

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

文法書

3000-3-F14-20

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)

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

Acrobat Distiller は、Adobe Systems Incorporated (アドビシステムズ社) の商標です。

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。

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation の米国およびその他の国における商標または登録商標です。

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。

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

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

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

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

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 3000-3-F14-20

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-F14-20) 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)

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	-

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

対象読者

このマニュアルは、クラスライブラリを利用してクライアントアプリケーションを開発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
- C++ 言語に関する知識
- XML 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 クラスライブラリのクラス、メソッドおよび関数
クラスライブラリで提供するクラス、メソッドおよび関数を簡単な機能概略付きの一覧表形式で説明しています。
- 第 2 章 クラスライブラリで使用するデータ型とクラスおよびメソッドの記述形式
クラスライブラリのリファレンスマニュアルで使用するデータ型の定義、およびクラス、メソッドの記述形式（ページレイアウト）について説明しています。
- 第 3 章 CdbCompound クラス（複合データを扱うクラス）
CdbCompound クラスの詳細とメソッドの文法について説明しています。
- 第 4 章 CdbConfiguredReferentialContainer クラス（バージョン付き構成管理コンテナを扱うクラス）
CdbConfiguredReferentialContainer クラスの詳細とメソッドの文法について説明しています。
- 第 5 章 CdbContainable クラス（包含要素を扱うクラス）
CdbContainable クラスの詳細とメソッドの文法について説明しています。
- 第 6 章 CdbCore クラス（エラー情報を取得するクラス）
CdbCore クラスの詳細とメソッドの文法について説明しています。
- 第 7 章 CdbDMA クラス（OID の設定やプロパティを扱うクラス）
CdbDMA クラスの詳細とメソッドの文法について説明しています。
- 第 8 章 CdbDocument クラス（バージョンなし文書を扱うクラス）
CdbDocument クラスの詳細とメソッドの文法について説明しています。
- 第 9 章 CdbEqStatement クラス（edmSQL 文を操作、実行するクラス）
CdbEqStatement クラスの詳細とメソッドの文法について説明しています。
- 第 10 章 CdbIndependentPersistence クラス（独立データを扱うクラス）
CdbIndependentPersistence クラスの詳細とメソッドの文法について説明しています。

はじめに

第 11 章 CdbPublicACL クラス (パブリック ACL を扱うクラス)

CdbPublicACL クラスの詳細とメソッドの文法について説明しています。

第 12 章 CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

CdbReferentialContainer クラスの詳細とメソッドの文法について説明しています。

第 13 章 CdbSession クラス (セッションを確立するクラス)

CdbSession クラスの詳細とメソッドの文法について説明しています。

第 14 章 CdbVariableArray クラス (可変長配列を扱うクラス)

CdbVariableArray クラスの詳細とメソッドの文法について説明しています。

第 15 章 CdbVersionable クラス (バージョン管理機能を提供するクラス)

CdbVersionable クラスの詳細とメソッドの文法について説明しています。

第 16 章 CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

CdbVersionableDocument クラスの詳細とメソッドの文法について説明しています。

第 17 章 CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

CdbVersionTraceableContainer クラスの詳細とメソッドの文法について説明しています。

第 18 章 CdbXmlTranslator クラス (XML 変換を扱うクラス)

CdbXmlTranslator クラスの詳細とメソッドの文法について説明しています。

第 19 章 CdbXmlTranslatorFactory クラス (XML の操作環境を管理するクラス)

CdbXmlTranslatorFactory クラスの詳細とメソッドの文法について説明しています。

第 20 章 関数

クラスライブラリが提供する関数について説明しています。

付録 A コーディング例

クラスライブラリの機能を使用してクライアントアプリケーションを作成する場合のコーディング例について説明しています。

付録 B DMA オブジェクトのプロパティ

DMA オブジェクトが保持しているプロパティのうち、クラスライブラリによって操作できるプロパティについて説明しています。

付録 C クラスライブラリのメソッドで使用する構造体

クラスライブラリのメソッドを使用するときに、オブジェクトに設定する情報を指定したり既存のオブジェクトの情報を取得したりするために使用する構造体について説明しています。

付録 D メソッドの実行に必要なパーミッション (アクセス制御機能を使用する場合)

アクセス制御機能を使用する場合に、それぞれのメソッドをコールするときに必要なパーミッションについて説明しています。

付録 E メソッド実行時の排他制御 (ロック)

クラスライブラリのメソッドの実行時に明示的に設定できるロックと暗黙で設定されるロックについて説明しています。

付録 F 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 G 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メソッドと関数

クラスライブラリのメソッドおよび関数で文字コード種別「UTF-8」をサポートしている範囲について説明しています。

す。

付録 H 戻り値一覧

クラスライブラリのメソッドを発行したときに返される戻り値について、一覧形式で説明しています。

付録 I 障害が発生した時に DocumentBroker が出力する情報

障害が発生した時に uCosminexus DocumentBroker が出力する情報の出力先などについて説明しています。

付録 J 用語解説

uCosminexus DocumentBroker Version 3 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

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 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。

DocumentBroker Rendering Option Version 3 (3020-3-N47)

Adobe Acrobat Distiller と連携した DocumentBroker Rendering Option の機能、環境設定、およびコマンドの文法について知りたい場合に参照してください。

DocumentBroker Rendering Option Version 3 (活文 PDFstaff 編)(3020-3-N49)

PDFstaff Runtime および PDFstaff SDK と連携した DocumentBroker Rendering Option の機能、環境設定、およびコマン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 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)

- スケーラブルデータベースサーバ HiRDB Version 6 SQL リファレンス (UNIX(R)/Windows(R) 用) (3000-6-237) ¹
- スケーラブルデータベースサーバ HiRDB Version 6 SQL リファレンス (Windows(R) 用) (3020-6-127) ¹
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (UNIX(R)/Windows(R) 用) (3000-6-277) ¹
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (Windows(R) 用) (3020-6-277) ¹
- スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス (3020-6-357) ¹
- 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) ²

- 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) ²

関連製品のマニュアル (その他)

- VisiBroker Version 3 VisiBroker for C++ プログラマーズガイド (3000-3-651) ³
- VisiBroker Version 3 VisiBroker for C++ プログラマーズガイド (Windows(R) 用)(3000-3-678) ³

注 1

このマニュアルでは、これらのマニュアルを「HiRDB SQL リファレンス」と表記しています。

注 2

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。

注 3

このマニュアルでは、これらのマニュアルを「VisiBroker for C++ プログラマーズガイド」と表記していません。

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。次に示す表を参考にして、お読みになる章を選択してください。

このマニュアルを読む目的		記述箇所
どんなクラスやメソッド、関数があるのかを知りたい。	クラスライブラリのクラス、メソッドおよび関数	1 章
	クラス一覧	1.1 節
	メソッド一覧	1.2 節
	関数一覧	1.3 節
クラスライブラリで使用するデータ型やリファレンスマニュアルの記述形式を知りたい。	クラスライブラリで使用するデータ型とクラスおよびメソッドの記述形式	2 章
	クラスライブラリで使用するデータ型	2.1 節
	クラスおよびメソッドの記述形式	2.2 節
各クラスの詳細とメソッドの文法について知りたい。	CdbrCompound クラス (複合データを扱うクラス)	3 章
	CdbrConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)	4 章
	CdbrContainable クラス (包含要素を扱うクラス)	5 章
	CdbrCore クラス (エラー情報を取得するクラス)	6 章
	CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)	7 章
	CdbrDocument クラス (バージョンなし文書を扱うクラス)	8 章
	CdbrEqlStatement クラス (edmSQL 文を操作、実行するクラス)	9 章
	CdbrIndependentPersistence クラス (独立データを扱うクラス)	10 章
	CdbrPublicACL クラス (パブリック ACL を扱うクラス)	11 章
	CdbrReferentialContainer クラス (バージョンなしコンテナを扱うクラス)	12 章
	CdbrSession クラス (セッションを確立するクラス)	13 章
CdbrVariableArray クラス (可変長配列を扱うクラス)	14 章	

このマニュアルを読む目的		記述箇所
	CdbrVersionable クラス (バージョン管理機能を提供するクラス)	15 章
	CdbrVersionableDocument クラス (バージョン付き文書を扱うクラス)	16 章
	CdbrVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)	17 章
	CdbrXmlTranslator クラス (XML 変換を扱うクラス)	18 章
	CdbrXmlTranslatorFactory クラス (XML の操作環境を管理するクラス)	19 章
関数の詳細を知りたい。	関数	20 章
コーディング例を見たい。	コーディング例	付録 A
	作成するコーディング例	A.1
	コーディング例	A.2
DMA オブジェクトのプロパティについて知りたい。	DMA オブジェクトのプロパティ	付録 B
	DMA のクラスのプロパティ一覧	B.1
	プロパティの説明	B.2
メソッドで使用する構造体について知りたい。	クラスライブラリのメソッドで使用する構造体	付録 C
	オブジェクトの情報を設定, 参照する構造体	C.1
	検索に使用する構造体	C.2
	構造体で指定する定数	C.3
メソッドの実行に必要なパーミッションについて知りたい。	メソッドの実行に必要なパーミッション (アクセス制御機能を使用する場合)	付録 D
	パーミッション一覧で使用する記号	D.1
	メソッドの実行に必要なパーミッション一覧	D.2
メソッド実行時の排他制御 (ロック) について知りたい。	メソッド実行時の排他制御 (ロック)	付録 E
	メソッド実行時の明示的なロックと暗黙のロック	E.1
	メソッド実行時のロック一覧	E.2
OIID 指定時の URL の文法について知りたい。	OIID 指定時の URL の文法	付録 F
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メソッドおよび関数について知りたい。	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メソッドと関数	付録 G
クラスライブラリの戻り値について知りたい。	戻り値一覧	付録 H
	クラスライブラリの戻り値	H.1
	DMA の戻り値	H.2
	クラスライブラリ独自の戻り値 (ERR_DBR の場合)	H.3
	データベースエラーが発生した場合の戻り値 (ERR_DB の場合)	H.4
	詳細メッセージの出力について	H.5
障害が発生した時に uCosminexus DocumentBroker が出力する情報について知りたい。	障害が発生した時に uCosminexus DocumentBroker が出力する情報	付録 I

このマニュアルを読む目的		記述箇所
uCosminexus DocumentBroker の用語を知りたい。	用語解説	付録 J

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
HiRDB/Single Server Version 7	
HiRDB/Single Server Version 8	
HiRDB/Parallel Server Version 6	
HiRDB/Parallel Server Version 7	
HiRDB/Parallel Server Version 8	
HiRDB Adapter for XML - Enterprise Edition	HiRDB Adapter for XML
HiRDB Adapter for XML - Standard Edition	
TPBroker Developer for C++	TPBroker
TPBroker for C++	
TPBroker	
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 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	

製品名称	略称
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 8
Windows(R) 8 Pro 日本語版 (32 ビット版)	
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)	Word
Microsoft(R) Office Word	
Microsoft(R) Word	
活文 PDFstaff Runtime	PDFstaff Runtime
活文 PDFstaff SDK	PDFstaff SDK

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。
- AIX を UNIX と表記することがあります。
- Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 R2 x64 Edition , Windows Server 2008 x86 , Windows Server 2008 R2 , 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

英略語	英字での表記
BOA	<u>B</u> asic <u>O</u> bject <u>A</u> dapter
CD-ROM	<u>C</u> ompact <u>D</u> isc <u>R</u> ead <u>O</u> nly <u>M</u> emory
CGI	<u>C</u> ommon <u>G</u> ateway <u>I</u> nterface
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CPU	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
CR	<u>C</u> arriage <u>R</u> eturn
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
DAP	<u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
DAT	<u>D</u> igital <u>A</u> udio <u>T</u> ape
DB	<u>D</u> atabase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DCD	<u>D</u> ocument <u>C</u> ontent <u>D</u> escription
DDE	<u>D</u> ynamic <u>D</u> ata <u>E</u> xchange
DIT	<u>D</u> irectory <u>I</u> nformation <u>T</u> ree
DLL	<u>D</u> ynamic <u>L</u> inking <u>L</u> ibrary
DMA	<u>D</u> ocument <u>M</u> anagement <u>A</u> lliance
DN	<u>D</u> istinguished <u>N</u> ame
EOF	<u>E</u> nd of <u>F</u> ile
ESIS-B	<u>E</u> lement <u>S</u> tructure <u>I</u> nformation <u>S</u> et- <u>B</u> inary Format
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
FAM	<u>F</u> ile <u>A</u> ccess <u>M</u> odule
GIF	<u>G</u> raphics <u>I</u> nterchange <u>F</u> ormat
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
GUID	<u>G</u> lobally <u>U</u> nique <u>I</u> dentifier
HTML	<u>H</u> ypertext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> ypertext <u>T</u> ransfer <u>P</u> rotocol
IANA	<u>I</u> nternet <u>A</u> ssigned <u>N</u> umbers <u>A</u> uthority
ID	<u>I</u> dentifier
IPF	<u>I</u> tanium(R) <u>P</u> rocessor <u>F</u> amily
ISO	<u>I</u> nternational <u>O</u> rganization for <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

英略語	英字での表記
OIID	<u>O</u> bject <u>I</u> nstance <u>I</u> dentifier
OLE	<u>O</u> bject <u>L</u> inking and <u>E</u> mbedding
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
ORDB	<u>O</u> bject <u>R</u> elational <u>D</u> atabase
OS	<u>O</u> perating <u>S</u> ystem
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
PC	<u>P</u> ersonal <u>C</u> omputer
PDF	<u>P</u> ortable <u>D</u> ocument <u>F</u> ormat
RDB	<u>R</u> elational <u>D</u> atabase
RDN	<u>R</u> elative <u>D</u> istinguished <u>N</u> ame
RFC	<u>R</u> equest for <u>C</u> omment
RTF	<u>R</u> ich <u>T</u> ext <u>F</u> ormat
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
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) [<u>A</u> B] 「何も指定しない」か「A または B を指定する」ことを示します。何も指定しない場合は A が仮定されます。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号 で区切られている場合は、そのうちの一つを選択します。 (例) { A B C } A, B または C のどれかを指定することを示します。

記号	意味
[]	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。

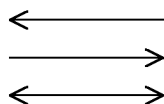
種類	定義
英字	A ~ Z a ~ z
英小文字	a ~ z
英大文字	A ~ Z
数字	0 ~ 9
英数字	A ~ Z a ~ z 0 ~ 9
記号	! " # \$ % & ' () + , _ . / : ; < = > @ [] ^ - { } タブ 空白

注 すべての半角文字を使用してください。

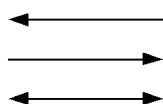
このマニュアルの図中で使用する記号

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。

●制御の流れ



●その他の流れ



●フォルダ、コンテナ



●文書



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), 必須(ひっす)

KB (キロバイト) などの単位表記について

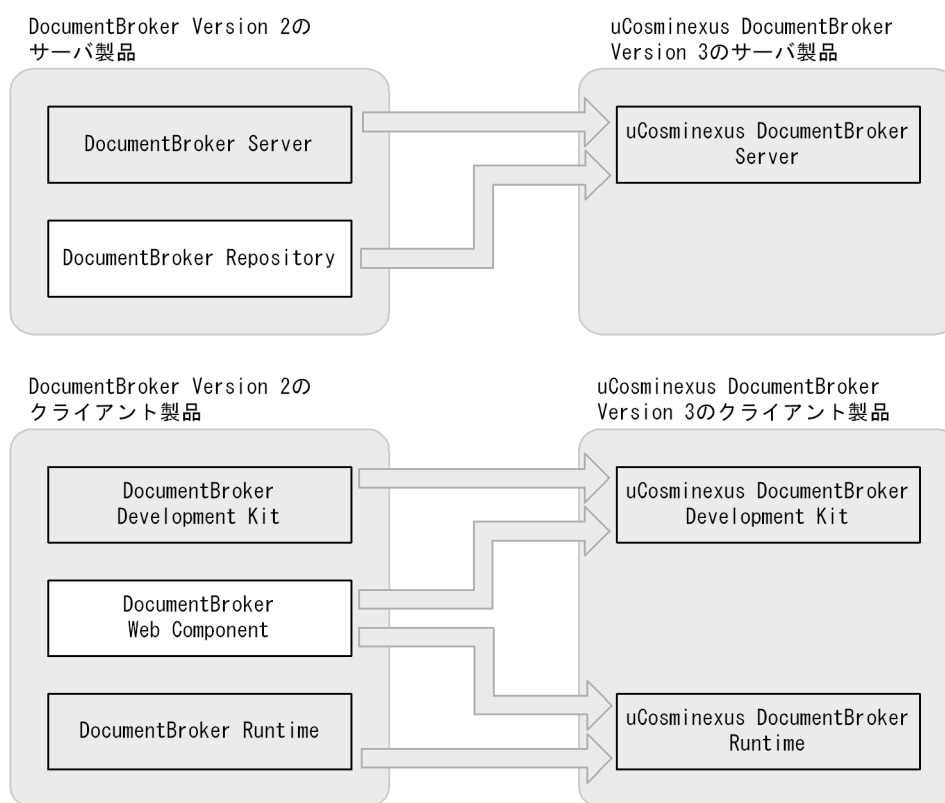
1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ 1,024 バイト, 1,024² バイト, 1,024³ バイト, 1,024⁴ バイトです。

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

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

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

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



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

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

Version 2 のマニュアル名称	Version 3 のマニュアル名称
DocumentBroker Version 2 システム導入・運用ガイド	uCosminexus DocumentBroker Version 3 システム導入・運用ガイド

Version 2 のマニュアル名称	Version 3 のマニュアル名称
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	クラスライブラリのクラス、メソッドおよび関数	1
1.1	クラス一覧	2
1.2	メソッド一覧	5
1.3	関数一覧	14
2	クラスライブラリで使用するデータ型とクラスおよびメソッドの記述形式	15
2.1	クラスライブラリで使用するデータ型	16
2.2	クラスおよびメソッドの記述形式	18
3	CdbrCompound クラス (複合データを扱うクラス)	21
	CdbrCompound クラスの詳細	22
	CdbrCompound::~CdbrCompound メソッド	23
	CdbrCompound::CdbrCompound メソッド	24
	CdbrCompound::GetCount メソッド	25
	CdbrCompound::GetDataType メソッド	26
	CdbrCompound::GetId メソッド	27
	CdbrCompound::GetValue メソッド	28
	CdbrCompound::SetValue メソッド	29
4	CdbrConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)	31
	CdbrConfiguredReferentialContainer クラスの詳細	33
	CdbrConfiguredReferentialContainer::~CdbrConfiguredReferentialContainer メソッド	39
	CdbrConfiguredReferentialContainer::CdbrConfiguredReferentialContainer メソッド	40
	CdbrConfiguredReferentialContainer::CreateObject メソッド	41
	CdbrConfiguredReferentialContainer::GetContainableList メソッド	45
	CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド	52
	CdbrConfiguredReferentialContainer::GetLinkPropertyValues メソッド	59
	CdbrConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッド	62
	CdbrConfiguredReferentialContainer::GetVTContaineelList メソッド	65
	CdbrConfiguredReferentialContainer::GetVTContaineelListAndLock メソッド	72
	CdbrConfiguredReferentialContainer::Link メソッド	79
	CdbrConfiguredReferentialContainer::LinkAndLock メソッド	82
	CdbrConfiguredReferentialContainer::LinkVTFix メソッド	86
	CdbrConfiguredReferentialContainer::LinkVTFixAndLock メソッド	89
	CdbrConfiguredReferentialContainer::LinkVTFloat メソッド	93

CdbrConfiguredReferentialContainer::LinkVTFloatAndLock メソッド	96
CdbrConfiguredReferentialContainer::PutLinkPropertyValues メソッド	99
CdbrConfiguredReferentialContainer::SetVTFix メソッド	101
CdbrConfiguredReferentialContainer::SetVTFloat メソッド	103
CdbrConfiguredReferentialContainer::Unlink メソッド	105
CdbrConfiguredReferentialContainer::UnlinkAndLock メソッド	107
CdbrConfiguredReferentialContainer::UnlinkVT メソッド	110
CdbrConfiguredReferentialContainer::UnlinkVTAndLock メソッド	112

5

CdbrContainable クラス (包含要素を扱うクラス)	115
CdbrContainable クラスの詳細	116
CdbrContainable::~~CdbrContainable メソッド	120
CdbrContainable::BindPublicACL メソッド	121
CdbrContainable::CdbrContainable メソッド	123
CdbrContainable::GetContainerList メソッド	124
CdbrContainable::GetContainerListAndLock メソッド	130
CdbrContainable::GetPublicACLList メソッド	137
CdbrContainable::GetPublicACLListAndLock メソッド	140
CdbrContainable::UnbindPublicACL メソッド	143

6

CdbrCore クラス (エラー情報を取得するクラス)	145
CdbrCore クラスの詳細	146
CdbrCore::~~CdbrCore メソッド	147
CdbrCore::CdbrCore メソッド	148
CdbrCore::GetDBError メソッド	149
CdbrCore::GetLastError メソッド	150

7

CdbrDMA クラス (OID の設定やプロパティを扱うクラス)	151
CdbrDMA クラスの詳細	152
CdbrDMA::~~CdbrDMA メソッド	154
CdbrDMA::CdbrDMA メソッド	155
CdbrDMA::ConnectObject メソッド	156
CdbrDMA::GetOID メソッド	158
CdbrDMA::GetPropertyValues メソッド	159
CdbrDMA::GetPropertyValuesAndLock メソッド	162
CdbrDMA::GetType メソッド	166
CdbrDMA::PutPropertyValues メソッド	168
CdbrDMA::ReleaseObject メソッド	172
CdbrDMA::RemoveObject メソッド	174

CdbrDMA::SetOID メソッド	177
----------------------	-----

8

CdbrDocument クラス (バージョンなし文書を扱うクラス)	179
CdbrDocument クラスの詳細	181
CdbrDocument::~CdbrDocument メソッド	187
CdbrDocument::AddRendition メソッド	188
CdbrDocument::CdbrDocument メソッド	195
CdbrDocument::ChangeMasterRendition メソッド	196
CdbrDocument::ConvertContentType メソッド	198
CdbrDocument::CreateIndex メソッド	203
CdbrDocument::CreateObject メソッド	206
CdbrDocument::CreateRelation メソッド	216
CdbrDocument::DeleteRendition メソッド	219
CdbrDocument::GetContent メソッド	221
CdbrDocument::GetContentAndLock メソッド	224
CdbrDocument::GetContentInfo メソッド	231
CdbrDocument::GetRelationList メソッド	234
CdbrDocument::GetRelationListAndLock メソッド	239
CdbrDocument::GetRenditionList メソッド	245
CdbrDocument::GetRenditionListAndLock メソッド	248
CdbrDocument::GetVersionableList メソッド	252
CdbrDocument::GetVersionableListAndLock メソッド	257
CdbrDocument::PutRelationPropertyValues メソッド	262
CdbrDocument::PutRenditionPropertyValues メソッド	265
CdbrDocument::RemoveIndex メソッド	268
CdbrDocument::RemoveRelation メソッド	270
CdbrDocument::UpdateContent メソッド	272
CdbrDocument::UpdateContentAndRenditionType メソッド	278

9

CdbrEqStatement クラス (edmSQL 文を操作, 実行するクラス)	287
CdbrEqStatement クラスの詳細	288
CdbrEqStatement::~CdbrEqStatement メソッド	290
CdbrEqStatement::CdbrEqStatement メソッド	291
CdbrEqStatement::ChangeACLMode メソッド	292
CdbrEqStatement::ChangeGetObjCount メソッド	294
CdbrEqStatement::ChangeLockType メソッド	295
CdbrEqStatement::DeleteResult メソッド	297
CdbrEqStatement::Execute メソッド	299
CdbrEqStatement::GetACLMode メソッド	302

CdbrEqlStatement::GetResult メソッド	304
CdbrEqlStatement::Initialize メソッド	307
CdbrEqlStatement::Set メソッド	308
CdbrEqlStatement::SetObjParam メソッド	310
CdbrEqlStatement::SetOIDParam メソッド	312
CdbrEqlStatement::SetParam メソッド	314
CdbrEqlStatement::Terminate メソッド	317

10 CdbrIndependentPersistence クラス (独立データを扱うクラス)	319
CdbrIndependentPersistence クラスの詳細	320
CdbrIndependentPersistence::~CdbrIndependentPersistence メソッド	324
CdbrIndependentPersistence::BindPublicACL メソッド	325
CdbrIndependentPersistence::CdbrIndependentPersistence メソッド	327
CdbrIndependentPersistence::CreateObject メソッド	328
CdbrIndependentPersistence::GetPublicACLList メソッド	332
CdbrIndependentPersistence::GetPublicACLListAndLock メソッド	335
CdbrIndependentPersistence::UnbindPublicACL メソッド	338

11 CdbrPublicACL クラス (パブリック ACL を扱うクラス)	341
CdbrPublicACL クラスの詳細	342
CdbrPublicACL::~CdbrPublicACL メソッド	345
CdbrPublicACL::CdbrPublicACL メソッド	346
CdbrPublicACL::CreateObject メソッド	347
CdbrPublicACL::GetBindObjectList メソッド	351
CdbrPublicACL::GetBindObjectListAndLock メソッド	356

12 CdbrReferentialContainer クラス (バージョンなしコンテナを扱うクラス)	361
CdbrReferentialContainer クラスの詳細	362
CdbrReferentialContainer::~CdbrReferentialContainer メソッド	366
CdbrReferentialContainer::CdbrReferentialContainer メソッド	367
CdbrReferentialContainer::CreateObject メソッド	368
CdbrReferentialContainer::GetContainableList メソッド	373
CdbrReferentialContainer::GetContainableListAndLock メソッド	379
CdbrReferentialContainer::GetLinkPropertyValues メソッド	386
CdbrReferentialContainer::GetLinkPropertyValuesAndLock メソッド	389
CdbrReferentialContainer::Link メソッド	392
CdbrReferentialContainer::LinkAndLock メソッド	395
CdbrReferentialContainer::PutLinkPropertyValues メソッド	399
CdbrReferentialContainer::Unlink メソッド	402

CdbrReferentialContainer::UnlinkAndLock メソッド	404
--	-----

13 CdbrSession クラス (セッションを確立するクラス) 407

CdbrSession クラスの詳細	409
CdbrSession::~CdbrSession メソッド	411
CdbrSession::Begin メソッド	412
CdbrSession::CdbrSession メソッド	413
CdbrSession::Commit メソッド	414
CdbrSession::Connect メソッド	415
CdbrSession::Disconnect メソッド	418
CdbrSession::GetClassType メソッド	419
CdbrSession::GetDocSpaceCharacterSet メソッド	421
CdbrSession::GetLastDetailError メソッド	422
CdbrSession::GetReferencePath メソッド	423
CdbrSession::GetUserInfo メソッド	424
CdbrSession::OrbComponentShutdown メソッド	427
CdbrSession::OrbComponentStart メソッド	429
CdbrSession::Rollback メソッド	431
CdbrSession::SetReferencePath メソッド	432

14 CdbrVariableArray クラス (可変長配列を扱うクラス) 435

CdbrVariableArray クラスの詳細	436
CdbrVariableArray::~CdbrVariableArray メソッド	437
CdbrVariableArray::Add メソッド	438
CdbrVariableArray::CdbrVariableArray メソッド	439
CdbrVariableArray::Delete メソッド	440
CdbrVariableArray::GetCount メソッド	441
CdbrVariableArray::GetData Type メソッド	442
CdbrVariableArray::GetValue メソッド	443
CdbrVariableArray::Resize メソッド	444
CdbrVariableArray::SetValue メソッド	445

15 CdbrVersionable クラス (バージョン管理機能を提供するクラス) 447

CdbrVersionable クラスの詳細	448
CdbrVersionable::~CdbrVersionable メソッド	452
CdbrVersionable::BindPublicACL メソッド	453
CdbrVersionable::CdbrVersionable メソッド	455
CdbrVersionable::DeleteVersion メソッド	456
CdbrVersionable::GetPropertyValues メソッド	459

CdbrVersionable::GetPropertyValuesAndLock メソッド	462
CdbrVersionable::GetPublicACLList メソッド	466
CdbrVersionable::GetPublicACLListAndLock メソッド	469
CdbrVersionable::GetReservationStatus メソッド	472
CdbrVersionable::GetReservationStatusAndLock メソッド	474
CdbrVersionable::GetVersionList メソッド	477
CdbrVersionable::GetVersionListAndLock メソッド	481
CdbrVersionable::PutPropertyValues メソッド	486
CdbrVersionable::UnbindPublicACL メソッド	490
CdbrVersionable::VersionCheckIn メソッド	493
CdbrVersionable::VersionCheckOut メソッド	495
CdbrVersionable::VersionRevoke メソッド	498

16 CdbrVersionableDocument クラス (バージョン付き文書を扱うクラス)	501
CdbrVersionableDocument クラスの詳細	503
CdbrVersionableDocument::~CdbrVersionableDocument メソッド	509
CdbrVersionableDocument::AddRendition メソッド	510
CdbrVersionableDocument::CdbrVersionableDocument メソッド	517
CdbrVersionableDocument::ChangeMasterRendition メソッド	518
CdbrVersionableDocument::ConvertContentType メソッド	521
CdbrVersionableDocument::CreateIndex メソッド	527
CdbrVersionableDocument::CreateObject メソッド	530
CdbrVersionableDocument::CreateRelation メソッド	540
CdbrVersionableDocument::DeleteRendition メソッド	543
CdbrVersionableDocument::GetContent メソッド	545
CdbrVersionableDocument::GetContentAndLock メソッド	548
CdbrVersionableDocument::GetContentInfo メソッド	555
CdbrVersionableDocument::GetRelationList メソッド	558
CdbrVersionableDocument::GetRelationListAndLock メソッド	563
CdbrVersionableDocument::GetRenditionList メソッド	569
CdbrVersionableDocument::GetRenditionListAndLock メソッド	573
CdbrVersionableDocument::PutRelationPropertyValues メソッド	577
CdbrVersionableDocument::PutRenditionPropertyValues メソッド	580
CdbrVersionableDocument::RemoveIndex メソッド	584
CdbrVersionableDocument::RemoveRelation メソッド	586
CdbrVersionableDocument::UpdateContent メソッド	588
CdbrVersionableDocument::UpdateContentAndRenditionType メソッド	594

17	CdbrVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)	603
	CdbrVersionTraceableContainer クラスの詳細	605
	CdbrVersionTraceableContainer::~CdbrVersionTraceableContainer メソッド	610
	CdbrVersionTraceableContainer::CdbrVersionTraceableContainer メソッド	611
	CdbrVersionTraceableContainer::CreateObject メソッド	612
	CdbrVersionTraceableContainer::GetVersionableList メソッド	616
	CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド	621
	CdbrVersionTraceableContainer::GetVTContaineelist メソッド	626
	CdbrVersionTraceableContainer::GetVTContaineelistAndLock メソッド	633
	CdbrVersionTraceableContainer::LinkVTFix メソッド	640
	CdbrVersionTraceableContainer::LinkVTFixAndLock メソッド	643
	CdbrVersionTraceableContainer::LinkVTFloat メソッド	647
	CdbrVersionTraceableContainer::LinkVTFloatAndLock メソッド	650
	CdbrVersionTraceableContainer::SetVTFix メソッド	653
	CdbrVersionTraceableContainer::SetVTFixAndLock メソッド	655
	CdbrVersionTraceableContainer::SetVTFloat メソッド	658
	CdbrVersionTraceableContainer::SetVTFloatAndLock メソッド	660
	CdbrVersionTraceableContainer::UnlinkVT メソッド	663
	CdbrVersionTraceableContainer::UnlinkVTAndLock メソッド	665
18	CdbrXmlTranslator クラス (XML 変換を扱うクラス)	667
	CdbrXmlTranslator クラスの詳細	668
	CdbrXmlTranslator::~CdbrXmlTranslator メソッド	670
	CdbrXmlTranslator::CdbrXmlTranslator メソッド	671
	CdbrXmlTranslator::GetDmaInfoList メソッド	672
	CdbrXmlTranslator::GetIndexData メソッド	676
19	CdbrXmlTranslatorFactory クラス (XML の操作環境を管理するクラス)	679
	CdbrXmlTranslatorFactory クラスの詳細	680
	CdbrXmlTranslatorFactory::~CdbrXmlTranslatorFactory メソッド	681
	CdbrXmlTranslatorFactory::CdbrXmlTranslatorFactory メソッド	682
	CdbrXmlTranslatorFactory::CreateTranslator メソッド	683
	CdbrXmlTranslatorFactory::Initialize メソッド	684
20	関数	687
	関数	688
	dbrDelete 関数	689

dbrDmaldFromString 関数	691
-----------------------	-----

付録	693
付録 A コーディング例	694
付録 A.1 作成するコーディング例	694
付録 A.2 コーディング例	696
付録 B DMA オブジェクトのプロパティ	722
付録 B.1 DMA のクラスのプロパティ一覧	722
付録 B.2 プロパティの説明	735
付録 C クラスライブラリのメソッドで使用する構造体	742
付録 C.1 オブジェクトの情報を設定, 参照する構造体	742
付録 C.2 検索に使用する構造体	767
付録 C.3 構造体で指定する定数	768
付録 D メソッドの実行に必要なパーミッション (アクセス制御機能を使用する場合)	770
付録 D.1 パーミッション一覧で使用する記号	770
付録 D.2 メソッドの実行に必要なパーミッション一覧	770
付録 E メソッド実行時の排他制御 (ロック)	783
付録 E.1 メソッド実行時の明示的なロックと暗黙のロック	783
付録 E.2 メソッド実行時のロッカー一覧	783
付録 F OIID 指定時の URL の文法	793
付録 G 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メソッドと関数	795
付録 G.1 文字コード種別が UTF-8 の場合に使用できるメソッド	795
付録 G.2 文字コード種別が UTF-8 の場合に使用できる関数	800
付録 H 戻り値一覧	801
付録 H.1 クラスライブラリの戻り値	801
付録 H.2 DMA の戻り値	801
付録 H.3 クラスライブラリ独自の戻り値 (ERR_DBR の場合)	806
付録 H.4 データベースエラーが発生した場合の戻り値 (ERR_DB の場合)	812
付録 H.5 詳細メッセージの出力について	812
付録 I 障害が発生した時に DocumentBroker が出力する情報	815
付録 J 用語解説	817

索引	833
-----------	-----

1

クラスライブラリのクラス，メソッドおよび関数

クラスライブラリのクラス，メソッドおよび関数を一覧形式で示します。

1.1 クラス一覧

1.2 メソッド一覧

1.3 関数一覧

1.1 クラス一覧

クラスライブラリで提供するクラスの一覧を次の表に示します。なお、それぞれのクラスの使用方法については、参照先欄に示す箇所（マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」の章・節・項番号と標題）で説明しています。

表 1-1 クラスライブラリで提供するクラス一覧

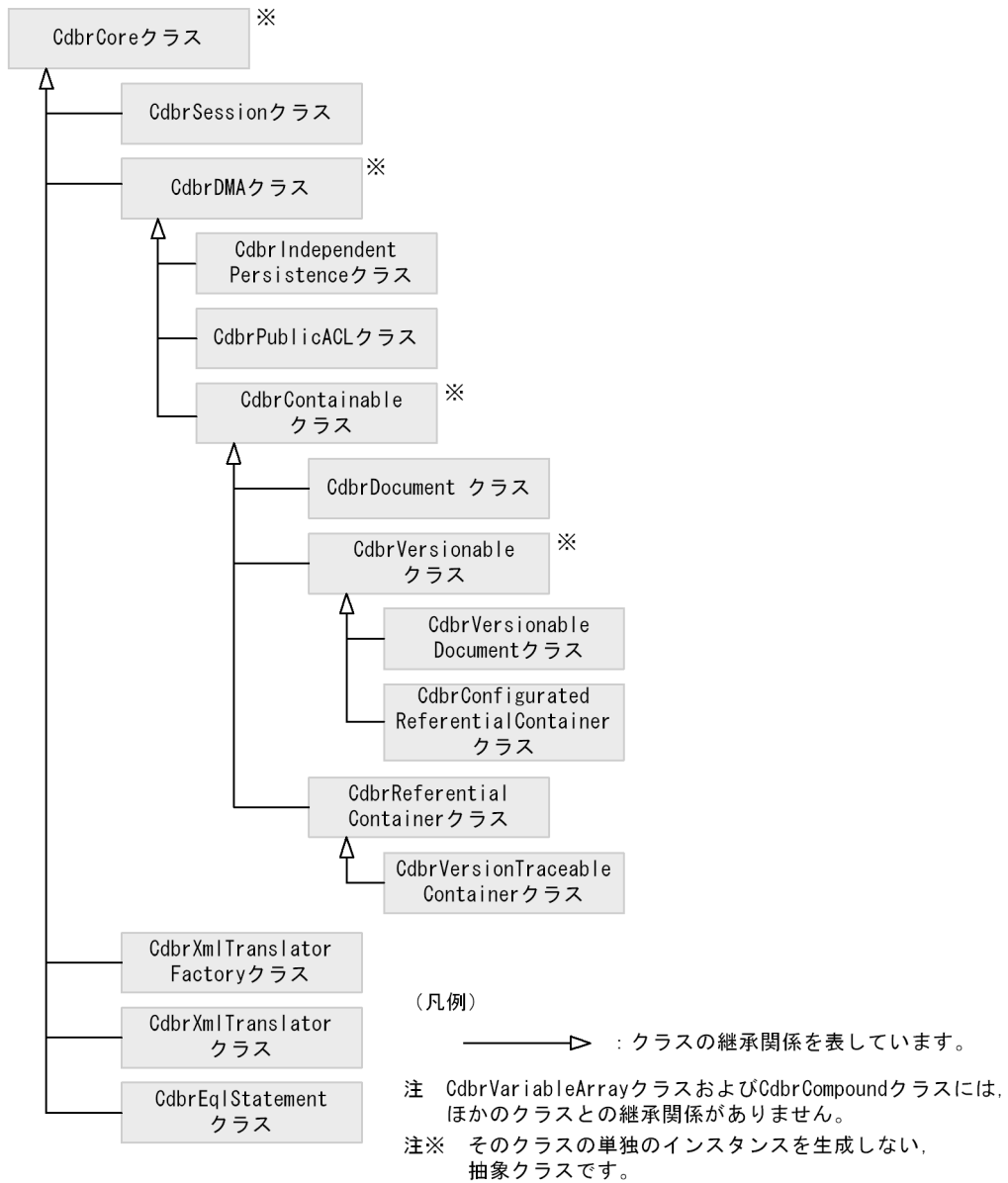
クラス名	説明	「DocumentBroker Version 3 クラスライブラリ C++ 解説」の参照先
文書空間と接続するためのクラス		
CdbrSession クラス	クライアント実行環境と文書空間との接続を管理する機能を提供するクラスです。また、ログイン管理もします。	2.3 文書空間との接続（セッションの確立）
文書を作成、管理するためのクラス		
CdbrVersionableDocument クラス	DocumentBroker で扱う文書のうち、バージョン管理をする文書を作成および管理する機能を提供するクラスです。	3.2 クラスライブラリで扱う文書
CdbrDocument クラス	DocumentBroker で扱う文書のうち、バージョン管理をしない文書や、バージョン管理をしている文書の個々のバージョンに当たる文書を作成および管理する機能を提供するクラスです。	
複数の文書をまとめて管理するためのクラス		
CdbrReferentialContainer クラス	文書をコンテナによって包含して管理する機能を提供するクラスです。	3.9 コンテナを使用した文書管理
文書の構成管理をするためのクラス		
CdbrConfiguratedReferentialContainer クラス	コンテナを使用して文書やコンテナのバージョン構成を管理したり、コンテナ自身のバージョンを管理したりする機能を提供するクラスです。	3.10 構成管理コンテナを使用した文書の構成管理
CdbrVersionTraceableContainer クラス	コンテナを使用して文書やコンテナのバージョン構成を管理する機能を提供するクラスです。また、バージョン管理しているコンテナの、個々のバージョンに当たるコンテナを操作する機能を提供します。	
XML 文書を管理するためのクラス		
CdbrXmlTranslator クラス	XML プロパティマッピングに使用する定義ファイル類を解析して、XML 文書管理機能を実行するための環境を管理する機能を提供するクラスです。	3.11 XML 文書の管理
CdbrXmlTranslatorFactory クラス	XML 文書管理を実行する機能を提供するクラスです。	
独立データを管理するためのクラス		

クラス名	説明	「DocumentBroker Version 3 クラスライブラリ C++ 解 説」の 参照先
CdbrIndependentPersistence クラス	独立したデータを扱う機能を提供するクラスです。	3.13 独立データの管理
文書やコンテナを検索するためのクラス		
CdbrEqlStatement クラス	edmSQLを基にした検索を実行する機能や、検索結果を取得する機能を提供するクラスです。	4. オブジェクトの検索
可変長な一次元配列を扱うためのクラス		
CdbrCompound クラス	異なる型の複数のデータから構成される複合データを扱う機能を提供するクラスです。	2.6.7 VariableArray 型プロパティの操作
CdbrVariableArray クラス	同じ型の複数の要素から構成される可変長の配列を扱う機能を提供するクラスです。なお、このクラスで扱うことができる同じ型の要素とは、CdbrCompound クラスを基に作成された CdbrCompound オブジェクトだけです。	
アクセス制御リストを管理するためのクラス		
CdbrPublicACL クラス	アクセス制御リストを管理して、これを複数の文書やコンテナで共有して使用する機能を提供するクラスです。	3.15 アクセス制御
サブクラスに共通な機能を継承させるためのクラス（抽象クラス）		
CdbrContainable クラス	コンテナの包含要素になる機能を提供するクラスです。	2.5.4 抽象クラス
CdbrCore クラス	クラスライブラリのクラスのうち、CdbrVariableArray クラスおよび CdbrCompound クラス以外のすべてのクラスのスーパークラスです。エラー管理機能を持ちます。	
CdbrDMA クラス	DMA のオブジェクトモデルの dmaClass_DMA クラスに対応するクラスです。OID の取得およびプロパティの参照や設定をするメソッドを定義します。	
CdbrVersionable クラス	バージョン管理機能を提供するクラスです。	

クラスライブラリで提供しているこれらのクラスには継承関係があり、上位にあるクラスで定義したプロパティやメソッドは下位のクラスに継承されます。クラスの継承関係を、次の図に示します。

1. クラスライブラリのクラス、メソッドおよび関数

図 1-1 クラスの継承関係



1.2 メソッド一覧

この節では、クラスライブラリの各クラスで提供するメソッドの一覧を示します。

メソッドは、上位のクラスで定義されたものがサブクラスに継承されます。例えば、CdbrDocument クラスでは、CdbrDocument クラスで定義されているメソッドのほか、スーパークラスである CdbrCore クラス、CdbrDMA クラスおよび CdbrContainable クラスで定義されているメソッドも使用できます。

なお、クラスおよびメソッドは、アルファベット順に説明します。

(1) CdbrCompound クラス

CdbrCompound クラスのメソッド一覧を次の表に示します。

表 1-2 CdbrCompound クラスのメソッド一覧

メソッド	機能
~CdbrCompound	デストラクタ
CdbrCompound	コンストラクタ
GetCount	要素数の取得
GetDataType	要素の型の取得
GetId	要素 ID の取得
GetValue	値の取得
SetValue	値の設定

