先ベースパスを指定できます。

コンテンツ格納先ベースパスの管理に応じたオブジェクトの削除方法を次の表に示します。

表 5-8 リファレンスファイル文書を扱う場合のオブジェクトの削除方法

管理するコンテンツ格納先ベースパスの数	削除対象のオブジェクト		削除方法
一つ	文書		dbrdeldoc コマンドを実行します。
	コンテナと包含される文書	再帰的に削除しない	dbrdelcntr コマンドを実行します。
		再帰的に削除する	-R オプションを指定して、dbrdelcntr コマンドを実行します。
複数	コンテナ内で同一 ¹	文書	-R オプションを指定して dbrdelcntr コマンドを実行するか、または次の手順で削除してください。 1. 削除対象を、同一のコンテンツ格納先ベースパスの文書に絞り込みます。 2. 同一のコンテンツ格納先ベースパスの文書に対して、dbrdeldoc コマンドを実行します。
		コンテナと包含される文書	再帰的に削除しない
	再帰的に削除する		コンテナの下位のコンテナに包含される文書のコンテンツ格納先ベースパスがすべて同一の場合 -R オプションおよび -i オプションを指定して、dbrdelcntr コマンドを実行します。 コンテナの下位のコンテナに包含される文書のコンテンツ格納先ベースパスが、コンテナごとに異なる場合 1. dbrlschild コマンドを実行して、コンテナに包含される削除対象のコンテナ一覧を取得します。 2. 取得した一覧のコンテナごとに、コンテナに包含される文書のコンテンツ格納先ベースパスをリファレンス情報ファイルに指定し、-i オプションを指定して dbrdelcntr コマンドを実行します。このとき、-R オプションは指定しないでください。
	コンテナ内に混在 ²	文書	

5. コマンドリファレンス

管理するコンテンツ格納先ベースパスの数	削除対象のオブジェクト		削除方法
	コンテナと包含される文書	再帰的に削除しない	次の手順で削除してください。 <ol style="list-style-type: none"> 1. 削除対象を、同一のコンテンツ格納先ベースパスの文書に絞り込みます。 2. 同一のコンテンツ格納先ベースパスの文書に対して、<code>dbrdeldoc</code> コマンドを実行します。 3. <code>dbrdelcntr</code> コマンドを実行して、コンテナを削除します。
		再帰的に削除する	次の手順で削除してください。 <ol style="list-style-type: none"> 1. 削除対象を、同一のコンテンツ格納先ベースパスの文書に絞り込みます。 2. 同一のコンテンツ格納先ベースパスの文書に対して、<code>dbrdeldoc</code> コマンドを実行します。 3. <code>dbrdelcntr</code> コマンドを実行して、コンテナを削除します。

注 1

コンテナ内で、コンテナに包含されるリファレンスファイル文書のコンテンツ格納先ベースパスがすべて同一の場合です。

注 2

コンテナ内で、コンテナに包含されるリファレンスファイル文書のコンテンツ格納先ベースパスが異なる場合です。

付録

付録 A OIID 指定時の URL の文法

付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲

付録 C 用語解説

付録 A OIID 指定時の URL の文法

ここでは、データベースに格納された DMA オブジェクトが保持している OIID を指定する場合の指定方法について説明します。

データベースに格納された DMA オブジェクト（永続オブジェクト）は、プロパティの値として NULL でない OIID（Object Instance Identifier）を保持しています。OIID は、論理的に永続オブジェクトへの URL（Uniform Resource Locator）であると考えられます。したがって、URL のシンタクスを使用して定義されます。OIID を定義する URL を、特に「DMA URL」といいます。

OIID の使用方法

OIID は、文書空間での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。文書空間中で OIID は、最大 2 の 48 乗 -1 個作成できます。

なお、OIID にはユーザアプリケーションから見た連続性、規則性はありません。このため、例えば、ユーザが文書やコンテナなどを指定するときに識別子として直接 OIID を入力するような使用方法は、OIID の使用方法として適しません。

(1) OIID の形式

OIID は URL のシンタクスを使用して定義します。

(a) 形式

OIID は次の形式で指定します。

`dma:///<system id>/<docspace id>/<object id>`

各項目について説明します。

`<system id>` ::= <テキスト化された DMA System の Dmald>

OIID を生成した DMA System を登録するために使われた GUID です。

「(2) OIID (DMA URL) の BNF 記法」で示す「`dmasystem`」に相当します。

`<docspace id>` ::= <テキスト化された文書空間の Dmald>

OIID を生成した文書空間を登録するために使われた GUID です。

「(2) OIID (DMA URL) の BNF 記法」で示す「`dmadocspace`」に相当します。

`<object id>` ::= <印刷可能な ASCII コード>

ASCII サブセットにコード化されたデータシーケンスです。

「(2) OIID (DMA URL) の BNF 記法」で示す「`dmaobjectid`」に相当します。

(b) 指定例

OIID の指定例を次に示します。

<指定例>

```
dma:///07a17522-a626-11d0-b11f-0020af27a837/a95414a4-fef7
-11d3-999f-080009bafc2d/f5d8e3a2-284c-11d2-9177
-0000e213036700000000000006852
```

(2) OIID (DMA URL) の BNF 記法

OIID (DMA URL) の BNF 記法を次に示します。

```
dmaurl= "dma:/" "/" dmasystem "/"dmadocspace "/" dmaobjectid
```

```
dmssystem = dmaguid
dmadocspace = dmaguid
dmaobjectid = oidchars
dmaguid = 1*dmaguidchar
dmaguidchar = hex | "-"
oidchars = *oidchar
oidchar = alpha | digit | safe | ":" | "@" | "&" | "="
alpha = lowalpha | hialpha
lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" |
          "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
          "y" | "z"
hialpha = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" |
          "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" |
          "W" | "X" | "Y" | "Z"
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
safe = "$" | "-" | "_" | "." | "+"
hex = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
```

付録 B 文書空間で使用する文字コード種別が UTF-8 の場合のサポート範囲

ここでは、文書空間で使用する文字コード種別が UTF-8 の場合にサポートする範囲について説明します。なお、文書空間で使用する文字コード種別が Shift-JIS の場合は、すべての機能をサポートしています。