(2) CdbrConfiguredReferentialContainer クラス

CdbrConfiguredReferentialContainer クラスのメソッド一覧を次の表に示します。

表 1-3 CdbrConfiguredReferentialContainer クラスのメソッド一覧

メソッド	機能
~CdbrConfiguredReferentialContainer	デストラクタ
CdbrConfiguredReferentialContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetContainableList	包含している Containable オブジェクトの一覧取得
GetContainableListAndLock	包含している Containable オブジェクトの一覧取得と同時に指定したロックの設定
GetLinkPropertyValues	リンクのプロパティの取得
GetLinkPropertyValuesAndLock	リンクのプロパティの取得と同時に指定したロックの設定
GetVTContaineelist	構成管理しているオブジェクトの一覧取得
GetVTContaineelistAndLock	構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定
Link	Containable オブジェクトとの関連づけ
LinkAndLock	Containable オブジェクトとの関連づけと同時に指定したロックの設定
LinkVTFix	構成管理するオブジェクトの FIX モードでの関連づけ

1. クラスライブラリのクラス、メソッドおよび関数

メソッド	機能
LinkVTFixAndLock	構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定
LinkVTFloat	構成管理するオブジェクトの FLOATING モードでの関連づけ
LinkVTFloatAndLock	構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定
PutLinkPropertyValues	リンクのプロパティの設定
SetVTFix	構成要素のバージョンの固定
SetVTFloat	構成要素のバージョン固定の解除
Unlink	Containable オブジェクトとの関連づけの解除
UnlinkAndLock	Containable オブジェクトとの関連づけの解除と同時に指定したロックの設定
UnlinkVT	構成管理しているオブジェクトとの関連づけの解除
UnlinkVTAndLock	構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

(3) CdbrContainable クラス

CdbrContainable クラスのメソッド一覧を次の表に示します。

表 1-4 CdbrContainable クラスのメソッド一覧

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

(4) CdbrCore クラス

CdbrCore クラスのメソッド一覧を次の表に示します。

表 1-5 CdbrCore クラスのメソッド一覧

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

(5) CdbrDMA クラス

CdbrDMA クラスのメソッド一覧を次の表に示します。

表 1-6 CdbrDMA クラスのメソッド一覧

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

(6) CdbrDocument クラス

CdbrDocument クラスのメソッド一覧を次の表に示します。

表 1-7 CdbrDocument クラスのメソッド一覧

メソッド	機能
~CdbrDocument	デストラクタ
AddRendition	レンディションの追加
CdbrDocument	コンストラクタ
ChangeMasterRendition	マスタレンディションへの変更
ConvertContentType	コンテンツ種別の変換
CreateIndex	全文検索インデックスの作成
CreateObject	オブジェクトの新規作成
CreateRelation	文書間リレーションの設定
DeleteRendition	レンディションの削除
GetContent	ファイルの取得
GetContentAndLock	ファイルの取得と同時に指定したロックの設定
GetContentInfo	マルチファイル文書の情報の取得
GetRelationList	文書のリレーション情報の一覧取得
GetRelationListAndLock	文書のリレーション情報の一覧取得と同時に指定したロックの設定
GetRenditionList	レンディション一覧の取得
GetRenditionListAndLock	レンディション一覧の取得と同時に指定したロックの設定
GetVersionableList	バージョン管理情報一覧の取得
GetVersionableListAndLock	バージョン管理情報一覧の取得と同時に指定したロックの設定

1. クラスライブラリのクラス、メソッドおよび関数

メソッド	機能
PutRelationPropertyValues	文書間リレーションのプロパティの設定
PutRenditionPropertyValues	レンディションのプロパティの設定
RemoveIndex	全文検索インデックスの削除
RemoveRelation	文書間リレーションの解除
UpdateContent	ファイルの更新
UpdateContentAndRenditionType	ファイルの更新および RenditionType の変更

(7) CdbrEqIStatement クラス

CdbrEqIStatement クラスのメソッド一覧を次の表に示します。

表 1-8 CdbrEqIStatement クラスのメソッド一覧

メソッド	機能
~CdbrEqIStatement	デストラクタ
CdbrEqIStatement	コンストラクタ
ChangeACLMode	アクセス制御モードの変更
ChangeGetObjCount	検索結果の取得件数の変更
ChangeLockType	ロック種別の変更
DeleteResult	検索結果の削除
Execute	設定済み edmSQL 文の実行
GetACLMode	アクセス制御モードの取得
GetResult	検索結果の取得
Initialize	オブジェクトの初期化とセッションオブジェクトへの登録
Set	edmSQL 文の設定
SetObjParam	edmSQL 文の ? パラメタ (DMA オブジェクト) の設定
SetOIIDParam	edmSQL 文の ? パラメタ (OIID) の設定
SetParam	edmSQL 文の ? パラメタの設定
Terminate	セッションオブジェクトへの登録抹消

(8) CdbrIndependentPersistence クラス

CdbrIndependentPersistence クラスのメソッド一覧を次の表に示します。

表 1-9 CdbrIndependentPersistence クラスのメソッド一覧

メソッド	機能
~CdbrIndependentPersistence	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrIndependentPersistence	コンストラクタ
CreateObject	オブジェクトの新規作成
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定

メソッド	機能
UnbindPublicACL	パブリック ACL のアンバインド

(9) CdbrPublicACL クラス

CdbrPublicACL クラスのメソッド一覧を次の表に示します。

表 1-10 CdbrPublicACL クラスのメソッド一覧

メソッド	機能
~CdbrPublicACL	デストラクタ
CdbrPublicACL	コンストラクタ
CreateObject	パブリック ACL の新規作成
GetBindObjectList	バインドしているオブジェクトの一覧情報の取得
GetBindObjectListAndLock	バインドしているオブジェクトの一覧情報の取得と同時に指定したロックの設定

(10) CdbrReferentialContainer クラス

CdbrReferentialContainer クラスのメソッド一覧を次の表に示します。

表 1-11 CdbrReferentialContainer クラスのメソッド一覧

メソッド	機能
~CdbrReferentialContainer	デストラクタ
CdbrReferentialContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetContainableList	包含要素の一覧取得
GetContainableListAndLock	包含要素の一覧取得と同時に指定したロックの設定
GetLinkPropertyValues	リンクのプロパティの取得
GetLinkPropertyValuesAndLock	リンクのプロパティの取得と同時に指定したロックの設定
Link	関連づけの設定
LinkAndLock	関連づけの設定と同時に指定したロックの設定
PutLinkPropertyValues	リンクのプロパティの設定
Unlink	関連づけの解除
UnlinkAndLock	関連づけの解除と同時に指定したロックの設定

(11) CdbrSession クラス

CdbrSession クラスのメソッド一覧を次の表に示します。

表 1-12 CdbrSession クラスのメソッド一覧

メソッド	機能
~CdbrSession	デストラクタ
Begin	トランザクションの開始
CdbrSession	コンストラクタ

1. クラスライブラリのクラス、メソッドおよび関数

メソッド	機能
Commit	永続オブジェクトの確定
Connect	DocSpace との接続
Disconnect	セッションの切断
GetClassType	指定した OIID のクラスの判定
GetDocSpaceCharacterSet	接続する文書空間の文字コード種別の取得
GetLastDetailError	詳細エラー情報の取得
GetReferencePath	コンテンツ格納先ベースパスの取得
GetUserInfo	ユーザ情報の取得
OrbComponentShutdown	TPBroker の ORB コンポーネントの終了
OrbComponentStart	TPBroker の ORB コンポーネントの開始
Rollback	最後に CdbrSession::Commit メソッドがコールされた状態まで戻す
SetReferencePath	コンテンツ格納先ベースパスの設定

(12) CdbrVariableArray クラス

CdbrVariableArray クラスのメソッド一覧を次の表に示します。

表 1-13 CdbrVariableArray クラスのメソッド一覧

メソッド	機能
~CdbrVariableArray	デストラクタ
Add	要素の追加
CdbrVariableArray	コンストラクタ
Delete	要素の削除
GetCount	要素数の取得
GetDataType	要素の型の取得
GetValue	値の取得
Resize	要素数の変更
SetValue	値の設定

(13) CdbrVersionable クラス

CdbrVersionable クラスのメソッド一覧を次の表に示します。

表 1-14 CdbrVersionable クラスのメソッド一覧

メソッド	機能
~CdbrVersionable	デストラクタ
BindPublicACL	パブリック ACL のバインド (バージョン指定あり)
CdbrVersionable	コンストラクタ
DeleteVersion	バージョンの削除
GetPropertyValues	プロパティの取得 (バージョン指定あり)
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定 (バージョン指定あり)

メソッド	機能
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得 (バージョン指定あり)
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定 (バージョン指定あり)
GetReservationStatus	チェックアウト状態の取得
GetReservationStatusAndLock	チェックアウト状態の取得と同時に指定したロックの設定
GetVersionList	バージョン情報一覧の取得
GetVersionListAndLock	バージョン情報一覧の取得と同時に指定したロックの設定
PutPropertyValues	プロパティの設定 (バージョン指定あり)
UnbindPublicACL	パブリック ACL のアンバインド (バージョン指定あり)
VersionCheckIn	バージョンのチェックイン
VersionCheckOut	バージョンのチェックアウト
VersionRevoke	チェックアウトの取り消し

(14) CdbVersionableDocument クラス

CdbVersionableDocument クラスのメソッド一覧を次の表に示します。

表 1-15 CdbVersionableDocument クラスのメソッド一覧

メソッド	機能
~CdbVersionableDocument	デストラクタ
AddRendition	レンディションの追加
CdbVersionableDocument	コンストラクタ
ChangeMasterRendition	マスタレンディションへの変更
ConvertContentType	コンテンツ種別の変換
CreateIndex	全文検索インデックスの作成
CreateObject	オブジェクトの新規作成
CreateRelation	文書間リレーションの設定
DeleteRendition	レンディションの削除
GetContent	ファイルの取得
GetContentAndLock	ファイルの取得と同時に指定したロックの設定
GetContentInfo	マルチファイル文書の情報の取得
GetRelationList	文書のリレーション情報の一覧取得
GetRelationListAndLock	文書のリレーション情報の一覧取得と同時に指定したロックの設定
GetRenditionList	レンディション一覧の取得
GetRenditionListAndLock	レンディション一覧の取得と同時に指定したロックの設定
PutRelationPropertyValues	文書間リレーションのプロパティの設定
PutRenditionPropertyValues	レンディションのプロパティの設定
RemoveIndex	全文検索インデックスの削除
RemoveRelation	文書間リレーションの解除
UpdateContent	ファイルの更新

1. クラスライブラリのクラス、メソッドおよび関数

メソッド	機能
UpdateContentAndRenditionType	ファイルの更新および RenditionType の変更

(15) CdbVersionTraceableContainer クラス

CdbVersionTraceableContainer クラスのメソッド一覧を次の表に示します。

表 1-16 CdbVersionTraceableContainer クラスのメソッド一覧

メソッド	機能
~CdbVersionTraceableContainer	デストラクタ
CdbVersionTraceableContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetVersionableList	バージョン管理情報一覧の取得
GetVersionableListAndLock	バージョン管理情報一覧の取得と同時に指定したロックの設定
GetVTContaineeList	構成管理しているオブジェクトの一覧取得
GetVTContaineeListAndLock	構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定
LinkVTFix	構成管理するオブジェクトの FIX モードでの関連づけ
LinkVTFixAndLock	構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定
LinkVTFloat	構成管理するオブジェクトの FLOATING モードでの関連づけ
LinkVTFloatAndLock	構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定
SetVTFix	構成要素のバージョンの固定
SetVTFixAndLock	構成要素のバージョンの固定と同時に指定したロックの設定
SetVTFloat	構成要素のバージョン固定の解除
SetVTFloatAndLock	構成要素のバージョン固定の解除と同時に指定したロックの設定
UnlinkVT	構成管理しているオブジェクトとの関連づけの解除
UnlinkVTAndLock	構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

(16) CdbXmlTranslator クラス

CdbXmlTranslator クラスのメソッド一覧を次の表に示します。

表 1-17 CdbXmlTranslator クラスのメソッド一覧

メソッド	機能
~CdbXmlTranslator	デストラクタ
CdbXmlTranslator	コンストラクタ
GetDmaInfoList	DMA 情報の取得
GetIndexData	検索インデクス作成用データの取得

(17) CdbXmlTranslatorFactory クラス

CdbXmlTranslatorFactory クラスのメソッド一覧を次の表に示します。

表 1-18 CdbXmlTranslatorFactory クラスのメソッド一覧

メソッド	機能
~CdbXmlTranslatorFactory	デストラクタ
CdbXmlTranslatorFactory	コンストラクタ
CreateTranslator	CdbXmlTranslator オブジェクトの生成
Initialize	オブジェクトの初期化

1.3 関数一覧

クラスライブラリで提供する関数の一覧を次の表に示します。

表 1-19 クラスライブラリで提供する関数の一覧

関数名	説明
dbrDelete	メモリの解放
dbrDmaIdFromString	文字列から DmaId 型への変換

2

クラスライブラリで使用するデータ型とクラスおよびメソッドの記述形式

クラスライブラリで使用するデータ型と、このマニュアルでのクラスおよびメソッドの記述形式について説明します。

2.1 クラスライブラリで使用するデータ型

2.2 クラスおよびメソッドの記述形式

2.1 クラスタイプライリで使用されるデータ型

データ型は DMA1.0 の定義に従います。このマニュアルで使用される型の定義を次の表に示します。

表 2-1 使用するデータ型

名前	定義	備考
DmaBoolean	unsigned short	-
DmaId	struct DmaId { DmaUInteger32 Data1; DmaUInteger16 Data2; DmaUInteger16 Data3; DmaUInteger8 Data4[8]; };	GUID (Globally Unique Identifiers)
DmaInteger8	signed char	-
DmaInteger16	short	-
DmaInteger32	long	-
Dmapv	void *	-
DmaTChar	char	-
DmaUInteger8	unsigned char	-
DmaUInteger16	unsigned short	-
DmaUInteger32	unsigned long	-
pDmaBinary	pDmaUInteger8	DmaBinary はない
pDmaBoolean	DmaBoolean *	-
pDmaId	DmaId *	-
pDmaInteger8	DmaInteger8 *	-
pDmaInteger16	DmaInteger16 *	-
pDmaInteger32	DmaInteger32 *	-
pDmapv	Dmapv *	-
pDmaString_T	DmaTChar*	DmaString_T はない
pDmaUInteger8	DmaUInteger8 *	-
pDmaUInteger16	DmaUInteger16 *	-
pDmaUInteger32	DmaUInteger32 *	-
ppDmaBinary	pDmaBinary *	-
ppDmaBoolean	pDmaBoolean *	-
ppDmaId	pDmaId *	-
ppDmaInteger8	pDmaInteger8 *	-
ppDmaInteger16	pDmaInteger16 *	-
ppDmaInteger32	pDmaInteger32 *	-
ppDmaString_T	pDmaString_T*	-
ppDmaUInteger8	pDmaUInteger8 *	-
ppDmaUInteger16	pDmaUInteger16 *	-
ppDmaUInteger32	pDmaUInteger32 *	-

(凡例)

- : なし

注 p はポインタ , pp はポインタへのポインタを意味します。

2.2 クラスおよびメソッドの記述形式

この節では、それぞれのクラスおよびメソッドで説明する項目の記述形式について示します。

各クラスの説明では、そのクラスで定義されるメソッドについてだけ説明しています。上位のクラスで定義されているメソッドについては、定義元のクラスを参照してください。

クラスで説明する項目

クラスの説明では、次の項目について説明しています。

スーパークラス

最上位からそのクラスまでの継承関係とともにスーパークラスを概念図で示しています。

プロパティ一覧

クラスライブラリで定義されているプロパティ一覧を記述しています。

プロパティ一覧の例を次の表に示します。

表 2-2 プロパティ一覧の例

プロパティ名	プロパティ識別子	データ型	読み取り専用
X	dbrProp_X	Integer32	

プロパティ名

プロパティの名称を記述しています。

プロパティ識別子

プロパティのシステム ID を記述しています。

データ型

プロパティのデータ型について記述しています。

読み取り専用

読み取り専用のプロパティか編集できるプロパティかを記述しています。

：読み取り専用のプロパティであることを示します。

-：編集できるプロパティであることを示します。

プロパティ詳細

各プロパティ値の詳細について説明しています。上位のクラスから継承されたプロパティについては、プロパティ名だけを記述しています。上位のクラスから継承されたプロパティの値の詳細については、継承元のプロパティ詳細を参照してください。

なお、DMA で規定されているプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

上位のクラスから継承されるメソッドおよびそのクラスで導入されるメソッドを、クラス別に表形式で示しています。

クラスの説明に続けて、そのクラスで導入されるメソッドをメソッド名のアルファベット順に説明しています。上位のクラスから継承されるメソッドの詳細については、継承元クラスのメソッドの説明を参照してください。

メソッドで説明する項目

メソッドの説明では、次の項目について説明しています。

機能

メソッドの概要を説明しています。

形式

メソッドの形式を説明しています。

詳細

メソッドの機能の詳細について説明しています。

引数

引数について次のように説明しています。

引数（入力）

指定する値について説明しています。

引数（出力）

指定した領域に設定される値について説明しています。

- 引数は「形式」で記述した文字列です。
- 省略可能であることを明記していない引数（構造体）は、すべて指定する必要があります。
- （入力）は、クライアントが値を指定することを示しています。
- （出力）は、DocumentBroker によって設定される値を格納する領域をクライアントが指定することを示しています。

戻り値

メソッドをコールした結果として返されるコードについて説明しています。

戻り値の詳細

戻り値が「DMA_FALSE」の場合に設定される、詳細な内容を示すコードについて説明しています。なお、戻り値の詳細には、major_code と minor_code が設定されます。コードの体系については、「付録 H 戻り値一覧」を参照してください。

3

CdbrCompound クラス（複合データを扱うクラス）

異なる型の複数のデータから構成される複合データを扱う機能を提供するクラスです。

主に、VariableArray 型のプロパティの Object 型要素に対して値を参照・設定するために使用します。

CdbrCompound クラスの詳細

CdbrCompound::~CdbrCompound メソッド

CdbrCompound::CdbrCompound メソッド

CdbrCompound::GetCount メソッド

CdbrCompound::GetDataType メソッド

CdbrCompound::GetId メソッド

CdbrCompound::GetValue メソッド

CdbrCompound::SetValue メソッド

CdbrCompound クラスの詳細

スーパークラス

なし

プロパティ一覧

なし

ただし、アクセス制御情報を表すプロパティの要素として使用する場合、次に示すプロパティを CdbrCompound クラスのプロパティとして使用します。

表 3-1 CdbrCompound クラスで使用するプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
Subject	dbrProp_Subject	String 型	
SubjectType	dbrProp_SubjectType	Integer32 型	
Permission	dbrProp_Permission	Integer32 型	
ACLIdElem	dbrProp_ACLIdElem	String 型	

プロパティ詳細

dbrProp_ACL プロパティ, dbrProp_SACL プロパティの要素として使用する場合

- Subject
ACL のサブジェクトを指定します。
- SubjectType
ACL のサブジェクト種別を指定します。
- Permission
ACL のパーミッションを指定します。

dbrProp_PublicACLIds プロパティの要素として使用する場合

- ACLIdElem
パブリック ACL の OIID を指定します。

メソッド一覧

CdbrCompound クラスで導入されるメソッド

メソッド	機能
~CdbrCompound	デストラクタ
CdbrCompound	コンストラクタ
GetCount	要素数の取得
GetDataType	要素の型の取得
GetId	要素 ID の取得
GetValue	値の取得
SetValue	値の設定

以降、CdbrCompound クラスで導入されるメソッドについて説明します。

CdbCompound::~~CdbCompound メソッド

機能

デストラクタ

形式

~CdbCompound ()

詳細

CdbCompound オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrCompound::CdbrCompound メソッド

機能

コンストラクタ

形式

CdbrCompound ()

詳細

CdbrCompound オブジェクトのコンストラクタです。

なお、生成直後、データの要素数は 0 です。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbCompound::GetCount メソッド

機能

要素数の取得

形式

```
DmaInteger32 GetCount()
```

詳細

複合データを構成する要素の個数を取得するメソッドです。要素がまったくない場合または要素の個数が算出できない場合は、戻り値に 0 が返ります。

引数

なし

戻り値

複合データの要素数 (0 以上)

戻り値の詳細

なし

CdbrCompound::GetDataType メソッド

機能

要素の型の取得

形式

```
DmaInteger32 GetDataType (  
    pDmaId pId)
```

詳細

複合データを構成する要素のうち、pId で指定した要素の型を取得するメソッドです。

引数

pId (入力)

型を取得する要素の ID を指定します。ID に該当する要素がない場合または ID に該当する要素が取得できない場合は、戻り値に 0 が返ります。

戻り値

返却される値は次のとおりです。

- DMA_DATATYPE_BOOLEAN : Boolean 型
- DMA_DATATYPE_ID : ID 型
- DMA_DATATYPE_INTEGER32 : Integer32 型
- DMA_DATATYPE_STRING : String 型
- 0 : 該当要素なし

戻り値の詳細

なし

CdbCompound::GetId メソッド

機能

要素 ID の取得

形式

```
pDmaId GetId(  
    DmaIndex32 iIndex)
```

詳細

iIndex に指定した位置の要素の ID を取得するメソッドです。エラーの場合、戻り値に NULL が返ります。

引数

iIndex (入力)

ID を取得する要素の位置を指定します。0 ~ (要素数 - 1) の値を指定してください。範囲外の値を指定するとエラーになります。なお、要素の位置は要素数の増減によって変化する場合があります。

戻り値

ID のポインタ 正常終了

NULL 異常終了

戻り値の詳細

なし

CdbCompound::GetValue メソッド

機能

値の取得

形式

```
// Boolean型の場合
DmaBoolean GetValue(pDmaId pId, pDmaBoolean pbRetVal)
// ID型の場合
DmaBoolean GetValue(pDmaId pId, pDmaId pRetVal)
// Integer32型の場合
DmaBoolean GetValue(pDmaId pId, pDmaInteger32 plRetVal)
// String型の場合
DmaBoolean GetValue(pDmaId pId, ppDmaString_T ppRetVal)
```

詳細

pId に指定した要素の値を取得するメソッドです。

引数

pId (入力)

取得する構成要素の ID を指定します。該当する要素がない場合はエラーになります。

pbRetVal, pRetVal, plRetVal, ppRetVal (出力)

取得した値が設定されます。値を格納する領域のアドレスを指定してください。

指定した領域の型と要素の型が異なる場合はエラーになります。取得する String 型プロパティに初期値として NULL が設定されている場合、戻り値に DMA_TRUE が返り、*ppRetVal には NULL が返ります。

String 型の場合、使用後に dbrDelete 関数によって解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

CdbCompound::SetValue メソッド

機能

値の設定

形式

```
// Boolean型の場合
DmaBoolean SetValue(pDmaId pId, DmaBoolean bNewValue)
// ID型の場合
DmaBoolean SetValue(pDmaId pId, DmaId& rNewValue)
// Integer32型の場合
DmaBoolean SetValue(pDmaId pId, DmaInteger32 lNewValue)
// String型の場合
DmaBoolean SetValue(pDmaId pId, pDmaString_T pNewValue)
```

詳細

pId で指定した要素に値を設定するメソッドです。

引数

pId (入力)

設定する要素の ID を指定します。ID に該当する要素がない場合、要素が新たに追加されます。要素がある場合、既存の値が上書きされます。

pId に NULL を指定した場合および String 型の pNewValue に NULL を指定した場合はエラーになります。

bNewValue, rNewValue, lNewValue, pNewValue (入力)

設定する値を指定します。指定する引数の型によって要素の型が決定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

4

CdbrConfiguratedReferentialContainer クラス（バージョン付き構成管理 コンテナを扱うクラス）

コンテナを使用して文書やコンテナのバージョン構成を管理したり，コンテナ自身のバージョンを管理したりする機能を提供するクラスです。
バージョン付き構成管理コンテナは，複数の構成要素をバージョン管理かつ構成管理できるコンテナです。また，バージョン付き構成管理コンテナ自身もバージョン管理および構成管理の対象になります。

CdbrConfiguratedReferentialContainer クラスの詳細

CdbrConfiguratedReferentialContainer::~CdbrConfiguratedReferentialContainer メソッド

CdbrConfiguratedReferentialContainer::~CdbrConfiguratedReferentialContainer メソッド

CdbrConfiguratedReferentialContainer::~CreateObject メソッド

CdbrConfiguratedReferentialContainer::~GetContainableList メソッド

CdbrConfiguratedReferentialContainer::~GetContainableListAndLock メソッド

CdbrConfiguratedReferentialContainer::~GetLinkPropertyValues メソッド

CdbrConfiguratedReferentialContainer::~GetLinkPropertyValuesAndLock メソッド

CdbrConfiguratedReferentialContainer::~GetVTContaineelList メソッド

CdbrConfiguratedReferentialContainer::~GetVTContaineelListAndLock メソッド

CdbrConfiguratedReferentialContainer::~Link メソッド

CdbrConfiguratedReferentialContainer::~LinkAndLock メソッド

CdbrConfiguratedReferentialContainer::~LinkVTFix メソッド

CdbrConfiguratedReferentialContainer::~LinkVTFixAndLock メソッド

CdbrConfiguratedReferentialContainer::~LinkVTFloat メソッド

CdbrConfiguratedReferentialContainer::~LinkVTFloatAndLock メソッド

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

CdbConfiguredReferentialContainer::PutLinkPropertyValues メソッド

CdbConfiguredReferentialContainer::SetVTFix メソッド

CdbConfiguredReferentialContainer::SetVTFloat メソッド

CdbConfiguredReferentialContainer::Unlink メソッド

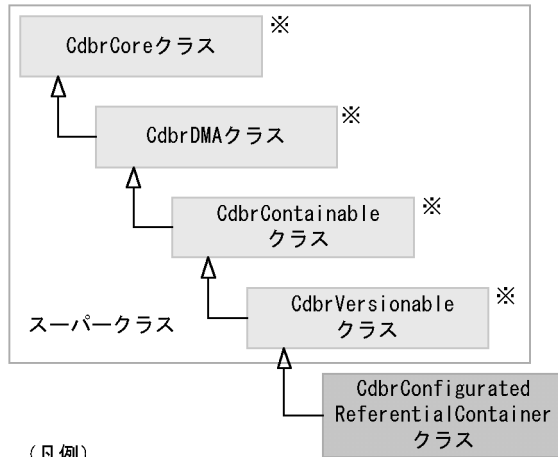
CdbConfiguredReferentialContainer::UnlinkAndLock メソッド

CdbConfiguredReferentialContainer::UnlinkVT メソッド

CdbConfiguredReferentialContainer::UnlinkVTAndLock メソッド

CdbrConfiguredReferentialContainer クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 4-1 CdbrConfiguredReferentialContainer クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ChildrenCount	dbrProp_ChildrenCount	Integer32 型	
ContaineesCount	dbrProp_ContaineesCount	Integer32 型	
ContaineesCountVT	dbrProp_ContaineesCountVT	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContainersCountVT	dbrProp_ContainersCountVT	Integer32 型	
CurrentVersion	dbrProp_CurrentVersion	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
VersionsCount	dbrProp_VersionsCount	Integer32 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 4-2 CdbrConfiguredReferentialContainer クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- ²
UserPermission	dbrProp_UserPermission	Integer32 型	

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用している場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用している場合)

- ACL
- EveryonePermission
- OwnerPermission
- PrimaryGroupId
- PrimaryGroupPermission
- PublicACLCount

- PublicACLIds
- SACL

これらのプロパティの詳細については、「CdbContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbVersionable クラスから継承されるプロパティ

- VersionsCount
- CurrentVersion
- ContainersCountVT

これらのプロパティの詳細については、「CdbVersionable クラスの詳細」のプロパティ詳細を参照してください。

CdbConfiguredReferentialContainer クラスで導入されるプロパティ

- ChildrenCount
カレントバージョンに対応するオブジェクトが、直接型のコンテインメントで包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) の個数を表します。直接型のコンテインメントとは、DMA オブジェクトの DirectContainmentRelationship オブジェクトによって設定された関連づけです。
- ContaineesCount
カレントバージョンに対応するオブジェクトが、参照型のコンテインメントで包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) の個数を表します。参照型のコンテインメントとは、DMA オブジェクトの ReferentialContainmentRelationship オブジェクトによって設定された関連づけです。
- ContaineesCountVT
カレントバージョンに対応するオブジェクトが、構成管理型のコンテインメントで包含している構成管理の対象となるオブジェクト (CdbVersionableDocument オブジェクトおよび CdbConfiguredReferentialContainer オブジェクト) の個数を表します。

なおこのほかに、CdbConfiguredReferentialContainer クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA のクラスの dmaClass_ConfigurationHistory クラスと edmClass_ContainerVersion クラス、およびこれらのサブクラスのプロパティを参照、設定できます。DMA のクラスのプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbCore クラスから継承されるメソッド

メソッド	機能
~CdbCore	デストラクタ
CdbCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbCore クラス (エラー情報を取得するクラス)」を参照してください。

4. CdbrConfiguratedReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrVersionable クラスから継承されるメソッド

メソッド	機能
~CdbrVersionable	デストラクタ
BindPublicACL	パブリック ACL のバインド (バージョン指定あり)
CdbrVersionable	コンストラクタ
DeleteVersion	バージョンの削除
GetPropertyValues	プロパティの取得 (バージョン指定あり)
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定 (バージョン指定あり)

4. CdbrConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

メソッド	機能
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得 (バージョン指定あり)
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定 (バージョン指定あり)
GetReservationStatus	チェックアウト状態の取得
GetReservationStatusAndLock	チェックアウト状態の取得と同時に指定したロックの設定
GetVersionList	バージョン情報一覧の取得
GetVersionListAndLock	バージョン情報一覧の取得と同時に指定したロックの設定
PutPropertyValues	プロパティの設定 (バージョン指定あり)
UnbindPublicACL	パブリック ACL のアンバインド (バージョン指定あり)
VersionCheckIn	バージョンのチェックイン
VersionCheckOut	バージョンのチェックアウト
VersionRevoke	チェックアウトの取り消し

これらのメソッドの詳細については、「15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)」を参照してください。

CdbrConfiguredReferentialContainer クラスで導入されるメソッド

メソッド	機能
~CdbrConfiguredReferentialContainer	デストラクタ
CdbrConfiguredReferentialContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetContainableList	包含している Containable オブジェクトの一覧取得
GetContainableListAndLock	包含している Containable オブジェクトの一覧取得と同時に指定したロックの設定
GetLinkPropertyValues	リンクのプロパティの取得
GetLinkPropertyValuesAndLock	リンクのプロパティの取得と同時に指定したロックの設定
GetVTContaineelist	構成管理しているオブジェクトの一覧取得
GetVTContaineelistAndLock	構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定
Link	Containable オブジェクトとの関連づけ
LinkAndLock	Containable オブジェクトとの関連づけと同時に指定したロックの設定
LinkVTFix	構成管理するオブジェクトの FIX モードでの関連づけ
LinkVTFixAndLock	構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定
LinkVTFloat	構成管理するオブジェクトの FLOATING モードでの関連づけ
LinkVTFloatAndLock	構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

メソッド	機能
PutLinkPropertyValues	リンクのプロパティの設定
SetVTFix	構成要素のバージョンの固定
SetVTFloat	構成要素のバージョン固定の解除
Unlink	Containable オブジェクトとの関連づけの解除
UnlinkAndLock	Containable オブジェクトとの関連づけの解除と同時に指定したロックの設定
UnlinkVT	構成管理しているオブジェクトとの関連づけの解除
UnlinkVTAndLock	構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

以降, CdbConfiguredReferentialContainer クラスで導入されるメソッドについて説明します。

CdbConfiguredReferentialContainer::~CdbConfiguredReferentialContainer メソッド

機能

デストラクタ

形式

```
virtual ~CdbConfiguredReferentialContainer()
```

詳細

CdbConfiguredReferentialContainer オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbConfiguredReferentialContainer::CdbConfiguredReferentialContainer メソッド

機能

コンストラクタ

形式

`CdbConfiguredReferentialContainer()`

詳細

`CdbConfiguredReferentialContainer` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrConfiguredReferentialContainer::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
CdbrConfiguredReferentialContainer::CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*  pDMAInfoList,
    ppDmaString_T     ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           // プロパティ識別子
    DmaInteger32       lType;           // データ型
    DmaInteger32       lCardinality;    // 基本単位
    DmaInteger32       lCount;         // 要素数
    union {           // プロパティの値
        pDmaBoolean    pbBoolean;      // Boolean型
        pDmaInteger32  plInteger32;    // Integer32型
        pDmaId         pId;           // ID型
        pDmapv         ppObject;      // Object型
        ppDmaString_T  ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*        pItem;
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAININFO {
    DmaId              ClassId;
    SDBR_PROPLIST     PropList;
};
```

詳細

オブジェクトを新規に作成するメソッドです。

SetOIID メソッドもしくは ConnectObject メソッドによってすでにオブジェクトと接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更されずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の値を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体である、SDBR_DMAININFO 構造体を指定します。DMA オブジェクトの情報として SDBR_DMAININFO 構造体には、1 番目の要素として dmaClass_ConfigurationHistory クラスまたはそのサブクラスの情報、2 番目の要素として edmClass_ContainerVersion クラスまたはそのサブクラスの情報を指定してください。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMAININFO 構造体については、「付録 C.1(3) SDBR_DMAININFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

ppOID (出力)

作成した CdbConfiguredReferentialContainer オブジェクトの OID へのポインタが設定されます。なお、*ppOID がポイントするメモリは、ユーザプログラムでは解放しないでください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1 文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1 文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1 文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetContainableList メソッド

機能

包含している Containable オブジェクトの一覧取得

形式

// 包含要素の一覧を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer::GetContainableList (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**       ppObjList )
```

// 種類別の包含要素の一覧とリンク情報を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer::GetContainableList (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lContainmentType,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*        pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList )
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32  lType;             // データ型
    DmaInteger32  lCardinality;      // 基本単位
    DmaInteger32  lCount;           // 要素数
    union {                // プロパティの値
        pDmaBoolean    pbBoolean;    // Boolean型
        pDmaInteger32  plInteger32;   // Integer32型
        pDmaId         pId;          // ID型
        pDmapv         ppObject;     // Object型
        ppDmaString_T  ppString;     // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32  lCount;
    SDBR_PROP*    pItem;
};
```

// オブジェクト情報構造体

```
struct SDBR_OBJINFO {
    pDmaString_T    pOIID;          // オブジェクトのOIID
    DmaInteger32   lType;          // オブジェクトの種別
    SDBR_PROPLIST  PropList;       // プロパティリスト構造体
};
```

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

```
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32    lCount;
    SDBR_OBJINFO*  pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO{
    pDmaString_T    pOIID;           // オブジェクトのOIID
    pDmaString_T    lType;          // オブジェクトの種別
    pDmaString_T    pLinkId;        // リンク識別子
    SDBR_PROPLIST   PropList;       // オブジェクトのプロパティリスト
    SDBR_PROPLIST   LinkPropList;   // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST{
    DmaInteger32    lCount; // リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};
```

詳細

pVersionId で指定した CdbConfiguredReferentialContainer オブジェクトが包含している、CdbConfigurable クラスのサブクラスのオブジェクトの一覧を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList で指定したユーザ定義のプロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態が反映されます。したがって、1 回目のコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対してメソッドをコールしても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbConfigurable クラス (包含要素を扱うクラス)」の CdbConfigurable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した値の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合、および pVersionId の値が前回取得時と異なる場合は、すべて DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 4-3 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
IContainmentType	×	
IPropDefListCount	×	
pPropDefList	×	
IObjListCount		
ppObjList		

(凡例)

：有効

×：無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものとして処理されます。

表 4-4 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
IContainmentType	×	
IClassType	×	
IPropDefListCount	×	
pPropDefList	×	
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppLinkedObjList		

(凡例)

：有効

×：無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものとして処理されます。

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。

バージョン識別子は CdbVersionable::GetVersionList メソッドまたは

CdbVersionable::VersionCheckOut メソッドで取得します。チェックアウト中のオブジェクトに対しては、仮のバージョン識別子を指定します。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

NULL を指定した場合は、カレントバージョンが対象になります。

IContainmentType (入力)

対象となるコンテインメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリの解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側はこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体の使用方法については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType(入力)

取得する要素のクラス識別子を指定します。これによって、指定されたクラス識別子の要素だけを取得できます。指定できる識別子を次に示します。

表 4-5 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbReferentialContainer クラス
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの OR 条件
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbReferentialContainer クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT
```

ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、リンクのプロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体の使用方法については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkedObjList (出力)

取得した情報を格納するリンク付きオブジェクトリスト構造体である、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。**ppLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録 C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetContainableListAndLock メソッド

機能

包含している Containable オブジェクトの一覧取得と同時に指定したロックの設定

形式

// 包含要素の一覧を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer::GetContainableListAndLock (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**       ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )
```

// 種類別の包含要素の一覧とリンク情報を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer::GetContainableListAndLock (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lContainmentType,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*        pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;          // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;          // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {
        pDmaBoolean pbBoolean;     // Boolean型
        pDmaInteger32 plInteger32; // Integer32型
        pDmaId       pId;          // ID型
        pDmapv       ppObject;     // Object型
        ppDmaString_T ppString;    // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};
```

// オブジェクト情報構造体

```
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;          // オブジェクトのOIID
};
```

```

    DmaInteger32      lType;           // オブジェクトの種別
    SDBR_PROPLIST     PropList;       // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32      lCount;
    SDBR_OBJINFO*     pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO{
    pDmaString_T      pOIID;          // オブジェクトのOIID
    pDmaString_T      lType;          // オブジェクトの種別
    pDmaString_T      pLinkId;        // リンク識別子
    SDBR_PROPLIST     PropList;       // オブジェクトのプロパティリスト
    SDBR_PROPLIST     LinkPropList;   // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST{
    DmaInteger32      lCount; // リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

```

詳細

ロックタイプを明示的に指定して、pVersionId で指定したオブジェクトが含まれている、CdbConfigurable クラスのサブクラスのオブジェクトの一覧を取得するメソッドです。

その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList で指定したユーザ定義プロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にコールした時点での状態が反映されます。したがって、1回目のメソッドのコール以降に変更した要素数やプロパティ値は、2回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対してメソッドをコールしても、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbConfigurable クラス (包含要素を扱うクラス)」の CdbConfigurable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した値の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合、および pVersionId の値が前回取得時と異なる場合は、すべて DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 4-6 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lContainmentType	×	
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		
lLockType	×	

(凡例)

：有効

×：無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものとして処理されます。

表 4-7 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lContainmentType	×	
lClassType	×	
lPropDefListCount	×	
pPropDefList	×	
lLinkPropDefListCount	×	
pLinkPropDefList	×	
lObjListCount		
ppLinkedObjList		
lLockType	×	

(凡例)

：有効

×：無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものとして処理されます。

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。チェックアウト中のオブジェクトに対しては、仮のバージョン識別子を指定します。

NULL を指定した場合は、カレントバージョンが対象になります。

IContainmentType (入力)

対象となるコンテインメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリはメソッドをコールした側が、dbrDelete 関数によって解放してください。なお、*ppObjList を指定したメモリの解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側はこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体の使用方法については、「付録 C.1(9)

SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

4. CdbConfiguratedReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

IClassType(入力)

指定されたクラス識別子の要素だけを取得します。指定できる識別子を次に示します。

表 4-8 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbReferentialContainer クラス
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの OR 条件
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbReferentialContainer クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、リンクのプロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティとして指定できる識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkedObjList (出力)

取得した情報を格納するリンク付きオブジェクトリスト構造体である、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。**ppLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリはメソッドをコールした側が、dbrDelete 関数によって解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録 C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetLinkProperty Values メソッド

機能

リンクのプロパティの取得

形式

```
DmaBoolean CdbConfiguredReferentialContainer::GetLinkPropertyValues (
    pDmaString_T      pLinkId,
    DmaInteger32      lLinkPropDefListCount,
    SDBR_PROPDEF*     pLinkPropDefList,
    SDBR_PROPLIST**   ppLinkPropList)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                     // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;       // ID型
        pDmapv           ppObject;  // Object型
        ppDmaString_T    ppString;  // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

リンク識別子を指定して、リンクのプロパティを取得するメソッドです。

引数

pLinkId (入力)

リンク識別子を指定します。

lLinkPropDefListCount (入力)

取得するプロパティの個数として、pLinkPropDefList の要素数を指定します。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子はユーザが定義したプロパティの識別子だけです。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定すると、エラーになります。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。

*ppLinkPropList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、プロパティがまったく取得されなかった場合は、NULL が設定されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッド

機能

リンクのプロパティの取得と同時に指定したロックの設定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::GetLinkPropertyValuesAndLock(
    pDmaString_T      pLinkId,
    DmaInteger32      lLinkPropDefListCount,
    SDBR_PROPDEF*     pLinkPropDefList,
    SDBR_PROPLIST**   ppLinkPropList,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;          // データ型
    DmaInteger32 lCardinality;   // 基本単位
    DmaInteger32 lCount;         // 要素数
    union {                       // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;        // ID型
        pDmapv           ppObject;   // Object型
        ppDmaString_T    ppString;   // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32      lCount;
    SDBR_PROP*       pItem;
};
```

詳細

ロックタイプを明示的に指定し、リンク識別子を指定して、リンクのプロパティを取得するメソッドです。

引数

pLinkId (入力)

リンク識別子を指定します。

lLinkPropDefListCount (入力)

取得するプロパティの個数として、pLinkPropDefList の要素数を指定します。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子はユーザが定義したプロパティの識別子だけです。なお、lLinkPropDefListCount

が正の整数のときに pLinkPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。

*ppLinkPropList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、プロパティがまったく取得されなかった場合は、NULL が設定されます。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetVTContaineelist メソッド

機能

構成管理しているオブジェクトの一覧取得

形式

//要素一覧を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer:: GetVTContaineelist (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTOBJLIST**    ppVTObjList)
```

//種類別の要素一覧とリンク情報を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer:: GetVTContaineelist (
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTLINKEDOBJLIST** ppVTLinkedObjList)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32  lType;             // データ型
    DmaInteger32  lCardinality;     // 基本単位
    DmaInteger32  lCount;           // 要素数
    union {                       // プロパティの値
        pDmaBoolean    pbBoolean;   // Boolean型
        pDmaInteger32  plInteger32;  // Integer32型
        pDmaId         pId;         // ID型
        pDmapv         ppObject;     // Object型
        ppDmaString_T  ppString;    // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32  lCount;
    SDBR_PROP*   pItem;
};
```

// 構成管理オブジェクト情報構造体

```
struct SDBR_VTOBJINFO{
    pDmaString_T  pOIID;           // オブジェクトのOIID
    pDmaString_T  pVerOIID;       // バージョントレースの対象になる
    pDmaString_T  pOIID;         // オブジェクトのOIID
    pDmaString_T  pLinkID;        // オブジェクトのリンク識別子
    DmaInteger32  lTraceMode;     // 構成管理モード
};
```

4. CdbConfiguratedReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

```
DmaInteger32 lType; // オブジェクトの種別
SDBR_PROPLIST PropList; // プロパティ情報のリスト
};

// 構成管理オブジェクトリスト構造体
struct SDBR_VTOBJLIST{
    DmaInteger32 lCount; // 構成管理オブジェクト情報の個数
    SDBR_VTOBJINFO* pItem; // 構成管理オブジェクト情報
};

// 構成管理リンク付きオブジェクト情報構造体
struct SDBR_VTLINKEDOBJINFO{
    pDmaString_T pOIID; // オブジェクトのOIID
    pDmaString_T pVerOIID; // バージョントレースの対象になる
    // オブジェクトのOIID
    pDmaString_T pLinkID; // リンク識別子
    DmaInteger32 lTraceMode; // 構成管理モード
    DmaInteger32 lType; // オブジェクトの種別
    SDBR_PROPLIST PropList; // プロパティリスト
    SDBR_PROPLIST LinkPropList; // リンクのプロパティリスト
};

// 構成管理リンク付きオブジェクトリスト構造体
struct SDBR_VTLINKEDOBJLIST{
    DmaInteger32 lCount; // リンク付きオブジェクト情報の個数
    SDBR_VTLINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};
```

詳細

CdbConfiguratedReferentialContainer オブジェクトの pVersionId で指定したバージョンが構成管理しているオブジェクトの一覧を取得するメソッドです。

pPropDefList でプロパティを指定した場合は、指しているバージョン (構成管理モードが FLOATING モードの場合は最新のバージョン, FIX モードの場合は固定のバージョン) のプロパティも取得します。また, pLinkPropDefList で指定したユーザ定義プロパティを持つリンクについては, そのプロパティの値も取得します。なお, リンクのプロパティとは, DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトのプロパティです。

初回のメソッドのコールですべての要素の取得が完了しなかった場合, *pbContinue に DMA_TRUE が設定されるので, 再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合, 取得される内容はすべて初めにコールした時点での状態が反映されます。したがって, 1 回目のメソッドのコール以降に変更した要素数やプロパティ値は, 2 回目以降のコールに反映されません。

また, 要素の個数に変更がなく, 同じオブジェクトに対してメソッドをコールしても, 取得される情報の順序は常に異なる可能性があります。

次に, 構成管理している CdbVersionableDocument オブジェクトの一覧を 10 個ずつ取得して, その OIID とバージョンが固定されているかどうかを出力する例を示します。なお, エラー処理については省略しています。

構成管理している CdbVersionableDocument オブジェクトの一覧を取得する例

```
SDBR_VTOBJLIST* pVTObjList = NULL;
DmaInteger32 lLoop;
DmaBoolean bContinue = DMA_UNKNOWN;
// すべての要素を取得します。
while (bContinue != DMA_FALSE){
    //情報の取得
    pDbrCfgRfct->GetVTContaineerList(&bContinue,
```

```

        NULL, 0, NULL, 10, &pVTObjList);
//取得数だけループして,
//OIIDとバージョンが固定されているかどうかを出力します。
for (lLoop = 0; lLoop < pVTObjList->lCount; lLoop++) {
    printf("OIID=%s¥n", pVTObjList->pItem[lLoop].pOIID);
    if (pVTObjList->pItem[lLoop].lTraceMode == DBR_MODE_FIX)
    {
        printf("Version=Fix¥n");
    }else{
        printf("Version=UnFix¥n");
    }
}
dbrDelete (pVTObjList);
}

```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合または前回の読み出しですべて取得した場合は、すべて DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドのコールの継続が必要かどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 4-9 ほかの引数の扱われ方 (SDBR_VTOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppVTObjList		

(凡例)

○ : 有効

× : 無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

表 4-10 ほかの引数の扱われ方 (SDBR_VTLINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lClassType	×	
lPropDefListCount	×	
pPropDefList	×	
lLinkPropDefListCount	×	
pLinkPropDefList	×	
lObjListCount		
ppLinkedObjList		

(凡例)

○ : 有効

× : 無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したもものとして処理されます。

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は、CdbVersionable::GetVersionList メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得したものを指定します。チェックアウト中のオブジェクトに対しては、仮のバージョン識別子を指定してください。NULL を指定した場合は、カレントバージョンが対象になります。

lPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、lPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppVTObjList (出力)

構成管理オブジェクトリスト構造体である、SDBR_VTOBJLIST 構造体へのポインタが設定されます。構成管理オブジェクトリスト構造体へのポインタを格納する領域を指定してください。

**ppVTObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザ

プログラムで `dbrDelete` 関数を使用して解放してください。

SDBR_VTOBJLIST 構造体および SDBR_VTOBJINFO 構造体に設定される内容については、「付録 C.1(26) SDBR_VTOBJLIST 構造体 (構成管理オブジェクトリスト構造体)」および「付録 C.1(25) SDBR_VTOBJINFO 構造体 (構成管理オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType(入力)

取得する要素のクラス識別子を指定します。これによって、指定されたクラス識別子の要素だけが取得できます。ここでは、指しているバージョンのオブジェクトに該当するクラス識別子を指定する必要があります。クラス識別子の一覧を次に示します。

表 4-11 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbDocument クラスの要素と CdbVersionTraceableContainer クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONTRACEABLECONTAINER
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

ILinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、ILinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppVTLinkedObjList (出力)

取得した情報を格納する構成管理リンク付きオブジェクトリスト構造体である、SDBR_VTLINKEDOBJLIST 構造体へのポインタが設定されます。**ppVTLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで `dbrDelete` 関数を使用して解放してください。

SDBR_VTLINKEDOBJLIST 構造体および SDBR_VTLINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(24) SDBR_VTLINKEDOBJLIST 構造体 (構成管理リンク付きオブジェクトリスト

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

構造体)」および「付録 C.1(23) SDBR_VTLINKEDOBJINFO 構造体 (構成管理リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::GetVTContaineerListAndLock メソッド

機能

構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定

形式

//要素一覧を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer:: GetVTContaineerListAndLock(
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTOBJLIST**     ppVTObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)
```

//種類別の要素一覧とリンク情報を取得する場合

```
DmaBoolean CdbConfiguredReferentialContainer:: GetVTContaineerListAndLock(
    pDmaBoolean          pbContinue,
    pDmaString_T         pVersionId,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*        pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTLINKEDOBJLIST** ppVTLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {         // プロパティの値
        pDmaBoolean pbBoolean;     // Boolean型
        pDmaInteger32 plInteger32; // Integer32型
        pDmaId       pId;          // ID型
        pDmapv       ppObject;     // Object型
        ppDmaString_T ppString;    // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};
```

// 構成管理オブジェクト情報構造体

```
struct SDBR_VTOBJINFO{
    pDmaString_T   pOIID;          // オブジェクトのOIID
    pDmaString_T   pVerOIID;      // バージョントレースの対象になる
    // オブジェクトのOIID
```



```

    pDmaString_T    pLinkID;    // オブジェクトのリンク識別子
    DmaInteger32    lTraceMode; // 構成管理モード
    DmaInteger32    lType;      // オブジェクトの種別
    SDBR_PROPLIST   PropList;   // プロパティ情報のリスト
};

// 構成管理オブジェクトリスト構造体
struct SDBR_VTOBJLIST{
    DmaInteger32    lCount;    // 構成管理オブジェクト情報の個数
    SDBR_VTOBJINFO* pItem;    // 構成管理オブジェクト情報
};

// 構成管理リンク付きオブジェクト情報構造体
struct SDBR_VTLINKEDOBJINFO{
    pDmaString_T    pOIID;      // オブジェクトのOIID
    pDmaString_T    pVerOIID;   // バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T    pLinkID;    // リンク識別子
    DmaInteger32    lTraceMode; // 構成管理モード
    DmaInteger32    lType;      // オブジェクトの種別
    SDBR_PROPLIST   PropList;   // プロパティリスト
    SDBR_PROPLIST   LinkPropList; // リンクのプロパティリスト
};

// 構成管理リンク付きオブジェクトリスト構造体
struct SDBR_VTLINKEDOBJLIST{
    DmaInteger32    lCount; // リンク付きオブジェクト情報の個数
    SDBR_VTLINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

```

詳細

ロックタイプを明示的に指定して、CdbConfiguredReferentialContainer オブジェクトの pVersionId で指定したバージョンが構成管理しているオブジェクトの一覧を取得するメソッドです。

pPropDefList でプロパティを指定した場合は、指しているバージョン (構成管理モードが FLOATING モードの場合は最新のバージョン、FIX モードの場合は固定のバージョン) のプロパティも取得します。

また、pLinkPropDefList で指定したユーザ定義プロパティを持つリンクについては、そのプロパティの値も取得します。なお、リンクのプロパティとは、DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトのプロパティです。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にコールした時点での状態が反映されます。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対してメソッドをコールしても、取得される情報の順序は常に異なる可能性があります。

次に、構成管理している CdbVersionableDocument オブジェクトの一覧を 10 個ずつ取得して、その OIID とバージョンが固定されているかどうかを出力する例を示します。なお、エラー処理については省略しています。

構成管理している CdbVersionableDocument オブジェクトの一覧を取得する例

```

SDBR_VTOBJLIST* pVTObjList = NULL;
DmaInteger32    lLoop;
DmaBoolean      bContinue = DMA_UNKNOWN;
// すべての要素を取得します。

```

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

```

while (bContinue != DMA_FALSE){
    //情報の取得
    pDbrCfgRfct->GetVTContaineelistAndLock(&bContinue,
        NULL, 0, NULL, 10, &pVTObjList,DMA_LOCK_WRITE);
    //取得数だけループして,
    //OIIDとバージョンが固定されているかどうかを出力します。
    for (lLoop = 0; lLoop < pVTObjList->lCount; lLoop++) {
        printf("OIID=%s¥n", pVTObjList->pItem[lLoop].pOIID);
        if (pVTObjList->pItem[lLoop].lTraceMode == DBR_MODE_FIX)
        {
            printf("Version=Fix¥n");
        }else{
            printf("Version=UnFix¥n");
        }
    }
    dbrDelete (pVTObjList);
}

```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合または前回の読み出しですべて取得した場合は、すべて DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドのコールの継続が必要かどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 4-12 ほかの引数の扱われ方 (SDBR_VTOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppVTObjList		
lLockType	×	

(凡例)

: 有効

× : 無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

表 4-13 ほかの引数の扱われ方 (SDBR_VTLINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
IClassType	×	
IPropDefListCount	×	
pPropDefList	×	
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppVTLinkedObjList		
ILockType	×	

(凡例)

○ : 有効

× : 無効

注

前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものとして処理されます。

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は、CdbVersionable::GetVersionList メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得したものを指定します。チェックアウト中のオブジェクトに対しては、仮のバージョン識別子を指定してください。NULL を指定した場合は、カレントバージョンが対象になります。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppVTObjList (出力)

構成管理オブジェクトリスト構造体である、SDBR_VTOBJLIST 構造体へのポインタが設定されます。構成管理オブジェクトリスト構造体へのポインタを格納する領域を指定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

**ppVTObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_VTOBJLIST 構造体および SDBR_VTOBJINFO 構造体に設定される内容については、「付録 C.1(26) SDBR_VTOBJLIST 構造体 (構成管理オブジェクトリスト構造体)」および「付録 C.1(25) SDBR_VTOBJINFO 構造体 (構成管理オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType (入力)

取得する要素のクラス識別子を指定します。これによって、指定されたクラス識別子の要素だけが取得できます。ここでは、指しているバージョンのオブジェクトに該当するクラス識別子を指定する必要があります。クラス識別子の一覧を次に示します。

表 4-14 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbDocument クラスの要素と CdbVersionTraceableContainer クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONTRACEABLECONTAINER
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

ILinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティとして指定できる識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、ILinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppVTLinkedObjList (出力)

取得した情報を格納する構成管理リンク付きオブジェクトリスト構造体である、SDBR_VTLINKEDOBJLIST 構造体へのポインタが設定されます。**ppVTLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_VTLINKEDOBJLIST 構造体および SDBR_VTLINKEDOBJINFO 構造体に設定される内容につ

いては、「付録 C.1(24) SDBR_VTLINKEDOBJLIST 構造体 (構成管理リンク付きオブジェクトリスト構造体)」および「付録 C.1(23) SDBR_VTLINKEDOBJINFO 構造体 (構成管理リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::Link メソッド

機能

Containable オブジェクトとの関連づけ

形式

```
// オブジェクトを指定して関連づける場合
DmaBoolean CdbConfiguredReferentialContainer::Link(
    pDmaString_T      pVersionId,
    CdbContainable*  pDbrContainable,
    DmaInteger32     lContainmentType
    = DBR_CONTAINMENT_REFERENTIAL)

// 関連づけと同時にリンクのプロパティを設定する場合
DmaBoolean CdbConfiguredReferentialContainer::Link(
    pDmaString_T      pVersionId,
    CdbContainable*  pDbrContainable,
    SDBR_PROPLIST*   pLinkPropList,
    ppDmaString_T    ppLinkId,
    DmaInteger32     lContainmentType
    = DBR_CONTAINMENT_REFERENTIAL)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;       // ID型
        pDmapv           ppObject;  // Object型
        ppDmaString_T    ppString;  // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

CdbConfiguredReferentialContainer オブジェクト自身を上位のオブジェクトとして、指定した Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) を下位のオブジェクトとして関連づけるメソッドです。

また、リンクにプロパティを設定することもできます。リンクのプロパティとは、DMA オブジェクトの DirectContainmentRelationship オブジェクト、ReferentialContainmentRelationship オブジェクトおよび VersionTraceableContainmentRelationship オブジェクトのプロパティです。

引数

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、

4. CdbConfiguratedReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得したものを指定します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。

NULL を指定した場合は、カレントバージョンが対象になります。

pDbrContainable (入力)

関連づけるオブジェクトを指定します。CdbContainable クラスのサブクラスのオブジェクトを指定します。

pLinkPropList (入力)

リンクに設定するプロパティリストを指定します。なお、リンクに設定できるプロパティは、ユーザが定義したプロパティだけです。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbConfiguratedReferentialContainer::GetLinkPropertyValues メソッド、CdbConfiguratedReferentialContainer::GetLinkPropertyValuesAndLock メソッドおよび CdbConfiguratedReferentialContainer::PutLinkPropertyValues メソッドでリンクのプロパティを参照、設定するときに使用します。また、CdbConfiguratedReferentialContainer::Unlink メソッドによってリンクの関連づけの解除をする場合にも指定できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

IContainmentType (入力)

関連づけで設定するコンテインメントの種類を指定します。次のどちらかを指定してください。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

なお、上位のコンテナを下位以下のコンテナの下位のオブジェクトとして関連づけることはできません。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::LinkAndLock メソッド

機能

Containable オブジェクトとの関連づけと同時に指定したロックの設定

形式

```
// オブジェクトを指定して関連づける場合
DmaBoolean CdbConfiguredReferentialContainer::LinkAndLock(
    pDmaString_T      pVersionId,
    CdbConfigurable* pDbrContainable,
    DmaInteger32      lContainmentType
                    = DBR_CONTAINMENT_REFERENTIAL,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// 関連づけと同時にリンクのプロパティを設定する場合
DmaBoolean CdbConfiguredReferentialContainer::LinkAndLock(
    pDmaString_T      pVersionId,
    CdbConfigurable* pDbrContainable,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T     ppLinkId,
    DmaInteger32      lContainmentType
                    = DBR_CONTAINMENT_REFERENTIAL,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32     plInteger32; // Integer32型
        pDmaId            pId;       // ID型
        pDmapv            ppObject;  // Object型
        ppDmaString_T     ppString;  // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;
    SDBR_PROP*       pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbConfiguredReferentialContainer オブジェクト自身を上位のオブジェクトとして、指定した Containable オブジェクト (CdbConfigurable クラスのサブクラスのオブジェクト) を下位のオブジェクトとして関連づけるメソッドです。

また、リンクにプロパティを設定することもできます。リンクのプロパティとは、DMA オブジェクトの DirectContainmentRelationship オブジェクト、ReferentialContainmentRelationship オブジェクトおよび VersionTraceableContainmentRelationship オブジェクトのプロパティです。

引数

pVersionId (入力)

対象となる CdbrConfiguratedReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbrVersionable::GetVersionList メソッド、CdbrVersionable::GetVersionListAndLock メソッドまたは CdbrVersionable::VersionCheckOut メソッドで取得したものを指定します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。

NULL を指定した場合は、カレントバージョンが対象になります。

pDbrContainable (入力)

関連づけるオブジェクトを指定します。CdbrContainable クラスのサブクラスのオブジェクトを指定します。

pLinkPropList (入力)

リンクに設定するプロパティリストを指定します。なお、リンクに設定できるプロパティは、ユーザが定義したプロパティだけです。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbrConfiguratedReferentialContainer::GetLinkPropertyValues メソッド、CdbrConfiguratedReferentialContainer::GetLinkPropertyValuesAndLock メソッドおよび CdbrConfiguratedReferentialContainer::PutLinkPropertyValues メソッドでリンクのプロパティを参照、設定するときに使用します。また、CdbrConfiguratedReferentialContainer::Unlink メソッドによってリンクの関連づけの解除をする場合にも指定できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

IContainmentType (入力)

関連づけで設定するコンテインメントの種類を指定します。次のどちらかを指定してください。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

なお、上位のコンテナを下位以下のコンテナの下位のオブジェクトとして関連づけることはできません。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrConfiguredReferentialContainer::LinkVTFix メソッド

機能

構成管理するオブジェクトの FIX モードでの関連づけ

形式

```
DmaBoolean CdbrConfiguredReferentialContainer::LinkVTFix(
    pDmaString_T      pVersionId,
    CdbrVersionable* pDbrVersionable,
    pDmaString_T      pCompoVersionId,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T     ppLinkId)
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;          // 要素数
    union {
        pDmaBoolean pbBoolean;      // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId      pId;            // ID型
        pDmapv      ppObject;       // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};
```

詳細

CdbrConfiguredReferentialContainer オブジェクトにバージョン管理するオブジェクトを、構成管理型のコンテナメントで関連づけるメソッドです。関連づけられるのは、次のオブジェクトです。

- CdbrVersionableDocument オブジェクト
- CdbrConfiguredReferentialContainer オブジェクト

このとき、構成管理モードとして常に固定されたバージョンの構成要素を指す FIX モードを設定します。

FIX モードを設定すると、構成要素がバージョンアップした場合も、

CdbrConfiguredReferentialContainer オブジェクトが指す構成要素のバージョンは変更されません。

CdbrConfiguredReferentialContainer オブジェクトの構成管理モードを、リンクが常にカレントバージョンの構成要素を指す FLOATING モードに変更する場合は、

CdbrConfiguredReferentialContainer::SetVTFloat メソッドを使用してください。

引数

pVersionId (入力)

対象になる CdbrConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。

バージョン識別子は CdbrVersionable::GetVersionList メソッド、

CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。NULLを指定した場合はカレントバージョンが対象になります。

pDbrVersionable (入力)

構成管理する CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを指定します。

pCompoVersionId (入力)

関連づける構成要素のバージョン識別子を指定します。NULLを指定した場合、カレントバージョンを指定したことになります。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリストを指定します。リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULLを指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbConfiguratedReferentialContainer::GetLinkPropertyValues メソッド、CdbConfiguratedReferentialContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbConfiguratedReferentialContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。

4. CdbcrConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::LinkVTFixAndLock メソッド

機能

構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::LinkVTFixAndLock(
    pDmaString_T      pVersionId,
    CdbVersionable*  pDbrVersionable,
    pDmaString_T      pCompoVersionId,
    SDBR_PROPLIST*   pLinkPropList,
    ppDmaString_T     ppLinkId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;       // ID型
        pDmapv           ppObject;  // Object型
        ppDmaString_T    ppString;  // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbConfiguredReferentialContainer オブジェクトにバージョン管理するオブジェクトを、構成管理型のコンテナメントで関連づけるメソッドです。関連づけられるのは、次のオブジェクトです。

- CdbVersionableDocument オブジェクト
- CdbConfiguredReferentialContainer オブジェクト

このとき、構成管理モードとして常に固定されたバージョンの構成要素を指す FIX モードを設定します。FIX モードを設定すると、構成要素がバージョンアップした場合も、CdbConfiguredReferentialContainer オブジェクトが指す構成要素のバージョンは変更されません。

CdbConfiguredReferentialContainer オブジェクトの構成管理モードを、リンクが常にカレントバージョンの構成要素を指す FLOATING モードに変更する場合は、CdbConfiguredReferentialContainer::SetVTFloat メソッドをコールしてください。

引数

pVersionId (入力)

対象になる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド, CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。NULL を指定した場合はカレントバージョンが対象になります。

pDbrVersionable (入力)

構成管理する CdbVersionableDocument オブジェクトまたは CdbConfiguredReferentialContainer オブジェクトを指定します。

pCompoVersionId (入力)

関連づける構成要素のバージョン識別子を指定します。NULL を指定した場合, カレントバージョンを指定したことになります。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリスト構造体を指定します。リンクにプロパティの値を設定できるのは, ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は, NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は, CdbConfiguredReferentialContainer::GetLinkPropertyValues メソッド, CdbConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbConfiguredReferentialContainer::PutLinkPropertyValues メソッドで, 処理対象とするリンクを指定する場合に使用します。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合, ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には, ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は, ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には, ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

CdbConfiguredReferentialContainer::LinkVTFloat メソッド

機能

構成管理するオブジェクトの FLOATING モードでの関連づけ

形式

```
DmaBoolean CdbConfiguredReferentialContainer::LinkVTFloat(
    pDmaString_T      pVersionId,
    CdbVersionable*  pDbrVersionable,
    SDBR_PROPLIST*   pLinkPropList,
    ppDmaString_T    ppLinkId)

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {
        pDmaBoolean   pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId        pId;          // ID型
        pDmapv        ppObject;     // Object型
        ppDmaString_T ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};
```

詳細

CdbConfiguredReferentialContainer オブジェクトとバージョン管理するオブジェクトを、構成管理型のコンテナメントで関連づけるメソッドです。関連づけられるのは、次のオブジェクトです。

- CdbVersionableDocument オブジェクト
- CdbConfiguredReferentialContainer オブジェクト

このとき、構成管理モードとして常にカレントバージョンの構成要素を指す FLOATING モードを設定します。FLOATING モードを設定すると、構成要素がバージョンアップしたときにコンテナが常にカレントバージョンにリンクをつなぎ替えます。

CdbConfiguredReferentialContainer オブジェクトの構成管理モードを、リンクを特定のバージョンで固定する FIX モードに変更する場合は、CdbConfiguredReferentialContainer::SetVTFix メソッドをコールしてください。

引数

pVersionId (入力)

対象になる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。

バージョン識別子は CdbVersionable::GetVersionList メソッド、

CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッド

4. CdbConfiguratedReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

で取得します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。NULLを指定した場合はカレントバージョンが対象になります。

pDbrVersionable (入力)

構成管理する CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを指定します。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbConfiguratedReferentialContainer::GetLinkPropertyValues メソッド、CdbConfiguratedReferentialContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbConfiguratedReferentialContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::LinkVTFloatAndLock メソッド

機能

構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::LinkVTFloatAndLock(
    pDmaString_T      pVersionId,
    CdbVersionable*  pDbrVersionable,
    SDBR_PROPLIST*   pLinkPropList,
    ppDmaString_T    ppLinkId,
    DmaInteger32     lLockType = DMA_LOCK_WRITE)
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;        // 要素数
    union {                    // プロパティの値
        pDmaBoolean    pbBoolean; // Boolean型
        pDmaInteger32  plInteger32; // Integer32型
        pDmaId         pId;       // ID型
        pDmapv         ppObject;  // Object型
        ppDmaString_T  ppString;  // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32    lCount;
    SDBR_PROP*     pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbConfiguredReferentialContainer オブジェクトとバージョン管理するオブジェクトを、構成管理型のコンテナメントで関連づけるメソッドです。関連づけられるのは、次のオブジェクトです。

- CdbVersionableDocument オブジェクト
- CdbConfiguredReferentialContainer オブジェクト

このとき、構成管理モードとして常にカレントバージョンの構成要素を指す FLOATING モードを設定します。FLOATING モードを設定すると、構成要素がバージョンアップしたときにコンテナが常に最新のバージョンにリンクをつなぎ替えます。

CdbConfiguredReferentialContainer オブジェクトの構成管理モードを、リンクを特定のバージョンで固定する FIX モードに変更する場合は、CdbConfiguredReferentialContainer::SetVTFix メソッドをコールしてください。

引数

pVersionId (入力)

対象になる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。

バージョン識別子は CdbVersionable::GetVersionList メソッド ,
CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッド
で取得します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定します。NULL
を指定した場合はカレントバージョンが対象になります。

pDbrVersionable (入力)

構成管理する CdbVersionableDocument オブジェクトまたは CdbConfiguredReferentialContainer
オブジェクトを指定します。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリスト構造体である , SDBR_PROPLIST 構造体を指定します。
リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子に
はユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構
造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を
参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、
CdbConfiguredReferentialContainer::GetLinkPropertyValues メソッド ,
CdbConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッドまたは
CdbConfiguredReferentialContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを
指定する場合に使用します。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログ
ラムで dbrDelete 関数を使用して解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定で
きません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read
ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除
されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後でこの操作を実行してください。

CdbConfiguredReferentialContainer::PutLinkPropertyValues メソッド

機能

リンクのプロパティの設定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::PutLinkPropertyValues (
    pDmaString_T      pLinkId,
    SDBR_PROPLIST*   pLinkPropList)
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;       // ID型
        pDmapv           ppObject;   // Object型
        ppDmaString_T    ppString;  // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};
```

詳細

リンク識別子を指定してプロパティを設定するメソッドです。

引数

pLinkId (入力)

リンク識別子を指定します。

pLinkPropList (入力)

リンクに設定するプロパティ情報をプロパティリスト構造体である、SDBR_PROPLIST 構造体として指定します。なお、リンクに値を設定できるプロパティは、ユーザが定義したプロパティだけです。プロパティ識別子には、ユーザが定義したプロパティの識別子を指定してください。同じプロパティ識別子を指定したプロパティ構造体を複数指定した場合は、エラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::SetVTFix メソッド

機能

構成要素のバージョンの固定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::SetVTFix(
    pDmaString_T      pLinkId,
    pDmaString_T      pCompoVersionId)
```

詳細

CdbConfiguredReferentialContainer オブジェクトの pLinkId で指定した構成要素のバージョンを、pCompoVersionId で指定したバージョンで固定する (構成管理モードに FIX モードを設定する) メソッドです。

このメソッドによってバージョンを固定したあとは、構成要素がバージョンアップしても、CdbConfiguredReferentialContainer オブジェクトは固定したバージョンの構成要素を参照し続けます。

すでに FIX モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は、FIX モードのまま正常終了します。

引数

pLinkId (入力)

バージョンを固定する構成要素のリンク識別子を指定します。リンク識別子は、CdbConfiguredReferentialContainer::GetVTContaineeList メソッドまたは CdbConfiguredReferentialContainer::GetVTContaineeListAndLock メソッドで取得します。

pCompoVersionId (入力)

固定するバージョンのバージョン識別子を指定します。

なお、DBR_VTVERSION を指定した場合には現時点で関連づけられているバージョンで固定されます。NULL を指定した場合にはカレントバージョンで固定されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::SetVTFloat メソッド

機能

構成要素のバージョン固定の解除

形式

```
DmaBoolean CdbConfiguredReferentialContainer::SetVTFloat (
    pDmaString_T    pLinkId)
```

詳細

CdbConfiguredReferentialContainer オブジェクトが構成管理している pLinkId で指定した構成要素のバージョンの固定を解除する (構成管理モードに FLOATING モードを設定する) メソッドです。

このメソッドをコールすると, CdbConfiguredReferentialContainer オブジェクトは, 常に構成要素のカレントバージョンを参照するようにリンクをつなぎ替えます。

すでに FLOATING モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は, FLOATING モードのまま正常終了します。

引数

pLinkId (入力)

バージョン固定を解除する構成要素のリンク識別子を指定します。リンク識別子は, CdbConfiguredReferentialContainer::GetVTContaineeList メソッドで取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して, ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて, 再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して, 障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて, 再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::Unlink メソッド

機能

Containable オブジェクトとの関連づけの解除

形式

```
// Containableオブジェクトを指定して解除する場合
DmaBoolean CdbConfiguredReferentialContainer::Unlink(
    pDmaString_T      pVersionId,
    CdbContainable*   pDbrContainable)

// リンク識別子を指定して解除する場合
DmaBoolean CdbConfiguredReferentialContainer::Unlink(
    pDmaString_T      pLinkId)
```

詳細

CdbConfiguredReferentialContainer オブジェクトが包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) を指定して、関連づけを解除するメソッドです。

また、リンク識別子を指定して関連づけを解除することもできます。

指定したオブジェクトと関連づけがされていない場合は、エラーになります。

なお、包含している Containable オブジェクトやそのリンク識別子は、CdbConfiguredReferentialContainer::GetContainableList メソッドまたは CdbConfiguredReferentialContainer::GetContainableListAndLock メソッドによって取得できます。

引数

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得したものを指定します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定してください。

NULL を指定すると、カレントバージョンが対象になります。

pDbrContainable (入力)

関連づけを解除するオブジェクトを指定します。CdbContainable クラスのサブクラスのオブジェクトを指定します。

pLinkId (入力)

関連づけを解除するオブジェクトのリンク識別子を指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::UnlinkAndLock メソッド

機能

Containable オブジェクトとの関連づけの解除と同時に指定したロックの設定

形式

```
// Containableオブジェクトを指定して解除する場合
DmaBoolean CdbConfiguredReferentialContainer::UnlinkAndLock(
    pDmaString_T      pVersionId,
    CdbContainable*   pDbrContainable,
    DmaInteger32      lLockType = DMA_LOCK_WRITE )

// リンク識別子を指定して解除する場合
DmaBoolean CdbConfiguredReferentialContainer::UnlinkAndLock(
    pDmaString_T      pLinkId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE )
```

詳細

ロックタイプを明示的に指定し、CdbConfiguredReferentialContainer オブジェクトが包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) を指定して、関連づけを解除するメソッドです。

また、リンク識別子を指定して関連づけを解除することもできます。

指定したオブジェクトと関連づけがされていない場合は、エラーになります。

なお、包含している Containable オブジェクトやそのリンク識別子は、CdbConfiguredReferentialContainer::GetContainableList メソッドまたは CdbConfiguredReferentialContainer::GetContainableListAndLock メソッドによって取得できます。

引数

pVersionId (入力)

対象となる CdbConfiguredReferentialContainer オブジェクトのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得したものを指定します。チェックアウト中のオブジェクトに対しては仮のバージョン識別子を指定してください。

NULL を指定すると、カレントバージョンが対象になります。

pDbrContainable (入力)

関連づけを解除するオブジェクトを指定します。CdbContainable クラスのサブクラスのオブジェクトを指定します。

pLinkId (入力)

関連づけを解除するオブジェクトのリンク識別子を指定します。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::UnlinkVT メソッド

機能

構成管理しているオブジェクトとの関連づけの解除

形式

```
DmaBoolean CdbConfiguredReferentialContainer::UnlinkVT(
    pDmaString_T      pLinkId)
```

詳細

CdbConfiguredReferentialContainer オブジェクトに関連づけて構成管理しているオブジェクトとの関連づけを解除するメソッドです。

指定したオブジェクトと関連づけされていないリンク識別子を指定した場合は、エラーになります。

引数

pLinkId (入力)

関連づけを解除する要素のリンク識別子を指定します。リンク識別子は、CdbConfiguredReferentialContainer::GetVTContaineeList メソッドまたは CdbConfiguredReferentialContainer::GetVTContaineeListAndLock メソッドで取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。

4. CdbConfiguredReferentialContainer クラス (バージョン付き構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbConfiguredReferentialContainer::UnlinkVTAndLock メソッド

機能

構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

形式

```
DmaBoolean CdbConfiguredReferentialContainer::UnlinkVTAndLock(  
    pDmaString_T      pLinkId,  
    DmaInteger32      lLockType = DMA_LOCK_WRITE)
```

詳細

ロックタイプを明示的に指定して、CdbConfiguredReferentialContainer オブジェクトに関連づけて構成管理しているオブジェクトとの関連づけを解除するメソッドです。

指定したオブジェクトと関連づけされていないリンク識別子を指定した場合は、エラーになります。

引数

pLinkId (入力)

関連づけを解除する要素のリンク識別子を指定します。リンク識別子は、CdbConfiguredReferentialContainer::GetVTContaineelist メソッドまたは CdbConfiguredReferentialContainer::GetVTContaineelistAndLock メソッドで取得します。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドを呼び出し後に、この操作を実行してください。

5

CdbrContainable クラス（包含要素を扱うクラス）

コンテナの包含要素になる機能を提供するクラスです。つまり、コンテナメントで包含される要素を表すクラスです。
このクラスはほかのクラスのスーパークラスとして存在する抽象クラスです。
このクラス単独のインスタンスは作成できません。

CdbrContainable クラスの詳細

CdbrContainable::~CdbrContainable メソッド

CdbrContainable::BindPublicACL メソッド

CdbrContainable::CdbrContainable メソッド

CdbrContainable::GetContainerList メソッド

CdbrContainable::GetContainerListAndLock メソッド

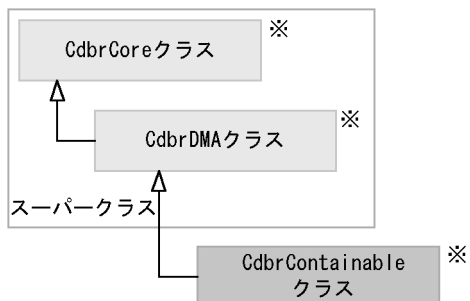
CdbrContainable::GetPublicACLList メソッド

CdbrContainable::GetPublicACLListAndLock メソッド

CdbrContainable::UnbindPublicACL メソッド

CdbrContainable クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 5-1 CdbbrContainable クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ParentCount	dbrProp_ParentCount	Integer32 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 5-2 CdbbrContainable クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- 2
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

• OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

• OwnerId

• UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスで導入されるプロパティ

• ContainersCount

参照型の包含関係でこのオブジェクトを包含している CdbrReferentialContainer オブジェクトおよび CdbrConfiguredReferentialContainer オブジェクトの個数を表します。

• ParentCount

直接型の包含関係でこのオブジェクトを包含している CdbrReferentialContainer オブジェクトおよび CdbrConfiguredReferentialContainer オブジェクトの個数を表します。

CdbrContainable クラスで導入されるプロパティ (アクセス制御機能を使用する場合)

• PrimaryGroupId

プライマリグループのグループ識別子を表します。

• ACL

オブジェクトの ACL (ローカル ACL) を表します。

• SACL

オブジェクトのセキュリティ ACL を表します。

• OwnerPermission

ACFlag のうち、所有者のパーミッションを表します。

• PrimaryGroupPermission

アクセス制御フラグ (ACFlag) のうち、プライマリグループのパーミッションを表します。

• EveryonePermission

アクセス制御フラグ (ACFlag) のうち、全ユーザのパーミッションを表します。

• PublicACLCount

オブジェクトがバインドしているパブリック ACL の数を表します。

• PublicACLIds

オブジェクトがバインドしているパブリック ACL の OIID のリストを表します。

パブリック ACL の OIID のリストは、OID を構成要素とした VariableArray 型プロパティとして作成します。まず、CdbrCompound クラスのメソッドを使用して、バインドするパブリック ACL の OIID を CdbrCompound クラスのオブジェクトの dbrProp_ACLIdElem プロパティに設定します。これを、CdbrVariableArray クラスのメソッドを使用して VariableArray 型プロパティの構成要素と

5. CdbrContainable クラス (包含要素を扱うクラス)

します。このプロパティの値を SDBR_PROP 構造体に設定してから、バインドするオブジェクトの CreateObject メソッドや PutPropertyValues メソッドの引数に指定してメソッドをコールします。これによって、パブリック ACL はオブジェクトにバインドできます。なお、すでにパブリック ACL をバインドしているオブジェクトに対してパブリック ACL を追加したい場合は、BindPublicACL メソッドを使用してください。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスで導入されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得

メソッド	機能
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

以降, CdbContainable クラスで導入されるメソッドについて次に説明します。

CdbriContainable::~CdbriContainable メソッド

機能

デストラクタ

形式

```
virtual ~CdbriContainable ()
```

詳細

CdbriContainable オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbmContainable::BindPublicACL メソッド

機能

パブリック ACL のバインド

形式

```
DmaBoolean BindPublicACL(
    DmaInteger32          lIdListCount,
    ppDmaString_T        ppIdList)
```

詳細

パブリック ACL をバインドするメソッドです。

ppIdList に複数のパブリック ACL を指定することで、一度に複数のパブリック ACL をバインドできます。

存在しない OIID を指定した場合、重複した OIID を指定した場合、すでにバインドされているパブリック ACL の OIID を指定した場合、およびバインド数の制限値を超えた場合は、エラーになります。

引数

lIdListCount (入力)

バインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

バインドするパブリック ACL の OIID 文字列のポインタ配列を指定します。NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。

5. CdbcrContainable クラス (包含要素を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されていません。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLIC_ACL_ALREADY_BOUND	指定されたパブリック ACL はすでにバインドされています。	すでにバインドされているパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbContainable::CdbContainable メソッド

機能

コンストラクタ

形式

`CdbContainable()`

詳細

CdbContainable オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbmContainable::GetContainerList メソッド

機能

コンテナの一覧取得

形式

```
// コンテナの一覧を取得する場合
DmaBoolean GetContainerList (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList )

// コンテナの一覧とそのリンク情報を取得する場合
DmaBoolean GetContainerList (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList )

// クラスを指定してコンテナの一覧とそのリンク情報を取得する場合
DmaBoolean GetContainerList (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          // プロパティ識別子
    DmaInteger32  lType;           // データ型
    DmaInteger32  lCardinality;    // 基本単位
    DmaInteger32  lCount;         // 要素数
    union {
        pDmaBoolean  pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32; // Integer32型
        pDmaId       pId;         // ID型
        pDmapv       ppObject;    // Object型
        ppDmaString_T ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;
    SDBR_PROP*   pItem;
};
```

```

};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T      pOIID;           // オブジェクトのOIID
    DmaInteger32      lType;          // オブジェクトの種別
    SDBR_PROPLIST     PropList;       // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32      lCount;
    SDBR_OBJINFO*     pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO{
    pDmaString_T      pOIID;           // オブジェクトのOIID
    DmaInteger32      lType;          // オブジェクトの種別
    pDmaString_T      pLinkId;        // リンクの識別子
    SDBR_PROPLIST     PropList;       // オブジェクトのプロパティリスト
    SDBR_PROPLIST     LinkPropList;   // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST{
    DmaInteger32      lCount; //リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; //リンク付きオブジェクト情報
};

```

詳細

このオブジェクトを包含しているコンテナ (CdbrReferentialContainer オブジェクトおよび CdbrVersionTraceableContainer オブジェクト) の一覧を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList に指定したプロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態を示しています。したがって、1 回目のメソッドのコール要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

次に、CdbrContainable クラスのサブクラスのオブジェクトである、CdbrVersionableDocument オブジェクトを包含する CdbrReferentialContainer オブジェクトの一覧を 10 個ずつ取得して、その OIID を出力する例を示します。なお、エラー処理は省略しています。

包含要素一覧の取得例

```

SDBR_OBJLIST* pObjList = NULL;
DmaInteger32 lLoop;
DmaBoolean   bContinue = DMA_UNKNOWN;
//すべての要素を取得します。
while (bContinue != DMA_FALSE) {
    //情報の取得
    pDbrVerDoc->GetContainerList(&bContinue,
    DBR_CONTAINMENT_REFERENTIAL, 0, NULL, 10, &pObjList);
    //取得数だけループして、OIIDを出力します。
    for (lLoop = 0; lLoop < pObjList->lCount; lLoop++) {
        printf("OIID=%s\n", pObjList->pItem[lLoop].pOIID);
    }
}

```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、DMA_FALSE を指定したもとして処理されます。また、前回実行したときと引数が異なるメソッドが実行された場合は、pbContinue の値を無視して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 5-3 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lContainmentType	×	
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		

(凡例)

：有効
 ×：無効

表 5-4 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lContainmentType	×	
lClassType	×	
lPropDefListCount	×	
pPropDefList	×	
lLinkPropDefListCount	×	
pLinkPropDefList	×	
lObjListCount		
ppLinkedObjList		

(凡例)

：有効
 ×：無効

IContainmentType (入力)

対象となるコンテナメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテナメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテナメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_VERSIONTRACEABLE :
構成管理型のコンテナメントを設定します。DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトを使用した関連づけです。

*pbContinue に DMA_TRUE を指定して継続取得する場合、継続取得の開始時に指定された種別が有効になり、この値は無視されます。

IClassType (入力)

指定したクラス識別子の要素だけを取得します。指定できる識別子を次に示します。

表 5-5 クラス識別子の一覧

クラス識別子	クラスライブラリのクラス
DBR_CLASS_REFERENTIALCONTAINER	CdbReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbVersionTraceableContainer クラスまたは CdbReferentialContainer クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_ALL	上記すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。

コンテナを表すすべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

情報を取得するオブジェクトがこのプロパティを持つ場合、(*ppObjList)->Item[n].PropList にプロパティ情報が設定されます。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

5. CdbContainable クラス (包含要素を扱うクラス)

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合はプロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティ識別子として指定できるのは、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

オブジェクト情報リストである、SDBR_OBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで、dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkedObjList (出力)

リンク付きオブジェクト情報リストである、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録 C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbmContainable::GetContainerListAndLock メソッド

機能

コンテナの一覧取得と同時に指定したロックの設定

形式

```
// コンテナの一覧を取得する場合
DmaBoolean GetContainerListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )

// コンテナの一覧とそのリンク情報を取得する場合
DmaBoolean GetContainerListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )

// クラスを指定してコンテナの一覧とそのリンク情報を取得する場合
DmaBoolean GetContainerListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32  lType;             // データ型
    DmaInteger32  lCardinality;     // 基本単位
    DmaInteger32  lCount;           // 要素数
    union {                          // プロパティの値
        pDmaBoolean  pbBoolean;     // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId       pId;           // ID型
        pDmapv       ppObject;      // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
```

```

struct SDBR_PROPLIST {
    DmaInteger32    lCount;
    SDBR_PROP*     pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T    pOIID;           // オブジェクトのOIID
    DmaInteger32    lType;           // オブジェクトの種別
    SDBR_PROPLIST   PropList;       // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32    lCount;
    SDBR_OBJINFO*   pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO{
    pDmaString_T    pOIID;           // オブジェクトのOIID
    DmaInteger32    lType;           // オブジェクトの種別
    pDmaString_T    pLinkId;        // リンクの識別子
    SDBR_PROPLIST   PropList;       // オブジェクトのプロパティリスト
    SDBR_PROPLIST   LinkPropList;  // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST{
    DmaInteger32    lCount; //リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; //リンク付きオブジェクト情報
};

```

詳細

ロックタイプを明示的に指定して、このオブジェクトを包含しているコンテナ (CdbReferentialContainer オブジェクトおよび CdbVersionTraceableContainer オブジェクト) の一覧を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList に指定したプロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態を示しています。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティの値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

次に、CdbContainable クラスのサブクラスのオブジェクトである、CdbVersionableDocument オブジェクトを包含する CdbReferentialContainer オブジェクトの一覧を 10 個ずつ取得して、その OIID を出力する例を示します。なお、エラー処理は省略しています。

包含要素一覧の取得例