文書空間で使用する文字コード種別が UTF-8 の場合に使用できるオブジェクト操作ツールのコマンドを次の表に示します。

表 B-1 文書空間で使用する文字コード種別が UTF-8 の場合に使用できるオブジェクト操作ツールのコマンド

コマンド名	機能	使用可否
dbrbind	パブリック ACL のバインド	
dbrcretacl	パブリック ACL の作成	
dbrcrentnr	コンテナの作成	
dbrcrtdata	独立データの作成	
dbrcrtdoc	文書の作成	
dbrcrtidx	全文検索インデクスの作成	
dbrdelacl	パブリック ACL の削除	
dbrdelcntr	コンテナの削除	
dbrdeldata	独立データの削除	
dbrdeldoc	文書の削除	
dbrdelidx	全文検索インデクスの削除	
dbrdowndoc	文書のダウンロード	
dbrexquery	問い合わせの実行	
dbrgetprop	プロパティの取得	
dbrlink	オブジェクト間の関連づけ	
dbrlsacl	バインドしているパブリック ACL 一覧またはバインドされているオブジェクト一覧の取得	
dbrlschild	コンテナに含まれるオブジェクト一覧の取得	
dbrlscompo	マルチファイル文書のコンテンツ情報一覧の取得	-
dbrlsparent	オブジェクトを包含するコンテナ一覧の取得	
dbrlsversion	バージョン一覧の取得	
dbrlsversionable	バージョン管理情報一覧の取得	
dbrnrndrdoc	文書のレンディション操作	
dbrsetmode	構成管理モードの設定	
dbrsetprop	プロパティの設定	
dbrtoolsetup	実行環境のセットアップ	
dbrunbind	パブリック ACL のアンバインド	
dbrunlink	オブジェクト間の関連づけの解除	
dbrupentr	コンテナのバージョンアップ	
dbrupdoc	文書のアップロード	

(凡例)

: 使用できます。

- : 使用できません。

注

File Link 文書の場合は、作成、削除、ダウンロード、またはアップロードできません。

付録 C 用語解説

DocumentBroker で使用する用語について説明します。

(英字)

ACE

Access Control Element の略です。アクセス制御エレメントのことです。

ACFlag

Access Control Flag の略です。アクセス制御フラグのことです。

ACL

Access Control List の略です。アクセス制御リストのことです。

AND-NOT 検索

検索オペレータに「AND-NOT」を使用する検索方法です。二つの検索条件を AND-NOT でつないで、左側のオペランドに指定した検索条件は成立するが、右側のオペランドに指定した検索条件は成立しない文書を検索します。例えば、「著者が『日立太郎』であるが、所属は『日立製作所』ではない文書を検索する」というような場合に使用できます。

AND 検索

検索オペレータに「AND」を使用して、検索条件同士の論理積を求める検索方法です。例えば、「著者が『日立太郎』で、文書中に『コンピュータ』という文字列を含む文書を検索する」というような場合に使用できます。

API (Application Programming Interface)

アプリケーションプログラムとのインターフェースを指します。

CdbrCompound オブジェクト

複合データを表すオブジェクトです。複合データとは、複数の異なる型によって表されるデータです。主に、VariableArray 型のプロパティとして設定されている Object 型の要素の値を参照、設定するときに使用します。

CdbrConfiguratedReferentialContainer オブジェクト

バージョン付き構成管理コンテナに相当するオブジェクトです。CdbrConfiguratedReferentialContainer クラスを基に作成します。

CdbrDocument オブジェクト

(バージョンなし)文書に相当するオブジェクトです。CdbrDocument クラスを基に作成します。CdbrVersionableDocument オブジェクトの個々のバージョンとしても操作できます。

CdbrEqStatement オブジェクト

SQL に基づいた文法で検索条件が記述できる edmSQL 検索を実行するためのオブジェクトです。CdbrEqStatement クラスを基に作成します。

CdbrIndependentPersistence オブジェクト

クラスライブラリで提供するクラスの階層構造に属さない、独立データを表すオブジェクトです。CdbrIndependentPersistence クラスを基に作成します。

CdbrPublicACL オブジェクト

パブリック ACL を表すオブジェクトです。CdbrPublicACL クラスを基に作成します。

CdbrReferentialContainer オブジェクト

フォルダや分類 (インデクス) を使用した文書管理を実現する、(バージョンなし)コンテナに相当するオブジェクトで

す。CdbReferentialContainer クラスを基に作成します。

CdbSession オブジェクト

文書空間とクライアント環境のセッションを確立するオブジェクトです。CdbSession クラスを基に作成します。

CdbVariableArray オブジェクト

可変長配列を表すオブジェクトです。主に、VariableArray 型のプロパティの値を参照、設定するときに使用します。

CdbVersionableDocument オブジェクト

DocumentBroker で扱うバージョン付き文書に相当するオブジェクトです。このオブジェクトによって表される文書は、バージョン管理ができます。CdbVersionableDocument クラスを基に作成します。

CdbVersionTraceableContainer オブジェクト

バージョンなし構成管理コンテナに相当するオブジェクトです。CdbVersionTraceableContainer クラスを基に作成します。CdbConfiguredReferentialContainer オブジェクトの個々のバージョンとしても操作できます。

Child

直接型のコンテインメント (DirectContainment) の場合、オブジェクト (Parent) に包含されるオブジェクトを指します。

ConfigurationHistory オブジェクト

バージョン管理に使用する最上位の DMA オブジェクトです。文書のバージョンを管理する VersionSeries オブジェクトを管理します。

Containable オブジェクト

コンテナの包含要素になるオブジェクトです。CdbContainable クラスのサブクラスを基に作成されたオブジェクトを指します。

Containee

参照型のコンテインメント (ReferentialContainment) の場合、オブジェクト (Container) に包含されるオブジェクトを指します。

Container

参照型のコンテインメントの場合、オブジェクト (Containee) を包含するオブジェクトを指します。

ContainerVersion オブジェクト

包含しているオブジェクト全体を一つの概念的なオブジェクトとして管理する DMA オブジェクトです。また、構成管理型のコンテインメントではオブジェクトを包含するオブジェクトとして利用できます。ContainerVersion オブジェクトは、バージョンを保持できるオブジェクトです。したがって、ContainerVersion オブジェクトのバージョンを上げることで、ある時点でのオブジェクトのまとまりを管理できます。

Container オブジェクト

コンテインメントを利用してオブジェクトを管理する場合に、オブジェクトを包含する DMA オブジェクトの一つです。dmaClass_Container クラスおよびユーザが定義した dmaClass_Container クラスのサブクラスを基に作成します。

ContentElement (dmaClass_ContentElement クラス)

文書のコンテンツにアクセスするために使われる DMA オブジェクトの抽象クラスです。ContentElement の四つのサブクラス (edmClass_ContentFileLink クラス, dmaClass_ContentTransfer クラス, edmClass_ContentTransfers クラス, および dmaClass_ContentReference クラス) は、コンテンツの格納とアクセス機能を提供します。