```

SDBR_OBJLIST* pObjList = NULL;
DmaInteger32  lLoop;
DmaBoolean    bContinue = DMA_UNKNOWN;
//すべての要素を取得します。
while (bContinue != DMA_FALSE) {
    //情報の取得

```

5. CdbContainerable クラス (包含要素を扱うクラス)

```

pDbrVerDoc->GetContainerListAndLock (&bContinue,
DBR_CONTAINMENT_REFERENTIAL, 0, NULL, 10, &pObjList,
DMA_LOCK_READ);
//取得数だけループして、OIDを出力します。
for (lLoop = 0; lLoop < pObjList->lCount; lLoop++) {
    printf("OID=%s¥n", pObjList->pItem[lLoop].pOID);
}
}

```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、すべて DMA_FALSE を指定したものととして処理されます。また、前回実行したときと引数が異なるメソッドが実行された場合は、pbContinue の値を無視して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

次に、2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について示します。

表 5-6 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lContainmentType	×	
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		
lLockType	×	

(凡例)

- : 有効
- × : 無効

表 5-7 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lContainmentType	×	
lClassType	×	
lPropDefListCount	×	
pPropDefList	×	

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppLinkedObjList		
ILockType	×	

(凡例)

- : 有効
- × : 無効

IContainmentType (入力)

対象となるコンテインメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_VERSIONTRACEABLE :
構成管理型のコンテインメントを設定します。DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトを使用した関連づけです。

*pbContinue に DMA_TRUE を指定して継続取得する場合、継続取得の開始時に指定された種別が有効になり、この値は無視されます。

IClassType (入力)

指定したクラス識別子の要素だけを取得します。指定できる識別子を次に示します。

表 5-8 クラス識別子の一覧

クラス識別子	クラスライブラリのクラス
DBR_CLASS_REFERENTIALCONTAINER	CdbrReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbrVersionTraceableContainer クラスまたは CdbrReferentialContainer クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_ALL	上記すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。CdbrReferentialContainer クラスと CdbrVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。

コンテナを表すすべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定し

5. CdbContainable クラス (包含要素を扱うクラス)

た場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、lPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

情報を取得するオブジェクトがこのプロパティを持つ場合、(*ppObjList)->Item[n].PropList にプロパティ情報が設定されます。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティ識別子として指定できるのは、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

オブジェクト情報リストである、SDBR_OBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkedObjList (出力)

リンク付きオブジェクト情報リストである、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録

C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体) を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

5. CdbcrContainable クラス (包含要素を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbmContainable::GetPublicACLList メソッド

機能

バインドしているパブリック ACL 一覧情報の取得

形式

```
DmaBoolean GetPublicACLList (
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*    pPropDefList,
    SDBR_OBJLIST**   ppObjList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean; //Boolean型
        pDmaInteger32    plInteger32; //Integer32型
        pDmaId           pId;        //ID型
        pDmapv           ppObject;    //Object型
        ppDmaString_T    ppString;   //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*   pItem;        //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T    pOID;       // オブジェクトのOID
    DmaInteger32    lType;      // オブジェクトの種別
    SDBR_PROPLIST  PropList;    // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32 lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

オブジェクトがバインドしているパブリック ACL の一覧情報は ppObjList に設定されます。

引数

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である, SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である, SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOIID にはパブリック ACL の OIID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbmContainable::GetPublicACLListAndLock メソッド

機能

バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetPublicACLListAndLock(
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*    pPropDefList,
    SDBR_OBJLIST**   ppObjList,
    DmaInteger32     lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*  pItem;         //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T    pOIID;      // オブジェクトのOIID
    DmaInteger32    lType;      // オブジェクトの種別
    SDBR_PROPLIST  PropList;    // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32 lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

ロックタイプを明示的に指定して、オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

オブジェクトがバインドしているパブリック ACL の一覧情報は ppObjList に設定されます。

引数

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOIID にはパブリック ACL の OIID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrContainable::UnbindPublicACL メソッド

機能

パブリック ACL のアンバインド

形式

```
// 指定したパブリックACLをアンバインドする場合
DmaBoolean UnbindPublicACL(
    DmaInteger32      lIdListCount,
    ppDmaString_T    ppIdList)

// すべてのパブリックACLをアンバインドする場合
DmaBoolean UnbindPublicACL()
```

詳細

オブジェクトからパブリック ACL をアンバインドするメソッドです。

複数のパブリック ACL を一度にアンバインドしたり、すべてのパブリック ACL を一度にアンバインドしたりできます。

指定したパブリック ACL の OIID がバインドされていない場合、または重複した OIID を指定した場合はエラーになります。

引数

lIdListCount (入力)

アンバインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

アンバインドするパブリック ACL の OIID のポインタ配列を指定します。

NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

5. CdbcrContainable クラス (包含要素を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLICACL_NOT_BOUND	指定されたパブリック ACL はバインドされていません。	バインドされているパブリック ACL を指定して処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

6

CdbrCore クラス（エラー情報を取得するクラス）

クラスライブラリのクラスのうち，CdbrVariableArray クラスおよび CdbrCompound クラス以外のすべてのクラスのスーパークラスです。このクラスは，エラー管理機能を提供します。
なお，このクラスは抽象クラスです。このクラス単独のインスタンスは作成しません。

CdbrCore クラスの詳細

CdbrCore::~CdbrCore メソッド

CdbrCore::CdbrCore メソッド

CdbrCore::GetDBError メソッド

CdbrCore::GetLastError メソッド

CdbrCore クラスの詳細

スーパークラス

なし

プロパティ一覧

なし

メソッド一覧

CdbrCore クラスで導入されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

以降, CdbrCore クラスで導入されるメソッドについて説明します。

CdbCore::~~CdbCore メソッド

機能

デストラクタ

形式

```
virtual CdbCore::~~CdbCore()
```

詳細

CdbCore オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrCore::CdbrCore メソッド

機能

コンストラクタ

形式

`CdbrCore::CdbrCore()`

詳細

CdbrCore オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbCore::GetDBError メソッド

機能

データベースのエラーの取得

形式

```
void CdbCore::GetDBError(  
    pDmaInteger32 pSQLCode,  
    ppDmaString_T ppSQLState)
```

詳細

データベースのエラーを取得するメソッドです。データベースでエラーが発生した場合、CdbCore::GetLastError メソッドによって `major_code` に「ERR_DB」が設定されます。このとき、このメソッドを使用してデータベースの詳細なエラー情報を取得してください。

引数

pSQLCode (出力)

データベースのエラーとして、SQLCODE が設定されます。SQLCODE はデータベースの種類に依存しません。詳細については、ご使用のデータベースのマニュアルを参照してください。

ppSQLState (出力)

データベースのエラーとして、SQLSTATE が設定されます。詳細については、ご使用のデータベースのマニュアルを参照してください。なお、ユーザプログラムでは、ppSQLState を解放しないでください。

戻り値

なし

戻り値の詳細

なし

CdbbrCore::GetLastError メソッド

機能

エラー番号の取得

形式

```
DmaInteger32 CdbbrCore::GetLastError(  
    pDmaInteger32 plMinorCode,  
    ppDmaString_T ppErrorString = NULL)
```

詳細

各オブジェクトで発生したエラー情報を取得するメソッドです。

エラー番号は、オブジェクトごとに管理されています。このため、エラー情報を取得するときには、エラーが発生したオブジェクトの `GetLastError` をコールしてください。

`plMinorCode` に `NULL` を設定した場合は、戻り値に `major_code` だけが返されます。

引数

`plMinorCode` (出力)

エラーの `major_code` が次のどれかの場合、戻り値の詳細の `minor_code` が設定されます。

- `ERR_DMA`
- `ERR_DBR`
- `ERR_DB`

`NULL` を指定した場合は、戻り値として `major_code` だけが設定されます。

`ppErrorString` (出力)

戻り値の詳細の `major_code` が `ERR_CORBA` のとき、`VisiBroker` によって返却される例外名を格納したポインタが設定されます。

`ppErrorString` を指定しない場合または `NULL` を指定した場合は、`CORBA` の例外名は設定されません。

なお、ユーザプログラムで `ppErrorString` を解放しないでください。

戻り値

`major_code`

戻り値の詳細

なし

7

CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)

DMA のオブジェクトモデルの dmaClass_DMA クラスに対応するクラスです。

ほかのクラスのスーパークラスとして存在する抽象クラスで、DMA オブジェクトに対する共通機能を提供し、OIID の取得およびプロパティの参照や設定をするメソッドを定義します。このクラス単独のインスタンスは作成しません。

CdbrDMA クラスの詳細

CdbrDMA::~CdbrDMA メソッド

CdbrDMA::CdbrDMA メソッド

CdbrDMA::ConnectObject メソッド

CdbrDMA::GetOIID メソッド

CdbrDMA::GetPropertyValues メソッド

CdbrDMA::GetPropertyValuesAndLock メソッド

CdbrDMA::GetType メソッド

CdbrDMA::PutPropertyValues メソッド

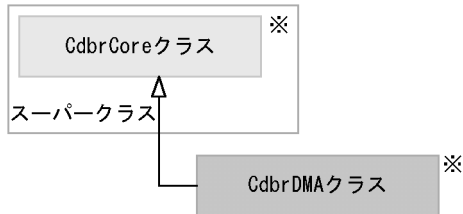
CdbrDMA::ReleaseObject メソッド

CdbrDMA::RemoveObject メソッド

CdbrDMA::SetOIID メソッド

CdbrDMA クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 7-1 CdbDMA クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 7-2 CdbDMA クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	-
UserPermission	dbrProp_UserPermission	Integer32 型	

注

変更は所有者またはセキュリティ管理者だけが実行できます。

プロパティ詳細

CdbrDMA クラスで導入されるプロパティ

- OIID
オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスで導入されるプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

- OwnerId
オブジェクトの所有者のユーザ識別子を表します。このプロパティを更新できるのは、オブジェクトの所有者またはセキュリティ管理者だけです。
- UserPermission
オブジェクトに付与されたログインユーザのパーミッションを表します。すなわち、オブジェクトに設定されている ACFlag、パブリック ACL などのアクセス制御情報のうち、ログインユーザまたは

ログインユーザが所属するグループに与えられているパーミッションの論理和を表します。
 なお、次に示すように、CdbrDMA のサブクラスのインスタンスによって、パーミッションの論理和
 を取る際に対象となるアクセス制御情報が異なります。

表 7-3 パーミッションの論理和を取る際に対象となるアクセス制御情報

クラス	対象となるアクセス制御情報
CdbrContainable クラスとそのサブクラス	ACFlag, ローカル ACL, およびバインドしているパブリック ACL
CdbrIndependentPersistence クラス	
CdbrPublicACL クラス	ローカル ACL (複数のオブジェクトに共有されている ACL)

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスで導入されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOID	OID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOID	OID の設定

以降、CdbrDMA クラスで導入されるメソッドについて説明します。

CdbbrDMA::~CdbbrDMA メソッド

機能

デストラクタ

形式

```
virtual ~CdbbrDMA()
```

詳細

CdbbrDMA オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrDMA::CdbrDMA メソッド

機能

コンストラクタ

形式

CdbrDMA ()

詳細

CdbrDMA オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrDMA::ConnectObject メソッド

機能

オブジェクトを指定したモードでロックして接続

形式

```
virtual DmaBoolean ConnectObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lLockType,
    pDmaString_T      pOID )
```

詳細

pOID で指定したオブジェクトと接続するメソッドです。オブジェクトに指定したロックを設定します。オブジェクトにすでにロックがかけられていて、指定したロックが設定できなかった場合は、ロックが設定できるまでウエイトするか、またはエラーとして直ちに終了するかを指定できます。ウエイトする指定をした場合でも、タイムアウトまたはデッドロックが発生した場合は、直ちにエラーとして終了します。このとき、このメソッドの終了時の戻り値として「DMA_FALSE」が返却され、戻り値の詳細として「DMA_DB」が設定されます。

このメソッドを、一つのオブジェクトに対して ReleaseObject メソッドを介さずに 2 回以上コールした場合、既存の接続は変更されずに、あとのメソッドのコールがエラーになります。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

lLockType (入力)

ロックの種別と、ロックを設定するときウエイトをするかどうかを指定します。次の値のどれかを指定してください。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックできなかった場合、ロックできるまでウエイトします。ただし、タイムアウトやデッドロックが発生した場合は終了します。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックできなかった場合、ロックできるまでウエイトします。ただし、タイムアウトやデッドロックが発生した場合は終了します。
- EDM_LOCK_READ_NOWAIT
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックできなかった場合には、直ちに終了します。
- EDM_LOCK_WRITE_NOWAIT
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックできなかった場合には、直ちに終了します。

pOID (入力)

接続するオブジェクトの OID を指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	引数で指定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_DISCONNECTED	文書空間との接続が切断されました。	セッションを接続してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbbrDMA::GetOIID メソッド

機能

OIID の取得

形式

```
DmaBoolean GetOIID(
    ppDmaString_T ppOIID)
```

詳細

オブジェクトの OIID を取得するメソッドです。ここで取得する OIID の実体は、対応する DMA オブジェクトの OIID です。

ppOIID に、OIID 文字列へのポインタが設定されます。この文字列領域はオブジェクト内に確保されて、オブジェクトの消滅とともに破棄されます。したがって、メソッドをコールした側が破棄する必要はありません。オブジェクトの消滅以降もこの値を使用したい場合は、必ず内容をほかの領域に複写するようにしてください。

引数

ppOIID (出力)

OIID 文字列へのポインタが設定されます。OIID が特定できない場合、NULL が設定されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxDMA::GetPropertyValues メソッド

機能

プロパティの取得

形式

```
virtual DmaBoolean GetPropertyValues (
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_PROPLIST**      ppPropList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32  lType;            // データ型
    DmaInteger32  lCardinality;     // 基本単位
    DmaInteger32  lCount;          // 要素数
    union {
        DmaBoolean  pbBoolean;     // Boolean型
        DmaInteger32 plInteger32;   // Integer32型
        DmaId       pId;           // ID型
        Dmapv       ppObject;      // Object型
        ppDmaString_T ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;           // プロパティ情報の個数
    SDBR_PROP*   pItem;           // プロパティ情報
};
```

詳細

プロパティの値を一括して取得するメソッドです。

次に、プロパティを取得して標準出力に表示する例を示します。

プロパティの値の取得例

```
SDBR_PROPDEF  PropDef [2];
SDBR_PROPLIST* pPropList = NULL;
SDBR_PROP*    pProp = NULL;
int           i, j;

// 取得したいプロパティのIDをリストに設定します。
PropDef [0].PropId = PROPERTY_ID_OF_NUMBER;
PropDef [1].PropId = PROPERTY_ID_OF_AUTHORS;

// プロパティの値を取得します。
if (pDbrDMA->GetPropertyValues (2, PropDef, &pPropList)
    == DMA_FALSE) {
    return DMA_FALSE;
}

// 型を判定して出力します。
for (i = 0; i < pPropList->lCount; i++) {
    pProp = &pPropList->pItem[i];
```

7. CdbrdMA クラス (OID の設定やプロパティを扱うクラス)

```

if (memcmp(&pProp->PropId,&PROPERTY_ID_OF_NUMBER
, sizeof(PROPERTY_ID_OF_NUMBER)) == 0) {
for(j = 0; j < pProp->lCount; j++) {
printf("Number=%d¥n", pProp->uniValue.plInteger32[j]);
}
}
else if (memcmp(&pProp->PropId,&PROPERTY_ID_OF_AUTHORS
, sizeof(PROPERTY_ID_OF_AUTHORS)) == 0) {
for(j = 0; j < pProp->lCount; j++) {
printf("Author=%s¥n", pProp->uniValue.ppString[j]);
}
}
}
// メモリの解放
dbrDelete (pPropList);

```

引数

lPropDefListCount (入力)

pPropDefList に指定した配列の要素数を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

取得するプロパティの定義情報の配列を、SDBR_PROPDEF 構造体として指定します。なお、lPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。SDBR_PROPDEF 構造体の指定内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppPropList がポイントする領域は、ライブラリが新たに確保したものです。この領域はメソッドをコールした側が dbrDelete 関数を使用して解放してください。なお、プロパティがまったく取得されなかった場合は、NULL が設定されます。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOID メソッドで設定されたOID が不正です。	正しいOID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_VALUE_NOT_SET	要求されたプロパティは値を保持していません。	この操作は実行できません。

CdbbrDMA::GetPropertyValuesAndLock メソッド

機能

プロパティの取得と同時に指定したロックの設定

形式

```
virtual DmaBoolean GetPropertyValuesAndLock(
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_PROPLIST**      ppPropList,
    DmaInteger32          lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;          // 要素数
    union {
        pDmaBoolean pbBoolean;      // Boolean型
        pDmaInteger32 plInteger32;   // Integer32型
        pDmaId       pId;            // ID型
        pDmapv        ppObject;      // Object型
        ppDmaString_T ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;           // プロパティ情報の個数
    SDBR_PROP*     pItem;           // プロパティ情報
};
```

詳細

ロックタイプを明示的に指定して、プロパティの値を一括して取得するメソッドです。

次に、プロパティを取得して標準出力に表示する例を示します。

プロパティの値の取得例

```
SDBR_PROPDEF PropDef[2];
SDBR_PROPLIST* pPropList = NULL;
SDBR_PROP*     pProp = NULL;
int            i, j;

// 取得したいプロパティのIDをリストに設定します。
PropDef[0].PropId = PROPERTY_ID_OF_NUMBER;
PropDef[1].PropId = PROPERTY_ID_OF_AUTHORS;

// プロパティの値を取得します。
if (pDbrDMA->GetPropertyValuesAndLock(2, PropDef,
    &pPropList, DMA_LOCK_READ) == DMA_FALSE) {
    return DMA_FALSE;
}

// 型を判定して出力します。
for (i = 0; i < pPropList->lCount; i++) {
```

```

pProp = &pPropList->pItem[j];
if (memcmp(&pProp->PropId,&PROPERTY_ID_OF_NUMBER
, sizeof(PROPERTY_ID_OF_NUMBER)) == 0) {
for(j = 0;j < pProp->lCount;j++) {
printf("Number=%d¥n", pProp->uniValue.plInteger32[j]);
}
}
else if (memcmp(&pProp->PropId,&PROPERTY_ID_OF_AUTHORS
, sizeof(PROPERTY_ID_OF_AUTHORS)) == 0) {
for(j = 0;j < pProp->lCount;j++) {
printf("Author=%s¥n", pProp->uniValue.ppString[j]);
}
}
}
// メモリの解放
dbrDelete (pPropList);

```

引数

IPropDefListCount (入力)

pPropDefList に指定した配列の要素数を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

取得するプロパティの定義情報の配列を、SDBR_PROPDEF 構造体として指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。SDBR_PROPDEF 構造体の指定内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppPropList がポイントする領域は、ライブラリが新たに確保したものです。この領域はメソッドをコールした側が dbrDelete 関数を使用して解放してください。なお、プロパティがまったく取得されなかった場合は、NULL が設定されます。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOID メソッドで設定された IID が不正です。	正しい IID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。

7. CdbbrDMA クラス (OID の設定やプロパティを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_VALUE_NOT_SET	要求されたプロパティは値を保持していません。	この操作は実行できません。

CdbxDMA::GetType メソッド

機能

オブジェクトの種別の取得

形式

```
DmaInteger32 GetType()
```

詳細

オブジェクトの種別を取得するメソッドです。オブジェクトの種別を示す値として、オブジェクトに対応するクラスライブラリのクラスを識別する定数が設定されます。あるクラスの識別定数と戻り値との論理積が 0 以外の場合、そのクラスのサブクラスであることを意味します。

なお、戻り値はクラスライブラリのクラスに関する情報を取得するものであり、クラスライブラリのクラスを継承したユーザ定義のクラスの識別や継承関係の判定には使用できません。例えば、CdbxVersionableDocument クラスを直接継承したユーザ定義のクラスからこのメソッドをコールすると、異なるクラスであっても戻り値は常に DBR_CLASS_VERSIONABLEDOCUMENT になります。

次に、オブジェクトの種別を取得して、その結果を出力する例を示します。

オブジェクトの種別の取得例

```
// 種別の取得
DmaInteger32 lValue = pDbrObj->GetType();
// 値の判定
if (lValue == DBR_CLASS_REFERENTIALCONTAINER) {
    printf("ReferentialContainerクラスのオブジェクトです。¥n");
}
else if ((lValue & DBR_CLASS_REFERENTIALCONTAINER)
        == DBR_CLASS_REFERENTIALCONTAINER) {
    printf("ReferentialContainerのサブクラスのオブジェクトです。¥n");
}
else {
    printf("ReferentialContainerクラスのオブジェクトではありません。¥n");
}
```

引数

なし

戻り値

オブジェクトに対応するクラスライブラリのクラスを識別する値が返ります。GetType メソッドの戻り値一覧を次に示します。

表 7-4 GetType メソッドの戻り値一覧

値	オブジェクトの種別 (クラス)
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbxConfiguredReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbxDocument クラス
DBR_CLASS_INDEPENDENTPERSISTENCE	CdbxIndependentPersistence クラス
DBR_CLASS_PUBLICACL	CdbxPublicACL クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbxReferentialContainer クラス

値	オブジェクトの種別 (クラス)
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_UNKNOWN	クラスの種別が不明

戻り値の詳細

なし

CdbxDMA::PutPropertyValues メソッド

機能

プロパティの設定

形式

```
DmaBoolean PutPropertyValues(
    SDBR_PROPLIST*      pPropList )

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          // プロパティ識別子
    DmaInteger32 lType;        // データ型
    DmaInteger32 lCardinality; // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean;    // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;          // ID型
        pDmapv           ppObject;     // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

プロパティに値を一括して設定するメソッドです。

次に、Number (Integer32 型) プロパティに値を設定する例を示します。プロパティ識別子は、PROP_ID_OF_NUMBER とします。

プロパティに値を設定する例

```
SDBR_PROP      Props;
SDBR_PROPLIST  PropList;
DmaInteger32   lNum = 25;

// 設定したいプロパティの識別子と値を用意します。
Props.PropId = PROPERTY_ID_OF_NUMBER;
Props.lType = DMA_DATATYPE_INTEGER32;
Props.lCardinality = DMA_CARDINALITY_SINGLE;
Props.uniValue.plInteger32 = &lNum;
Props.lCount = 1;
PropList.lCount = 1;
PropList.pItem = &Props;

// プロパティに値を設定します。
pDbrDMA->PutPropertyValues(&PropList);
```

引数

pPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に指定する内容については、「付録 C.1(14)

SDBR_PROPLIST 構造体 (プロパティリスト構造体) および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

同じプロパティ識別子を指定したプロパティ構造体を複数指定した場合は、エラーになります。

また、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbrCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。

7. CdbrdMA クラス (OID の設定やプロパティを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOID メソッドで設定された OID が不正です。	正しい OID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbbrDMA::ReleaseObject メソッド

機能

オブジェクトとの接続の解除

形式

```
virtual DmaBoolean ReleaseObject ()
```

詳細

オブジェクトとの接続を解除するメソッドです。すでに接続が解除されている場合は、何もしないで正常終了します。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_DISCONNECTED	文書空間との接続が切断されました。	セッションを接続してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。

7. CdbbrDMA クラス (OID の設定やプロパティを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbbrDMA::RemoveObject メソッド

機能

オブジェクトの削除

形式

```
virtual DmaBoolean RemoveObject()

// リファレンスファイル文書の場合
DmaBoolean RemoveObject(
    SDBR_REFERENCE_PATHINFO& ReferencePath)

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO{
    DmaInteger32 lContentOperateMode; // 操作モード
    pDmaString_T pEntry; // ファイルのパスおよびロケーション
    pDmaString_T pTargetPath; // コンテンツ格納先パス
    pDmaString_T pDeleteRootPath; // 削除ディレクトリのルートパス
};
```

詳細

オブジェクトを削除するメソッドです。対応する DMA オブジェクトを削除します。対応する DMA オブジェクトに関連づけ (リンク) が設定されていた場合、その関連づけ (リンク) も同時に削除します。

リファレンスファイル文書の場合

リファレンスファイル管理機能を使用した文書オブジェクトを削除する場合、コンテンツの登録先のディレクトリも同時に削除します。

コンテンツの登録先ディレクトリの削除は、削除ディレクトリのルートパスで指定されたディレクトリの下位のディレクトリまでの間のディレクトリが削除の対象となります。最下位層のディレクトリから順にディレクトリを削除します。削除するディレクトリやファイルにアクセス権がない場合や削除するディレクトリにファイルやフォルダが存在する場合、その時点までのディレクトリは削除され、処理は終了します。

この指定が、コンテンツロケーションのパスと一致していない場合、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。また、文書オブジェクト以外のオブジェクトに対して実行すると、DMARC_NOT_SUPPORTED のエラーになります。

なお、UpdateContentAndRenditionType またはこのメソッドで ERR_FILE_REFERENCE_MISMATCH_STATUS エラーになった場合、このメソッドを再実行することで、オブジェクトは正常に削除されます。

引数

ReferencePath (入力)

リファレンスファイル文書を削除するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。

コンテンツの削除と同時にコンテンツを格納したディレクトリを削除するときは、SDBR_REFERENCE_PATHINFO 構造体のメンバ pDeleteRootPath に、削除するディレクトリのルートパスを指定します。この指定したルートパスより下のディレクトリを削除します。この指定は、DocumentBroker サーバが Windows の場合 UNC 形式のパスで指定することもできます。ただし、通常のパス形式または DocumentBroker サーバが Windows の場合 UNC 形式でのパス表記は、コンテンツ格納先の表記と同じである必要があります。

なお、pDeleteRootPath に NULL を指定した場合、DocumentBroker がコンテンツを管理するための

ディレクトリと登録されているコンテンツを削除します。また、メンバ `pDeleteRootPath` で指定するパスの中に、カレントディレクトリの一つ上位のディレクトリを示す `..` は指定できません。指定した場合、`DMARC_BAD_PARAMETER` のエラーになります。

リファレンスファイル文書を削除する場合、`SDBR_REFERENCE_PATHINFO` 構造体のメンバ `pEntry` および `pTargetPath` への値の指定は、無効になります。

`SDBR_REFERENCE_PATHINFO` 構造体に設定する内容については、「付録 C.1(15)

`SDBR_REFERENCE_PATHINFO` 構造体 (リファレンスパス情報構造体) を参照してください。

戻り値

`DMA_TRUE` 正常終了

`DMA_FALSE` 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新 (カレントバージョンを削除した場合は一つ前のバージョン) から順に確認し、コンテンツが存在しないバージョンを削除してください。
ERR_DBR	ERR_FILE_REFERENCE_MISMATCH_STATUS	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生し、オブジェクトとコンテンツが不整合な状態となりました。	再度実行して文書を削除してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_LAST_VERSION	削除対象となるオブジェクトは、バージョン付き文書の唯一のバージョンであるため、削除できません。	この操作は実行できません。

7. CdbrrDMA クラス (OID の設定やプロパティを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOID メソッドで設定された IID が不正です。	正しい IID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrDMA::SetOIID メソッド

機能

OIID の設定

形式

```
DmaBoolean SetOIID(
    CdbrSession*    pDbrSession,
    pDmaString_T    pOIID)
```

詳細

操作対象とするオブジェクトの OIID を設定するメソッドです。オブジェクトを、指定した識別子で特定されるオブジェクトとして操作できます。ただし、このメソッドを発行したときに、指定したオブジェクトが存在するかどうかのチェックはしません。

また、クラスライブラリのオブジェクトを操作するために SetOIID メソッドをコールした場合は、操作後に ReleaseObject メソッドをコールして、オブジェクトを解放してください。ReleaseObject メソッドをコールしないで続けて SetOIID メソッドをコールした場合は、エラーになります。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

pOIID (入力)

操作対象とするオブジェクトの識別子を指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_OIID	引数で指定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

7. CdbrDMA クラス (OID の設定やプロパティを扱うクラス)

8

CdbrDocument クラス (バージョンなし文書を扱うクラス)

DocumentBroker で扱う文書のうち、バージョンなし文書 (バージョン管理をしない文書およびバージョン管理をしている文書の個々のバージョンに当たる文書) を作成および管理する機能を提供するクラスです。

CdbrDocument クラスの詳細

CdbrDocument::~CdbrDocument メソッド

CdbrDocument::AddRendition メソッド

CdbrDocument::CdbrDocument メソッド

CdbrDocument::ChangeMasterRendition メソッド

CdbrDocument::ConvertContentType メソッド

CdbrDocument::CreateIndex メソッド

CdbrDocument::CreateObject メソッド

CdbrDocument::CreateRelation メソッド

CdbrDocument::DeleteRendition メソッド

CdbrDocument::GetContent メソッド

CdbrDocument::GetContentAndLock メソッド

CdbrDocument::GetContentInfo メソッド

CdbrDocument::GetRelationList メソッド

CdbrDocument::GetRelationListAndLock メソッド

CdbrDocument::GetRenditionList メソッド

CdbrDocument::GetRenditionListAndLock メソッド

CdbrDocument::GetVersionableList メソッド

CdbrDocument::GetVersionableListAndLock メソッド

8. CdbrDocument クラス (バージョンなし文書を扱うクラス)

CdbrDocument::PutRelationPropertyValues メソッド

CdbrDocument::PutRenditionPropertyValues メソッド

CdbrDocument::RemoveIndex メソッド

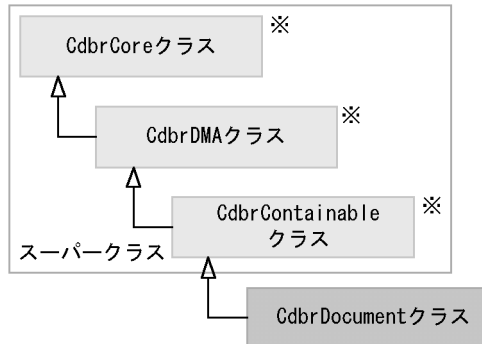
CdbrDocument::RemoveRelation メソッド

CdbrDocument::UpdateContent メソッド

CdbrDocument::UpdateContentAndRenditionType メソッド

CdbrDocument クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 8-1 CdbrDocument クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContainersCountVT	dbrProp_ContainersCountVT	Integer32 型	
RenditionType	dbrProp_RenditionType	String 型	
RetrievalName	dbrProp_RetrievalName	String 型	-
HeadRelationsCount	dbrProp_HeadRelationsCount	Integer32 型	
TailRelationsCount	dbrProp_TailRelationsCount	Integer32 型	
RenditionStatus	dbrProp_RenditionStatus	Integer32 型	
ContentType	dbrProp_ContentType	Integer32 型	
ReferenceType	dbrProp_ReferenceType	Integer32 型	
ContentLocation	dbrProp_ContentLocation	String 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 8-2 CdbrDocument クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- ²
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- ³

8. CdbrDocument クラス (バージョンなし文書を扱うクラス)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermissi on	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は ,() に基本単位を示します。

注 2

変更は, そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は, そのオブジェクトの所有者, セキュリティ ACL でアクセス制御情報変更権を与えられたユーザおよびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については, 「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティの詳細については, 「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- PrimaryGroupId
- ACL
- SACL
- OwnerPermission
- PrimaryGroupPermission
- EveryonePermission
- PublicACLCount
- PublicACLIds

これらのプロパティの詳細については、「CdbContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbDocument クラスで導入されるプロパティ

- ContainersCountVT
構成管理型の包含関係でこのオブジェクトを包含しているオブジェクトの個数を表します。
- RenditionType
RenditionType を表す文字列を表します。
- RetrievalName
登録されているファイルのファイル名を表す文字列を表します。
CreateObject メソッドまたは UpdateContent メソッドでファイルを登録・更新する場合、指定された URL からパス名を除いた文字列が設定されます。例えば、UNIX の場合に URL が "file:///tmp/sample.doc" のファイルを登録すると、文字列 "sample.doc" が設定されます。Windows の場合に URL が "file:///c:/temp/sample.doc" のファイルを登録すると、文字列 "sample.doc" が設定されます。ファイルが未登録の場合は、長さ 0 の文字列が設定されます。値の変更もできます。
- HeadRelationsCount
このオブジェクトが文書間リレーションで関連づけしているリレーション先文書の数を表します。
- TailRelationsCount
このオブジェクトが文書間リレーションでほかの文書からリレーション先として関連づけされている数を表します。
- RenditionStatus
マルチレンディション文書のサブレンディションのプロパティで、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。例えば、このプロパティを指定して GetRenditionList メソッドをコールすると、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す次のような値が設定されます。なお、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 8-3 RenditionStatus プロパティの値 (状態フラグ : 下位 2 バイト)

定数	対応する値 (16 進数)	説明
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 8-4 RenditionStatus プロパティの値 (変換フラグ : 上位 2 バイト)

定数	対応する値 (16 進数)	説明
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。

定数	対応する値 (16進数)	説明
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。

- ContentType

レンディションのコンテンツ種別を表すプロパティです。コンテンツの種別から、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、File Link 文書のコンテンツか、またはそれ以外のコンテンツかを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されます。

ユーザは、このプロパティを設定できません。このプロパティを参照する場合は、GetRenditionList メソッドおよび GetRenditionListAndLock メソッドを使用します。また、マスタレンディションの場合は、GetPropertyValues メソッド、GetPropertyValuesAndLock メソッド、または一部の一覧取得メソッドで参照できます。

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 8-5 ContentType プロパティの値とコンテンツ種別の関係

dbrProp_ContentType 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。

- ReferenceType

コンテンツのリファレンス種別を表すプロパティです。

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 8-6 ReferenceType プロパティの値とリファレンス種別の関係

dbrProp_ReferenceType 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。

- ContentLocation

コンテンツの格納先を示す情報 (コンテンツロケーション) を管理するプロパティです。

なお、このほかに、CdbDocument クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA のクラスの dmaClass_DocVersion クラスおよびそのサブクラスのプロパティを参照、設定できます。DMA のクラスのプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrDocument クラスで導入されるメソッド

8. CdbrDocument クラス (バージョンなし文書を扱うクラス)

メソッド	機能
~CdbrDocument	デストラクタ
AddRendition	レンディションの追加
CdbrDocument	コンストラクタ
ChangeMasterRendition	マスタレンディションへの変更
ConvertContentType	コンテンツ種別の変換
CreateIndex	全文検索インデクスの作成
CreateObject	オブジェクトの新規作成
CreateRelation	文書間リレーションの設定
DeleteRendition	レンディションの削除
GetContent	ファイルの取得
GetContentAndLock	ファイルの取得と同時に指定したロックの設定
GetContentInfo	マルチファイル文書の情報の取得
GetRelationList	文書のリレーション情報の一覧取得
GetRelationListAndLock	文書のリレーション情報の一覧取得と同時に指定したロックの設定
GetRenditionList	レンディション一覧の取得
GetRenditionListAndLock	レンディション一覧の取得と同時に指定したロックの設定
GetVersionableList	バージョン管理情報一覧の取得
GetVersionableListAndLock	バージョン管理情報一覧の取得と同時に指定したロックの設定
PutRelationPropertyValues	文書間リレーションのプロパティの設定
PutRenditionPropertyValues	レンディションのプロパティの設定
RemoveIndex	全文検索インデクスの削除
RemoveRelation	文書間リレーションの解除
UpdateContent	ファイルの更新
UpdateContentAndRenditionType	ファイルの更新および RenditionType の変更

以降, CdbrDocument クラスで導入されるメソッドについて説明します。

CdbrDocument::~~CdbrDocument メソッド

機能

デストラクタ

形式

```
virtual ~CdbrDocument()
```

詳細

CdbrDocument オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbDocument::AddRendition メソッド

機能

レンディションの追加

形式

```
// シングルファイル文書であるコンテンツを
// レンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pFilePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList)

// シングルファイル文書であるコンテンツを
// レンディション変換機能を使って登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pFilePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList,
    DmaInteger32          lConvertType)

// リファレンスファイル文書であるコンテンツをレンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList)

// File Link文書であるコンテンツをレンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList)

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;          // 要素数
    union {                // プロパティの値
        pDmaBoolean   pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId        pId;          // ID型
        pDmapv        ppObject;     // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32   lContentOperateMode; // 操作モード
    pDmaString_T   pEntry;              // ファイルのパスおよびロケーション
    pDmaString_T   pTargetPath;        // コンテント格納先パス
    pDmaString_T   pDeleteRootPath;    // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
```

```

pDmaString_T  pFilePath;           // 登録ファイルまたはディレクトリのURL
pDmaString_T  pFamName;           // FAM名
pDmaString_T  pFamFilePath;       // コンテンツ保管先パス
DmaInteger32  lContentDeleteMode; // コンテンツ削除モード
DmaInteger32  lFileNamingMode;    // コンテンツ名自動生成モード
};

```

詳細

バージョンなし文書に、指定されたレンディションと、コンテンツを追加するメソッドです。

このメソッドを使うことで、レンディションを複数持つ、マルチレンディション文書にすることができます。

なお、レンディションを複数持つマルチレンディション文書の場合、すべてのレンディションに空でない RenditionType が設定されている必要があります。つまり、RenditionType が NULL または空文字列 ("") のレンディションは存在できません。したがって、マスタレンディションの RenditionType が設定されていない (NULL または空文字列 ("") の場合) か、このメソッドの引数 pRenditionType が NULL または空文字列 ("") の場合、このメソッドはエラーになります。

レンディション変換機能を使ってコンテンツを登録する場合、コンテンツが空のレンディションを追加できます。ただし、レンディション変換機能を使うには、DocumentBroker Rendering Option が必要です。

このメソッドは、マルチファイルを管理するバージョンなし文書に対しては実行できません。

リファレンスファイル文書であるコンテンツを登録する場合

登録するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書であるコンテンツを登録する場合、CdbSession::SetReferencePath メソッドでコンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合は、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

File Link 文書であるコンテンツを登録する場合

登録するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、登録するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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登録できます。また、すでに HiRDB File Link で管理されているデータをコンテンツとして登録することもできます。

File Link 文書であるコンテンツを登録する場合、このメソッドで、ERR_FILELINK_FAILED のエラーが発生したときは、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pFilePath (入力)

追加するサブレンディションに登録するファイルのパス名を URL 形式で指定します。

例えば、レンディションの追加と同時にコンテンツを登録する場合で、UNIX の場合にファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合にファイル "c:\temp\sample.doc" を登録するには、"file:///c:\temp\sample.doc" と指定します。有効なプロトコルは "file:" だけです。"file:" 以外のプロトコルを指定した場合、指定を省略した場合または NULL を指定した場合、エラーになります。

なお、レンディション変換機能を使ってコンテンツを登録する場合 (lConvertType に DBR_CONVERT_TYPE_BATCH を指定した場合) だけ、pFilePath に NULL を指定できます。

pRenditionType (入力)

追加するサブレンディションの RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。NULL または空文字列 ("") は指定できません。

レンディションを追加しようとする文書に、この引数で指定する RenditionType と同じレンディションがすでにある場合、エラーになります。

pPropList (入力)

追加するレンディションに設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。プロパティを設定しない場合は NULL を指定してください。

なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

このメソッドで値を設定できるプロパティのプロパティ識別子を次に示します。

- dbrProp_RetrievalName

IConvertType (入力)

レンディション変換の方法を指定します。

次に示す値のどちらかを指定してください。

- DBR_CONVERT_TYPE_USER
レンディション変換機能を使用しないでコンテンツを登録する場合に指定します。この場合、コンテンツとして登録するファイルはユーザ自身で用意し、pFilePath にそのパス名を指定します。
- DBR_CONVERT_TYPE_BATCH
このメソッドの実行後に DocumentBroker Rendering Option が提供するコマンドでレンディション変換を実行し、変換されたファイルをコンテンツとして自動的に登録する場合に指定します。
なお、レンディション変換機能を使用しない場合は、DBR_CONVERT_TYPE_USER を指定してください。
また、マスタレンディションがシングルファイル文書の場合だけ、DBR_CONVERT_TYPE_BATCH を指定してレンディションの追加を行うことができます。マスタレンディションがシングルファイル文書でない文書に対して、DBR_CONVERT_TYPE_BATCH を指定した場合は、DocumentBroker Rendering Option によるレンディション変換で、major_code として ERR_DBR、minor_code として ERR_CONTENTTYPE_MISMATCH のエラーになります。

ReferencePath (入力)

リファレンスファイル文書を登録するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。SDBR_REFERENCE_PATHINFO 構造体のメンバ IContentOperateMode に DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定し、pEntry に追加するサブレンディションに登録するファイルのパス名を URL 形式で指定します。

また、SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath に、コンテンツ格納先パスを、CdbSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを基点とする相対パスで指定します。DocumentBroker サーバが Windows の場合、UNC 形式のパスで指定することもできます。

なお、lContentOperateMode に DBR_OPERATEMODE_NONE を指定した場合、コンテンツを持たないオブジェクトが作成されます。

このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023
バイト (AIX の場合) 以上のときにこのメソッドを実行すると、
ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023
バイト (AIX の場合) 以上のときにこのメソッドを実行すると、
ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書を登録する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ
pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15)
SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を登録するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に
NULL を設定した場合には、DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5)
SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_LIMITNUM_OF_RENDITIONS	保持できるレンディション数の上限に達しています。	この操作は実行できません。
ERR_DBR	ERR_MASTERRENDTYPE_NOT_SET	マスタレンディションの RenditionType に値が設定されていません。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_RENDITIONTYPE_DUPLICATED	RenditionType が重複しています。	重複しない RenditionType を指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> • ネットワーク環境を確認してください。 • _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 • 時間を置いて、再度実行してください。 • ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbDocument::CdbDocument メソッド

機能

コンストラクタ

形式

CdbDocument ()

詳細

CdbDocument オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbDocument::ChangeMasterRendition メソッド

機能

マスタレンディションへの変更

形式

```
DmaBoolean ChangeMasterRendition(
    pDmaString_T      pRenditionType )
```

詳細

マルチレンディション文書の指定されたレンディションをマスタレンディションに変更します。変更した場合、既存のマスタレンディションはサブレンディションになります。

なお、メソッド実行時の RenditionStatus プロパティの状態フラグの値によって、このメソッドの実行可否に関して次に示す制限があります。

メソッド実行時の RenditionStatus プロパティの状態フラグの値	ChangeMasterRendition メソッドの実行可否
DBR_RENDSTATUS_SUBREND_EXIST	実行できる
DBR_RENDSTATUS_NO_SUBREND	実行できない
DBR_RENDSTATUS_MASTERREND_UPDATE	変換フラグが DBR_RENDSTATUS_CONVERT_NOTREQUIRED であれば実行できる

このメソッドは、マルチファイル文書に対して実行できません。

引数

pRenditionType (入力)

対象文書のサブレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象文書のマスタレンディションの場合、または対象文書のサブレンディションとして存在しない場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENT_IS_EMPTY	コンテンツが変換処理されてなくて空です。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_IS_MASTERRENDITION	指定された RenditionType はマスタレンディションです。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_RENDITION_NOT_CONVERTED	コンテンツが更新処理されていません。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbDocument::ConvertContentType メソッド

機能

コンテンツ種別の変換

形式