ContentFileLink オブジェクト

文書のコンテンツを管理するために使用する DMA オブジェクトです。File Link 連携機能を使用する場合に使用します。dmaClass_ContentElement クラスのサブクラスである edmClass_ContentFileLink クラスを基に作成します。

ContentReference オブジェクト

文書のコンテンツを管理するために使用する DMA オブジェクトです。コンテンツをデータベースに格納しないで、位置情報を永続化して管理する場合に使用します。dmaClass_ContentElement クラスのサブクラスである dmaClass_ContentReference クラスを基に作成します。

ContentTransfers オブジェクト

文書のコンテンツを管理するために使用する DMA オブジェクトです。一つの文書のコンテンツとして複数のファイルをデータベースに格納（永続化）して管理する場合に使用します。dmaClass_ContentElement クラスのサブクラスである edmClass_ContentTransfers クラスを基に作成します。

ContentTransfer オブジェクト

文書のコンテンツを管理するために使用する DMA オブジェクトです。コンテンツをデータベースに格納（永続化）して管理する場合に使用します。dmaClass_ContentElement クラスのサブクラスである dmaClass_ContentTransfer クラスを基に作成します。

CORBA (Common Object Request Broker Architecture)

OMG (Object Management Group) が提唱するオブジェクト間の通信メカニズムを提供する ORB (Object Request Broker) の標準アーキテクチャです。

DMA (Document Management Alliance)

文書管理インターフェースの標準化を図る団体 AIIM (Association for Information and Image Management International) によって定義される共通インターフェースです。

DMA URL

DMA オブジェクトの OIID (Object Instance Identifier) を定義する URL です。

DMA オブジェクト

DMA が規定するオブジェクトモデルに基づいたオブジェクトです。DMA で規定されたクラス (クラス名が dmaClass_ で始まるクラス) , DocumentBroker で拡張したクラス (クラス名が edmClass_ で始まるクラス) , およびこれらのクラスからユーザが定義したサブクラスを基に作成されます。

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に使用するファイルです。

DocVersion オブジェクト

DocumentBroker で扱う文書に相当する DMA オブジェクトです。dmaClass_DocVersion クラスおよびユーザが定義した dmaClass_DocVersion クラスのサブクラスを基に作成します。

edmClass_Relationship クラス

文書間リレーションを表す DMA クラスです。単独では作成できない、文書に従属する Relationship オブジェクトの基となるクラスです。

edmSQL

DocumentBroker のオブジェクトを検索するための検索条件式を表現するための文法です。SQL の文法に基づいていません。

edmSQL 検索

検索条件に、SQL ライクの文法で記述できる edmSQL 文を指定して実行する検索のことです。

FAM (File Access Module)

HiRDB File Link で使用するファイルサーバ上で、コンテンツを管理するためのプログラムです。

File Link 文書

HiRDB File Link で管理しているファイルサーバに格納されている一つのファイル、または一つのディレクトリおよびそのサブディレクトリ下のすべてのファイルをコンテンツとして持つ文書のことです。データベースでは、コンテンツへのリンク情報を管理しています。CdbDocument クラスまたは CdbVersionableDocument クラスを使用して操作します。

File Link 連携機能

HiRDB File Link と連携してコンテンツを管理する機能です。

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。

ID ファイル

オブジェクト操作ツールを使用して複数のオブジェクトを一括して操作する場合、操作対象となるオブジェクトの OIID などを記述するファイルです。

LDAP

Lightweight Directory Access Protocol の略です。TCP/IP 上で動作する解放型 DAP を提供し、X.500 のデータモデルを保持しています。

NOT 条件

指定したキーワードとの不一致を求める検索条件です。例えば、「作成者が『日立太郎』ではない文書を検索する」というような場合に使用できます。

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。

OR 検索

検索オペレータに「OR」を使用して、検索条件同士の論理和を求める検索方法です。例えば、「作成者が『日立太郎』であるオブジェクトか、作成者の所属が『日立製作所』であるオブジェクトを検索する」というような場合に使用できます。

Parent

直接型のコンテインメントの場合、オブジェクト (Child) を包含するオブジェクトを指します。

Rendition オブジェクト

文書の表現形式 (HTML, PDF など) を管理する DMA オブジェクトです。

Reservation オブジェクト

新しいバージョンを作成する権利を VersionSeries オブジェクトに予約する DMA オブジェクトです。

Scalar 型

プロパティの基本単位の一つです。基本単位が Scalar 型であるプロパティは、データ型に従った値を一つだけ持ちます。

SystemManager オブジェクト

System オブジェクトを管理する DMA オブジェクトです。

System オブジェクト

サーバ内の文書空間を管理する DMA オブジェクトです。

UOC (User Own Coding)

ユーザによって作成されたプログラムのことです。DocumentBroker では、ユーザ認証や、アクセス制御機能のためのユーザ情報取得に UOC を使用できます。

VariableArray 型

プロパティの基本単位の一つです。基本単位が VariableArray 型であるプロパティは、データ型に従った複数の値を可変長な一次元配列として持ちます。また配列の要素は、構造体で管理できます。

Versionable オブジェクト

バージョン管理の対象になるオブジェクトです。CdbVersionable クラスのサブクラスを基に作成されたオブジェクトを指します。

VersionDescription オブジェクト

文書のバージョン (更新履歴) を管理するために使用する DMA オブジェクトです。VersionSeries オブジェクトと Versionable オブジェクトを接続するために使用します。

VersionSeries オブジェクト

文書のバージョン (更新履歴) を管理するために使用する DMA オブジェクトです。連続的な履歴を持つバージョンの構成を保持するオブジェクトです。

VersionTracedDocVersion オブジェクト

直接型、参照型および構成管理型のコンテインメントで、コンテナに包含される対象となるためのプロパティを持つ DMA オブジェクトです。dmaClass_DocVersion クラスのサブクラスの edmClass_VersionTracedDocVersion クラスを基に作成します。

VTContaineer

構成管理型のコンテインメントの場合、オブジェクト (VTContainer) に包含されるオブジェクトを指します。

VTContainer

構成管理型のコンテインメントの場合、オブジェクト (VTContaineer) を包含するオブジェクトを指します。

W3C (World Wide Web Consortium)

HTML や XML など、WWW に関する技術の標準化を推進する非営利団体です。

(ア行)

アクセス権

オブジェクトを作成したり、すでに作成されているオブジェクトにアクセスしたりする権利です。

アクセス制御エレメント (ACE : Access Control Element)

アクセス制御リスト (ACL) の要素です。一つのサブジェクトと一つのパーミッションの組で構成され、指定されたサブジェクトに対して指定されたパーミッションの範囲のアクセス権を与えることを示す情報です。

アクセス制御機能

DocumentBroker の文書空間でのオブジェクトの作成や、管理されている文書やコンテナなどのオブジェクトに対する操作を、ユーザやグループごとに許可または制限する機能です。

アクセス制御機能付き検索

アクセス制御機能を利用した文書空間で検索を実行した場合に、ユーザにアクセス権がないオブジェクトは検索結果として取得しない検索です。

アクセス制御情報

アクセス制御されている文書空間で、ユーザがメソッドを発行する際に、アクセス権の判定に使用される情報です。

アクセス制御情報変更権

オブジェクトに設定されているアクセス制御情報 (ACFlag および ACL) を変更する権利です。また、パブリック ACL をアクセス制御対象オブジェクトにバインドすることを許可する権利も含まれます。なお、パブリック ACL のアクセス情報変更権には、パブリック ACL のユーザ定義プロパティを変更する権利が含まれます。

アクセス制御フラグ (ACFlag : Access Control Flag)

オブジェクトの所有者、プライマリグループおよび全ユーザという区分でパーミッションを設定できるアクセス制御情報の一つです。

アクセス制御モデル

アクセス制御機能を利用して運用されている文書管理モデルです。

アクセス制御リスト (ACL : Access Control List)

任意のユーザまたはグループにパーミッションを設定できるアクセス制御情報の一つです。アクセス制御エレメント (ACE) のリストで構成されます。

アドレス呼び出し

文書空間内に構成されている独立した永続オブジェクトに付けられた ID (OID) を利用して、そのオブジェクトを探索することです。

インデクス情報ファイル

ユーザが追加するプロパティにインデクスを定義する場合に、定義するインデクスの情報を記述するファイルです。

永続オブジェクト

データベースに格納されたオブジェクトを指します。

永続プロパティ

データベースに存在するプロパティを指します。

オブジェクト作成権

オブジェクト作成権限で、文書空間にオブジェクトを作成する権利を与えるパーミッションです。ユーザ権限定義ファイルで使用します。

オブジェクト作成権限

文書空間にオブジェクトを作成する権限で、ユーザ権限の一つです。ユーザ権限定義ファイルで定義します。オブジェクト作成権限を与えられたユーザおよびグループに属するユーザは、オブジェクトを作成するメソッド (CreateObject) を実行できます。

オブジェクト操作権限

文書空間内のすべてのオブジェクトを、与えられた権限の範囲で操作する権利で、ユーザ権限の一つです。ユーザ権限定義ファイルで定義します。例えば、オブジェクト操作権限としてプロパティ参照権を与えられたユーザおよびグループに属するユーザは、文書空間内のすべてのオブジェクトのプロパティを参照できます。

オブジェクト操作ツール

DocumentBroker が提供するクラスライブラリを使用したクライアントアプリケーションです。クラスライブラリを使用してユーザがクライアントアプリケーションを開発する場合に、必要であると思われる機能をコマンド群として提供します。

オブジェクトリファレンス

DMA オブジェクトへのリファレンスを示す Object 型プロパティの値です。例えば、dmaProp_ParentContainer プロパティの値がこれに当たります。

(力行)

仮のバージョン識別子

チェックアウト中の仮のバージョンを識別するための識別子です。
 チェックアウト中のオブジェクトを参照・更新する時に使用します。この識別子はチェックアウト時に DocumentBroker によって設定される識別子であり、仮のバージョンに該当するオブジェクトの OIID とは異なります。

関連オブジェクト

コンテインメントの型 (直接型, 間接型および構成管理型) を定義するための DMA オブジェクトです。
 直接型のコンテインメントの場合, オブジェクトを包含するオブジェクト (Parent) とオブジェクトに包含されるオブジェクト (Child) を DirectContainmentRelationship オブジェクトを使用して関連づけます。参照型のコンテインメントの場合, オブジェクトを包含するオブジェクト (Container) とオブジェクトに包含されるオブジェクト (Containee) を ReferentialContainmentRelationship オブジェクトを使用して関連づけます。
 構成管理型のコンテインメントの場合, オブジェクトを包含するオブジェクト (VTContainer) とオブジェクトに包含されるオブジェクト (VTContainee) を VersionTraceableContainmentRelationship オブジェクトを使用して関連づけます。

基本コンテンツ更新権

基本パーミッションの一つで, オブジェクトのコンテンツを更新する権利を与えるパーミッションです。基本プロパティ参照権を含みます。文書に対して設定する場合は, 全文検索インデックスを作成, 削除する権利を含みます。

基本コンテンツ参照権

基本パーミッションの一つで, オブジェクトのコンテンツを参照する権利を与えるパーミッションです。基本プロパティ参照権を含みます。文書に対して設定する場合は, 全文検索を実行する権利を含みます。

基本削除権

基本パーミッションの一つで, オブジェクトを削除する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本単位

DMA で規定されているプロパティは, データ型に従った値を 1 個持つか複数個持つかが決められています。これを基本単位といいます。

基本バージョン管理権

基本パーミッションの一つで, バージョン管理されているオブジェクトのバージョンを追加, 削除する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本パーミッション

ユーザおよびグループがオブジェクトに対して実行できる操作の範囲を定めるパーミッションの基本単位です。オブジェクト操作権限, アクセス制御フラグおよびアクセス制御エレメントで, ユーザおよびグループに許可する操作の範囲を定めるときに使用します。例えば, あるユーザに対して, 文書の更新と削除を許可する場合は, 更新と削除を許可するために, 基本コンテンツ更新権と基本削除権という二つのパーミッションを設定します (一つのパーミッションで一つの権利を与える)。基本パーミッションには, 基本プロパティ参照権, 基本プロパティ更新権, 基本コンテンツ参照権, 基本コンテンツ更新権, 基本リンク権, 基本バージョン管理権および基本削除権があります。

基本プロパティ更新権

基本パーミッションの一つで, オブジェクトのプロパティを更新する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本プロパティ参照権

基本パーミッションの一つで, オブジェクトのプロパティを参照する権利を与えるパーミッションです。そのほかのすべての基本パーミッションに含まれます。コンテナに対して設定する場合は, 関連オブジェクトのユーザ定義プロパティの参照と管理されている要素を参照する権利を含みます。

基本リンク権

基本パーミッションの一つで、コンテンツの設定、変更および関連オブジェクトのユーザ定義プロパティを変更する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

近傍条件検索

検索オペレータに「Prox」を使用する拡張検索（全文検索）です。指定する検索タームが同時に存在する場合に、その検索ターム間の距離を条件として検索します。例えば、「『文書管理』という検索タームと『ドキュメント』という検索タームを含み、これらの検索タームがこのとおりの順序で出現し、かつ検索ターム間に入る文字が 5 文字以内である文書を検索する」というような場合に使用できます。