```
// リファレンスファイル文書に変換する場合
DmaBoolean ConvertContentType(
    DmaInteger32                lExecuteMode,
    DmaInteger32                lSourceScope,
    DmaBoolean                  bIsChangeMaster,
    pDmaString_T                pRenditionType,
    DmaInteger32                lSourceContentType,
    SDBR_RENDITION_COMMENTINFO& CommentInfo,
    SDBR_REFERENCE_PATHINFO&   ReferencePath )

// シングルファイル文書に変換する場合
DmaBoolean ConvertContentType(
    DmaInteger32                lExecuteMode,
    DmaInteger32                lSourceScope,
    DmaBoolean                  bIsChangeMaster,
    pDmaString_T                pRenditionType,
    DmaInteger32                lSourceContentType,
    SDBR_RENDITION_COMMENTINFO& CommentInfo )

// レン디션コメント情報構造体
struct SDBR_RENDITION_COMMENTINFO {
    DmaBoolean    bIsInvestSourceComment; // コメントの付与フラグ
    DmaInteger32  lInvestMode;           // コメントの付与方式
    pDmaString_T  pRenditionComment;    // レン디션コメント
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32    lContentOperateMode; // 操作モード
    pDmaString_T    pEntry;              // ファイルのパスおよびロケーション
    pDmaString_T    pTargetPath;        // コンテンツ格納先パス
    pDmaString_T    pDeleteRootPath;    // 削除ディレクトリのルートパス
};
```

詳細

コンテンツの格納先 (コンテンツ種別) を変換するメソッドです。

このメソッドを使用することでコンテンツの格納先を変換したレンディションが追加されます。

このメソッドを実行する前にコンテンツ格納先ベースパスを CdbSession::SetReferencePath メソッドで設定しておく必要があります。設定されていない場合は、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

lSourceScope, pRenditionType, および lSourceContentType で指定した変換対象のコンテンツが存在しない場合は、ERR_CONVERT_CONTENT_TARGET_NOTFOUND のエラーになります。

このメソッドで変換可能なコンテンツ種別の組み合わせを次に示します。

表 8-7 ConvertContentType メソッドで変換できるコンテンツ種別

変換前のコンテンツ種別 (lSourceContentType で指定する値)	変換後のコンテンツ種別 (メソッドの形式)
シングルファイル文書 (DBR_CONTENTTYPE_CONTENT)	リファレンスファイル文書 (リファレンスファイル文書に変換する場合の形式)

変換前のコンテンツ種別 (ISourceContentType で指定する値)	変換後のコンテンツ種別 (メソッドの形式)
リファレンスファイル文書 (DBR_CONTENTTYPE_REFERENCE)	シングルファイル文書 (シングルファイル文書に変換する場合の形式)

シングルファイル文書からリファレンスファイル文書に変換する場合

コンテンツが登録されていないシングルファイル文書をリファレンスファイル文書に変換すると、リファレンスファイル文書の edmProp_ReferenceType には、DBR_REFERENCETYPE_NONE が設定され、コンテンツは登録されません。

また、このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、

ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、

ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

リファレンスファイル文書からシングルファイル文書に変換する場合

リファレンスファイル文書の ContentLocation プロパティと CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパスを合わせた文字列長 (パスの区切り文字も含む) が、Windows の場合 260 バイト以上、AIX の場合 1023 バイト以上であると

ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

引数

IExecuteMode (入力)

変換モードを指定します。

この引数には DBR_CONVERT_MODE_VERBOSE を指定します。

ISourceScope (入力)

変換対象とするレンディションの範囲を指定します。

- DBR_CONVERT_SOURCE_ALL
すべてのレンディションを対象とします。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

- DBR_CONVERT_SOURCE_SEPARATE

単一のレンディションを対象とします。変換するレンディションのレンディションタイプは、pRenditionType で指定します。

blsChangeMaster (入力)

マスタレンディションのコンテンツを変換する場合に、変換して追加されるレンディションをマスタレンディションにするかしないかを指定します。

- DMA_TRUE

変換して追加されるレンディションをマスタレンディションに変更します。

- DMA_FALSE

マスタレンディションの変更は行いません。

pRenditionType (入力)

lSourceScope に DBR_CONVERT_SOURCE_SEPARATE を指定した場合は、変換の対象とするレンディションのレンディションタイプを表す文字列を指定します。指定したレンディションのコンテンツが変換されます。

この引数で指定したレンディションタイプが、対象文書のレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は NULL を使用することもできます。空文字列 ("") は指定できません。

ただし、lSourceScope に DBR_CONVERT_SOURCE_ALL が指定されている場合は、pRenditionType への値の指定は無効となります。

lSourceContentType (入力)

変換対象とするコンテンツ種別を指定します。

- DBR_CONTENTTYPE_CONTENT

シングルファイル文書のコンテンツを対象とします。リファレンスファイル文書に変換する形式のメソッドの場合に指定します。メソッドの形式が一致しない場合、DMARC_BAD_PARAMETER のエラーになります。

- DBR_CONTENTTYPE_REFERENCE

リファレンスファイル文書のコンテンツを対象とします。シングルファイル文書に変換する形式のメソッドの場合に指定します。メソッドの形式が一致しない場合、DMARC_BAD_PARAMETER のエラーになります。

- DBR_CONTENTTYPE_OTHER

コンテンツの種別に関係なく変換を行います。ただし、表 8-7 の ConvertContentType メソッドで変換できるコンテンツ種別に示す組み合わせを対象とします。変換対象外のコンテンツ種別は無視されません。

CommentInfo (入力)

サブレンディションに付与するコメントの情報を SDBR_RENDITION_COMMENTINFO 構造体で指定します。

SDBR_RENDITION_COMMENTINFO 構造体については、「付録 C.1(18)

SDBR_RENDITION_COMMENTINFO 構造体 (レンディションコメント情報構造体)」を参照してください。

ReferencePath (入力)

リファレンスファイル文書に変換するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書に変換するには、SDBR_REFERENCE_PATHINFO 構造体のメンバ lContentOperateMode に DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定します。

また、SDBR_REFERENCE_PATHINFO 構造体の pTargetPath に、コンテンツ格納先パスを、CdbSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを基点とした相対パスで指定します。

pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書に変換する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry、および pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体については、「付録 C.1(15)

SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_LIMITNUM_OF_RENDITIONS	保持できるレンディション数の上限に達しています。	この操作は実行できません。
ERR_DBR	ERR_MASTERRENDITIONTYPE_NOT_SET	マスタレンディションタイプ値が設定されていません。	マスタレンディションタイプ値を設定してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_RENDITIONTYPE_DUPLICATED	レンディションタイプが重複しています。	正しいレンディションタイプを設定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECT	次の要因が考えられます。 • セッションが接続されていません。 • EDMRefresher コマンドが実行中です。	要因に応じて、次のどちらかの対策を実施してください。 • セッションを接続してください。 • EDMRefresher コマンドが終了したあと、再度実行してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DBR	ERR_CONVERT_CONTENT_TARGET_NOT_FOUND	変換対象のコンテンツは存在しません。	オブジェクトのコンテンツ種別または指定した変換対象とするコンテンツ種別 (ISourceContentType) を見直してください。
ERR_DBR	ERR_CONVERT_CONTENT_NOT_RETRIEVALNAME	変換対象コンテンツの dmaProp_RetrievalName プロパティの値が設定されていない (値が NULL または、空文字列 ("")) のため、変換処理を実行できません。	dmaProp_RetrievalName プロパティの値を設定してメソッドを実行してください。

CdbrDocument::CreateIndex メソッド

機能

全文検索インデクスの作成

形式

```
DmaBoolean CreateIndex(
    const pDmaString_T      pFilePath)
```

詳細

指定したパス名のファイルから、全文検索インデクスを作成するメソッドです。パス名に NULL を指定した場合は、登録されている文書 (マルチレンディション文書の場合はマスタレンディション) に含まれているコンテンツから全文検索インデクスが作成されます。ただし、マルチファイル文書、リファレンスファイル文書または File Link 文書の場合、NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

既存の全文検索インデクスを、別のファイルで更新する場合も、このメソッドを実行してください。これによって、インデクスの遅延更新が実現できます。

このメソッドは、全文検索機能付き文書クラスから作成されている DocVersion オブジェクトを構成要素とした CdbrDocument オブジェクトにだけ実行できます。それ以外のオブジェクトを対象に実行した場合は、エラーになります。

また、インデクスを作成できるのは、テキストファイルとして作成されたファイルだけです。それ以外の形式の文書を指定してこのメソッドを実行した場合の動作は保証されません。場合によっては、データベースがロールバックする可能性があります。この場合は、データベースエラーが出力されますので、データベースのエラーコードを参照して、データを再登録してください。

なお、登録されている文書とインデクスを作成するためのファイルとの対応づけは、上位で行います。

UNIX の場合に、ファイル /tmp/sample.txt から全文検索インデクスを作成する例を次に示します。

Windows の場合は、ファイル c:\%temp%\sample.txt から全文検索インデクスを作成する例として、例中の "file:///tmp/sample.txt" を "file:///c:\%temp%\sample.txt" に読み替えてください。

全文検索インデクスを作成する例

```
pDbrDoc->CreateIndex("file:///tmp/sample.txt");
```

引数

pFilePath (入力)

全文検索インデクスを作成するファイルのパス名を設定します。登録されている文書 (マルチレンディション文書の場合はマスタレンディション) に含まれるコンテンツからインデクスを作成する場合には NULL を設定します。それ以外のファイルからインデクスを作成する場合には、ファイルのパス名を URL 形式で指定します。例えば、UNIX の場合、全文検索インデクスをファイル "/tmp/sample.txt" から作成するには、"file:///tmp/sample.txt" と指定します。Windows の場合、全文検索インデクスをファイル "c:\%temp%\sample.txt" から作成するには、"file:///c:\%temp%\sample.txt" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbrDocument::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
// シングルファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    pDmaString_T      pFilePath,
    pDmaString_T      pRenditionType
    ppDmaString_T      ppOIID,
    DmaInteger32      lCreateIndexFlg = DBR_CREATE_INDEX)

// マルチファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_PATHLIST&    PathList,
    pDmaString_T      pRenditionType,
    ppDmaString_T      ppOIID)

// リファレンスファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T      pRenditionType,
    ppDmaString_T      ppOIID)

// File Link文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T      pRenditionType,
    ppDmaString_T      ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality; // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean;      // Boolean型
        pDmaInteger32    plInteger32;     // Integer32型
        pDmaId           pId;            // ID型
        pDmapv           ppObject;       // Object型
        ppDmaString_T    ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;
    SDBR_PROP*        pItem;
};
```

```

// DMAオブジェクト生成用の構造体
struct SDBR_DMAINFO {
    DmaId          ClassId;
    SDBR_PROPLIST  PropList;
};

// ファイルパス情報構造体
struct SDBR_PATHINFO {
    pDmaString_T   pFilePath;          // ファイルのURL
    pDmaString_T   pComponentType;    // ファイルを識別する文字列
};

// ファイルパス情報リスト構造体
struct SDBR_PATHLIST {
    DmaInteger32   lCount;             // 登録ファイル数
    SDBR_PATHINFO* pItem;            // ファイルパス情報構造体のポインタ
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32   lContentOperateMode; // 操作モード
    pDmaString_T   pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T   pTargetPath;        // コンテンツ格納先パス
    pDmaString_T   pDeleteRootPath;    // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T   pFilePath;          // 登録ファイルまたはディレクトリのURL
    pDmaString_T   pFamName;          // FAM名
    pDmaString_T   pFamFilePath;      // コンテンツ保管先パス
    DmaInteger32   lContentDeleteMode; // コンテンツ削除モード
    DmaInteger32   lFileNamingMode;   // コンテンツ名自動生成モード
};

```

詳細

オブジェクトを新規に作成するメソッドです。

SetOID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更されずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

Windows の場合、次のことに注意してください。

- GetContentAndLock によるコンテンツの取得時、大文字と小文字を区別しないと同一になるファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。
- GetContentAndLock によるコンテンツの取得時、ロングファイル名とショートファイル名の関係で同一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合

作成するオブジェクトが次の条件に当てはまるときに、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデックスを作成できます。

- DMA オブジェクト生成用の構造体である、SDBR_DMAINFO 構造体の ClassId に、dmaClass_DocVersion クラスのサブクラスとして全文検索機能付き文書クラスが指定されている。
- RenditionType がテキスト型 (MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

ファイルの登録が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合にパス名が "file:///tmp/sample.doc" のファイルを登録すると、RetrievalName プロパティには "sample.doc" が設定されます。Windows の場合にパス名が "file:///c:¥temp¥sample.doc" のファイルを登録すると、RetrievalName プロパティには "sample.doc" が設定されます。

UNIX の場合に、CdbrDocument オブジェクトを新規に作成して接続する例を次に示します。この例では、ClassID_DocVer で指定した DMA DocVersion のクラス識別子で文書オブジェクトを新規に作成します。そして、ファイル /tmp/sample.txt を RenditionType に "MIME::text/plain" として登録して、そのポインタを取得します。なお、Windows の場合は、ファイル c:¥temp¥sample.txt を登録する例として、例中の "file:///tmp/sample.txt" を "file:///c:¥temp¥sample.txt" と読み替えてください。

CdbrDocument オブジェクトを新規に作成して接続する例

```
// 変数宣言
CdbrDocument DbrDoc;
SDBR_DMAINFO          DMAInfoList [1];
pDmaString             pOIID;

// DMAクラス情報の作成
DMAInfoList[0].ClassId = ClassID_DocVer;
DMAInfoList[0].PropList.pItem = NULL;
DMAInfoList[0].PropList.lCount = 0;
// オブジェクトの新規作成
DbrDoc.CreateObject(pDbrSession,
    1, DMAInfoList,
    "file:///tmp/sample.txt", "MIME::text/plain", &pOIID,
    DBR_CREATE_INDEX);
```

マルチファイル文書の場合

登録するコンテンツの情報として、SDBR_PATHLIST 構造体を指定します。

マルチファイル文書を作成する場合は、次のことに注意してください。

- マルチファイル文書では、テキストファイルの部分が特定できないため、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、マルチファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- dbrProp_RetrievalName プロパティに値を設定する場合は、SDBR_DMAINFO 構造体のメンバ PropList に設定された SDBR_PROPLIST 構造体に設定してください。SDBR_PROPLIST 構造体に dbrProp_RetrievalName を設定していない場合、データベースには NULL 値が格納されます。
- SDBR_PATHLIST 構造体の lCount に指定したファイル数より、SDBR_PATHINFO の配列の個数が多い場合は、lCount に指定したファイル数分のファイルが登録されます。
- 同じファイル名を持つファイルを格納しようとした場合は、ERR_RETRIEVALNAME_DUPLICATED のエラーになります。なお、ファイル名は、英字の大文字と小文字を区別して比較されます。

リファレンスファイル文書の場合

登録するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。

リファレンスファイル文書を作成する場合は、次のことに注意してください。

- リファレンスファイル文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、リファレンスファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツに登録するファイルの種類を表すレンディションタイプは、登録したファイルの拡張子

からは設定されません。レンディションタイプは、CreateObject メソッドの引数として明示的に指定してください。この引数に指定した値が、レンディションごとのファイルの種類を表すプロパティとして設定されます。

File Link 文書の場合

登録するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、登録するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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登録できます。また、すでに HiRDB File Link で管理されているデータをコンテンツとして登録することもできます。

File Link 文書を作成する場合は、次のことに注意してください。

- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツに登録するファイルの種類を表すレンディションタイプは、登録したファイルの拡張子からは設定されません。レンディションタイプは、CreateObject メソッドの引数として明示的に指定してください。この引数に指定した値が、レンディションごとのファイルの種類を表すプロパティとして設定されます。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の値を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体である、SDBR_DMAINFO 構造体の配列を指定します。DMA オブジェクトの情報として SDBR_DMAINFO 構造体には、dmaClass_DocVersion クラスまたはそのサブクラスの情報を指定してください。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMAINFO 構造体については、「付録 C.1(3) SDBR_DMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

素を指定した場合

- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

pFilePath (入力)

登録するファイルのパス名を指定します。パス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:¥temp¥sample.doc" を登録するには、"file:///c:¥temp¥sample.doc" と指定します。有効なプロトコルは "file:" だけです。それ以外のプロトコルを指定した場合はエラーになります。NULL を指定した場合、ファイルは登録されません。

pRenditionType (入力)

RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。

NULL を指定した場合、ファイルの拡張子から RenditionType を決定する処理はされません。したがって、RenditionType には何も設定されません。

ppOID (出力)

作成した CdbDocument オブジェクトの構成要素である、DMA オブジェクトの DocVersion オブジェクトの OID へのポインタが設定されます。ポインタを格納する領域を指定してください。なお、*ppOID がポイントするメモリは、ユーザプログラムでは解放しないでください。

lCreateIndexFlg (入力)

全文検索インデックスを作成するかどうかを指定します。

次のどちらかを指定してください。省略した場合は、DBR_CREATE_INDEX を指定したものとして処理されます。

- DBR_CREATE_INDEX
pFilePath に指定したパス名のファイルから全文検索インデックスを作成します。
- DBR_NOT_CREATE_INDEX
全文検索インデックスを作成しません。

なお、SDBR_DMAININFO 構造体の ClassId に全文検索機能付きクラス以外のクラスを指定した場合またはファイルの RenditionType がテキスト型でない場合は、lCreateIndexFlg の値は無視されます。

PathList (入力)

登録するファイルのファイルパス情報が設定された SDBR_PATHLIST 構造体を指定します。

SDBR_PATHLIST 構造体については、「付録 C.1(11) SDBR_PATHLIST 構造体 (ファイルパス情報リスト構造体)」を参照してください。

ReferencePath (入力)

リファレンスファイル文書を登録するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書を登録するには、SDBR_REFERENCE_PATHINFO 構

構造体のメンバ `lContentOperateMode` に `DBR_OPERATEMODE_USER_RELATIVE_CONTENT` を指定し、`pEntry` に登録するファイルのパス名を URL 形式で指定します。また、`SDBR_REFERENCE_PATHINFO` 構造体のメンバ `pTargetPath` に、コンテンツ格納先パスを、`CdbSession::SetReferencePath` メソッドで設定したコンテンツ格納先ベースパスを基点とする相対パスで指定します。DocumentBroker サーバが Windows の場合 UNC 形式のパスで指定することもできます。なお、`lContentOperateMode` に `DBR_OPERATEMODE_NONE` を指定した場合、コンテンツがないオブジェクトが作成されます。

このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の `ReferenceStorageMode` エントリに `Origin` を指定した場合次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、`ERR_FILE_REFERENCE_OPERATION_FAILED` のエラーになります。

- `CdbSession::SetReferencePath` メソッドで指定したコンテンツ格納先ベースパス
- `pTargetPath` で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- `pEntry` で指定したファイル名

DocumentSpace 構成定義の `ReferenceStorageMode` エントリに `Divide` を指定した場合次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、`ERR_FILE_REFERENCE_OPERATION_FAILED` のエラーになります。

- `CdbSession::SetReferencePath` メソッドで指定したコンテンツ格納先ベースパス
- `pTargetPath` で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、`pTargetPath` で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す `".."` は指定できません。指定した場合、`DMARC_BAD_PARAMETER` のエラーになります。

リファレンスファイル文書を登録する場合、`SDBR_REFERENCE_PATHINFO` 構造体のメンバ `pDeleteRootPath` への値の指定は、無効になります。

`SDBR_REFERENCE_PATHINFO` 構造体に設定する内容については、「付録 C.1(15) `SDBR_REFERENCE_PATHINFO` 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を登録するために必要な情報を格納した、`SDBR_FILELINK_PATHINFO` 構造体を指定します。

なお、`SDBR_FILELINK_PATHINFO` 構造体のメンバ `pFilePath` とメンバ `pFamFilePath` の両方に `NULL` を設定した場合には、登録コンテンツがないオブジェクトが作成されます。

`SDBR_FILELINK_PATHINFO` 構造体に設定する内容については、「付録 C.1(5) `SDBR_FILELINK_PATHINFO` 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

`DMA_TRUE` 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。または、マルチファイル文書で指定したファイルのサイズの合計が、BLOB の定義長を超えました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_BAD_CONTENTSCOUNT	登録しようとするファイルの数が多過ぎます。	登録するファイルの数を見直してください。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLICACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none">ネットワーク環境を確認してください。_HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。時間を置いて、再度実行してください。ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbrDocument::CreateRelation メソッド

機能

文書間リレーションの設定

形式

```
// バージョンなし文書と関連づける場合
DmaBoolean CreateRelation(
    CdbrDocument*      pDbrDocument,
    SDBR_PROPLIST*    pRelPropList,
    ppDmaString_T      ppRelId)

// バージョン付き文書と関連づける場合
DmaBoolean CreateRelation(
    CdbrVersionableDocument* pDbrVersionableDocument,
    SDBR_PROPLIST*          pRelPropList,
    ppDmaString_T           ppRelId)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality; // 基本単位
    DmaInteger32 lCount;        // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;        // ID型
        pDmapv           ppObject;    // Object型
        ppDmaString_T    ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

文書間のリレーションを設定するメソッドです。リレーション先には文書系のオブジェクトを指定します。指定できる文書系のオブジェクトを次に示します。

- CdbrDocument オブジェクト (バージョンなし文書)
- CdbrVersionableDocument オブジェクト (バージョン付き文書)

なお、リレーションには、ユーザ設定のプロパティを設定できます。

引数

pDbrDocument (入力)

このオブジェクトと関連づける CdbrDocument オブジェクト (バージョンなし文書) を指定します。

pDbrVersionableDocument (入力)

このオブジェクトと関連づける CdbrVersionableDocument オブジェクト (バージョン付き文書) を指定します。

pRelPropList (入力)

リレーションに設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。プロパティを設定しない場合は、NULL を指定してください。

なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppRelId (出力)

リレーション識別子が設定されます。リレーション識別子が格納される領域を指定してください。

このリレーション識別子は、文書間リレーションを解除するための RemoveRelation メソッドや、リレーションにプロパティを設定するための PutRelationPropertyValues メソッドで使用します。なお、*ppRelId がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrDocument::DeleteRendition メソッド

機能

レンディションの削除

形式

```
DmaBoolean DeleteRendition(
    pDmaString_T          pRenditionType )
```

詳細

マルチレンディション文書の指定されたサブレンディションとそのコンテンツを削除します。

リファレンスファイル文書であるレンディションとそのコンテンツを削除する場合、CdbrSession::SetReferencePath メソッドでコンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

このメソッドは、マルチファイルを管理するバージョンなし文書に対しては実行できません。

引数

pRenditionType (入力)

削除するサブレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が対象文書のマスタレンディションの場合、または対象文書のサブレンディションとして存在しない場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_IS_MASTERRENDITION	指定されたレンディションはマスタレンディションです。	この操作は実行できません。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrDocument::GetContent メソッド

機能

ファイルの取得

形式

```
// マスタレンディションのファイルを取得する場合
DmaBoolean GetContent(
    const pDmaString_T      pFilePath)

// RenditionTypeを指定してファイルを取得する場合
DmaBoolean GetContent(
    const pDmaString_T      pFilePath,
    pDmaString_T            pRenditionType )
```

詳細

文書空間に登録されているファイルを、指定したパスに複写するメソッドです。

マルチレンディション文書の場合、pRenditionType を指定することで、対応する RenditionType を持つレンディションのファイルが取得できます。

UNIX の場合に、登録されているファイルを /tmp/sample.doc に複写して取得する例を次に示します。Windows の場合は、c:¥temp¥sample.doc に複写して取得する例として、例中の "file:///tmp/sample.doc" を "file:///c:¥temp¥sample.doc" と読み替えてください。

ファイルを取得する例

```
pDbrVerDoc->GetContent("file:///tmp/sample.doc");
```

引数

pFilePath (入力)

取得したファイルを複写する複写先のパス名を指定します。

ファイルのパス名は URL 形式で設定します。ただし、プロトコル名は "file:" だけが指定できます。例えば、UNIX の場合、ファイル "/tmp/sample.doc" に複写するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:¥temp¥sample.doc" に複写するには、"file:///c:¥temp¥sample.doc" と指定します。

指定したパスのファイルが存在する場合は上書きします。ファイル名は、RenditionType プロパティまたは RetrievalName プロパティを参考にして、メソッドをコールした側が任意に決定してください。

指定したパスのファイルが存在しない場合、ディレクトリが存在し、かつファイルの作成が可能ならば新規にファイルを作成します。それ以外の場合はエラーになります。

pRenditionType (入力)

RenditionType を指定してファイルを取得する場合、取得するレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象文書のレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は NULL を指定するか、「マスタレンディションのファイルを取得する場合」の形式を使用することもできます。

空文字列 ("") は指定できません。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA		CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbDocument::GetContentAndLock メソッド

機能

ファイルの取得と同時に指定したロックの設定

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの取得)
DmaBoolean GetContentAndLock(
    const pDmaString_T      pFilePath,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの取得)
DmaBoolean GetContentAndLock(
    const pDmaString_T      pFilePath,
    pDmaString_T            pRenditionType,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// マルチファイル文書の場合
DmaBoolean GetContentAndLock(
    pDmaString_T            pDirPath,
    SDBR_CONTENTLIST&      ContentList,
    pDmaString_T            pRenditionType,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// リファレンスファイル文書の場合
DmaBoolean GetContentAndLock(
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T            pRenditionType,
    pDmaInteger32           plReferenceType,
    ppDmaString_T           ppContentLocation,
    DmaInteger32            lLockType=DMA_LOCK_WRITE)

// File Link文書の場合
DmaBoolean GetContentAndLock(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T            pRenditionType,
    DmaBinaryValue**        ppbvFileHandle,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// コンテント情報構造体
struct SDBR_CONTENTINFO {
    pDmaString_T    pComponentType; // ファイルを識別する文字列
    pDmaString_T    pRetrievalName; // ファイル名
    DmaUInteger32  lContentSize;   // ファイルのサイズ
};

// コンテント情報リスト構造体
struct SDBR_CONTENTLIST {
    DmaInteger32    lCount;          // 取得ファイル数
    SDBR_CONTENTINFO* pItem;       // コンテント情報構造体の
                                    // ポインタ
    DmaInteger32    lValidMode;     // 有効モード
    DmaBoolean      bIsMultiContents; // マルチファイル識別
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32    lContentOperateMode; // 操作モード
    pDmaString_T    pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T    pTargetPath;        // コンテント格納先パス
    pDmaString_T    pDeleteRootPath;    // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
```



```
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T pFilePath;    // 取得先のファイルまたはディレクトリのURL
    pDmaString_T pFamName;
    pDmaString_T pFamFilePath;
    DmaInteger32 lContentDeleteMode;
    DmaInteger32 lFileNamingMode;
};
```

詳細

ロックタイプを明示的に指定して、文書空間に登録されているファイルを、指定したパスに複写するメソッドです。マルチレンディション文書の場合、pRenditionType を指定すると、対応する RenditionType を持つレンディションのファイルが取得できます。

Windows の場合、次のことに注意してください。

- 大文字と小文字を区別しないと同一になるファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。
- ロングファイル名とショートファイル名の関係で同一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合

取得したコンテンツが、引数 pFilePath に指定したファイルに複写されます。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取得する場合は、引数 pRenditionType で対象のレンディションを指定します。

UNIX の場合に、登録されているファイルを /tmp/sample.doc に複写して取得する例を次に示します。Windows の場合は、c:\temp\sample.doc に複写して取得する例として、例中の "file:///tmp/sample.doc" を "file:///c:\temp\sample.doc" と読み替えてください。なお、この例で取得できるのは、マスタレンディションのコンテンツです。

ファイルを取得する例

```
pDbrVerDoc->GetContentAndLock("file:///tmp/sample.doc",
                               DMA_LOCK_READ);
```

マルチファイル文書の場合

引数 ContentList で指定した複数のファイルをコンテンツとして取得します。取得したコンテンツは、引数 pDirPath で指定したディレクトリに出力されます。ファイル名は、SDBR_CONTENTINFO 構造体に設定している pRetrievalName の値になります。

マルチファイル文書の場合、次のことに注意してください。

- GetContentInfo メソッドで取得した SDBR_CONTENTLIST 構造体を使用して、このメソッドを実行する場合、lValidMode は変更しないでください。
- この形式の GetContent メソッドを、マルチファイル文書ではないバージョンなし文書に対して実行すると、ERR_NOT_MULTI_CONTENTS のエラーになります。

リファレンスファイル文書の場合

取得したコンテンツが、引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry に指定したファイルに出力されます。

リファレンスファイル文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentReference オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_REFERENCE の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、

ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

取得したコンテンツが、引数 FileLinkPath の SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に指定したファイルまたはディレクトリに出力されます。

コンテンツとしてディレクトリが登録されている文書のレンディションに対してこのメソッドを実行すると、引数に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に指定した名前のディレクトリが作成され、その下に登録されているディレクトリおよびサブディレクトリ下のすべてのファイルが取得されます。コンテンツとしてディレクトリが登録されている文書のレンディションとは、AddRendition メソッド、CreateObject メソッド、UpdateContent メソッドまたは UpdateContentAndRenditionType メソッドの引数 FileLinkPath に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に、ディレクトリを指定して登録または更新した文書のレンディションのことです。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツとして空のディレクトリが登録されているレンディションに対してこのメソッドを実行すると、ERR_CONTENT_NOT_REGISTERED のエラーになります。コンテンツとして空のディレクトリが登録されているレンディションとは、AddRendition メソッド、CreateObject メソッド、UpdateContent メソッドまたは UpdateContentAndRenditionType メソッドの引数 FileLinkPath に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に、空のディレクトリを指定して作成した文書のレンディションのことです。
- 引数に指定する SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath で指定したファイル名のファイル、またはディレクトリパスのディレクトリ下に取得しようとするファイルがすでに存在する場合、ERR_FILELINK_FAILED のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pFilePath (入力)

取得したファイルを複写する複写先のパス名を指定します。

ファイルのパス名は URL 形式で設定します。ただし、プロトコル名は "file:" だけが指定できます。例えば、UNIX の場合、ファイル "/tmp/sample.doc" に複写するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" に複写するには、"file:///c:\temp\sample.doc" と指定します。

指定したパスのファイルが存在する場合は上書きします。ファイル名は、RenditionType プロパティまたは RetrievalName プロパティを参考にして、メソッドをコールした側が任意に決定してください。

指定したパスのファイルが存在しない場合、ディレクトリが存在し、かつファイルの作成が可能ならば新規にファイルを作成します。それ以外の場合はエラーになります。

pRenditionType (入力)

RenditionType を指定してファイルを取得する場合、取得するレンディションの RenditionType を表す文

字列を指定します。

この引数で指定した RenditionType が、対象文書のレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は NULL を指定するか、「マスタレンディションのファイルを取得する場合」の形式を使用することもできます。なお、空文字列 ("") は指定できません。

マルチファイル文書の場合、NULL または「シングルファイル文書の場合 (マスタレンディションのファイルの取得)」の形式を指定してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

pDirPath (入力)

出力されるファイルの格納先ディレクトリを、URL 形式で指定します。

指定したパスのディレクトリが存在し、かつそのディレクトリへの実行権限、書き込み権限があれば、取得したファイルを格納します。それ以外の場合、エラーになります。

ContentList (入力)

取得するコンテンツの情報を設定したコンテンツ情報リスト構造体を指定します。コンテンツ情報リスト構造体については、「付録 C.1(2) SDBR_CONTENTLIST 構造体 (コンテンツ情報リスト構造体)」を参照してください。

ReferencePath (入力)

リファレンスファイル文書を取得するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。ContentLocation プロパティと CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパスを合わせた文字列長 (パスの区切り文字も含む) が、Windows の場合は 260 バイト以上、AIX の場合は 1023 バイト以上であると ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

リファレンスファイル文書を取得する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath および pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

piReferenceType (出力)

コンテンツのリファレンス種別 (ReferenceType) を取得する場合、リファレンス種別を格納する領域のポインタを指定します。NULL を指定した場合、リファレンス種別は返却されません。

ppContentLocation (出力)

コンテンツロケーション (ContentLocation プロパティ) を取得する場合、コンテンツロケーションへのポインタを格納する領域を指定します。NULL を指定した場合、コンテンツロケーションは返却されません。取得した領域は、dbrDelete 関数で解放する必要があります。dbrDelete 関数の詳細については、「20. 関数」の「dbrDelete 関数」を参照してください。

FileLinkPath (出力)

SDBR_FILELINK_PATHINFO 構造体の pFilePath メンバだけが有効となります。そのほかのメンバの設定値は無視されます。pFilePath に、コンテンツの取得先となるクライアントのファイルパスまたはディレクトリパスを指定してください。

SDBR_FILELINK_PATHINFO 構造体の指定内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

ppbvFileHandle (出力)

ppbvFileHandle にファイルハンドル情報が返却されます。この引数には NULL を指定してください。取得した領域は、dbrDelete 関数で解放する必要があります。dbrDelete 関数の詳細については、「20. 関数」の「dbrDelete 関数」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_COMPONENTTYPE_DUPLICATED	ComponentType が重複して指定されています。	ComponentType が重複しないように指定してください。
ERR_DBR	ERR_CONTENT_NOT_REGISTERED	コンテンツが登録されていません。	コンテンツを登録してください。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_NOT_MULTI_CONTENTS	マルチファイル文書ではありません。	マルチファイル管理機能を使用しない場合の形式で実行してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_REFERENCETYPE_MISMATCH	オブジェクト種別が一致していません。	オブジェクト種別または指定した操作モード (IContentOperateMode) を見直してください。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

8. CdbbrDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbrDocument::GetContentInfo メソッド

機能

マルチファイル文書の情報の取得

形式

```
DmaBoolean GetContentInfo(
    pDmaString_T      pRenditionType,
    SDBR_CONTENTLIST** ppContentList,
    DmaInteger32      lLockType = DMA_LOCK_READ)

//コンテンツ情報構造体
struct SDBR_CONTENTINFO {
    pDmaString_T      pComponentType; // ファイルを識別する文字列
    pDmaString_T      pRetrievalName; // ファイル名
    DmaUInteger32     lContentSize;   // ファイルのサイズ
};

//コンテンツ情報リスト構造体
struct SDBR_CONTENTLIST {
    DmaInteger32      lCount;          // 取得ファイル数
    SDBR_CONTENTINFO* pItem;          // コンテンツ情報構造体の
                                        // ポインタ
    DmaInteger32      lValidMode;      // 有効モード
    DmaBoolean        bIsMultiContents; // マルチファイル識別
};
```

詳細

指定したレンディションのコンテンツの情報を取得します。取得した情報は、引数 ppContentList で指定した SDBR_CONTENTLIST 構造体の領域に格納されます。返却された SDBR_CONTENTLIST 構造体のメンバに設定される値については、「付録 C.1(2) SDBR_CONTENTLIST 構造体 (コンテンツ情報リスト構造体)」を参照してください。

この ppContentList に返却された SDBR_CONTENTLIST 構造体は、直接 GetContentAndLock メソッドの引数として指定できます。ただし、返却された SDBR_CONTENTLIST 構造体の lCount が 0 でないことを確認してから、GetContentAndLock メソッドの引数として指定してください。lCount が 0 である SDBR_CONTENTLIST 構造体を GetContentAndLock メソッドの引数として指定すると、DMARC_BAD_PARAMETER のエラーになります。

引数

pRenditionType (入力)

レンディションの RenditionType を表す文字列を指定します。指定したレンディションのコンテンツの情報が取得されます。

NULL を指定した場合、マスタレンディションが処理対象になります。

ppContentList (出力)

コンテンツ情報リスト構造体のポインタを格納する領域を指定します。このポインタが示す領域はクラスライブラリが確保するため、使用後はこのポインタを dbrDelete 関数で解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ

read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

- DMA_LOCK_WRITE

排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbDocument::GetRelationList メソッド

機能

文書のリレーション情報の一覧取得

形式

```
DmaBoolean GetRelationList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lRelPropDefListCount,
    SDBR_PROPDEF*       pRelPropDefList,
    DmaInteger32         lRelationListCount,
    SDBR_RELATIONLIST** ppRelationList,
    DmaInteger32         lRelationendType,
    DmaInteger32         lRelationendStatus)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32   lType;          //データ型
    DmaInteger32   lCardinality;   //基本単位
    DmaInteger32   lCount;        //値の個数
    union {
        pDmaBoolean pbBoolean;    //Boolean型
        pDmaInteger32 plInteger32; //Integer32型
        pDmaId       pId;         //ID型
        pDmapv       ppObject;    //Object型
        ppDmaString_T ppString;   //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;        //プロパティ情報の個数
    SDBR_PROP*    pItem;        //プロパティ情報
};

// リレーションリスト構造体
struct SDBR_RELATIONLIST {
    DmaInteger32   lCount; //リレーションしている数
    SDBR_RELATIONINFO* pItem; //リレーション情報構造体へのポインタ
};

// リレーション情報構造体
struct SDBR_RELATIONINFO {
    pDmaString_T   pOIID;        //文書オブジェクトのOIID
    DmaInteger32   lType;        //オブジェクトの種別
    pDmaString_T   pRelId;      //リレーション識別子
    SDBR_PROPLIST  PropList;     //オブジェクトのプロパティリスト
    SDBR_PROPLIST  RelPropList; //リレーションのプロパティリスト
};
```

詳細

文書間リレーションで関連づけられている文書の一覧情報を取得するメソッドです。

引数 lRelationendType に DBR_RELATIONEND_HEAD を設定することによって、このオブジェクトに

関連づけられているリレーション先文書の一覧情報を取得できます。また、lRelationendType に DBR_RELATIONEND_TAIL を設定することによって、このオブジェクトをリレーション先に持つリレーション元文書の一覧情報を取得できます。

初回取得時は、引数 pbContinue に DMA_FALSE を設定してメソッドをコールします。1回のメソッドのコールで一覧の取得が完了しなかった場合、引数 pbContinue に DMA_TRUE が設定されます。残りの要素は、引数 pbContinue に DMA_TRUE を設定して取得します。引数 pbContinue が DMA_FALSE になるまで、繰り返してこのメソッドをコールしてください。

リレーション先のオブジェクトが削除されていて、RemoveRelation メソッドで文書間リレーションが解除されないままの場合、リレーション情報は残っている状態になります。この場合も、引数 lRelationendStatus に DBR_RELATIONEND_STATUS_NOT_EXIST または DBR_RELATIONEND_STATUS_ALL を設定することによって、削除されたオブジェクトのリレーション情報を取得できます。オブジェクトが削除されていてリレーション情報だけが残っている場合、リレーション情報構造体である、SDBR_RELATIONINFO 構造体の内容は、pOIID = NULL、lType = DBR_CLASS_UNKNOWN、PropList.lCount = 0、PropList.pItem = NULL になり、pRelId と RelPropList だけに正当な値が設定されます。

なお、このメソッドで取得したリレーション識別子は、トランザクションが終了するまで有効とは限りません。ほかのトランザクションから削除されたり、リレーションのプロパティを変更されたりする可能性があります。

引数

pbContinue (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した要素の残りを破棄して、新たに先頭から取得します。

なお、オブジェクトの生成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、DMA_FALSE が指定されたものとして動作します。

前回の読み出しと異なる引数を持つメソッドがコールされた場合は、pbContinue の値を無視し、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- DMA_TRUE
継続可能 (まだ取得していない残りの要素があります)
- DMA_FALSE
継続不可 (すべての要素を取得しました)

lClassType (入力)

指定されたクラス識別子の要素だけを取得します。ただし、関連づけの対象オブジェクトがすでに削除されていて、リレーションだけが残っているリレーション情報については lClassType の指定とは無関係にすべて取得されます。

指定できるクラス識別子を次に示します。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

クラス識別子	クラスの種別
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件とした取得もできます。例えば、CdbDocument クラスと CdbVersionableDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONABLEDOCUMENT
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IPropDefListCount が 0 の場合だけ指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの個数 (引数 pRelPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pRelPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IRelPropDefListCount が 0 の場合だけ指定できます。IRelPropDefListCount が正の整数のときに pRelPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelationListCount (入力)

取得する文書間リレーション情報の数を 1 以上の値で指定します。0 以下の値を指定するとエラーになります。

ppRelationList (出力)

取得した文書間リレーション情報一覧へのポインタである、SDBR_RELATIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。**ppRelationList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してく

ださい。なお、**ppRelationList を指定してメモリを解放すると、構造体のメンバ (**ppRelationList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RELATIONLIST 構造体および SDBR_RELATIONINFO 構造体に設定される内容については、「付録 C.1(17) SDBR_RELATIONLIST 構造体 (リレーションリスト構造体)」および「付録 C.1(16) SDBR_RELATIONINFO 構造体 (リレーション情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IRelationendType (入力)

このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得するか、このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得するかを指定します。

次に示す値のどちらかを指定してください。

- DBR_RELATIONEND_HEAD
このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得する場合に指定します。
- DBR_RELATIONEND_TAIL
このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得する場合に指定します。

IRelationendStatus (入力)

関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得するか、すでに関連づけの対象オブジェクトが削除されていて、リレーションだけ残っているリレーション情報の一覧を取得するかを指定します。

次に示す値のどれかを指定してください。

- DBR_RELATIONEND_STATUS_EXIST
関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得する場合に指定します。
- DBR_RELATIONEND_STATUS_NOT_EXIST
すでに関連づけの対象オブジェクトが削除されていて、リレーションだけが残っているリレーション情報の一覧を取得する場合に指定します。
この場合、IClassType でのクラス種別の指定は無効になります。つまり、IClassType でクラス種別を指定しているかどうかに関係なく、関連づけの対象オブジェクトが削除されているリレーション情報はすべて取得されます。
- DBR_RELATIONEND_STATUS_ALL
すべてのリレーション情報の一覧を取得する場合に指定します。
この場合、関連づけの対象オブジェクトが存在しているリレーションに関しては、IClassType でのクラス種別の指定が有効です。ただし、関連づけの対象オブジェクトが削除されているリレーション情報は、IClassType でのクラス種別の指定とは無関係にすべて取得されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbDocument::GetRelationListAndLock メソッド

機能

文書のリレーション情報の一覧取得と同時に指定したロックの設定

形式

```
DmaBoolean GetRelationListAndLock(
    DmaBoolean          pbContinue,
    DmaInteger32        lClassType,
    DmaInteger32        lPropDefListCount,
    SDBR_PROPDEF*      pPropDefList,
    DmaInteger32        lRelPropDefListCount,
    SDBR_PROPDEF*      pRelPropDefList,
    DmaInteger32        lRelationListCount,
    SDBR_RELATIONLIST** ppRelationList,
    DmaInteger32        lRelationendType,
    DmaInteger32        lRelationendStatus,
    DmaInteger32        lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32  lType;           //データ型
    DmaInteger32  lCardinality;    //基本単位
    DmaInteger32  lCount;         //値の個数
    union {
        DmaBoolean  pbBoolean;    //Boolean型
        DmaInteger32 plInteger32;  //Integer32型
        DmaId        pId;         //ID型
        Dmapv        ppObject;    //Object型
        ppDmaString_T ppString;   //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;          //プロパティ情報の個数
    SDBR_PROP*   pItem;          //プロパティ情報
};

// リレーションリスト構造体
struct SDBR_RELATIONLIST {
    DmaInteger32  lCount; //リレーションしている数
    SDBR_RELATIONINFO* pItem; //リレーション情報構造体へのポインタ
};

// リレーション情報構造体
struct SDBR_RELATIONINFO {
    pDmaString_T  pOIID;          //文書オブジェクトのOIID
    DmaInteger32  lType;         //オブジェクトの種別
    pDmaString_T  pRelId;       //リレーション識別子
    SDBR_PROPLIST PropList;     //オブジェクトのプロパティリスト
    SDBR_PROPLIST RelPropList;  //リレーションのプロパティリスト
};
```