組み合わせパーミッション

基本パーミッションを複数組み合わせた権利を与えるパーミッションの単位です。アクセス制御フラグおよびアクセス制御エレメントでユーザおよびグループに許可する操作の範囲を定めるときに使用します。組み合わせパーミッションには、プロパティ参照権、参照権、プロパティ更新権、参照更新権、削除権、リンク権、バージョン管理権およびフルコントロールがあります。例えば、あるユーザに対して、ある文書の参照更新権という組み合わせパーミッションを設定すると、そのユーザは基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権および基本コンテンツ更新権が設定されたのと同じ範囲の操作を、その文書に対して実行できます。

クラス定義情報ファイル

DocumentBroker サーバで定義されている DMA オブジェクトのクラスまたはそのサブクラスのクラス名またはプロパティ名から GUID、データ型、プロパティの基本単位などの情報を取得するために使用するファイルです。

EDMCrtSimMeta コマンドで作成できます。オブジェクト操作ツールを使用する場合に必要です。

継承 (Inheritance)

既存のクラスを利用して新しいクラスを定義するオブジェクト指向の技術です。

構成管理型のコンテナメント (VersionTraceableContainment)

コンテナを利用した文書管理の方法です。構成管理型のコンテナメントで使われるコンテナは、バージョンを作成できるオブジェクトの構成を管理します。

構成管理コンテナ

構成管理型のコンテナメントで使われるコンテナです。

包含されるオブジェクトのバージョンを固定して管理したり、常に最新のバージョンを追跡して管理したりすることで、包含されるオブジェクト全体の構成を管理するために使用するコンテナです。CdbVersionTraceableContainer に相当します。DMA オブジェクトでは、ContainerVersion オブジェクトに相当します。

構造指定検索

XML 文書を管理している場合に、文書の論理構造（エレメント）やエレメントの属性をキーとして検索する方法です。例えば、「タイトルに『コンピュータ』という単語が含まれ、章に『XML』という単語が含まれる文書を検索する」というような場合に使用します。また、「『document』というエレメントに設定されている属性『status』の属性値が『public』である文書を検索する」というような場合にも使用できます。

コンテナメント (包含)

コンテナを使用したオブジェクトの包含関係を示します。コンテナメントには直接型 (DirectContainment)、参照型 (ReferentialContainment) および構成管理型 (VersionTraceableContainment) の 3 種類があります。

コンテナ

コンテナメントで、オブジェクトを包含できるオブジェクトの総称です。CdbReferentialContainer オブジェクト、CdbConfiguredReferentialContainer オブジェクトおよび CdbVersionTraceableContainer オブジェクトがこれに当たります。また、DMA オブジェクトとしては、Container オブジェクト、VersionTraceableContainer オブジェクトおよび ContainerVersion オブジェクトがあります。

コンテンツ

一般的には、属性に対する文書のデータ部分を指します。DMA では、ある文書に関連づけられていて、DMA で規定さ

れている Content モデルに従ってアクセスされるオブジェクトの集合を指します。また、アクセスされるオブジェクトの実体（例えば、report.doc、document.htm など）をコンテンツデータといいます。

コンテンツ格納先パス

リファレンスファイル管理機能を使用する場合に、コンテンツ格納先ベースパスを基点とする相対パスのことです。

コンテンツ格納先ベースパス

リファレンスファイル管理機能を使用する場合に、コンテンツ格納先の基点となるディレクトリパスのことです。

コンテンツ種別

レンディションのコンテンツが、シングルファイル文書のコンテンツか、マルチファイル文書のコンテンツか、リファレンスファイル文書のコンテンツか、または File Link 文書のコンテンツかを示します。

コンテンツロケーション

リファレンスファイル管理機能を使用する場合に、コンテンツの格納先を示す情報のことです。

(サ行)

削除権

組み合わせパーミッションの一つです。基本削除権と同じ操作を許可するパーミッションです。

サブクラス

あるクラスから派生するクラスのことです。または、それ自身がサブクラスとして参照されているクラスのことです。

サブジェクト (Subject)

アクセス権を与えるユーザまたはグループです。

サブジェクト種別 (SubjectType)

アクセス権を与えるサブジェクトが、ユーザなのか、グループなのかまたはシステムなのかを識別するための情報です。

サブレンディション

マルチレンディション文書に、追加登録されたレンディションのことです。マスタレンディション以外のレンディションを指します。なお、サブレンディションは、登録後にマスタレンディションに変更できます。

参照型のコンテインメント (ReferentialContainment)

コンテナを利用した文書管理の方法です。参照型のコンテインメントで使われるコンテナは、文書にはり付けるインデックスの働きをします。

参照権

組み合わせパーミッションの一つです。基本コンテンツ参照権と同じ操作を許可するパーミッションです。

参照更新権

組み合わせパーミッションの一つです。基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権および基本コンテンツ更新権を組み合わせたパーミッションです。すなわち、参照更新権を設定することで、プロパティを参照、更新する権利とコンテンツを参照、更新する権利を設定できます。

システム管理者

DocumentBroker の運用管理者です。

実行環境制御ファイル

オブジェクト操作ツールの実行環境を定義するファイルです。

状態フラグ

マルチレンディション文書の、マスタレンディションに対するサブレンディションのコンテンツの状態を表すフラグで

す。マスタレンディションとサブレンディションのコンテンツの状態が一致している、マスタレンディションのコンテンツが更新されたのに対してサブレンディションのコンテンツが更新されていない、またはサブレンディションのコンテンツが存在しない、という 3 種類の状態が表されます。dbrProp_RenditionStatus プロパティの下位 2 バイトに設定されます。

所有者 (Owner)

オブジェクトの所有者として設定されているユーザです。アクセス制御フラグでパーミッションを与えられます。所有者に設定されているユーザは、そのオブジェクトのアクセス制御フラグで所有者に与えられたパーミッションの範囲の操作をそのオブジェクトに対して実行できます。また、そのオブジェクトの所有者およびセキュリティ ACL の値を変更できます。

シングルファイル文書

データベースに登録されている一つのファイルをコンテンツとして持つ文書のことです。CdbDocument クラスまたは CdbVersionableDocument クラスを使用して操作します。