詳細

ロックタイプを明示的に指定して、文書間リレーションで関連づけられている文書の一覧情報を取得するメソッドです。

引数 lRelationendType に DBR_RELATIONEND_HEAD を設定することによって、このオブジェクトに関連づけられているリレーション先文書の一覧情報を取得します。また、lRelationendType に DBR_RELATIONEND_TAIL を設定することによって、このオブジェクトをリレーション先に持つリレーション元文書の一覧情報を取得できます。

初回取得時は、引数 pbContinue に DMA_FALSE を設定してメソッドをコールします。1回のメソッドのコールで一覧の取得が完了しなかった場合、引数 pbContinue に DMA_TRUE が設定されます。残りの要素は、引数 pbContinue に DMA_TRUE を設定して取得します。引数 pbContinue が DMA_FALSE になるまで、繰り返してこのメソッドをコールしてください。

リレーション先のオブジェクトが削除されていて、RemoveRelation メソッドで文書間リレーションが解除されないままの場合、リレーション情報は残っている状態になります。この場合も、引数 lRelationendStatus に DBR_RELATIONEND_STATUS_NOT_EXIST または DBR_RELATIONEND_STATUS_ALL を設定することによって、削除されたオブジェクトのリレーション情報を取得できます。オブジェクトが削除されていてリレーション情報だけが残っている場合、リレーション情報構造体である、SDBR_RELATIONINFO 構造体の内容は、pOIID = NULL、lType = DBR_CLASS_UNKNOWN、PropList.lCount = 0、PropList.pItem = NULL になり、pRelId と RelPropList だけに正当な値が設定されます。

なお、このメソッドで取得したリレーション識別子は、トランザクションが終了するまで有効とは限りません。ほかのトランザクションから削除されたり、リレーションのプロパティを変更されたりする可能性があります。

引数

pbContinue (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した要素の残りを破棄して、新たに先頭から取得します。

なお、オブジェクトの生成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、DMA_FALSE が指定されたものとして動作します。

前回の読み出しと異なる引数を持つメソッドがコールされた場合は、pbContinue の値無視し、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- DMA_TRUE
継続可能 (まだ取得していない残りの要素があります)
- DMA_FALSE
継続不可 (すべての要素を取得しました)

lClassType (入力)

指定されたクラス識別子の要素だけを取得します。ただし、関連づけの対象オブジェクトがすでに削除されていて、リレーションだけが残っているリレーション情報については lClassType の指定とは無関係にすべて取得されます。

指定できるクラス識別子を次に示します。

クラス識別子	クラスの種別
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件とした取得もできます。例えば、CdbDocument クラスと CdbVersionableDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONABLEDOCUMENT
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IPropDefListCount が 0 の場合だけ指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの個数 (引数 pRelPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pRelPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IRelPropDefListCount が 0 の場合だけ指定できます。IRelPropDefListCount が正の整数のときに pRelPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelationListCount (入力)

取得する文書間リレーション情報の数を 1 以上の値で指定します。0 以下の値を指定するとエラーになります。

ppRelationList (出力)

取得した文書間リレーション情報一覧へのポインタである、SDBR_RELATIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。**ppRelationList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してく

ださい。なお、**ppRelationList を指定してメモリを解放すると、構造体のメンバ (**ppRelationList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RELATIONLIST 構造体および SDBR_RELATIONINFO 構造体に設定される内容については、「付録 C.1(17) SDBR_RELATIONLIST 構造体 (リレーションリスト構造体)」および「付録 C.1(16) SDBR_RELATIONINFO 構造体 (リレーション情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IRelationendType (入力)

このオブジェクトが文書間リレーションで関連づけているリレーション先オブジェクトの一覧を取得するか、このオブジェクトが文書間リレーションで関連づけられているリレーション元オブジェクトの一覧を取得するかを指定します。

次に示す値のどちらかを指定してください。

- DBR_RELATIONEND_HEAD
このオブジェクトが文書間リレーションで関連づけているリレーション先オブジェクトの一覧を取得する場合に指定します。
- DBR_RELATIONEND_TAIL
このオブジェクトが文書間リレーションで関連づけられているリレーション元オブジェクトの一覧を取得する場合に指定します。

IRelationendStatus (入力)

関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得するか、すでに関連づけの対象オブジェクトが削除されていて、リレーションだけ残っているリレーション情報の一覧を取得するかを指定します。

次に示す値のどれかを指定してください。

- DBR_RELATIONEND_STATUS_EXIST
関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得する場合に指定します。
- DBR_RELATIONEND_STATUS_NOT_EXIST
すでに関連づけの対象オブジェクトが削除されていて、リレーションだけが残っているリレーション情報の一覧を取得する場合に指定します。
この場合、IClassType でのクラス種別の指定は無効になります。つまり、IClassType でクラス種別を指定しているかどうかに関係なく、関連づけの対象オブジェクトが削除されているリレーション情報はすべて取得されます。
- DBR_RELATIONEND_STATUS_ALL
すべてのリレーション情報の一覧を取得する場合に指定します。
この場合、関連づけの対象オブジェクトが存在しているリレーションに関しては、IClassType でのクラス種別の指定が有効です。ただし、関連づけの対象オブジェクトが削除されているリレーション情報は、IClassType でのクラス種別の指定とは無関係にすべて取得されます。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ

read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

- DMA_LOCK_WRITE

排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbDocument::GetRenditionList メソッド

機能

レンディション一覧の取得

形式

```
DmaBoolean GetRenditionList(
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_RENDITIONLIST** ppRenditionList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32  lType;           //データ型
    DmaInteger32  lCardinality;    //基本単位
    DmaInteger32  lCount;         //値の個数
    union {
        DmaBoolean  pbBoolean;    //Boolean型
        DmaInteger32 plInteger32;  //Integer32型
        DmaId       pId;          //ID型
        Dmapv       ppObject;     //Object型
        ppDmaString_T ppString;   //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;          //プロパティ情報の個数
    SDBR_PROP*    pItem;         //プロパティ情報
};

// レンディションリスト構造体
struct SDBR_RENDITIONLIST {
    DmaInteger32  lCount;          //レンディション情報の数
    SDBR_RENDITIONINFO* pItem;   //レンディション情報構造体
};

// レンディション情報構造体
struct SDBR_RENDITIONINFO {
    ppDmaString_T  pRenditionType; //RenditionTypeを表す文字列
    SDBR_PROPLIST  PropList;       //プロパティリスト構造体
};
```

詳細

文書に登録されているレンディションの一覧情報とそのプロパティを取得するメソッドです。

マルチレンディション文書の場合、マスタレンディションの情報は、レンディション情報構造体である、SDBR_RENDITIONINFO 構造体の配列要素の 1 番目に格納され、サブレンディションの情報は 2 番目以降に格納されます。

引数

IPropDefListCount (入力)

レンディションの一覧と同時に取得するプロパティの数 (pPropDefList に指定する配列の要素数) を指定します。0 を指定した場合は、プロパティは取得されません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

レンディションの一覧と同時に取得するプロパティの定義情報である, SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。lPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName
レンディションの RetrievalName を取得する場合に指定します。
- dbrProp_RenditionStatus
レンディションの RenditionStatus を取得する場合に指定します。なお, 設定される RenditionStatus の値については, 「CdbDocument クラスの詳細」の「RenditionStatus プロパティの値」を参照してください。
- dbrProp_ContentType
レンディションのコンテンツ種別を取得する場合に指定します。なお, 設定される ContentType の値については, 「CdbDocument クラスの詳細」の「ContentType プロパティの値とコンテンツ種別の対応」を参照してください。

SDBR_PROPDEF 構造体については, 「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppRenditionList (出力)

取得したレンディション一覧へのポインタである, SDBR_RENDITIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。*ppRenditionList がポイントするメモリは, ライブラリが新たに確保したものです。このメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。なお, *ppRenditionList を指定してメモリを解放すると, 構造体のメンバ (*ppRenditionList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RENDITIONLIST 構造体および SDBR_RENDITIONINFO 構造体に設定される内容については, 「付録 C.1(20) SDBR_RENDITIONLIST 構造体 (レンディションリスト構造体)」および「付録 C.1(19) SDBR_RENDITIONINFO 構造体 (レンディション情報構造体)」を参照してください。

また, プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して, ユーザプログラムを終了してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbDocument::GetRenditionListAndLock メソッド

機能

レンディション一覧の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetRenditionListAndLock(
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_RENDITIONLIST** ppRenditionList,
    DmaInteger32          lLockType = DMA_LOCK_WRITE )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32   lType;           //データ型
    DmaInteger32   lCardinality;    //基本単位
    DmaInteger32   lCount;         //値の個数
    union {
        DmaBoolean pbBoolean;      //Boolean型
        DmaInteger32 plInteger32;   //Integer32型
        DmaId       pId;           //ID型
        Dmapv       ppObject;      //Object型
        ppDmaString_T ppString;    //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;          //プロパティ情報の個数
    SDBR_PROP*     pItem;          //プロパティ情報
};

// レンディションリスト構造体
struct SDBR_RENDITIONLIST {
    DmaInteger32   lCount;          //レンディション情報の数
    SDBR_RENDITIONINFO* pItem;     //レンディション情報構造体
};

// レンディション情報構造体
struct SDBR_RENDITIONINFO {
    pDmaString_T   pRenditionType; //RenditionTypeを表す文字列
    SDBR_PROPLIST  PropList;       //プロパティリスト構造体
};
```

詳細

ロックタイプを明示的に指定して、文書に登録されているレンディションの一覧情報とそのプロパティを取得するメソッドです。

マルチレンディション文書の場合、マスタレンディションの情報は、レンディション情報構造体である、SDBR_RENDITIONINFO 構造体の配列要素の 1 番目に格納され、サブレンディションの情報は 2 番目以降に格納されます。

引数

IPropDefListCount (入力)

レンディションの一覧と同時に取得するプロパティの数 (pPropDefList に指定する配列の要素数) を指定します。0 を指定した場合は、プロパティは取得されません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

レンディションの一覧と同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。IPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName
レンディションの RetrievalName を取得する場合に指定します。
- dbrProp_RenditionStatus
レンディションの RenditionStatus を取得する場合に指定します。なお、設定される RenditionStatus の値については、「CdbDocument クラスの詳細」の「RenditionStatus プロパティの値」を参照してください。
- dbrProp_ContentType
レンディションのコンテンツ種別を取得する場合に指定します。なお、設定される ContentType の値については、「CdbDocument クラスの詳細」の「ContentType プロパティの値とコンテンツ種別の対応」を参照してください。

SDBR_PROPDEF 構造体については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppRenditionList (出力)

取得したレンディション一覧へのポインタである、SDBR_RENDITIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。*ppRenditionList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppRenditionList を指定してメモリを解放すると、構造体のメンバ (*ppRenditionList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RENDITIONLIST 構造体および SDBR_RENDITIONINFO 構造体に設定される内容については、「付録 C.1(20) SDBR_RENDITIONLIST 構造体 (レンディションリスト構造体)」および「付録 C.1(19) SDBR_RENDITIONINFO 構造体 (レンディション情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE

排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbDocument::GetVersionableList メソッド

機能

バージョン管理情報一覧の取得

形式

```
// バージョン識別子なしで取得する場合
DmaBoolean GetVersionableList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList )

// バージョン識別子付きで取得する場合
DmaBoolean GetVersionableList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VROBJLIST**   ppVrObjList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId                PropId;                // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;                // プロパティ識別子
    DmaInteger32         lType;                 // データ型
    DmaInteger32         lCardinality;          // 基本単位
    DmaInteger32         lCount;               // 要素数
    union {
        // プロパティの値
        pDmaBoolean      pbBoolean;           // Boolean型
        pDmaInteger32    plInteger32;         // Integer32型
        pDmaId           pId;                 // ID型
        pDmapv           ppObject;           // Object型
        ppDmaString_T    ppString;           // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T         pOIID;                // オブジェクトのOIID
    DmaInteger32         lType;                // オブジェクトの種別
    SDBR_PROPLIST        PropList;            // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32         lCount;
    SDBR_OBJINFO*       pItem;
};

// バージョン識別子付きオブジェクト情報構造体
struct SDBR_VROBJINFO {
```

```

    pDmaString_T      pOIID;          // オブジェクトのOIID
    DmaInteger32      lType;          // オブジェクトの種別
    pDmaString_T      pVersionId;     // バージョン識別子
    SDBR_PROPLIST     PropList;       // プロパティリスト構造体
};

// バージョン識別子付きオブジェクトリスト構造体
struct SDBR_VROBJLIST {
    DmaInteger32      lCount;
    SDBR_VROBJINFO*  pItem;
};

```

詳細

CdbrDocument オブジェクトを 1 バージョンとして包含している CdbrVersionableDocument オブジェクトの一覧を取得するメソッドです。このオブジェクトを表すバージョン識別子も同時に取得できます。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。ただし、このオブジェクトが仮のバージョンに該当するオブジェクトの場合は、CdbrVersionableDocument オブジェクトを取得できません。

初回にメソッドをコールした時点ですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態を示しています。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対するコールであっても、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbrContainable クラス (包含要素を扱うクラス)」の CdbrContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次にある要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されます。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 8-8 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lPropDefListCount	×	

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pPropDefList	×	
lObjListCount		
ppObjList		
ppVrObjList		

(凡例)

：有効

×：無効

lPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。lPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報として、オブジェクト情報リスト構造体である、SDBR_OBJLIST 構造体の配列へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリを解放すると、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppVrObjList (出力)

取得した情報として、バージョン識別子付きオブジェクトリスト構造体である、SDBR_VROBJLIST 構造体の配列へのポインタが設定されます。*ppVrObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppVrObjList を指定したメモリを解放すると、構造体のメンバ (*ppVrObjList)->pItem およびその先に確

保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_VROBJLIST 構造体および SDBR_VROBJINFO 構造体に設定される内容については、「付録 C.1(22) SDBR_VROBJLIST 構造体 (バージョン識別子付きオブジェクトリスト構造体)」および「付録 C.1(21) SDBR_VROBJINFO 構造体 (バージョン識別子付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbDocument::GetVersionableListAndLock メソッド

機能

バージョン管理情報一覧の取得と同時に指定したロックの設定

形式

```
// バージョン識別子なしで取得する場合
DmaBoolean GetVersionableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// バージョン識別子付きで取得する場合
DmaBoolean GetVersionableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VROBJLIST**    ppVrObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;          // 要素数
    union {
        pDmaBoolean pbBoolean;      // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId       pId;           // ID型
        pDmapv       ppObject;      // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;           // オブジェクトのOIID
    DmaInteger32   lType;           // オブジェクトの種類
    SDBR_PROPLIST PropList;        // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32   lCount;
    SDBR_OBJINFO* pItem;
};

// バージョン識別子付きオブジェクト情報構造体
```

8. CdbrDocument クラス (バージョンなし文書を扱うクラス)

```

struct SDBR_VROBJINFO {
    pDmaString_T      pOIID;           // オブジェクトのOIID
    DmaInteger32      lType;           // オブジェクトの種別
    pDmaString_T      pVersionId;     // バージョン識別子
    SDBR_PROPLIST     PropList;       // プロパティリスト構造体
};

// バージョン識別子付きオブジェクトリスト構造体
struct SDBR_VROBJLIST {
    DmaInteger32      lCount;
    SDBR_VROBJINFO*  pItem;
};

```

詳細

ロックタイプを明示的に指定して、CdbrDocument オブジェクトを 1 バージョンとして包含している CdbrVersionableDocument オブジェクトの一覧を取得するメソッドです。このオブジェクトを表すバージョン識別子も同時に取得できます。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。ただし、このオブジェクトが仮のバージョンに該当するオブジェクトの場合は、CdbrVersionableDocument オブジェクトを取得できません。

初回にメソッドをコールした時点ですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態を示しています。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対するコールであっても、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbrContainable クラス (包含要素を扱うクラス)」の CdbrContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次にある要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されます。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について次に示します。

表 8-9 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lPropDefListCount	×	

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pPropDefList	×	
lObjListCount		
ppObjList		
ppVrObjList		
lLockType	×	

(凡例)

○ : 有効

× : 無効

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティは取得しません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。IPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報として、オブジェクト情報リスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリを解放すると、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppVrObjList (出力)

取得した情報として、バージョン識別子付きオブジェクトリスト構造体である、SDBR_VROBJLIST 構造体へのポインタが設定されます。*ppVrObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、

*ppObjList を指定したメモリを解放すると、構造体のメンバ (*ppVrObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_VROBJLIST 構造体および SDBR_VROBJINFO 構造体に設定される内容については、「付録 C.1(22) SDBR_VROBJLIST 構造体 (バージョン識別子付きオブジェクトリスト構造体)」および「付録 C.1(21) SDBR_VROBJINFO 構造体 (バージョン識別子付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbDocument::PutRelationPropertyValues メソッド

機能

文書間リレーションのプロパティの設定

形式

```
DmaBoolean PutRelationPropertyValues (
    pDmaString_T          pRelId,
    SDBR_PROPLIST*       pRelPropList)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           //プロパティ識別子
    DmaInteger32         lType;           //データ型
    DmaInteger32         lCardinality;    //基本単位
    DmaInteger32         lCount;         //値の個数
    union {              //プロパティの値
        pDmaBoolean      pbBoolean;     //Boolean型
        pDmaInteger32    plInteger32;    //Integer32型
        pDmaId           pID;           //ID型
        pDmapv           ppObject;      //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;         //プロパティ情報の個数
    SDBR_PROP*          pItem;         //プロパティ情報
};
```

詳細

リレーション識別子を指定して、文書間リレーションのプロパティを設定するメソッドです。

引数

pRelId (入力)

プロパティを設定する文書間リレーションのリレーション識別子を指定します。

指定するリレーション識別子は、CreateRelation メソッドの引数 ppRelId で返却される値です。また、GetRelationList メソッドまたは GetRelationListAndLock メソッドを使用しても取得できます。

pRelPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。このメソッドで値を設定できる文書間リレーションのプロパティは、ユーザ設定のプロパティだけです。

NULL を指定するとエラーになります。なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体を設定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbDocument::PutRenditionPropertyValues メソッド

機能

レンディションのプロパティの設定

形式

```
DmaBoolean PutRenditionPropertyValues (
    pDmaString_T      pRenditionType,
    SDBR_PROPLIST*   pPropList )

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           //プロパティ識別子
    DmaInteger32      lType;            //データ型
    DmaInteger32      lCardinality;     //基本単位
    DmaInteger32      lCount;          //値の個数
    union {
        pDmaBoolean   pbBoolean;       //Boolean型
        pDmaInteger32 plInteger32;     //Integer32型
        pDmaId         pId;             //ID型
        pDmapv         ppObject;       //Object型
        ppDmaString_T ppString;        //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;           //プロパティ情報の個数
    SDBR_PROP*       pItem;           //プロパティ情報
};
```

詳細

サブレンディションのプロパティを設定するメソッドです。

マルチレンディション文書のサブレンディションのプロパティ (RenditionType 以外) を設定, 変更する場合, このメソッドを使用します。

なお, RenditionType を変更する場合は, CdbDocument::UpdateContentAndRenditionType メソッドを使用してください。

引数

pRenditionType (入力)

プロパティを設定するレンディションの RenditionType を表す文字列を指定します。この引数で指定した RenditionType が対象文書のレンディションとして存在しない場合はエラーになります。NULL を指定した場合, マスタレンディションが処理対象になります。空文字列 ("") は指定できません。

pPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である, SDBR_PROPLIST 構造体を指定します。NULL を指定するとエラーになります。

なお, SDBR_PROPLIST 構造体に, 同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName

RetrievalName プロパティを設定する場合に指定します。

- dbrProp_RenditionStatus

RenditionStatus プロパティを設定する場合に指定します。なお、RenditionStatus プロパティの値としてユーザが設定できるのは上位 2 バイトの変換フラグだけで、下位 2 バイトの状態フラグは設定できません。ユーザが設定できる変換フラグの値を次に示します。

DBR_RENDSTATUS_CONVERT_REQUIRED

レンディション変換機能の対象になる状態、つまり、レンディション変換機能によって、マスタレンディションの内容から、コンテンツを自動更新できる状態です。DocumentBroker Rendering Option が提供するコマンドを実行することで、コンテンツが自動更新されます。サブレンディションのコンテンツ登録後にマスタレンディションを更新した場合のような、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のときに、レンディション変換機能を使ってサブレンディションを更新したい場合に指定します。ただし、レンディション変換機能を使うには、DocumentBroker Rendering Option が必要です。

DBR_RENDSTATUS_CONVERT_NOTREQUIRED

レンディション変換機能の対象にならない状態です。DocumentBroker Rendering Option が提供するコマンドを実行しても、コンテンツは更新されません。ただし、変換フラグの値の設定に関しては、メソッド実行時の RenditionStatus プロパティの値によって、次に示す制限があります。

メソッド実行時の RenditionStatus プロパティの値	変換フラグの値の設定に関する制限の内容
状態フラグが DBR_RENDSTATUS_NO_SUBREND の場合	DBR_RENDSTATUS_CONVERT_NOTREQUIRED を設定できません。
変換フラグが DBR_RENDSTATUS_CONVERT_ERROR の場合	DBR_RENDSTATUS_CONVERT_REQUIRED または DBR_RENDSTATUS_CONVERT_NOTREQUIRED に変更できません。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrDocument::RemoveIndex メソッド

機能

全文検索インデクスの削除

形式

DmaBoolean RemoveIndex()

詳細

CdbrDocument オブジェクトに作成されている全文検索インデクスだけを削除します。インデクスが作成されていない CdbrDocument オブジェクトに対してこのメソッドをコールした場合は、何もしないで正常終了します。ただし、全文検索機能付き文書クラスから作成された DMA オブジェクトの DocVersion オブジェクトを構成要素としない CdbrDocument オブジェクトに対してこのメソッドを実行した場合は、エラーになります。

CdbrDocument オブジェクトに作成されている全文検索インデクスを削除する例を次に示します。

全文検索インデクスの削除の例

```
pDbrDoc->RemoveIndex();
```

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECT	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbDocument::RemoveRelation メソッド

機能

文書間リレーションの解除

形式

```
DmaBoolean RemoveRelation(
    DmaInteger32      lRelIdCount,
    ppDmaString_T    ppRelIds)
```

詳細

リレーションで結び付けられた文書間の関連づけを解除するメソッドです。

引数 ppRelIds のポインタ配列で指定された文書間のリレーションをすべて解除します。

リレーション識別子で示されるリレーション先のオブジェクトを削除しても、このメソッドで文書間リレーションを解除しないかぎり、リレーション情報は削除されないまま残ります。不要なリレーション情報を削除する場合にも、このメソッドをコールしてください。

引数

lRelIdCount (入力)

解除する文書間リレーションの個数を指定します。0 以下の値を指定した場合はエラーになります。0 より大きい値または次に示す値を指定してください。

- DBR_RELATION_ALL
このオブジェクトがリレーション元である文書間リレーションをすべて解除します。
- DBR_RELATION_END_NOT_EXIST
リレーション先のオブジェクトが削除されているリレーション情報をすべて削除します。

ppRelIds (入力)

解除する文書間リレーションのリレーション識別子のポインタ配列を指定します。配列の要素で示される文書間リレーションが解除 (リレーション情報が削除) されます。

lRelIdCount が 1 以上の場合に、NULL を指定するとエラーになります。

指定するリレーション識別子は、CreateRelation メソッドの引数 ppRelId で返却される値です。また、GetRelationList メソッドまたは GetRelationListAndLock メソッドを使用しても取得できます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrDocument::UpdateContent メソッド

機能

ファイルの更新

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの更新)
DmaBoolean UpdateContent(
    const pDmaString_T      pFilePath,
    DmaInteger32            lCreateIndexFlg = DBR_CREATE_INDEX)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの更新)
DmaBoolean UpdateContent(
    const pDmaString_T      pFilePath,
    pDmaString_T            pRenditionType,
    DmaInteger32            lCreateIndexFlg)

// マルチファイル文書の場合
DmaBoolean UpdateContent(
    SDBR_PATHLIST&         PathList,
    pDmaString_T           pRenditionType)

// File Link文書の場合
DmaBoolean UpdateContent(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T            pRenditionType)

// ファイルパス情報構造体
struct SDBR_PATHINFO {
    pDmaString_T    pFilePath;          // ファイルのURL
    pDmaString_T    pComponentType;    // ファイルを識別する文字列
};

// ファイルパス情報リスト構造体
struct SDBR_PATHLIST {
    DmaInteger32    lCount;             // 登録ファイル数
    SDBR_PATHINFO* pItem;             // ファイルパス情報構造体のポインタ
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T    pFilePath;          // 登録ファイルまたはディレクトリのURL
    pDmaString_T    pFamName;          // FAM名
    pDmaString_T    pFamFilePath;      // コンテンツ保管先パス
    DmaInteger32    lContentDeleteMode; // コンテンツ削除モード
    DmaInteger32    lFileNamingMode;   // コンテンツ名自動生成モード
};
```

詳細

登録されている CdbrDocument オブジェクトの内容を更新するメソッドです。CdbrDocument クラスに格納したコンテンツを、指定したパスのファイルで更新します。

このメソッドは、リファレンスファイルを管理する文書に対しては実行できません。

Windows の場合、次のことに注意してください。

- GetContentAndLock によるコンテンツの取得時、大文字と小文字を区別しないと同一になるファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。
- GetContentAndLock によるコンテンツの取得時、ロングファイル名とショートファイル名の関係で同

一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合 (マスタレンディションのファイルの更新)

シングルファイル文書 (マルチレンディション文書の場合はマスタレンディションであるシングルファイル文書) を更新する場合は、更新に使用するファイルのパスと全文検索インデクスを作成するかどうかを指定します。

内容を更新する CdbDocument オブジェクト (マルチレンディション文書の場合はマスタレンディション) が次の条件に当てはまる場合に、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデクスを作成できます。

- 全文検索機能付き文書クラスを基に作成された DocVersion オブジェクトを構成要素とする CdbDocument オブジェクトである。
- RenditionType がテキスト型 (MIME::text/ で始まる MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

更新が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合に、ファイル "/tmp/sample.doc" で更新すると、"sample.doc" が設定されます。Windows の場合に、ファイル "c:\temp\sample.doc" で更新すると、"sample.doc" が設定されます。UNIX の場合に、文書に登録されたコンテンツの内容をファイル "/tmp/sample.doc" で更新する例を次に示します。Windows の場合は、ファイル "c:\temp\sample.doc" で更新する例として、例中の "file:///tmp/sample.doc" を "file:///c:\temp\sample.doc" と読み替えてください。

文書の内容 (コンテンツ) を更新する例

```
pDbrDoc->UpdateContent ("file:///tmp/sample.doc");
```

シングルファイル文書の場合 (RenditionType を指定したファイルの更新)

マルチレンディション文書のサブレンディションであるシングルファイル文書を更新する場合は、更新に使用するファイルのパスのほかに、更新対象のサブレンディションを pRenditionType で指定します。なお、サブレンディションの内容を更新する場合、全文検索インデクスは作成できません。

マルチファイル文書の場合

マルチファイル文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_PATHLIST 構造体を指定します。

マルチファイル文書の場合、次のことに注意してください。

- CdbDocument クラスに格納した複数のファイルを、SDBR_PATHINFO 構造体に設定したファイルパスのファイルで、一括して更新します。SDBR_PATHINFO 構造体は、引数 PathList で指定する SDBR_PATHINFO 構造体に設定します。なお、特定のコンテンツだけを更新することはできません。また、更新後は、更新前のファイルは使用できません。
- 更新前のファイルを格納した BLOB のデータが書き換えられるため、BLOB のデータの長さは、更新後のファイルサイズの合計になります。
- この形式の UpdateContent を、マルチファイル文書ではないバージョンなし文書に対して実行すると、ERR_NOT_MULTI_CONTENTS のエラーになります。
- マルチファイル文書では、テキストファイルの部分が特定できないため、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、マルチファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

File Link 文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、更新に使用するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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でコンテンツを更新できます。また、すでに HiRDB File Link で管理されているデータでコンテンツを更新することもできます。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効です。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。
- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。

引数

pFilePath (入力)

更新に使用するファイルのパス名を指定します。ファイルのパス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" を登録するには、"file:///c:\temp\sample.doc" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

なお、ここで指定するパスは、GetContent メソッドで取得したファイルである必要はありません。ただし、RenditionType が異なるファイルは指定できません。

pRenditionType (入力)

RenditionType を指定してレンディションの内容を更新する場合、内容を更新するレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象文書のサブレンディションとして存在しない場合はエラーになります。

マスタレンディションの内容を更新する場合は NULL を指定するか、「マスタレンディションのファイルを更新する場合」の形式を使用することもできます。なお、空文字列 ("") は指定できません。

マルチファイル文書の場合、NULL または「シングルファイル文書の場合 (マスタレンディションのファイルの更新)」の形式を指定してください。

ICreateIndexFlg (入力)

全文検索インデクスを作成するかどうかを指定します。

次のどちらかを指定してください。この値を省略した場合は、DBR_CREATE_INDEX が指定されたものとして処理されます。

- DBR_CREATE_INDEX
pFilePath に指定したパス名のファイルから全文検索インデクスが作成されます。

- DBR_NOT_CREATE_INDEX
全文検索インデクスが作成されません。

なお、CdbDocument オブジェクトの構成要素が全文検索機能付きクラス以外から作成された DocVersion オブジェクトである場合、およびファイルの RenditionType がテキスト型でない場合は、lCreateIndexFlg の値は無視されます。

また、マルチレンディション文書の場合、lCreateIndexFlg はマスタレンディションの内容を更新する場合だけ有効です。

PathList (入力)

更新に使用するファイルのファイルパス情報が設定された SDBR_PATHLIST 構造体を指定します。SDBR_PATHLIST 構造体については、「付録 C.1(11) SDBR_PATHLIST 構造体 (ファイルパス情報リスト構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を更新するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。または、マルチファイル文書で指定したファイルのサイズの合計が、BLOB の定義長を超えました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_CONTENTSCOUNT	登録しようとするファイルの数が多過ぎます。	登録するファイルの数を見直してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_NOT_MULTI_CONTENTS	マルチファイル文書の文書ではありません。	マルチファイル管理機能を使用しない場合の形式で実行してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。

8. CnbrDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> • ネットワーク環境を確認してください。 • _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 • 時間を置いて、再度実行してください。 • ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbDocument::UpdateContentAndRenditionType メソッド

機能

ファイルの更新および RenditionType の変更

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの更新)
DmaBoolean UpdateContentAndRenditionType(
    const pDmaString_T      pFilePath,
    pDmaString_T           pRenditionType,
    DmaInteger32            lCreateIndexFlg = DBR_CREATE_INDEX)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの更新)
DmaBoolean UpdateContentAndRenditionType(
    const pDmaString_T      pFilePath,
    pDmaString_T           pBeforeRenditionType,
    pDmaString_T           pAfterRenditionType,
    DmaInteger32            lCreateIndexFlg)

// リファレンスファイル文書の場合
DmaBoolean UpdateContentAndRenditionType(
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T             pBeforeRenditionType,
    pDmaString_T             pAfterRenditionType)

// File Link文書の場合
DmaBoolean UpdateContentAndRenditionType(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T             pBeforeRenditionType,
    pDmaString_T             pAfterRenditionType)

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32  lContentOperateMode; // 操作モード
    pDmaString_T  pEntry;              // ファイルのパスおよびロケーション
    pDmaString_T  pTargetPath;        // コンテント格納先パス
    pDmaString_T  pDeleteRootPath;    // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T  pFilePath;          // 登録ファイルまたはディレクトリのURL
    pDmaString_T  pFamName;          // FAM名
    pDmaString_T  pFamFilePath;      // コンテント保管先パス
    DmaInteger32  lContentDeleteMode; // コンテント削除モード
    DmaInteger32  lFileNamingMode;   // コンテント名自動生成モード
};
```

詳細

登録されている CdbDocument オブジェクトの内容を、指定したパスのファイルで更新するメソッドです。更新するときには、RenditionType も変更できます。

シングルファイル文書の場合 (マスタレンディションのファイルの更新)

シングルファイル文書 (マルチレンディション文書の場合はマスタレンディションであるシングルファイル文書) を更新する場合は、更新に使用するファイルのパスと全文検索インデクスを作成するかどうかを指定できます。RenditionType を変更する場合は、pRenditionType に変更後の RenditionType を指定します。

内容を更新する CdbDocument オブジェクト (マルチレンディション文書の場合はマスタレンディション) が次の条件に当てはまる場合に、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデックスを作成できます。

- 全文検索機能付き文書クラスを基に作成された DocVersion オブジェクトを構成要素とする CdbDocument オブジェクトである。
- RenditionType がテキスト型 (MIME::text/ で始まる MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

更新が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合に、ファイル "/tmp/sample.doc" で更新すると、"sample.doc" が設定されます。Windows の場合に、ファイル "c:\temp\sample.doc" で更新すると、"sample.doc" が設定されます。UNIX の場合に、文書に登録されたコンテンツの内容をファイル "/tmp/sample.html" で更新して RenditionType を html 形式に変更する例を次に示します。Windows の場合は、ファイル "c:\temp\sample.html" の内容で更新する例として、例中の "file:///tmp/sample.html" を "file:///c:\temp\sample.html" と読み替えてください。

文書の内容 (コンテンツ) を更新して RenditionType を変更する例

```
pDbrDoc->UpdateContentAndRenditionType
("file:///tmp/sample.html", "MIME::text/html");
```

シングルファイル文書の場合 (RenditionType を指定したファイルの更新)

マルチレンディション文書のサブレンディションであるシングルファイル文書を更新する場合は、更新に使用するファイルのパスのほかに、更新対象のサブレンディションを引数 pBeforeRenditionType で指定します。RenditionType を変更する場合は、引数 pAfterRenditionType に変更後の RenditionType を指定します。なお、サブレンディションの内容を更新する場合、全文検索インデックスは作成できません。

リファレンスファイル文書の場合

リファレンスファイル文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。

リファレンスファイル文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentReference オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_REFERENCE の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

File Link 文書を更新する場合は、更新に使用するコンテンツの情報として、

SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、更新に使用するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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でコンテンツを更新できます。また、すでに HiRDB File Link で管理されているデータでコンテンツを更新することもできます。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効です。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

引数

pFilePath (入力)

更新に使用するファイルのパス名を指定します。ファイルのパス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" を登録するには、"file:///c:\temp\sample.doc" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

なお、ここで指定するパスは、GetContent メソッドで取得したファイルである必要はありません。ただし、RenditionType が異なるファイルは指定できません。

pRenditionType (入力)

文書 (マルチレンディション文書の場合はマスタレンディション) の内容を更新すると同時にその RenditionType を変更する場合、変更後の RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。NULL を指定した場合、RenditionType は変更されません。また、pFilePath に指定したファイルの拡張子から DocumentBroker が RenditionType を決定することはありません。

なお、マルチレンディション文書 (レンディションが複数ある文書) の場合、空文字列 ("") は指定できません。

pBeforeRenditionType (入力)

RenditionType を指定してレンディションの内容を更新する場合、更新対象のレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象文書のサブレンディションとして存在しない場合はエラーになります。

マスタレンディションの内容を更新する場合は NULL を指定するか、「マスタレンディションのファイルを更新する場合」の形式を使用することもできます。

pAfterRenditionType (入力)

RenditionType を指定してレンディションの内容を更新すると同時にその RenditionType を変更する場合、変更後の RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象文書のサブレンディションとしてすでに登録されている場合はエラーになります。ただし、pBeforeRenditionType 値と pAfterRenditionType 値とが同一値であってもエラーになりません。NULL を指定した場合は、サブレンディションの RenditionType は変更されません。また、pFilePath または ReferencePath の pEntry メンバに指定したファイルの拡張子から DocumentBroker が RenditionType を決定することはありません。

なお、空文字列 ("") は指定できません。

ICreateIndexFlg (入力)

全文検索インデクスを作成するかどうかを指定します。

次のどちらかを指定してください。この値を省略した場合は、DBR_CREATE_INDEX が指定されたものとして処理されます。

- DBR_CREATE_INDEX
pFilePath に指定したパス名のファイルから全文検索インデクスが作成されます。
- DBR_NOT_CREATE_INDEX
全文検索インデクスが作成されません。

なお、CdbDocument オブジェクトの構成要素が全文検索機能付きクラス以外から作成された DocVersion オブジェクトである場合、およびファイルの RenditionType がテキスト型でない場合は、ICreateIndexFlg の値は無視されます。

また、マルチレンディション文書の場合、ICreateIndexFlg はマスタレンディションの内容を更新する場合だけ有効です。

ReferencePath (入力)

リファレンスファイル文書を更新するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。

コンテンツを持たないオブジェクトを更新する場合

SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath に、コンテンツ格納先パスを、コンテンツ格納先ベースパスを基点とした相対パスで指定します。ContentLocation プロパティに格納されるパスは、次のように条件によって異なります。

- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合 ContentLocation プロパティには、pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ、および pEntry で指定したファイル名が連結されて格納されます。
- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合 ContentLocation プロパティには、文書登録時に pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ、DocumentBroker が指定するファイル名が連結されて格納されます。

コンテンツを持つオブジェクトを更新する場合

ContentLocation プロパティに格納されるパスは、次のように条件によって異なります。

- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合 ContentLocation プロパティには、更新前の ContentLocation プロパティのファイル名を除くパスに、SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry で指定したファイル名が連結されて格納されます。
- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合 ContentLocation プロパティには、文書登録時に pTargetPath で指定したパス、DocumentBroker がコンテンツを管理するためのディレクトリ、DocumentBroker が指定するファイル名が連結されて格納されます。

なお、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合 次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツを持たないオブジェクトにコンテンツを登録すると、コンテンツロケーションの指定で ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

8. CdbbrDocument クラス (バージョンなし文書を扱うクラス)

- CdbbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

また、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツ登録済みのオブジェクトのコンテンツを更新すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツを持たないオブジェクトにコンテンツを登録すると、コンテンツロケーションの指定で ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツ登録済みのオブジェクトのコンテンツを更新すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したパス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書を更新する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を更新するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には、DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INV ALID	クラスライブラリのオブジェク トは無効です。	有効なクラスライブラリのオブ ジェクトを指定してください。
ERR_DBR	ERR_CONTENTTYPE_MIS MATCH	指定したレンディションタイプ のコンテンツに対して実行でき ません。	正しい文書オブジェクトを指定 してメソッドを実行してくださ い。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリ がロードできません。	詳細メッセージを基に対処して ください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発 生しました。	詳細メッセージを基に対処して ください。
ERR_DBR	ERR_FILE_REFERENCE_ MISMATCH_STATUS	リファレンスファイル管理機能 で扱うサーバマシンでのコンテ ントの操作でエラーが発生し、 オブジェクトとコンテンツが不 整合な状態となりました。	文書を削除したあと、更新後の コンテンツを使用して文書を作 成してください。
ERR_DBR	ERR_FILE_REFERENCE_ OPERATION_FAILED	リファレンスファイル管理機能 で扱うサーバマシンでのコンテ ントの操作でエラーが発生しま した。	詳細メッセージを基に対処して ください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDI TION	指定された RenditionType は 存在しません。	RenditionType を見直してくだ さい。
ERR_DBR	ERR_OBJECT_NOT_CONN ECT	オブジェクトは接続されていま せん。	オブジェクトを接続してくださ い。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールし てから、このメソッドをコール してください。
ERR_DBR	ERR_REFERENCETYPE_ MISMATCH	オブジェクト種別が一致してい ません。	オブジェクト種別または指定し た操作モード (IContentOperateMode) を見 直してください。

8. CdbbrDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。

8. CdbDocument クラス (バージョンなし文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none">• ネットワーク環境を確認してください。• <code>_HIEDMS_FTPMODE=STATIC</code> を指定してファイル転送サービスを起動しているかどうかを、確認してください。• 時間を置いて、再度実行してください。• ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

9

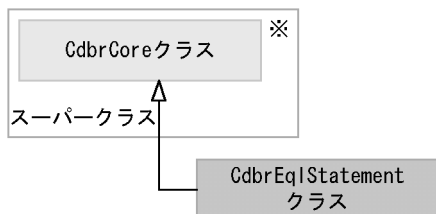
CdbrEqStatement クラス (edmSQL 文を操作，実行するクラス)

検索機能として，edmSQL で記述した edmSQL 文を操作，実行する機能を提供するクラスです。

CdbrEqStatement クラスの詳細
CdbrEqStatement::~CdbrEqStatement メソッド
CdbrEqStatement::CdbrEqStatement メソッド
CdbrEqStatement::ChangeACLMode メソッド
CdbrEqStatement::ChangeGetObjCount メソッド
CdbrEqStatement::ChangeLockType メソッド
CdbrEqStatement::DeleteResult メソッド
CdbrEqStatement::Execute メソッド
CdbrEqStatement::GetACLMode メソッド
CdbrEqStatement::GetResult メソッド
CdbrEqStatement::Initialize メソッド
CdbrEqStatement::Set メソッド
CdbrEqStatement::SetObjParam メソッド
CdbrEqStatement::SetOIIDParam メソッド
CdbrEqStatement::SetParam メソッド
CdbrEqStatement::Terminate メソッド

CdbxEqlStatement クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

なし

メソッド一覧

CdbxEore クラスから継承されるメソッド

メソッド	機能
~CdbxEore	デストラクタ
CdbxEore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbxEore クラス (エラー情報を取得するクラス)」を参照してください。

CdbxEqlStatement クラスで導入されるメソッド

メソッド	機能
~CdbxEqlStatement	デストラクタ
CdbxEqlStatement	コンストラクタ
ChangeACLMode	アクセス制御モードの変更
ChangeGetObjCount	検索結果の取得件数の変更
ChangeLockType	ロック種別の変更
DeleteResult	検索結果の削除
Execute	設定済み edmSQL 文の実行
GetACLMode	アクセス制御モードの取得
GetResult	検索結果の取得
Initialize	オブジェクトの初期化とセッションオブジェクトへの登録
Set	edmSQL 文の設定
SetObjParam	edmSQL 文の ? パラメタ (DMA オブジェクト) の設定

メソッド	機能
SetOIIDParam	edmSQL 文の ? パラメタ (OIID) の設定
SetParam	edmSQL 文の ? パラメタの設定
Terminate	セッションオブジェクトへの登録抹消

以降, Cdbreqlstatement クラスで導入されるメソッドについて説明します。

CdbxEqlStatement::~~CdbxEqlStatement メソッド

機能

デストラクタ

形式

~CdbxEqlStatement ()

詳細

CdbxEqlStatement オブジェクトのデストラクタです。

CdbxEqlStatement オブジェクトを削除します。また, CdbxEqlStatement クラスで管理している情報も削除します。

引数

なし

戻り値

なし

戻り値の詳細

なし

Cdbreqlstatement::Cdbreqlstatement メソッド

機能

コンストラクタ

形式

`Cdbreqlstatement ()`

詳細

`Cdbreqlstatement` オブジェクトのコンストラクタです。

`Cdbreqlstatement` オブジェクトを生成します。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbxEqlStatement::ChangeACLMode メソッド

機能

アクセス制御モードの変更

形式