スーパークラス

あるクラスのクラス定義に使われたクラスを、派生したクラスのスーパークラスといいます。

セキュリティ ACL

オブジェクトに設定されたアクセス制御情報へのアクセスを制御するためのアクセス制御リストです。任意のユーザまたはグループにアクセス制御情報変更権を設定できます。

セキュリティ運用者

DocumentBroker のアクセス制御の運用情報の管理者です。セキュリティ定義ファイルを保守します。

セキュリティ管理者

アクセス制御機能を利用した文書空間で、アクセス判定を受けることなく、すべてのオブジェクトに自由にアクセスする特権を持ち、文書空間のすべてのオブジェクトを保守するユーザです。セキュリティ定義ファイルに定義します。

セキュリティ定義ファイル

アクセス制御の運用情報を定義するファイルです。セキュリティ管理者、ユーザ権限定義ファイル名およびオブジェクト作成時に、アクセス制御フラグにデフォルトで設定されるパーミッションを定義します。

全文検索

文書に含まれるキーワードを条件 (全文検索条件) として、キーワードを含む文書を検索する方法です。

全文検索インデクス

全文検索の対象になるテキストデータに対応するプロパティです。edmProp_TextIndex プロパティ, edmProp_StIndex プロパティ, edmProp_ConceptTextIndex プロパティ, edmProp_ConceptStIndex プロパティおよび edmProp_Content プロパティに相当します。

全文検索機能付き文書クラス

全文検索の対象となる文書を作成するための dmaClass_DocVersion クラスのサブクラスです。メタ情報ファイルを新規に作成したり、既存のメタ情報ファイルを編集したりして、全文検索のためのプロパティを追加したユーザ定義のクラスです。

(夕行)

チェックアウト (check-out)

文書またはコンテナにバージョンを追加するために、次バージョンの追加を予約して、最新バージョンのコピーを要求することです。

DMA オブジェクトでは、VersionSeries オブジェクトに新規 DocVersion オブジェクト (文書) を作成するための権利を予約して、新規 DocVersion オブジェクトの追加対象となる VersionSeries オブジェクトで管理している最新の DocVersion オブジェクトのコピーを要求することです。

チェックイン (check-in)

文書またはコンテナのバージョンの追加を確定することです。
DMA オブジェクトでは、VersionSeries オブジェクトに、新規 DocVersion オブジェクト (文書) を追加することです。
クライアントは特定の VersionSeries オブジェクトに対して、Reservation オブジェクトを保持しておく必要があります。

直接型のコンテインメント (DirectContainment)

コンテナを利用した文書管理の方法です。直接型のコンテインメントで使われるコンテナは、文書を格納するフォルダの働きをします。

定義情報ファイル

サブクラスおよびプロパティを追加するときに、追加するオブジェクトの定義情報を記述するファイルです。

ディレクトリサービス

ネットワーク上にあるユーザや組織の情報などの資源とその属性を記憶し、検索できるようにしたシステムです。
DocumentBroker では、Active Directory や Sun Java System Directory Server などの製品を使用した LDAP 対応のディレクトリサービスと連携できます。
LDAP 対応のディレクトリサービスとして使用できる製品の詳細については、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。

動作環境メタ情報ファイル

DocumentBroker を起動するときに参照するメタ情報です。実行環境ディレクトリ /etc/meta_files に格納されます。

独立データ

CdbrIndependentPersistence オブジェクトに相当します。ほかのオブジェクトに依存しない、独立したオブジェクトです。

特権

アクセス制御機能を利用した文書空間で、アクセス判定を受けることなく、すべてのオブジェクトに自由にアクセスする権利です。セキュリティ定義ファイルにセキュリティ管理者として定義されたユーザに与えられます。特権の有無は、ログイン時にセキュリティ定義ファイルが参照され、ログインユーザごとに作成されるユーザ情報に保持されます。

トップオブジェクト

クラスライブラリのオブジェクトを構成する複数の DMA オブジェクトのうち、最上位に位置するオブジェクトです。
例えば、CdbrDocument オブジェクトの場合は、DMA オブジェクトの DocVersion オブジェクトがトップオブジェクトです。

(ナ行)

ナビゲーション

ある独立した永続オブジェクトから、別の関連する独立した永続オブジェクトに対する横断的な参照関係を指します。

(ハ行)

バージョン管理権

組み合わせパーミッションの一つです。基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権、基本コンテンツ更新権および基本バージョン管理権を組み合わせたパーミッションです。

バージョン付き構成管理コンテナ

バージョンを保持できる構成管理コンテナです。CdbrConfiguredReferentialContainer オブジェクトに相当します。
DMA オブジェクトでは、ConfigurationHistory オブジェクトに相当します。

バージョンなし構成管理コンテナ

バージョンを保持しない構成管理コンテナです。CdbVersionTraceableContainer オブジェクトに相当します。DMA オブジェクトでは、ContainerVersion オブジェクトに相当します。

パーミッション

オブジェクトの作成，オブジェクトのプロパティ参照，オブジェクトのコンテンツ更新などの実行可能な操作の範囲を表す値です。オブジェクト作成権限を与えるパーミッション，オブジェクトの操作の範囲を定めるパーミッションがあります。オブジェクトの操作の範囲を定めるパーミッションには，基本パーミッションと組み合わせパーミッションがあります。

パブリック ACL

文書空間にオブジェクトとして存在するアクセス制御リスト（ACL）です。複数のオブジェクトが共有できます。

ファイルサーバ

HiRDB File Link で構築したファイルサーバを指します。

フィルタリング定義ファイル（TFD）

Preprocessing Library for Text Search の正規化機能によって構造指定検索用データ（ESIS-B データ）から不要なタグを除去したり，不要なタグ間のテキストをタグごと除去したりする場合に必要な定義ファイルです。

プライマリグループ

アクセス制御フラグ（ACFlag）でパーミッションを与えるグループです。

フルコントロール

組み合わせパーミッションの一つです。すべての基本パーミッションを組み合わせたパーミッションです。オブジェクトに対するすべての操作を許可します。

プロパティ更新権

組み合わせパーミッションの一つです。基本プロパティ更新権と同じ操作を許可するパーミッションです。

プロパティ参照権

組み合わせパーミッションの一つです。基本プロパティ参照権と同じ操作を許可するパーミッションです。

文書

CdbVersionableDocument オブジェクトおよび CdbDocument オブジェクトに相当します。

CdbVersionableDocument オブジェクトを，「バージョン付き文書」と呼びます。CdbDocument オブジェクトを，「バージョンなし文書」と呼びます。

DMA オブジェクトでは，dmaClass_DocVersion クラスおよびそのサブクラスを基に作成するオブジェクトのことで

す。

文書空間

DMA オブジェクトモデルを実装するリポジトリです。

変換フラグ

マルチレンディション文書の，サブレンディションのコンテンツを，レンディション変換の対象にするかどうかを表すフラグです。DocumentBroker Rendering Option を使用してレンディション変換を実行する場合に使用します。また，DocumentBroker Rendering Option によるレンディション変換でエラーが発生した場合には，エラーを示すフラグとしても使われます。dbrProp_RenditionStatus プロパティの上位 2 バイトに設定されます。

（マ行）

マスタレンディション

マルチレンディション文書に，最初に登録されたレンディションのことです。マルチレンディション文書を参照・更新

するときには、レンディション形式を指定しますが、レンディション形式を指定しない場合は、マスタレンディションが対象になります。なお、マスタレンディションとして扱うレンディションは、登録後に変更できません。

マルチファイル管理機能

一つの文書に、複数のファイルを登録して管理する機能です。

マルチファイル文書

データベースに登録されている複数のファイルをコンテンツとして持つ文書のことです。CdbDocument クラスまたは CdbVersionableDocument クラスを使用して操作します。

マルチレンディション機能

一つの文書に、同一内容の複数の異なる形式のコンテンツを登録する機能です。

マルチレンディション文書

複数のレンディションを登録している文書のことです。一つの同じ内容を表す複数の形式のコンテンツを保持する文書です。CdbDocument クラスまたは CdbVersionableDocument クラスを使用して操作します。

メタ情報ファイル

DocumentBroker が利用する DMA のクラス、プロパティ、検索オペレータなどの詳細情報を定義したファイルです。クラスおよびプロパティを追加、変更する場合に使用できます。

メタデータ (metadata)

クラス、プロパティおよびオペレータに関する詳細情報を定義するデータです。

メタデータ空間

単一の継承関係を形成するクラスの集合のことです。

(ヤ行)

ユーザ権限

文書空間にオブジェクトを作成する権利 (オブジェクト作成権限) と、文書空間内のすべてのオブジェクトに対する操作の範囲 (オブジェクト操作権限) をユーザまたはグループ単位で定めるアクセス制御情報の一つです。ユーザ権限定義ファイルに定義します。ユーザ権限の内容は、ログイン時にユーザ権限定義ファイルが参照され、ログインユーザごとに作成されるユーザ情報に保持されます。

ユーザ権限定義ファイル

ユーザ権限 (オブジェクト作成権限およびオブジェクト操作権限) を定義するためのファイルです。

ユーザ情報

ログインユーザのユーザ識別子、所属グループ、特権およびユーザ権限を表す情報です。ログイン時にユーザごとに生成され、アクセス権の判定に使用されます。

(ラ行)

ランキング検索

全文検索条件に対する適応度をスコアとして算出して、スコアを基に検索結果集合の要素 (文書) をソートして出力する検索です。

リファレンスファイル管理機能

DocumentBroker サーバから接続可能なファイルシステムの任意のディレクトリで文書のコンテンツを管理し、文書のプロパティおよびコンテンツの格納先の情報をデータベースで管理する機能です。

リファレンスファイル文書

DocumentBroker サーバから接続可能なファイルシステムの任意のディレクトリに格納されているファイルをコンテンツとして持つ文書のことです。データベースでは、文書のプロパティとコンテンツの格納先の情報を管理しています。CdbDocument クラスまたは CdbVersionableDocument クラスを使用して操作します。

リレーション

DMA オブジェクトの Relationship オブジェクトに相当します。文書と文書間の参照関係を表すオブジェクトです。

リレーション先文書

文書間リレーションを設定される、参照先になる文書です。

リレーション識別子

文書間に設定されたリレーションを識別するための識別子です。

リレーションを削除したり、リレーションのプロパティを更新したりする時に使用します。この識別子は、リレーションを設定した時に DocumentBroker によって設定される識別子です。同じ文書から同じ文書に対して二つのリレーションを設定した場合は、それぞれ異なるリレーション識別子が設定されます。

リレーション元文書

文書間リレーションを設定する元になる文書です。

リンク

DMA オブジェクトの DirectContainmentRelationship オブジェクト、ReferentialContainmentRelationship オブジェクトまたは VersionTraceableContainmentRelationship オブジェクトによって表される、文書とコンテナの関連づけです。

リンク権

組み合わせパーミッションの一つです。基本リンク権と同じ操作を許可するパーミッションです。

リンク識別子

コンテナオブジェクトへのリンク（関連づけ）を識別するための識別子です。

リンクを解除したり、リンクのプロパティの参照または更新したり、構成管理モードを変更したりする時に使用します。この識別子は関連づけをした時に DocumentBroker によって設定される識別子です。同じコンテナオブジェクトに、同じ要素を 2 度関連づけした場合は、それぞれ異なるリンク識別子が設定されます。

レンディション

文書のコンテンツの形式およびそのコンテンツをあわせてレンディションと呼びます。DMA オブジェクトの Rendition オブジェクトおよび ContentTransfer オブジェクトに相当します。

レンディションタイプ

Word などのアプリケーションで編集したファイル、HTML 形式のファイル、GIF などの画像データのファイルのように、登録した文書のコンテンツのファイルの形式を表す文字列です。レンディションごとに設定できます。

DocumentBroker では、レンディションタイプとして、MIME 名を指定することを推奨しています。

レンディション変換

マルチレンディション文書の、マスタレンディションのコンテンツの文書形式を変換して、サブレンディションのコンテンツを作成、登録することです。

ローカル ACL

オブジェクトごとに設定できるアクセス制御リスト (ACL) です。VariableArray 型のプロパティとして設定されます。

ロケール (Locale)

言語や使用する文字コードの種別、特定の国や地域で特別な意味を持つ属性などの定義のことです。ロケールは地域化した形でアプリケーションの拡張性を提供するために使われます。

索引

A

ACE 180
ACFlag 180
ACL 180
AND-NOT 検索 180
AND 検索 180
API 180

C

CdbrCompound オブジェクト 180
CdbrConfiguredReferentialContainer オブジェク
ト 180
CdbrDocument オブジェクト 180
CdbrEqStatement オブジェクト 180
CdbrIndependentPersistence オブジェクト 180
CdbrPublicACL オブジェクト 180
CdbrReferentialContainer オブジェクト 180
CdbrSession オブジェクト 181
CdbrVariableArray オブジェクト 181
CdbrVersionableDocument オブジェクト 181
CdbrVersionTraceableContainer オブジェクト 181
Child 181
ConfigurationHistory オブジェクト 181
Containable オブジェクト 181
Containee 181
Container 181
ContainerVersion オブジェクト 181
Container オブジェクト 181
ContentElement 181
ContentFileLink オブジェクト 181
ContentReference オブジェクト 182
ContentTransfers オブジェクト 182
ContentTransfer オブジェクト 182
CORBA 182