```
DmaBoolean ChangeACLMode (
    DmaInteger32          lACLMode)
```

詳細

アクセス制御機能付きで検索 (edmSQL 文) を実行するかどうかを示すアクセス制御モードを変更するメソッドです。

このメソッドで変更したモードは, CdbxEqlStatement::Execute メソッドのコール時に有効になります。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールするとエラーになります。また, 戻り値が DMA_FALSE の場合, 設定したアクセス制御モードは保証できません。

なお, アクセス制御モードの設定は, 文書空間でアクセス制御機能を使用している場合だけ有効になります。

引数

lACLMode (入力)

検索 (edmSQL 文) 実行時のアクセス制御モードを指定します。

次のどちらかを指定してください。ただし, アクセス制御機能をサポートしていない文書空間の使用時に DBR_QUERY_WITH_ACL を指定した場合にはエラーになります。

- DBR_QUERY_WITH_ACL
アクセス制御機能付き検索モードで実行します。このモードの場合, ユーザはアクセス権がないオブジェクトを検索結果として取得できません。
- DBR_QUERY_WITHOUT_ACL
アクセス制御機能なし検索モードで実行します。このモードの場合, ユーザはアクセス権がないオブジェクトでも検索結果として取得できます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御対応のデータベースを使用していません。	アクセス制御付き検索の実行モードを変更する場合は, アクセス制御対応のデータベースを使用してください。
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。 または、サポートされていないデータ型です。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxEqlStatement::ChangeGetObjCount メソッド

機能

検索結果の取得件数の変更

形式

```
DmaBoolean ChangeGetObjCount (
    DmaInteger32          lGetObjCount)
```

詳細

一度に取得する検索結果 (edmSQL 文の実行結果) の件数を変更するメソッドです。

このメソッドで変更した取得件数は , CdbxEqlStatement::Execute メソッドのコール時に有効になります。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールすると , エラーになります。また , 戻り値が DMA_FALSE の場合 , 設定した取得件数は保証できません。

引数

lGetObjCount (入力)

一度の CdbxEqlStatement::GetResult メソッドのコールで取得する検索結果 (edmSQL 文の実行結果) の取得件数を指定します。0 より大きい値か - 1 を指定してください。- 1 を指定した場合は , すべての実行結果が取得されます。- 1 より小さい値 , または 0 を指定した場合にはエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZE	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

CdbxEqlStatement::ChangeLockType メソッド

機能

ロック種別の変更

形式

```
DmaBoolean ChangeLockType(
    DmaInteger32      lLockType)
```

詳細

検索結果 (edmSQL 文の実行結果) として取得するオブジェクトに設定するロック種別を変更するメソッドです。

このメソッドで変更したロック種別は, CdbxEqlStatement::Execute メソッドのコール時に有効になります。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールすると, エラーになります。また, 戻り値が DMA_FALSE の場合, 設定したロック種別は保証できません。

引数

lLockType

検索結果 (edmSQL 文の実行結果) として取得したオブジェクトに設定するロックの種別を指定します。

次に示す値のどれかを指定してください。これら以外のロック種別を指定した場合にはエラーになります。

- DBR_WITHOUT_LOCK
ロックを設定しません。
- DBR_LOCK_READ
read ロックを設定します。このロックを設定した場合, ほかのクライアントから write ロックを設定できません。
- DBR_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合, ほかのクライアントから read ロックおよび write ロックを設定できません。
- DBR_LOCK_READ_FOR_UPDATE
read ロックを設定します。ただしこの場合は, 検索結果として取得したオブジェクトを更新または削除できます。
このロックは, 次に示すような理由で write ロックを設定できない場合で, 検索結果を分割して取得 (CdbxEqlStatement::ChangeGetObjCount メソッドで結果取得件数に - 1 以外の値を指定し, かつ, CdbxEqlStatement::GetResult メソッドの第 1 引数 bSearchContinue を DMA_TRUE にして実行) し, その取得したオブジェクトを更新または削除するようなときに指定します。
 - 検索対象クラスを結合している
 - write ロックを設定したくない

注意

- DBR_WITHOUT_LOCK を指定した場合, 検索 (edmSQL 文) 実行時にデータベースはロックされません。このため, 取得した結果のオブジェクトを使用する段階で, すでにデータベースの内容が書き換えられている可能性があります。取得結果のオブジェクトとデータベースのデータの整合性を確保したい場合は, ロックタイプを指定し, 明示的にロックを設定するようにしてください。
- DBR_LOCK_READ_FOR_UPDATE を指定した場合には, データベース内に作業表 (検索結果の

整合性を更新作業から守るために検索結果を一時的にコピーしておく作業領域)が作成されます。このため, DBR_LOCK_READ を指定した場合より処理速度が遅くなる可能性があります。したがって, このロックタイプは, 検索対象クラスを結合していて, かつ, 取得した結果のオブジェクトを更新または削除する場合に限って使用するようになっています。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

CdbxEqlStatement::DeleteResult メソッド

機能

検索結果の削除

形式

```
DmaBoolean DeleteResult()
```

詳細

保持している検索結果 (edmSQL 文の実行結果) を削除するメソッドです。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールすると、エラーになります。CdbxEqlStatement::Initialize メソッドをコールしたあとであれば、検索結果 (edmSQL 文の実行結果) が存在しない状態でこのメソッドをコールしてもエラーにはなりません。何もせずに正常終了します。

戻り値が DMA_FALSE の場合、検索結果 (edmSQL 文の実行結果) は正常に削除できていません。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_EQL_NOT_INITIALIZ E	初期化が実行されていません。	初期化を実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CON NECT	セッションが接続されていま せん。	セッションを接続してくださ い。

9. CdbxEqlStatement クラス (edmSQL 文を操作 , 実行するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbxEqlStatement::Execute メソッド

機能

設定済み edmSQL 文の実行

形式

```
DmaBoolean Execute (
    ppDmaString_T      ppParseMessage = NULL)
```

詳細

検索 (edmSQL 文) を実行するメソッドです。CdbxEqlStatement::Set メソッドで登録されている edmSQL 文に, CdbxEqlStatement::SetObjParam メソッド, CdbxEqlStatement::SetOIIDParam メソッドまたは CdbxEqlStatement::SetParam メソッドで設定された ? パラメタの値を設定して SQL 文を生成し, 実行します。

edmSQL 文が正常に実行できた場合, CdbxEqlStatement::Initialize メソッドの初期値 (デフォルト値は - 1 で全件取得するという意味) または CdbxEqlStatement::ChangeGetObjCount メソッドで設定される件数分の検索結果を取得して, クライアントに保持します。ただし, この時点では検索結果をクライアントに保持するだけ なので, その取得には CdbxEqlStatement::GetResult メソッドを使用してください。

注

このような処理をするのは, CdbxEqlStatement::GetResult メソッドの 1 回目のコール時に, サーバと通信しなくても結果を取得できるようにするためです (通信によるオーバーヘッドを 1 回分削減できる)。2 回目以降 (結果の継続取得時) はサーバと通信して, 結果を取得します。

次に示すメソッドがコールされていない状態でこのメソッドをコールした場合は, エラーになります。

- CdbxEqlStatement::Initialize メソッド
- CdbxEqlStatement::Set メソッド

戻り値が DMA_FALSE の場合, 検索 (edmSQL 文) 実行後の内容は保証できません。

edmSQL 文中の ? パラメタに関する注意

edmSQL 文中に ? パラメタが存在する場合, CdbxEqlStatement::SetObjParam メソッド, CdbxEqlStatement::SetOIIDParam メソッドまたは CdbxEqlStatement::SetParam メソッドで設定された値で SQL 文を実行しますが, edmSQL 文中の ? パラメタと設定された ? パラメタの情報に不整合がある場合の動作を次に示します。

- edmSQL 文中の ? パラメタの値が未設定の場合
NULL を仮定して動作します。
- edmSQL 文中の ? パラメタの個数より大きいインデクス (出現順) で値が設定されている場合
エラーになり, major_code として ERR_DMA, minor_code として EDMRC_QPARAM_INDEX_OUT_OF_RANGE が返却されます。
- 指定できるデータ型以外の値が設定されている場合
エラーになり, major_code として ERR_DMA, minor_code として EDMRC_QPARAM_BAD_VALUE が返却されます。

edmSQL 文のエラーチェックについて

構文解析中に edmSQL 文のエラーを検出した場合には, major_code として ERR_DBR, minor_code

として ERR_EQL_BAD_STATEMENT が返却され, edmSQL 文の構文解析の実行結果メッセージが ppParseMessage に設定されます。ただし, ppParseMessage が指定されていない (省略したまたは NULL を指定した) 場合は, メッセージは設定されません。ppParseMessage が指定されていて, 構文解析が正常に終了した場合には NULL が設定されます。なお, ppParseMessage がポイントするメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。

構文にエラーがあった場合の対処については, マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」の「4. オブジェクトの検索」を参照してください。

edmSQL のキャッシュ機能について

一度実行した edmSQL 文の構文解析の実行結果は, キャッシュされます。そして, 次に実行する edmSQL 文と照合され, すでにキャッシュされている情報と同内容であると, 解析済みの情報が再利用されます。このため, 一度構文解析した edmSQL 文を再度, 構文解析するときには処理が高速になります。キャッシュされた情報は, 同一のセッション内で保持されます。

edmSQL の制限について

「dbrProp_」で始まるクラスライブラリ固有のプロパティは, edmSQL 文に指定できません。また, Binary 型, Float 型, DateTime 型のデータに関する処理は未サポートです。このような edmSQL の制限については, マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」の「4. オブジェクトの検索」を参照してください。また, 最終的に実行される SQL のバイト長制限など, 実行可能な SQL 文については, データベースの仕様に従います。データベース仕様の詳細については, マニュアル「HiRDB SQL リファレンス」を参照してください。

アクセス制御機能に関する注意

- アクセス制御機能を利用するかどうかは, 設定されているアクセス制御モードに従います。
- アクセス制御機能付き検索モードで実行する場合, アクセス制御に必要な情報が, 内部的に検索条件の SELECT 句に追加されます。このため, 取得できるプロパティ数は, アクセス制御機能なし検索を実行する場合よりも少なくなります。

オブジェクトを削除するメソッドと同時に実行する場合の注意

RemoveObject メソッドや DeleteVersion メソッドなどのオブジェクトを削除するメソッドと同時に実行する場合, 次の点に注意してください。

- オブジェクトを削除するメソッドを実行してそのトランザクションが完了する前に, 削除したオブジェクトに対しこのメソッドを実行すると, major_code として ERR_DMA, minor_code として DMARC_NOT_FOUND が返却されます。

引数

ppParseMessage

edmSQL 文の構文解析の実行結果メッセージを格納する領域へのアドレスを指定します。

アドレスが指定されていない場合 (NULL を指定した場合) には, 構文解析で edmSQL 文のエラーが検出されてもメッセージは設定されません。

省略した場合には NULL を指定したときと同じ動作をします。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_EQL_BAD_STATEMENT	不正な edmSQL 文を実行しようとした。	edmSQL 文を見直してください。必要に応じて構文解析結果情報を取得してください。
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DBR	ERR_EQL_STATEMENT_NOT_SET	実行する edmSQL 文が登録されていません。	CdbxEqlStatement::Set メソッドで実行する edmSQL 文を登録してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。または、サポートされていないデータ型です。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。
ERR_DMA	EDMRC_QPARAM_BAD_VALUE	? パラメタの設定で指定した値が、不正である可能性があります。	設定した ? パラメタの値、データ型を見直してください。
ERR_DMA	EDMRC_QPARAM_INDEX_OUT_OF_RANGE	? パラメタの設定で指定した値の個数を超える出現順序が指定されています。	? パラメタの設定を見直して、正しい出現順序を指定してください。

CdbxEqlStatement::GetACLMode メソッド

機能

アクセス制御モードの取得

形式

```
DmaBoolean GetACLMode(
    pDmaInteger32          plACLMode)
```

詳細

アクセス制御機能付きで検索 (edmSQL 文) を実行するかどうかを示すアクセス制御モードを取得するメソッドです。

このメソッドで取得するモードは、CdbxEqlStatement::Execute メソッド実行時に確定したモード (実際に動作しているモード) です。CdbxEqlStatement::Execute メソッド実行前に

CdbxEqlStatement::ChangeACLMode メソッドで指定したモードは返却されないので注意してください。

なお、一度も CdbxEqlStatement::Execute メソッドを実行していない状態でこのメソッドを実行した場合、DBR_QUERY_WITHOUT_ACL が返却されます。CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールすると、エラーになります。また、アクセス制御機能を使用していない文書空間に接続しているときにこのメソッドをコールするとエラーになります。

戻り値が DMA_FALSE の場合は、取得したアクセス制御モードは保証できません。

引数

plACLMode (出力)

取得したアクセス制御モードが設定されます。モードが設定される領域を指定してください。領域が指定されていない場合 (NULL である場合) にはエラーになります。

次のどちらかが設定されます。

- DBR_QUERY_WITH_ACL
アクセス制御機能付きで検索 (edmSQL 文) を実行するモードが設定されていることを示します。
- DBR_QUERY_WITHOUT_ACL
アクセス制御機能なしで検索 (edmSQL 文) を実行するモードが設定されていることを示します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御対応のデータベースを使用していません。	アクセス制御付き検索の実行モードを変更する場合は, アクセス制御対応のデータベースを使用してください。
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

CdbxEqlStatement::GetResult メソッド

機能

検索結果の取得

形式

```
DmaBoolean GetResult(
    DmaBoolean          bSearchContinue,
    pDmaInteger32       plSearchCount,
    SDBR_QUERYRESULT** ppQueryResult)

    struct SDBR_PROPERTIES {
        DmaInteger32    lType;           //データ型
        DmaInteger32    lCardinality;    //基本単位
        DmaInteger32    lCount;         //プロパティ値の個数
        union {          //プロパティ値を
                        // 格納する共用体
            pDmaBoolean    pbBoolean;   //Boolean
            pDmaId         pId;         //ID
            pDmaInteger32  plInteger;    //Integer
            pDmapv         pplObject;   //Object
            ppDmaString_T  ppString;    //String
        } uValue;
    };

// 検索結果構造体
    struct SDBR_QUERYRESULT {
        DmaInteger32    lPropCount;     //プロパティ個数
        SDBR_PROPERTIES* pProperties;   //プロパティ情報
    };
```

詳細

直前に実行した CdbxEqlStatement::Execute メソッドで得られた検索結果 (edmSQL 文の実行結果) を取得するメソッドです。

取得できたデータの件数は plSearchCount に , データを格納した検索結果構造体の配列のポインタ値は ppQueryResult に , それぞれ設定されます。

検索結果データを格納するための構造体のメモリは , このメソッド内で確保されます。このメモリは , ユーザプログラムで dbrDelete 関数を使用して解放してください。なお , このメソッドをコールするたびにメモリは確保されるので , コールするごとにメモリを解放する必要があります。

検索結果が 0 件の場合には , 検索結果データの構造体は NULL です。ただし , 途中でエラーが発生した場合には , 実行結果データの構造体のアドレスはユーザが指定した値のままで , NULL は設定されません。

SDBR_QUERYRESULT 構造体のメンバである SDBR_PROPERTIES 構造体の uValue (プロパティ値を格納する共用体) には , Boolean 型 ・ ID 型 ・ Integer 型 ・ Object 型 ・ String 型の値が格納されます。

検索結果が可変長配列の場合は , CdbxEqlStatement クラスのオブジェクト (Object 型) が格納されます。このとき lType にはデータ型として DMA_DATATYPE_OBJECT が , lCardinality には基本単位として EDM_DMA_CARDINALITY_VARRAY が格納されます。

戻り値が DMA_FALSE の場合 , ppQueryResult のポインタ値およびそのポインタ値が示す検索結果データの構造体の内容は保証できません。

なお , アクセス制御機能付きで検索 (edmSQL 文) を実行した場合は , 検索結果集合のうち , アクセス権

がある検索結果だけが取得できます。

次に示すメソッドがコールされていない状態でこのメソッドをコールした場合は, エラーになります。

- CdbxEqlStatement::Initialize メソッド
- CdbxEqlStatement::Execute メソッド

引数

bSearchContinue (入力)

検索結果の取得方法を指定します。次に示す値のどちらかを指定してください。ただし, 検索実行後, 初めて検索結果を取得する場合は, どちらの値を指定しても検索結果の先頭から取得されます。

- DMA_FALSE
検索結果の先頭から取得します。
- DMA_TRUE
前回取得した検索結果の次の検索結果から取得します。

plSearchCount (出力)

実際に取得した検索結果の件数が設定されます。件数が設定される領域を指定してください。この値が, CdbxEqlStatement::ChangeGetObjCount メソッドで設定した取得件数よりも小さい場合は, その値を超える検索結果は存在しません。つまり, 検索結果の最終行の取り出しが終了しています。

件数が設定される領域が指定されていない場合 (NULL を指定した場合) はエラーになります。

ppQueryResult (出力)

検索結果を格納する検索結果構造体である, SDBR_QUERYRESULT 構造体へのポインタが設定されます。SDBR_QUERYRESULT 構造体およびそのプロパティ情報が設定される SDBR_PROPERTIES 構造体については, 「付録 C.2(1) 検索結果として取得する構造体」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して, ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて, 再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して, 障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて, 再度実行してください。
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。

9. CdbxEqlStatement クラス (edmSQL 文を操作 , 実行するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_STATEMENT_NOT_EXEC	edmSQL 文が実行されていません。	CdbxEqlStatement::Execute メソッドを実行し , edmSQL 文を実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては , 一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて , 処理を再実行してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に , この操作を実行してください。

CdbxEqlStatement::Initialize メソッド

機能

オブジェクトの初期化とセッションオブジェクトへの登録

形式

```
DmaBoolean Initialize(
    CdbxEsession* pDbxEsession)
```

詳細

CdbxEqlStatement クラスを初期化处理し , CdbxEsession オブジェクトとの関連づけをするメソッドです。

初期化处理の内容を次に示します。

クラスの要素	初期値	内容
ロック種別	DBR_WITHOUT_LOCK	ロックを設定しない
検索結果の取得件数	-1	全行取得
アクセス制御モード	接続した文書空間に依存したモード	アクセス制御機能をサポートしている文書空間の場合には DBR_QUERY_WITH_ACL , サポートしていない文書空間の場合には DBR_QUERY_WITHOUT_ACL

戻り値が DMA_FALSE の場合 , クラスの初期化は正常に終了していません。

引数

pDbxEsession (入力)

CdbxEsession オブジェクトを指定します。CdbxEsession オブジェクトが指定されていない場合 (NULL を指定した場合) はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_ALREADY_INITIALIZED	すでに初期化されています。	この操作は実行できません。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

CdbxEqlStatement::Set メソッド

機能

edmSQL 文の設定

形式

```
DmaBoolean Set (
    pDmaString_T      pEqlStatement)
```

詳細

実行予定の edmSQL 文を設定するメソッドです。

このメソッドで設定した edmSQL 文は , CdbxEqlStatement::Execute メソッドのコール時に有効になります。

なお , このメソッドでは設定された edmSQL 文のチェックはしません。このメソッドは実行予定の edmSQL 文を設定するだけです。edmSQL 文のチェックは CdbxEqlStatement::Execute メソッドで実施されます。したがって , edmSQL 文の構文に関するエラーは CdbxEqlStatement::Execute メソッドで返却されます。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールした場合は , エラーになります。

戻り値が DMA_FALSE の場合 , 設定された edmSQL 文は保証できません。

edmSQL 文の仕様と文法については , マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」の「4. オブジェクトの検索」を参照してください。

引数

pEqlStatement (入力)

実行予定の edmSQL 文を指定します。edmSQL 文が指定されていない場合 (NULL を指定した場合) はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZE	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて , 処理を再実行してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxEqlStatement::SetObjParam メソッド

機能

edmSQL 文の ? パラメタ (DMA オブジェクト) の設定

形式

```
DmaBoolean SetObjParam (
    DmaInteger32      lParamIndex,
    pDmaString_T     pOIID)
```

詳細

edmSQL 文中の ? パラメタに DMA オブジェクトを設定するメソッドです。なお, DMA オブジェクトは直接指定できないため, オブジェクトの識別子である OIID を指定します。

このメソッドで設定した ? パラメタの値 (DMA オブジェクト) は, CdbxEqlStatement::Execute メソッドのコール時に有効になります。? パラメタが存在しない edmSQL 文, または ? パラメタに DMA オブジェクトを設定しない edmSQL 文を実行する場合は, このメソッドをコールする必要はありません。

? パラメタの値 (DMA オブジェクト) は, 最後に設定した値が有効になります。設定し直していない ? パラメタの値は有効なままです。ただし, CdbxEqlStatement::Set メソッドがコールされた場合には, すでに設定されている ? パラメタのすべての値が無効になります。つまり, CdbxEqlStatement::Set メソッドで設定した edmSQL 文の, すべての ? パラメタについて値を設定し直す必要があります。

なお, ? パラメタに OIID を設定する場合には CdbxEqlStatement::SetOIIDParam メソッドを使用し, OIID でも DMA オブジェクトでもない値を設定する場合には CdbxEqlStatement::SetParam メソッドを使用してください。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールした場合は, エラーになります。

戻り値が DMA_FALSE の場合, 設定された ? パラメタの値 (DMA オブジェクト) は保証できません。

引数

lParamIndex (入力)

edmSQL 文中の DMA オブジェクトを設定する ? パラメタの出現順序を指定します。

1 ~ (edmSQL 文中の ? パラメタの数) の範囲で指定してください。0 以下の値を指定した場合には, エラーになります。0 より大きい値で範囲外の値を指定した場合は, CdbxEqlStatement::Execute メソッドのコール時にエラーになります。

また, 同一の ? パラメタに対して, 複数回, 値を設定した場合には, 最後に設定した値が有効になります。

pOIID (入力)

edmSQL 文中の ? パラメタに設定する値を指定します。DMA オブジェクトの OIID を指定してください。

指定した値が OIID でない場合および NULL の場合にはエラーになります。OIID かどうかは, 次に示す観点からチェックされます。

- 指定された文字列が「dma:///」で始まっているかどうか
- 指定された文字列長が正しい (133 バイト) かどうか

指定した値が、構文不正な値となる場合は、CdbxEqlStatement::Execute メソッドのコール時にエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZE	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxEqlStatement::SetOIIDParam メソッド

機能

edmSQL 文の ? パラメタ (OIID) の設定

形式

```
DmaBoolean SetOIIDParam (
    DmaInteger32      lParamIndex,
    pDmaString_T     pOIID)
```

詳細

edmSQL 文中の ? パラメタに OIID を設定するメソッドです。

このメソッドで設定した ? パラメタの値 (OIID) は, CdbxEqlStatement::Execute メソッドのコール時に有効になります。? パラメタが存在しない edmSQL 文, または ? パラメタに OIID を設定しない edmSQL 文を実行する場合は, このメソッドをコールする必要はありません。

設定した ? パラメタの値 (OIID) は, 最後に設定した値が有効になります。設定し直していない ? パラメタの値は有効なままです。ただし, CdbxEqlStatement::Set メソッドがコールされた場合には, すでに設定されている ? パラメタのすべての値が無効になります。つまり, CdbxEqlStatement::Set メソッドで設定した edmSQL 文の, すべての ? パラメタについて値を設定し直す必要があります。

なお, ? パラメタに DMA オブジェクトを設定する場合には CdbxEqlStatement::SetObjParam メソッドを使用し, OIID でも DMA オブジェクトでもない値を設定する場合には CdbxEqlStatement::SetParam メソッドを使用してください。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールした場合は, エラーになります。

戻り値が DMA_FALSE の場合, 設定された ? パラメタの値 (OIID) は保証できません。

引数

lParamIndex (入力)

edmSQL 文中の OIID を設定する ? パラメタの出現順序を指定します。

1 ~ (edmSQL 文中の ? パラメタの数) の範囲で指定してください。0 以下の値を指定した場合には, エラーになります。0 より大きい値で範囲外の値を指定した場合は, CdbxEqlStatement::Execute メソッドのコール時にエラーになります。

また, 同一の ? パラメタに対して, 複数回, 値を設定した場合には, 最後に設定した値が有効になります。

pOIID (入力)

edmSQL 文中の ? パラメタに設定する値を指定します。OIID を指定してください。

指定した値が OIID でない場合および NULL の場合にはエラーになります。OIID かどうかは, 次に示す観点からチェックされます。

- 指定された文字列が「dma:///」で始まっているかどうか
- 指定された文字列長が正しい (133 バイト) かどうか

指定した値が, 構文不正な値となる場合は, CdbxEqlStatement::Execute メソッドのコール時にエラーに

なります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxEqlStatement::SetParam メソッド

機能

edmSQL 文の ? パラメタの設定

形式

```
// Boolean型の値を設定する場合
DmaBoolean SetParam (
    DmaInteger32    lParamIndex,
    DmaBoolean      bParam)

// Integer32型の値を設定する場合
DmaBoolean SetParam (
    DmaInteger32    lParamIndex,
    DmaInteger32    lParam)

// String型の値を設定する場合
DmaBoolean SetParam (
    DmaInteger32    lParamIndex,
    pDmaString_T    pParam)

// BinaryValue型の値を設定する場合
DmaBoolean SetParam (
    DmaInteger32    lParamIndex,
    DmaBinaryValue* pbvParamParam)

struct DmaBinaryValue{
    DmaUInteger32    cbBytes;
    pDmaBinary      pbBytes;
};
```

詳細

edmSQL 文中の ? パラメタの値を設定するメソッドです。

このメソッドで設定できる値を次に示します。

- Boolean 型の定数
- Integer 型の定数
- String 型の定数
- BinaryValue 型の定数

このメソッドで設定した ? パラメタの値は , CdbxEqlStatement::Execute メソッドのコール時に有効になります。 ? パラメタが存在しない edmSQL 文を実行する場合は , このメソッドをコールする必要はありません。

? パラメタの値は , 最後に設定した値が有効になります。設定し直していない ? パラメタの値は有効なままです。ただし , CdbxEqlStatement::Set メソッドがコールされた場合には , すでに設定されている ? パラメタのすべての値が無効になります。つまり , CdbxEqlStatement::Set メソッドで設定した edmSQL 文の , すべての ? パラメタについて値を設定し直す必要があります。

なお , ? パラメタに OIID を設定する場合には CdbxEqlStatement::SetOIIDParam メソッドを使用し , DMA オブジェクトを設定する場合には CdbxEqlStatement::SetObjParam メソッドを使用してください。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールした場合は , エラーになります。

戻り値が DMA_FALSE の場合 , 設定された ? パラメタの値は保証できません。

edmSQL 文中の ? パラメタに BinaryValue 型の値を設定する例を次に示します。

BinaryValue 型の値を設定する例

```

///// 実行するedmSQL文をセット
brc = cEql->Set( (pDmaString_T)
  "Select score_concept(edmProp_ConceptStIndex) "
  " From usrClass_ContentSearchDV"
  " Where concept_with_score"
  " ( edmProp_ConceptStIndex, ? AS BINARY(5M)) IS TRUE" );
if ( DMA_TRUE != brc ) {
  //エラー処理
}
//?パラメタをセット( ? : {"この文字列を検索します"})
DmaBinaryValue pbvBinary;
char sCondition[1024];
strcpy(sCondition, "{¥"この文字列を検索します¥}");
pbvBinary.cbBytes =
  (DmaUInteger32)sizeof("{¥"この文字列を検索します¥}");
pbvBinary.pbBytes = (pDmaBinary)sCondition;
brc = cEql.SetParam(1, &pbvBinary);
if ( DMA_TRUE != brc ) {
  //エラー処理
}
///// edmSQL文の実行
brc = cEql->Execute();
if ( DMA_TRUE != brc ) {
  //エラー処理
}

```

引数

IParamIndex (入力)

edmSQL 文中の ? パラメタの出現順序を指定します。

1 ~ (edmSQL 文中の ? パラメタの数) の範囲で指定してください。0 以下の値を指定した場合には、エラーになります。0 より大きい値で範囲外の値を指定した場合は、CdbEqlStatement::Execute メソッドのコール時にエラーになります。

また、同一の ? パラメタに対して、複数回、値を設定した場合には、最後に設定した値が有効になります。

bParam, IParam, pParam, pbvParamParam (入力)

edmSQL 文中の ? パラメタに設定する値を指定します。

指定した値の型と要素の型が一致しないなどで構文不正となる場合には、CdbEqlStatement::Execute メソッドのコール時にエラーになります。

なお、このメソッドでは、この引数で指定された値が妥当かどうかのチェックはしません。ただし、String 型の値を設定するときに値として NULL を指定した場合は、エラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_EQL_NOT_INITIALIZE	初期化が実行されていません。	初期化を実行してください。

9. CdbxEqlStatement クラス (edmSQL 文を操作 , 実行するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbxEqlStatement::Terminate メソッド

機能

セッションオブジェクトへの登録抹消

形式

Terminate ()

詳細

CdbxEqlStatement クラスの終了処理をするメソッドです。CdbxEqlStatement オブジェクトで管理している情報を削除し、CdbxEsession オブジェクトと CdbxEqlStatement オブジェクトとの関連を切り離します。したがって、このメソッドをコールしたあとは、CdbxEqlStatement::Initialize メソッドをコールし、CdbxEsession オブジェクトとの関連づけをしないと、CdbxEqlStatement クラスのそのほかのメソッドが使用できなくなります。

CdbxEqlStatement::Initialize メソッドがコールされていない状態でこのメソッドをコールした場合は、エラーになります。

戻り値が DMA_FALSE の場合、CdbxEsession オブジェクトへの登録抹消処理は保証できません。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

9. CdbxEqlStatement クラス (edmSQL 文を操作 , 実行するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

10 CdbriIndependentPersistence クラス (独立データを扱うクラス)

独立したデータを扱う機能を提供するクラスです。

CdbriIndependentPersistence クラスの詳細

CdbriIndependentPersistence::~~CdbriIndependentPersistence メソッド

CdbriIndependentPersistence::BindPublicACL メソッド

CdbriIndependentPersistence::CdbriIndependentPersistence メソッド

CdbriIndependentPersistence::CreateObject メソッド

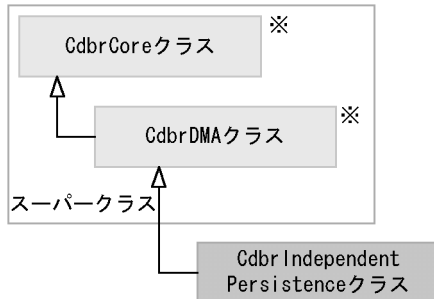
CdbriIndependentPersistence::GetPublicACLList メソッド

CdbriIndependentPersistence::GetPublicACLListAndLock メソッド

CdbriIndependentPersistence::UnbindPublicACL メソッド

CdbrIndependentPersistence クラスの詳細

スーパークラス



（凡例）

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 10-1 CdbrIndependentPersistence クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 10-2 CdbrIndependentPersistence クラスのプロパティ（アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ）

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- 2
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザおよびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbriIndependentPersistence クラスで導入されるプロパティ (アクセス制御機能を使用する場合)

- PrimaryGroupId

プライマリグループのグループ識別子を表します。

- ACL

オブジェクトのローカル ACL を表します。

- SACL

オブジェクトのセキュリティ ACL を表します。

- OwnerPermission

ACFlag のうち、所有者のパーミッションを表します。

- PrimaryGroupPermission

ACFlag のうち、プライマリグループのパーミッションを表します。

- EveryonePermission

ACFlag のうち、全ユーザのパーミッションを表します。

- PublicACLCount

オブジェクトがバインドしているパブリック ACL の数を表します。

- PublicACLIds

オブジェクトがバインドしているパブリック ACL の OIID のリストを表します。

パブリック ACL の OIID のリストは、OID を構成要素とした VariableArray 型プロパティとして作成します。まず、CdbrCompound クラスのメソッドを使用して、バインドするパブリック ACL の OIID を CdbrCompound クラスのオブジェクトの dbrProp_ACLIdElem プロパティに設定します。

これを、CdbrVariableArray クラスのメソッドを使用して VariableArray 型プロパティの構成要素とします。このプロパティの値を SDBR_PROP 構造体に設定してから、バインドするオブジェクトの CreateObject メソッドや PutPropertyValues メソッドの引数に指定してメソッドをコールします。

これによって、パブリック ACL はオブジェクトにバインドできます。

なお、すでにパブリック ACL をバインドしているオブジェクトに対してパブリック ACL を追加したい場合は、BindPublicACL メソッドを使用してください。

10. CdbrIndependentPersistence クラス (独立データを扱うクラス)

なお、CdbrIndependentPersistence クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA のクラスの edmClass_IndependentPersistence クラスおよびサブクラスのプロパティを参照、設定できます。DMA のクラスのプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrIndependentPersistence クラスで導入されるメソッド

メソッド	機能
~CdbrIndependentPersistence	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrIndependentPersistence	コンストラクタ
CreateObject	オブジェクトの新規作成
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定

メソッド	機能
UnbindPublicACL	パブリック ACL のアンバインド

以降, CdbriIndependentPersistence クラスで導入されるメソッドについて説明します。

CdbriIndependentPersistence::~~CdbriIndependentPersistence メソッド

機能

デストラクタ

形式

```
virtual ~CdbriIndependentPersistence()
```

詳細

CdbriIndependentPersistence オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbriIndependentPersistence::BindPublicACL メソッド

機能

パブリック ACL のバインド

形式

```
DmaBoolean BindPublicACL(
    DmaInteger32          lIdListCount,
    ppDmaString_T        ppIdList)
```

詳細

パブリック ACL をバインドするメソッドです。

ppIdList に複数のパブリック ACL を指定することで、一度に複数のパブリック ACL をバインドできます。

存在しない OIID を指定した場合、重複した OIID を指定した場合、すでにバインドしているパブリック ACL の OIID を指定した場合、およびバインド数の制限値を超えて指定した場合はエラーになります。

引数

lIdListCount (入力)

バインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

バインドするパブリック ACL の OIID 文字列のポインタ配列を指定します。NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。

10. CdbriIndependentPersistence クラス (独立データを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は、10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されていません。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLIC_ACL_ALREADY_BOUND	指定されたパブリック ACL はすでにバインドされています。	すでにバインドされているパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。または、メモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbriIndependentPersistence::CdbriIndependentPersistence メソッド

機能

コンストラクタ

形式

`CdbriIndependentPersistence()`

詳細

`CdbriIndependentPersistence` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrIndependentPersistence::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
CdbrIndependentPersistence::CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*  pDMAInfoList,
    ppDmaString_T     ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean;    // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;          // ID型
        pDmapv           ppObject;     // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;
    SDBR_PROP*       pItem;
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAININFO {
    DmaId      ClassId;
    SDBR_PROPLIST PropList;
};
```

詳細

オブジェクトを新規に作成するメソッドです。

SetOIID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更されずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の値を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体である、

SDBR_DMAININFO 構造体の配列を指定します。DMA オブジェクトの情報として SDBR_DMAININFO 構造体には、edmClass_IndependentPersistence クラスまたはそのサブクラスの情報を指定してください。NULL を指定した場合は、edmClass_IndependentPersistence クラスを使用して DMA オブジェクトが作成されます。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMAININFO 構造体については、「付録 C.1(3) SDBR_DMAININFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

ppOID (出力)

作成した独立データのオブジェクトの OID へのポインタが設定されます。ポインタを格納する領域を指定してください。

*ppOID がポイントするメモリを、ユーザプログラムで解放しないでください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。

10. CdbrlndependentPersistence クラス (独立データを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbrlCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1 文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1 文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1 文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECT	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbriIndependentPersistence::GetPublicACLList メソッド

機能

バインドしているパブリック ACL 一覧情報の取得

形式

```
DmaBoolean GetPublicACLList(
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*    pPropDefList,
    SDBR_OBJLIST**   ppObjList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;       //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;          //プロパティ情報の個数
    SDBR_PROP*       pItem;          //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T      pOIID;          // オブジェクトのOIID
    DmaInteger32      lType;         // オブジェクトの種別
    SDBR_PROPLIST     PropList;      // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32      lCount;
    SDBR_OBJINFO*    pItem;
};
```

詳細

オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、`pPropDefList` で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

オブジェクトがバインドしているパブリック ACL の一覧情報は `ppObjList` に設定されます。

引数

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOIID にはパブリック ACL の OIID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

10. CdbIndependentPersistence クラス (独立データを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbriIndependentPersistence::GetPublicACLListAndLock メソッド

機能

バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetPublicACLListAndLock(
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*    pPropDefList,
    SDBR_OBJLIST**   ppObjList,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {
        DmaBoolean      pbBoolean;    //Boolean型
        DmaInteger32    plInteger32;   //Integer32型
        DmaId           pId;          //ID型
        Dmapv           ppObject;     //Object型
        ppDmaString_T   ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;          //プロパティ情報の個数
    SDBR_PROP*       pItem;          //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    ppDmaString_T     pOIID;          // オブジェクトのOIID
    DmaInteger32      lType;          // オブジェクトの種別
    SDBR_PROPLIST     PropList;      // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32      lCount;
    SDBR_OBJINFO*    pItem;
};
```

詳細

ロックタイプを明示的に指定して、オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

オブジェクトがバインドしているパブリック ACL の一覧情報は ppObjList に設定されます。

引数

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになりません。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOIID にはパブリック ACL の OIID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_SESSIONS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbriIndependentPersistence::UnbindPublicACL メソッド

機能

パブリック ACL のアンバインド

形式

//指定したパブリックACLをアンバインドする場合

```
DmaBoolean UnbindPublicACL(
    DmaInteger32      lIdListCount,
    ppDmaString_T    ppIdList)
```

//すべてのパブリックACLをアンバインドする場合

```
DmaBoolean UnbindPublicACL()
```

詳細

オブジェクトからパブリック ACL をアンバインドするメソッドです。

複数のパブリック ACL を一度にアンバインドしたり、すべてのパブリック ACL を一度にアンバインドしたりできます。

指定した OIID のパブリック ACL がバインドされていない場合、および重複した OIID を指定した場合はエラーになります。

引数

lIdListCount (入力)

アンバインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

アンバインドするパブリック ACL の OIID のポインタ配列を指定します。

NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLICACL_NOT_BOUND	指定されたパブリック ACL はバインドされていません。	バインドされているパブリック ACL を指定して処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

11

CdbrPublicACL クラス（パブリック ACL を扱うクラス）

アクセス制御リストを管理して、これを複数の文書やコンテナで共有して使用する機能を提供するクラスです。

CdbrPublicACL クラスの詳細

CdbrPublicACL::~CdbrPublicACL メソッド

CdbrPublicACL::CdbrPublicACL メソッド

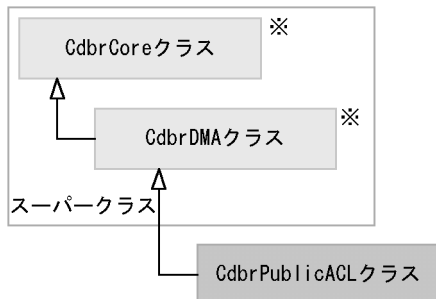
CdbrPublicACL::CreateObject メソッド

CdbrPublicACL::GetBindObjectList メソッド

CdbrPublicACL::GetBindObjectListAndLock メソッド

CdbPublicACL クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 11-1 CdbPublicACL クラスのプロパティ

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OIID	dmaProp_OIID	String 型	
OwnerId	dbrProp_OwnerId	String 型	- 2
UserPermission	dbrProp_UserPermission	Integer32 型	
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
BindObjectCount	dbrProp_BindObjectCount	Integer32 型	

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザおよびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbDMA クラスから継承されるプロパティ

- OwnerId

- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrPublicACL クラスで導入されるプロパティ

- SACL
オブジェクトのセキュリティ ACL を表します。
- ACL
オブジェクトのアクセス制御リスト (ACL) を表します。
- BindObjectCount
パブリック ACL をバインドしているクラスライブラリのオブジェクトの数を表します。

なおこのほかに、CdbrPublicACL クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA クラスの edmClass_PublicACL クラスのプロパティが使用できます。edmClass_PublicACL クラスで定義されているプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

また、CdbrPublicACL クラスのオブジェクトには、ユーザ定義プロパティも設定できます。ただし、ユーザ定義プロパティの値の変更は、そのパブリック ACL の所有者か、セキュリティ管理者だけが実行できます。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除

11. CdbrPublicACL クラス (パブリック ACL を扱うクラス)

メソッド	機能
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrPublicACL クラスで導入されるメソッド

メソッド	機能
~CdbrPublicACL	デストラクタ
CdbrPublicACL	コンストラクタ
CreateObject	パブリック ACL の新規作成
GetBindObjectList	バインドしているオブジェクトの一覧情報の取得
GetBindObjectListAndLock	バインドしているオブジェクトの一覧情報の取得と同時に指定したロックの設定

以降, CdbrPublicACL クラスで導入されるメソッドについて説明します。

CdbrPublicACL::~CdbrPublicACL メソッド

機能

デストラクタ

形式

`~CdbrPublicACL()`

詳細

CdbrPublicACL オブジェクトのデストラクタです。

インスタンスが文書空間中のパブリック ACL オブジェクトに接続中の場合、接続を解除します。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbPublicACL::CdbPublicACL メソッド

機能

コンストラクタ

形式

`CdbPublicACL()`

詳細

CdbPublicACL オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrPidPublicACL::CreateObject メソッド

機能

パブリック ACL の新規作成

形式

```
DmaBoolean CreateObject (
    CdbrPidSession*    pDbrPidSession,
    DmaInteger32       lDMAInfoListCount,
    SDBR_DMAINFO*     pDMAInfoList,
    ppDmaString_T     ppOID )

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;                // プロパティ識別子
    DmaInteger32       lType;                // データ型
    DmaInteger32       lCardinality;         // 基本単位
    DmaInteger32       lCount;              // 要素数
    union {
        pDmaBoolean    pbBoolean;          // Boolean型
        pDmaInteger32  plInteger32;        // Integer32型
        pDmaId         pId;                // ID型
        pDmapv         ppObject;          // Object型
        ppDmaString_T  ppString;          // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*        pItem;
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAINFO {
    DmaId              ClassId;
    SDBR_PROPLIST     PropList;
};
```

詳細

パブリック ACL を新規に作成するメソッドです。

SetOID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更されずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

引数

pDbrPidSession (入力)

CdbrPidSession オブジェクトを指定します。NULL を指定するとエラーになります。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の値を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体である、

SDBR_DMAINFO 構造体の配列を指定します。DMA オブジェクトの情報として SDBR_DMAINFO 構造体には, edmClass_PublicACL クラスの情報を指定してください。NULL を指定した場合, edmClass_PublicACL クラスを使用して DMA オブジェクトが作成されます。

また, IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると, エラーになります。

SDBR_DMAINFO 構造体については, 「付録 C.1(3) SDBR_DMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

また, プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppOID (出力)

作成した CdbPublicACL オブジェクトの OID へのポインタが設定されます。ポインタを格納する領域を指定してください。

*ppOID がポイントするメモリを, ユーザプログラムで解放しないでください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して, ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて, 再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して, 障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて, 再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_PERMISSION	パーミッションの指定が不正です。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbCompound オブジェクトを指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLICACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1 文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1 文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1 文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

11. CdbPublicACL クラス (パブリック ACL を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbPublicACL::GetBindObjectList メソッド

機能

バインドしているオブジェクトの一覧情報の取得

形式