D

DBR_DETAIL_ERRORLOG 11, 13
dbrbind 79
dbrertacl 81
dbrertentr 83
dbrertdata 88
dbrertdoc 90
dbrertidx 101
dbrdelacl 103
dbrdelentr 104

dbrdeldata 111
dbrdeldoc 112
dbrdelidx 116
dbrdowndoc 117
dbrexquery 120
dbrgetprop 122
dbrlink 125
dbrlsacl 128
dbrlschild 131
dbrlscompo 134
dbrlsparent 136
dbrlsversion 139
dbrlsversionable 141
dbrnrdrdoc 143
dbrsetmode 151
dbrsetprop 155
DBRTOOLS DIR 11, 12
dbrtoolsetup 158
dbrunbind 160
dbrunlink 162
dbrupentr 165
dbrupdoc 166
directory エントリ 39
DMA 182
DMA URL 182
DMA オブジェクト 182
DocumentBroker Development Kit をバージョン
アップした場合 9
DocumentBroker Runtime をバージョンアップした
場合 9
DocumentSpace 構成定義ファイル 182
DocVersion オブジェクト 182

E

edmClass_Relationship クラス 182
edmSQL 182
edmSQL 検索 182

F

FAM 182
filelink_contentpath エントリ 41
filelink_famname エントリ 41
FileLink セクション 21
File Link 文書 183
File Link 連携機能 183
files エントリ 40

G

GUID 183

I

ID ファイル 18, 32, 183

L

LANG 10
LDAP 183
LIBPATH 11
link エントリ 38
Log セクション 20

M

mime エントリ 41
mode エントリ 38

N

NODISCLAIM 12
NOT 条件 183

O

OID 183
OID (DMA URL) のBNF 記法 176
OID 指定時の URL の文法 176
OID の形式 176
OID ファイル 32
OR 検索 183
Output セクション 21

P

Parent 183
PATH 11, 12
PSALLOC 12

R

reference_contentbasepath エントリ 43
reference_contentpath エントリ 44
reference_type エントリ 42
Reference セクション 24
Rendition オブジェクト 183
Reservation オブジェクト 183

S

Scalar 型 183

Session セクション 20
SystemManager オブジェクト 183
System オブジェクト 183

T

TZ 11

U

UOC 184
url エントリ 38

V

VariableArray 型 184
Versionable オブジェクト 184
VersionDescription オブジェクト 184
VersionSeries オブジェクト 184
VersionTracedDocVersion オブジェクト 184
VTContainee 184
VTContainer 184

W

W3C 184

X

XDK_HOME 11

あ

アクセス権 184
アクセス制御エレメント 184
アクセス制御機能 184
アクセス制御機能付き検索 184
アクセス制御情報 184
アクセス制御情報変更権 185
アクセス制御フラグ 185
アクセス制御モデル 185
アクセス制御リスト 185
アップロード情報ファイル 57
アドレス呼び出し 185

い

インデクス情報ファイル 185

え

永続オブジェクト 185
永続プロパティ 185

お

オブジェクト作成権 185
 オブジェクト作成権限 185
 オブジェクト操作権限 185
 オブジェクト操作ツール 185
 オブジェクト操作ツールで使用できる文字コード種別
 6
 オブジェクトリファレンス 185

か

仮のバージョン識別子 186
 関連オブジェクト 186

き

基本コンテンツ更新権 186
 基本コンテンツ参照権 186
 基本削除権 186
 基本単位 186
 基本バージョン管理権 186
 基本パーミッション 186
 基本プロパティ更新権 186
 基本プロパティ参照権 186
 基本リンク権 187
 近傍条件検索 187

く

組み合わせパーミッション 187
 クラス定義情報ファイル 13, 18, 28, 187

け

継承 187

こ

構成管理型のコンテナメント 187
 構成管理コンテナ 187
 構造指定検索 187
 コマンド一覧 75
 コマンド実行ログ 64
 コンテナメント 187
 コンテナ 187
 コンテンツ 187
 コンテンツ格納先パス 188
 コンテンツ格納先ベースパス 188
 コンテンツ種別 188
 コンテンツロケーション 188

さ

削除権 188
 サブクラス 188
 サブジェクト 188
 サブジェクト種別 188
 サブレンディション 188
 参照型のコンテナメント 188
 参照権 188
 参照更新権 188

し

システム管理者 188
 実行環境制御ファイル 13, 18, 19, 188
 実行環境制御ファイルの記述例 26
 出力ログ情報 65
 障害対策 70
 状態フラグ 188
 所有者 189
 シングルファイル文書 189

す

スーパークラス 189

せ

セキュリティ ACL 189
 セキュリティ運用者 189
 セキュリティ管理者 189
 セキュリティ定義ファイル 189
 全文検索 189
 全文検索インデクス 189
 全文検索機能付き文書クラス 189

ち

チェックアウト 189
 チェックイン 190
 直接型のコンテナメント 190

て

定義情報ファイル 190
 ディレクトリサービス 190

と

問い合わせファイル 18, 55
 動作環境メタ情報ファイル 190
 独立データ 190
 特権 190

トップオブジェクト 190

な

ナビゲーション 190

は

バージョン管理権 190

バージョン識別子ファイル 32

バージョン付き構成管理コンテナ 190

バージョンなし構成管理コンテナ 191

パーミッション 191

パブリック ACL 191

ふ

ファイルサーバ 191

フィルタリング定義ファイル (TFD) 191

プライマリグループ 191

フルコントロール 191

プロパティエントリ 36

プロパティ更新権 191

プロパティ参照権 191

プロパティ情報ファイル 18, 34

プロパティ情報ファイルの記述例 49

文書 191

文書空間 191

へ

変換フラグ 191

ま

マスタレンディション 191

マルチファイル管理機能 192

マルチファイル文書 192

マルチレンディション機能 192

マルチレンディション文書 192

め

メタ情報ファイル 192

メタデータ 192

メタデータ空間 192

ゆ

ユーザ権限 192

ユーザ権限定義ファイル 192

ユーザ情報 192

ら

ランキング検索 192

り

リファレンス情報ファイル 60

リファレンスファイル管理機能 192

リファレンスファイル管理機能を使用する場合のオブジェクトの削除 173

リファレンスファイル文書 193

リレーション 193

リレーション先文書 193

リレーション識別子 193

リレーション元文書 193

リンク 193

リンク権 193

リンク識別子 193

リンク識別子ファイル 32

れ

レンディション 193

レンディションタイプ 193

レンディション変換 193

ろ

ローカル ACL 193

ロケール 193