```
DmaBoolean GetBindObjectList(
    pDmaBoolean      pbContinue,
    DmaInteger32     lClassType,
    DmaInteger32     lPropDefListCount,
    SDBR_PROPDEF*   pPropDefList,
    DmaInteger32     lObjListCount,
    SDBR_OBJLIST**  ppObjList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;           // データ型
    DmaInteger32 lCardinality;    // 基本単位
    DmaInteger32 lCount;          // 要素数
    union {                       // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;        // ID型
        pDmapv           ppObject;    // Object型
        ppDmaString_T    ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;           // プロパティ情報の個数
    SDBR_PROP*   pItem;           // プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T    pOIID;         // オブジェクトのOIID
    DmaInteger32    lType;         // オブジェクトの種別
    SDBR_PROPLIST  PropList;      // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32    lCount;
    SDBR_OBJINFO*  pItem;
};
```

詳細

パブリック ACL オブジェクトをバインドしているオブジェクトに関する一覧情報を取得するメソッドです。また、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

1 回のメソッドのコールで一覧の取得が完了しなかった場合、引数 pbContinue に DMA_TRUE が設定されるので、引数 pbContinue が DMA_FALSE になるまで繰り返しメソッドをコールして残りを取得して

ください。

複数回に分けて情報を取得する場合、取得される内容はすべて、1 回目のメソッドのコール時点の状態に依存します。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄し、新たに先頭から取得します。

なお、オブジェクトの作成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、DMA_FALSE が指定されたものとして動作します。

前回メソッドをコールした時と引数が異なるメソッドをコールした場合は、pbContinue の値は無視され、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- DMA_TRUE
継続可能 (まだ取得していない残りの要素があります)
- DMA_FALSE
継続不可 (すべての要素を取得しました)

IClassType (入力)

対象となるオブジェクトのクラス種別を、クラス識別子によって指定します。これによって、指定されたクラス識別子の要素だけが取得できます。

クラス識別子の一覧を次に示します。

表 11-2 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_INDEPENDENTPERSISTENCE	CdbIndependentPersistence クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの OR 条件
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件として指定した取得もできます。例えば、CdbReferentialContainer クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT
```

ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

なお、特定のクラス種別を指定すると、内部で該当するオブジェクトの選別が実行されるため処理が遅くなります。このため、クラス種別はできるだけ DBR_CLASS_ALL を指定するようにしてください。例えば、パブリック ACL をバインドしているのが CdbReferentialContainer オブジェクトだけだとわかっている場合、クラス種別として DBR_CLASS_REFERENTIALCONTAINER を指定するのではなく、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。0 を指定した場合、プロパティは取得されません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

IPropDefCount が 0 の場合、この引数は無視されます。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

情報を取得するオブジェクトの数を指定します。1 以上の値を指定してください。0 以下の値を指定するとエラーになります。

ppObjList (出力)

取得したオブジェクト情報を格納するリストである、SDBR_OBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。

*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbPublicACL::GetBindObjectListAndLock メソッド

機能

バインドしているオブジェクトの一覧情報の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetBindObjectListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32  lType;            // データ型
    DmaInteger32  lCardinality;     // 基本単位
    DmaInteger32  lCount;          // 要素数
    union {
        pDmaBoolean  pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32; // Integer32型
        pDmaId       pID;          // ID型
        pDmapv       ppObject;     // Object型
        ppDmaString_T ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;           // プロパティ情報の個数
    SDBR_PROP*   pItem;           // プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T  pOIID;           // オブジェクトのOIID
    DmaInteger32  lType;          // オブジェクトの種別
    SDBR_PROPLIST PropList;       // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32  lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

ロックタイプを明示的に指定して、パブリック ACL オブジェクトをバインドしているオブジェクトに関する一覧情報を取得するメソッドです。また、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

1 回のメソッドのコールで一覧の取得が完了しなかった場合、引数 pbContinue に DMA_TRUE が設定さ

れるので、引数 pbContinue が DMA_FALSE になるまで繰り返しメソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて、1回目のメソッドのコール時点の状態に依存します。したがって、1回目のメソッドのコール以降に変更した要素数やプロパティ値は、2回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbrContainable クラス (包含要素を扱うクラス)」の CdbrContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄し、新たに先頭から取得します。

なお、オブジェクトの作成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、DMA_FALSE が指定されたものとして動作します。

前回メソッドをコールした時と引数が異なるメソッドがコールされた場合は、pbContinue の値を無視し、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- DMA_TRUE
継続可能 (まだ取得していない残りの要素があります)
- DMA_FALSE
継続不可 (すべての要素を取得しました)

IClassType (入力)

対象となるオブジェクトを、クラス識別子によって指定します。これによって、指定されたクラス識別子の要素だけが取得できます。

クラス識別子の一覧を次に示します。

表 11-3 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbrConfiguredReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_INDEPENDENTPERSISTENCE	CdbrIndependentPersistence クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbrReferentialContainer クラス
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbrReferentialContainer クラスと CdbrVersionTraceableContainer クラスの OR 条件

クラス識別子	クラスの種別
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbReferentialContainer クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT

ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

なお、特定のクラス種別を指定すると、内部で該当するオブジェクトの選別を行なうため処理が遅くなります。このため、クラス種別はできるだけ DBR_CLASS_ALL を指定するようにしてください。例えば、パブリック ACL をバインドしているのが CdbReferentialContainer オブジェクトだけだとわかっている場合、クラス種別として DBR_CLASS_REFERENTIALCONTAINER を指定するのではなく、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。0 を指定した場合、プロパティを取得しません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

IPropDefCount が 0 の場合、この引数は無視されます。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

情報を取得するオブジェクトの数を指定します。1 以上の値を指定してください。0 以下の値を指定するとエラーになります。

ppObjList (出力)

取得したオブジェクト情報を格納するリストである、SDBR_OBJLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。

*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を指定します。

- DMA_LOCK_READ

read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

- DMA_LOCK_WRITE

排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。

11. CdbPublicACL クラス (パブリック ACL を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

12 CdbrReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

文書をコンテナによって包含して管理する機能を提供するクラスです。

CdbrReferentialContainer クラスの詳細

CdbrReferentialContainer::~CdbrReferentialContainer メソッド

CdbrReferentialContainer::CdbrReferentialContainer メソッド

CdbrReferentialContainer::CreateObject メソッド

CdbrReferentialContainer::GetContainableList メソッド

CdbrReferentialContainer::GetContainableListAndLock メソッド

CdbrReferentialContainer::GetLinkPropertyValues メソッド

CdbrReferentialContainer::GetLinkPropertyValuesAndLock メソッド

CdbrReferentialContainer::Link メソッド

CdbrReferentialContainer::LinkAndLock メソッド

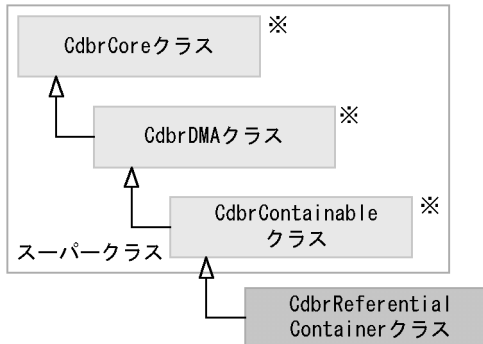
CdbrReferentialContainer::PutLinkPropertyValues メソッド

CdbrReferentialContainer::Unlink メソッド

CdbrReferentialContainer::UnlinkAndLock メソッド

CdbReferentialContainer クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を示します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 12-1 CdbReferentialContainer クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContaineesCount	dbrProp_ContaineesCount	Integer32 型	
ChildrenCount	dbrProp_ChildrenCount	Integer32 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 12-2 CdbReferentialContainer クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- 2
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermissi on	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- ³

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- PrimaryGroupId
- ACL
- SACL
- OwnerPermission
- PrimaryGroupPermission
- EveryonePermission
- PublicACLCount
- PublicACLIds

このプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbReferentialContainer クラスで導入されるプロパティ

- ContaineesCount

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

参照型のコンテインメントで包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) の個数を表します。

参照型のコンテインメントとは, DMA オブジェクトの ReferentialContainmentRelationship オブジェクトによって設定された関連づけです。

- ChildrenCount

直接型のコンテインメントで包含している Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) の個数を表します。直接型のコンテインメントとは, DMA オブジェクトの DirectContainmentRelationship オブジェクトによって設定された関連づけです。

なおこのほかに, CdbReferentialContainer クラスでは, 構成要素である DMA オブジェクトを作成する基になる DMA のクラスの dmaClass_Container クラスおよびそのサブクラスのプロパティを参照, 設定できます。DMA のクラスのプロパティについては, 「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbCore クラスから継承されるメソッド

メソッド	機能
~CdbCore	デストラクタ
CdbCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については, 「6. CdbCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbDMA クラスから継承されるメソッド

メソッド	機能
~CdbDMA	デストラクタ
CdbDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については, 「7. CdbDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrReferentialContainer クラスで導入されるメソッド

メソッド	機能
~CdbrReferentialContainer	デストラクタ
CdbrReferentialContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetContainableList	包含要素の一覧取得
GetContainableListAndLock	包含要素の一覧取得と同時に指定したロックの設定
GetLinkPropertyValues	リンクのプロパティの取得
GetLinkPropertyValuesAndLock	リンクのプロパティの取得と同時に指定したロックの設定
Link	関連づけの設定
LinkAndLock	関連づけの設定と同時に指定したロックの設定
PutLinkPropertyValues	リンクのプロパティの設定
Unlink	関連づけの解除
UnlinkAndLock	関連づけの解除と同時に指定したロックの設定

以降, CdbrReferentialContainer クラスで導入されるメソッドについて説明します。

CdbreferentialContainer::~~CdbreferentialContainer メソッド

機能

デストラクタ

形式

```
virtual ~CdbreferentialContainer()
```

詳細

CdbreferentialContainer オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbreferentialContainer::CdbreferentialContainer メソッド

機能

コンストラクタ

形式

`CdbreferentialContainer ()`

詳細

`CdbreferentialContainer` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbReferentialContainer::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
DmaBoolean CreateObject (
CdbSession*      pDbrSession,
DmaInteger32     lDMAInfoListCount,
SDBR_DMAININFO* pDMAInfoList,
ppDmaString_T   ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;      // プロパティ識別子
    DmaInteger32 lType;     // データ型
    DmaInteger32 lCardinality; // 基本単位
    DmaInteger32 lCount;    // 要素数
    union {                // プロパティの値
        pDmaBoolean      pbBoolean;    // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;          // ID型
        pDmapv           ppObject;     // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAININFO {
    DmaId      ClassId;
    SDBR_PROPLIST PropList;
};
```

詳細

オブジェクトを新規に作成するメソッドです。

SetOIID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更せずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

次に、CdbReferentialContainer オブジェクトを新規に作成して接続する例を示します。この例では、変数 ClassId_Sample で指定されたクラス識別子で CdbReferentialContainer オブジェクトを新規に作成して、そのポインタを取得します。

CdbReferentialContainer オブジェクトを新規に作成して接続する例

```
//変数宣言
SDBR_DMAININFO DMAInfoList;
pDmaString_T   pOIID;

//テンプレートの作成
DMAInfoList.ClassId = ClassId_Sample;
DMAInfoList.PropList.pItem = NULL;
DMAInfoList.PropList.lCount = 0;
```



```
//オブジェクトの新規作成
DbrRfct.CreateObject (pDbrSession, 1, &DMAInfoList ,
    &pOIID);
```

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の数を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体の配列を指定します。DMA オブジェクトの情報として SDBR_DMMAINFO 構造体には、dmaClass_Container クラスまたはそのサブクラスの情報を指定してください。NULL を指定した場合は、dmaClass_Container クラスを使用して DMA オブジェクトが作成されます。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMMAINFO 構造体については、「付録 C.1(3) SDBR_DMMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

ppOIID (出力)

作成された CdbrReferentialContainer オブジェクトの OIID へのポインタが設定されます。なお、*ppOIID がポイントするメモリは、ユーザプログラムでは解放しないでください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::GetContainableList メソッド

機能

包含要素の一覧取得

形式

```
//包含要素の一覧を取得する場合
DmaBoolean GetContainableList (
    pDmaBoolean      pbContinue,
    DmaInteger32     lContainmentType,
    DmaInteger32     lPropDefListCount,
    SDBR_PROPDEF*   pPropDefList,
    DmaInteger32     lObjListCount,
    SDBR_OBJLIST**  ppObjList )

//種類別の包含要素の一覧とリンク情報を取得する場合
DmaBoolean GetContainableList (
    pDmaBoolean      pbContinue,
    DmaInteger32     lContainmentType,
    DmaInteger32     lClassType,
    DmaInteger32     lPropDefListCount,
    SDBR_PROPDEF*   pPropDefList,
    DmaInteger32     lLinkPropDefListCount,
    SDBR_PROPDEF*   pLinkPropDefList,
    DmaInteger32     lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;          // データ型
    DmaInteger32 lCardinality;   // 基本単位
    DmaInteger32 lCount;         // 要素数
    union {                       // プロパティの値
        pDmaBoolean      pbBoolean;   // Boolean型
        pDmaInteger32    plInteger32;  // Integer32型
        pDmaId           pId;         // ID型
        pDmapv           ppObject;    // Object型
        ppDmaString_T    ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T  pOIID;         // オブジェクトのOIID
    DmaInteger32 lType;         // オブジェクトの種別
    SDBR_PROPLIST PropList;     // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32 lCount;
};
```

```

    SDBR_OBJINFO*    pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO {
    pDmaString_T    pOIID;           // オブジェクトのOIID
    DmaInteger32    lType;           // オブジェクトの種別
    pDmaString_T    pLinkId;        // リンク識別子
    SDBR_PROPLIST   PropList;        // オブジェクトのプロパティリスト
    SDBR_PROPLIST   LinkPropList;    // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST {
    DmaInteger32    lCount; // リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

```

詳細

このオブジェクトが包含している、CdbrContainable クラスのサブクラスのオブジェクトの一覧を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList で指定したプロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にコールした時点での状態が反映されます。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対してこのメソッドをコールしても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbrContainable クラス (包含要素を扱うクラス)」の CdbrContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した値の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合、または前回の読み出しですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されます。

また、前回と異なる引数を指定してメソッドをコールした場合は、前回取得した値の残りを破棄して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 12-3 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IContainmentType	×	
IPropDefListCount	×	
pPropDefList	×	
IObjListCount		
ppVTObjList		

(凡例)

: 有効

× : 無効

注 前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

表 12-4 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IContainmentType	×	
IClassType	×	
IPropDefListCount	×	
pPropDefList	×	
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppLinkedObjList		

(凡例)

: 有効

× : 無効

注 前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

IContainmentType (入力)

対象となるコンテインメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

*pbContinue に DMA_TRUE を指定して継続して取得する場合は、1 回目の取得の開始時に指定された種類のコンテインメントが対象となり、この値は無視されます。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクト情報リスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリの解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType(入力)

指定されたクラス識別子の要素だけを取得します。指定できる識別子を次に示します。

表 12-5 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbrConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbrReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbrReferentialContainer クラスと CdbrVersionTraceableContainer クラスの OR 条件
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbrReferentialContainer ク

クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT

ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の数を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティとして指定できる識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkedObjList (出力)

取得した情報を格納するリンク付きオブジェクトリスト構造体である、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。**ppLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録 C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::GetContainableListAndLock メソッド

機能

包含要素の一覧取得と同時に指定したロックの設定

形式

// 包含要素の一覧を取得する場合

```
DmaBoolean GetContainableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )
```

// 種類別の包含要素の一覧とリンク情報を取得する場合

```
DmaBoolean GetContainableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lContainmentType,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_LINKEDOBJLIST** ppLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;          // 要素数
    union {
        pDmaBoolean pbBoolean;      // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId       pId;           // ID型
        pDmapv        ppObject;     // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};
```

// オブジェクト情報構造体

```
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;           // オブジェクトのOIID
    DmaInteger32   lType;           // オブジェクトの種類
    SDBR_PROPLIST  PropList;       // プロパティリスト構造体
};
```

```

};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32    lCount;
    SDBR_OBJINFO*  pItem;
};

// リンク付きオブジェクト情報構造体
struct SDBR_LINKEDOBJINFO {
    pDmaString_T    pOIID;           // オブジェクトのOIID
    DmaInteger32    lType;           // オブジェクトの種別
    pDmaString_T    pLinkId;        // リンク識別子
    SDBR_PROPLIST   PropList;        // オブジェクトのプロパティリスト
    SDBR_PROPLIST   LinkPropList;    // リンクのプロパティリスト
};

// リンク付きオブジェクトリスト構造体
struct SDBR_LINKEDOBJLIST {
    DmaInteger32    lCount; // リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

```

詳細

ロックタイプを明示的に指定して、このオブジェクトが包含している CdbContainable クラスのサブクラスのオブジェクトの一覧を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティの値も取得します。また、pLinkPropDefList で指定したプロパティを持つリンクについては、そのプロパティの値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて初めにコールした時点での状態が反映されます。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

要素の個数に変更がなく、同じオブジェクトに対してこのメソッドをコールしても、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した値の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合、または前回の読み出しですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されます。

また、前回と異なる引数を指定してメソッドをコールした場合は、前回取得した値の残りを破棄して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 12-6 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IContainmentType	×	
IPropDefListCount	×	
pPropDefList	×	
IObjListCount		
ppVTObjList		
ILockType	×	

(凡例)
 : 有効
 × : 無効

表 12-7 ほかの引数の扱われ方 (SDBR_LINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IContainmentType	×	
IClassType	×	
IPropDefListCount	×	
pPropDefList	×	
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppLinkedObjList		
ILockType	×	

(凡例)
 : 有効
 × : 無効

IContainmentType (入力)

対象となるコンテインメントの種類を指定します。次のどちらかの値を指定します。

- DBR_CONTAINMENT_DIRECT :
 直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
 参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

*pbContinue に DMA_TRUE を指定して継続して取得する場合は、1 回目の取得の開始時に指定された種類のコンテインメントが対象となり、この値は無視されます。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合はプロパティを取得しません。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクト情報リスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリの解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType(入力)

指定されたクラス識別子の要素だけを取得します。指定できる識別子を次に示します。

表 12-8 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbrConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbrReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbrReferentialContainer クラスと CdbrVersionTraceableContainer クラスの OR 条件
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件を指定した取得もできます。例えば、CdbrReferentialContainer ク

クラスの要素と CdbDocument クラスの要素を取得したい場合は、次のように指定します。

DBR_CLASS_REFERENTIALCONTAINER | DBR_CLASS_DOCUMENT

ただし、CdbReferentialContainer クラスと CdbVersionTraceableContainer クラスの要素を取得したい場合は、DBR_CLASS_RFCT_OR_VTCT を指定してください。DBR_CLASS_RFCT_OR_VTCT とほかのクラスのクラス識別子を OR で指定することもできます。

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合、プロパティを取得しません。負の数を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。リンクのプロパティとして指定できる識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合も、エラーになります。

SDBR_PROPDEF 構造体については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkedObjList (出力)

取得した情報を格納するリンク付きオブジェクトリスト構造体である、SDBR_LINKEDOBJLIST 構造体へのポインタが設定されます。**ppLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_LINKEDOBJLIST 構造体および SDBR_LINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)」および「付録 C.1(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)」を参照してください。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE

排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::GetLinkPropertyValues メソッド

機能

リンクのプロパティの取得

形式

```
DmaBoolean CdbReferentialContainer::GetLinkPropertyValues (
    pDmaString_T      pLinkId,
    DmaInteger32      lLinkPropDefListCount,
    SDBR_PROPDEF*     pLinkPropDefList,
    SDBR_PROPLIST**   ppLinkPropList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;          // データ型
    DmaInteger32 lCardinality;    // 基本単位
    DmaInteger32 lCount;         // 要素数
    union {
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32     plInteger32; // Integer32型
        pDmaId            pId;        // ID型
        pDmapv            ppObject;    // Object型
        ppDmaString_T     ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32      lCount;
    SDBR_PROP*        pItem;
};
```

詳細

リンク識別子を指定して、リンクのプロパティを取得します。

引数

pLinkId (入力)

リンク識別子を指定します。

lLinkPropDefListCount (入力)

取得する pLinkPropDefList の要素数を指定します。なお、負の数を指定した場合は、エラーになります。

pLinkPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

指定できるリンクのプロパティの識別子はユーザが定義したプロパティ識別子だけです。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタを格納する領域を指定します。

*ppLinkPropList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INV ALID	クラスライブラリのオブジェク トは無効です。	有効なクラスライブラリのオブ ジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除され ています。	正しいオブジェクトを指定して ください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールし てから、このメソッドをコール してください。
ERR_DBR	ERR_SESSION_NOT_CON NECT	セッションが接続されていませ ん。	セッションを接続してくださ い。
ERR_DMA	DMARC_BAD_PARAMETE R	不正な引数が入力されました。	正しい引数を指定してくださ い。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正で す。	正しいプロパティ識別子を指定 してください。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::GetLinkPropertyValuesAndLock メソッド

機能

リンクのプロパティの取得と同時に指定したロックの設定

形式

```
DmaBoolean CdbReferentialContainer::GetLinkPropertyValuesAndLock(
    pDmaString_T      pLinkId,
    DmaInteger32      lLinkPropDefListCount,
    SDBR_PROPDEF*     pLinkPropDefList,
    SDBR_PROPLIST**   ppLinkPropList,
    DmaInteger32 lLockType = DMA_LOCK_WRITE)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;           // データ型
    DmaInteger32 lCardinality;    // 基本単位
    DmaInteger32 lCount;          // 要素数
    union {                       // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;        // ID型
        pDmapv           ppObject;   // Object型
        ppDmaString_T    ppString;   // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*   pItem;
};
```

詳細

ロックタイプを明示的に指定し、リンク識別子を指定して、リンクのプロパティを取得するメソッドです。

引数

pLinkId (入力)

リンク識別子を指定します。

lLinkPropDefListCount (入力)

取得する pLinkPropDefList の要素数を指定します。負の数を指定した場合は、エラーになります。

pLinkPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

指定できるリンクのプロパティ識別子はユーザが定義したプロパティだけです。なお、lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定すると、エラーになります。

す。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppLinkPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタを格納する領域を指定します。

*ppLinkPropList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::Link メソッド

機能

関連づけの設定

形式

```
// オブジェクトを指定して関連づける場合
DmaBoolean Link(
    CdbContainable*      pDbrContainable,
    DmaInteger32         lContainmentType
                        = DBR_CONTAINMENT_REFERENTIAL )

// 関連づけと同時にリンクのプロパティを設定する場合
DmaBoolean Link(
    CdbContainable*      pDbrContainable,
    SDBR_PROPLIST*      pPropList,
    ppDmaString_T        ppLinkId,
    DmaInteger32         lContainmentType
                        = DBR_CONTAINMENT_REFERENTIAL )

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           // プロパティ識別子
    DmaInteger32         lType;           // データ型
    DmaInteger32         lCardinality;    // 基本単位
    DmaInteger32         lCount;         // 要素数
    union {               // プロパティの値
        pDmaBoolean      pbBoolean;     // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pID;           // ID型
        pDmapv           ppObject;      // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};
```

詳細

CdbReferentialContainer オブジェクト自身を上位のオブジェクトとして、CdbContainable クラスのサブクラスのオブジェクトを下位のオブジェクトとして関連づけを設定するメソッドです。その際、リンクにプロパティを設定することもできます。なお、リンクとは、DMA オブジェクトの DirectContainmentRelationship オブジェクトおよび ReferentialContainmentRelationship オブジェクトです。

引数

pDbrContainable (入力)

関連づける Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) を指定します。

lContainmentType (入力)

関連づけで設定するコンテインメントの種類を指定します。次のどちらかを指定してください。

- **DBR_CONTAINMENT_DIRECT :**
直接型のコンテインメントを設定します。DMA オブジェクトの `DirectContainmentRelationship` オブジェクトを使用した関連づけです。
なお、上位のコンテナを下位以下のコンテナの下位のオブジェクトとして関連づけることはできません。
- **DBR_CONTAINMENT_REFERENTIAL :**
参照型のコンテインメントを設定します。DMA オブジェクトの `ReferentialContainmentRelationship` オブジェクトを使用した関連づけです。

pPropList (入力)

リンクに設定するプロパティのリストを指定します。なお、リンクに値を設定できるプロパティは、ユーザが定義したプロパティだけです。プロパティ識別子には、ユーザが定義したプロパティの識別子を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に指定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、`CdbReferentialContainer::GetLinkPropertyValues` メソッド、`CdbReferentialContainer::GetLinkPropertyValuesAndLock` メソッドおよび `CdbReferentialContainer::PutLinkPropertyValues` メソッドなどによって、リンクのプロパティを参照、設定するときに使用します。また、`CdbReferentialContainer::Unlink` メソッドによってリンクによる関連づけを解除する場合も使用できます。

*ppLinkId がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで `dbrDelete` 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::LinkAndLock メソッド

機能

関連づけの設定と同時に指定したロックの設定

形式

```
// オブジェクトを指定して関連づける場合
DmaBoolean LinkAndLock(
    CdbContainable*      pDbrContainable,
    DmaInteger32         lContainmentType
                        = DBR_CONTAINMENT_REFERENTIAL,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// 関連づけと同時にリンクのプロパティを設定する場合
DmaBoolean LinkAndLock(
    CdbContainable*      pDbrContainable,
    SDBR_PROPLIST*      pPropList,
    ppDmaString_T       ppLinkId,
    DmaInteger32         lContainmentType
                        = DBR_CONTAINMENT_REFERENTIAL,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           // プロパティ識別子
    DmaInteger32         lType;           // データ型
    DmaInteger32         lCardinality;    // 基本単位
    DmaInteger32         lCount;         // 要素数
    union {              // プロパティの値
        pDmaBoolean      pbBoolean;     // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;           // ID型
        pDmapv           ppObject;      // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbReferentialContainer オブジェクト自身を上位のオブジェクトとして、CdbContainable クラスのサブクラスのオブジェクトを下位のオブジェクトとして関連づけを設定するメソッドです。

また、その際、リンクにプロパティを設定することもできます。なお、リンクとは、DMA オブジェクトの DirectContainmentRelationship オブジェクトおよび ReferentialContainmentRelationship オブジェクトです。

引数

pDbrContainable (入力)

関連づける Containable オブジェクト (CdbContainable クラスのサブクラスのオブジェクト) を指定します。

IContainmentType (入力)

関連づけで設定するコンテインメントの種類を指定します。次のどちらかを指定してください。

- DBR_CONTAINMENT_DIRECT :
直接型のコンテインメントを設定します。DMA オブジェクトの DirectContainmentRelationship オブジェクトを使用した関連づけです。
- DBR_CONTAINMENT_REFERENTIAL :
参照型のコンテインメントを設定します。DMA オブジェクトの ReferentialContainmentRelationship オブジェクトを使用した関連づけです。

なお、上位のコンテナを下位以下のコンテナの下位のオブジェクトとして関連づけることはできません。

pPropList (入力)

リンクに設定するプロパティのリストを指定します。なお、リンクに値を設定できるプロパティは、ユーザが定義したプロパティだけです。プロパティ識別子には、ユーザが定義したプロパティの識別子を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に指定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbReferentialContainer::GetLinkPropertyValues メソッド、CdbReferentialContainer::GetLinkPropertyValuesAndLock メソッドおよび CdbReferentialContainer::PutLinkPropertyValues メソッドなどによって、リンクのプロパティを参照、設定するときに使用します。また、CdbReferentialContainer::Unlink メソッドによってリンクによる関連づけを解除する場合も使用できます。

*ppLinkId がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbReferentialContainer::PutLinkPropertyValues メソッド

機能

リンクのプロパティの設定

形式

```
DmaBoolean CdbReferentialContainer::PutLinkPropertyValues (
    pDmaString_T      pLinkId,
    SDBR_PROPLIST*    pLinkPropList)

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           // プロパティ識別子
    DmaInteger32       lType;           // データ型
    DmaInteger32       lCardinality;    // 基本単位
    DmaInteger32       lCount;         // 要素数
    union {            // プロパティの値
        pDmaBoolean    pbBoolean;      // Boolean型
        pDmaInteger32  plInteger32;    // Integer32型
        pDmaId         pId;            // ID型
        pDmapv         ppObject;       // Object型
        ppDmaString_T  ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*        pItem;
};
```

詳細

リンク識別子を指定してプロパティを設定するメソッドです。

引数

pLinkId (入力)

リンク識別子を指定します。

pLinkPropList (入力)

リンクに設定するプロパティ情報をプロパティリスト構造体である、SDBR_PROPLIST 構造体として指定します。

なお、リンクに値を設定できるプロパティは、ユーザが定義したプロパティだけです。プロパティ識別子には、ユーザが定義したプロパティの識別子を指定してください。同じプロパティ識別子を指定したプロパティ構造体を複数指定した場合は、エラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。

12. CdbReferentialContainer クラス (バージョンなしコンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrReferentialContainer::Unlink メソッド

機能

関連づけの解除

形式

```
// Containableオブジェクトを指定して解除する場合
DmaBoolean Unlink(
    CdbrContainable*      pDbrContainable)

// リンク識別子を指定して解除する場合
DmaBoolean Unlink(
    pDmaString_T         pLinkId)
```

詳細

CdbrReferentialContainer オブジェクトが包含している Containable オブジェクト (CdbrContainable クラスのサブクラスのオブジェクト) を指定して、CdbrReferentialContainer オブジェクトとの関連づけを解除するメソッドです。また、リンク識別子を指定して関連づけを解除することもできます。

指定したオブジェクトと関連づけがされていない場合は、何もしないで正常終了します。

なお、包含している Containable オブジェクトやそのリンク識別子は、CdbrReferentialContainer::GetContainableList メソッドまたは CdbrReferentialContainer::GetContainableListAndLock メソッドによって取得できます。

引数

pDbrContainable (入力)

関連づけを解除するオブジェクトを指定します。CdbrContainable クラスのサブクラスのオブジェクトを指定します。

pLinkId (入力)

関連づけを解除するオブジェクトのリンク識別子を指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrReferentialContainer::UnlinkAndLock メソッド

機能

関連づけの解除と同時に指定したロックの設定

形式

```
// Containableオブジェクトを指定して解除する場合
DmaBoolean UnlinkAndLock(
    CdbrContainable*      pDbrContainable,
    DmaInteger32          lLockType = DMA_LOCK_WRITE)

// リンク識別子を指定して解除する場合
DmaBoolean UnlinkAndLock(
    DmaInteger32*         pLinkId,
    DmaInteger32          lLockType = DMA_LOCK_WRITE)
```

詳細

ロックタイプを明示的に指定し、Containable オブジェクト (CdbrContainable クラスのサブクラスのオブジェクト) を指定して、CdbrReferentialContainer オブジェクトとの関連づけを解除するメソッドです。

また、リンク識別子を指定して関連づけを解除することもできます。

指定したオブジェクトと関連づけがされていない場合は、何もしないで正常終了します。

なお、包含している Containable オブジェクトやそのリンク識別子は、CdbrReferentialContainer::GetContainableList メソッドまたは CdbrReferentialContainer::GetContainableListAndLock メソッドによって取得できます。

引数

pDbrContainable (入力)

関連づけを解除するオブジェクトを指定します。CdbrContainable クラスのサブクラスのオブジェクトを指定します。

pLinkId (入力)

関連づけを解除するオブジェクトのリンク識別子を指定します。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

13 CdbrSession クラス (セッションを確立するクラス)

クライアント実行環境と文書空間とのセッションを確立する機能を提供するクラスです。また、セッションを確立した時、与えられたユーザ名、パスワードでログイン管理します。

CdbrReferentialContainer クラス、

CdbrConfiguratedReferentialContainer クラス、

CdbrVersionableDocument クラスなどの機能を使用するときに

CdbrSession クラスのポインタを設定することによって、これらのクラスのインスタンスは接続されたセッションを使用することができます。

CdbrSession クラスの詳細

CdbrSession::~CdbrSession メソッド

CdbrSession::~Begin メソッド

CdbrSession::~CdbrSession メソッド

CdbrSession::~Commit メソッド

CdbrSession::~Connect メソッド

CdbrSession::~Disconnect メソッド

CdbrSession::~GetClassType メソッド

CdbrSession::~GetDocSpaceCharacterSet メソッド

CdbrSession::~GetLastDetailError メソッド

CdbrSession::~GetReferencePath メソッド

CdbrSession::~GetUserInfo メソッド

CdbrSession::~OrbComponentShutdown メソッド

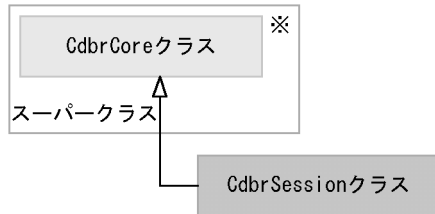
CdbrSession::~OrbComponentStart メソッド

CdbrSession::~Rollback メソッド

CdbSession::SetReferencePath メソッド

CdbSession クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

なし

メソッド一覧

CdbCore クラスから継承されるメソッド

メソッド	機能
~CdbCore	デストラクタ
CdbCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbSession クラスで導入されるメソッド

メソッド	機能
~CdbSession	デストラクタ
Begin	トランザクションの開始
CdbSession	コンストラクタ
Commit	永続オブジェクトの確定
Connect	DocSpace との接続
Disconnect	セッションの切断
GetClassType	指定した OIID のクラスの判定
GetDocSpaceCharacterSet	接続する文書空間の文字コード種別の取得
GetLastDetailError	詳細エラー情報の取得
GetReferencePath	コンテンツ格納先ベースパスの取得
GetUserInfo	ユーザ情報の取得
OrbComponentShutdown	TPBroker の ORB コンポーネントの終了

13. CdbSession クラス (セッションを確立するクラス)

メソッド	機能
OrbComponentStart	TPBroker の ORB コンポーネントの開始
Rollback	最後に CdbSession::Commit メソッドがコールされた状態まで戻す
SetReferencePath	コンテンツ格納先ベースパスの設定

以降, CdbSession クラスで導入されるメソッドについて説明します。

CdbSession::~CdbSession メソッド

機能

デストラクタ

形式

```
CdbSession::~CdbSession()
```

詳細

CdbSession オブジェクトのデストラクタです。

セッションが接続されている場合は、CdbSession::Disconnect メソッドで切断してから、オブジェクトを破棄してください。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbSession::Begin メソッド

機能

トランザクションの開始

形式

```
DmaBoolean CdbSession::Begin()
```

詳細

トランザクションを開始するメソッドです。CdbSession::Connect メソッドで文書空間と接続したあとでこのメソッドをコールします。永続オブジェクトへのアクセス, 作成, 変更, 削除, および問い合わせは, トランザクション中に実行する必要があります。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して, ユーザプログラムを終了してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して, 障害の原因を取り除いてください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては, 一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ILLEGAL_OPERATION	DMA 規約で禁止されている操作です。	この操作は実行できません。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて, 処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrSession::CdbrSession メソッド

機能

コンストラクタ

形式

```
CdbrSession::CdbrSession()
```

詳細

CdbrSession オブジェクトのコンストラクタです。

このオブジェクトを作成しただけでは、セッションは接続されません。セッションを接続するためには、CdbrSession::Connect メソッドをコールする必要があります。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbSession::Commit メソッド

機能

永続オブジェクトの確定

形式

```
DmaBoolean CdbSession::Commit()
```

詳細

更新したオブジェクトを確定するメソッドです。

CdbSession::Rollback メソッドをコールして回復する場合は、最後に CdbSession::Commit メソッドをコールした時点まで回復されます。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されています。	セッションを接続してください。
ERR_DMA	DMARC_ILLEGAL_OPERATION	DMA 規約で禁止されている操作です。	この操作は実行できません。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::Connect メソッド

機能

DocSpace との接続

形式

```
// 接続先の実行環境識別子を指定しない場合
DmaBoolean CdbSession::Connect (
    pDmaId          pDocSpaceID,
    pDmaString_T   username,
    pDmaString_T   password
)

// 接続先の実行環境識別子を指定する場合
DmaBoolean CdbSession::Connect (
    pDmaId          pDocSpaceID,
    pDmaString_T   username,
    pDmaString_T   password,
    DmaInteger32   lServerEnvId
)
```

詳細

pDocSpaceId に指定した文書空間に接続するメソッドです。一つの文書空間に複数の実行環境を構築している場合は、接続先の実行環境を選択できます。

タイムアウトによってセッションが切断されたあとで CdbSession::Connect メソッドを実行すると、major_code が ERR_DMA, minor_code が DMARC_NETWORK_UNAVAILABLE のエラーになります。また、CdbSession::Connect メソッド以外のメソッドをコールすると、major_code が ERR_DBR, minor_code が ERR_SESSION_NOT_CONNECT のエラーになります。これらのエラーになったあと、切断された CdbSession オブジェクトを再接続する場合は、CdbSession::Disconnect メソッドを実行してから CdbSession::Connect メソッドを実行してください。

引数

pDocSpaceId (入力)

文書空間識別子を指定します。

pUserName (入力)

ユーザ名を指定します。

pPassword (入力)

パスワードを指定します。

lServerEnvId (入力)

接続先の実行環境識別子を指定します。

一つの文書空間に複数の実行環境を作成して同時に稼働させている場合に、クライアントでどの実行環境下のサーバに接続するかを選択します。

指定できる値は、0 ~ 254 です。

指定できる値以外を指定した場合は、接続先の実行環境識別子を指定しない場合の形式と同じ動作になります。

なお、接続先の選択方法には、接続先の実行環境識別子を環境変数 `DBR_CONNECTSV_ENVID` に指定する方法もあります。環境変数の設定については、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。

クライアントで接続先を選択する場合は、接続先の `DocumentSpace` 構成定義ファイルの `SelectServerInMultiServer` エントリに `Yes` を指定して起動しておく必要があります。指定していない場合、`major_code` が `ERR_DMA`、`minor_code` が `DMARC_NETWORK_UNAVAILABLE` のエラーになります。

戻り値

`DMA_TRUE` 正常終了

`DMA_FALSE` 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
<code>ERR_CORBA</code>	<code>CORBA</code> の戻り値が設定されます。	<code>CORBA</code> でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
<code>ERR_DBR</code>	<code>ERR_NO_SERVICE</code>	使用できるサービスプロセスがありません。	時間を置いて再度実行してください。
<code>ERR_DBR</code>	<code>ERR_NOTSUPPORT_CLIENT_VERSION</code>	互換性のないサーバに接続しようとしてしました。	サーバのバージョンに対応したクライアントライブラリを利用してください。
<code>ERR_DBR</code>	<code>ERR_SESSION_ALREADY_CONNECTED</code>	セッションがすでに接続されています。	処理を継続してください。または、いったん切断してから、再度このメソッドをコールしてください。
<code>ERR_DBR</code>	<code>ERR_SESSION_OVERFLOW</code>	制限を超えて Connect しようとしてしました。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、 <code>DocumentSpace</code> 構成定義ファイルの <code>SessionMax</code> エントリを見直してください。
<code>ERR_DMA</code>	<code>DMARC_BAD_DOC_SPACE</code>	指定された識別子の文書空間は存在しません。	正しい文書空間を指定してください。
<code>ERR_DMA</code>	<code>DMARC_FAILED</code>	<code>DocumentSpace</code> 構成定義ファイルの <code>ProcessOrbBoaOption</code> エントリに、 <code>-OathreadMax</code> オプションを 129 以上の値で指定している環境で、同時に 129 以上のクライアントから <code>CdbSession::Connect</code> メソッドが発行されました。	時間を置いて再度実行するか、 <code>-OathreadMax</code> オプションの値を 128 以下にしてください。
<code>ERR_DMA</code>	<code>DMARC_NETWORK_ERROR</code>	ネットワーク環境でエラーが発生しました。	ネットワーク環境を確認してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NETWORK_UNAVAILABLE	この操作を実行するために必要なネットワークが利用できません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 クライアント側で接続する Document Broker サーバを選択している場合は、接続先の Document Broker サーバの docspace.ini の [Entry0001] セクションに「SelectServerInMultiServer=Yes」の定義があるか確認してください。クライアント側で接続する DocumentBroker サーバを選択していない場合は、「SelectServerInMultiServer=No」と定義しているか確認してください。 DocumentBroker サーバが起動しているか確認してください。 EDMRefresher コマンドが実行中の場合は、コマンド終了後、再度実行してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_AUTHENTICATED	ユーザは認証されていません。	正しいユーザ ID およびパスワードを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::Disconnect メソッド

機能

セッションの切断

形式

```
DmaBoolean CdbSession::Disconnect()
```

詳細

セッションを切断するメソッドです。接続していないオブジェクトに対してこのメソッドをコールした場合は、エラーになります。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::GetClassType メソッド

機能

指定した OIID のクラスの判定

形式

```
DmaBoolean CdbSession::GetClassType (
    pDmaString_T    pOIID,
    pDmaInteger32  plClassType )
```

詳細

オブジェクトの OIID を指定して、クラスライブラリのクラスを判定するメソッドです。

オブジェクトを削除するメソッドと同時に実行する場合の注意

RemoveObject メソッドや DeleteVersion メソッドなどのオブジェクトを削除するメソッドと同時に実行する場合、次の点に注意してください。

- オブジェクトを削除するメソッドを実行してそのトランザクションが完了する前に、削除したオブジェクトに対しこのメソッドを実行すると、major_code として ERR_DMA, minor_code として DMARC_NOT_FOUND が返却されます。

引数

pOIID (入力)

オブジェクトの OIID を指定します。

plClassType (出力)

クラスライブラリのクラス識別子が設定されます。

設定されるクラス識別子とクラスライブラリのクラスの対応を次に示します。

表 13-1 クラス識別子とクラスライブラリのクラスの対応

クラス識別子	クラスライブラリのクラス
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbConfiguredReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_INDEPENDENTPERSISTENCE	CdbIndependentPersistence クラス
DBR_CLASS_PUBLICACL	CdbPublicACL クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbReferentialContainer クラス
DBR_CLASS_UNKNOWN	クラスの種別が不明
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbVersionTraceableContainer クラス

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DMA	DMARC_BAD_OIID	引数で指定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::GetDocSpaceCharacterSet メソッド

機能

接続する文書空間の文字コード種別の取得

形式

```
DmaBoolean GetDocSpaceCharacterSet (
    pDmaInteger32  plDocSpaceCharacterSet )
```

詳細

セッションが接続している文書空間で使用している文字コード種別を取得するメソッドです。

一つのユーザプログラムで文字コード種別の異なる複数の DocumentBroker サーバに接続する場合に、このメソッドをコールすることで動的に接続先の DocumentBroker サーバの文字コード種別に合わせた処理の分岐が可能となります。

引数

plDocSpaceCharacterSet (出力)

セッションが接続している文書空間の文字コード種別を格納する領域を指定します。NULL を指定すると DMARC_BAD_PARAMETER のエラーになります。

正常終了すると次のいずれかの値になります。

- 接続先文書空間の文字コード種別が SJIS の場合
DBR_DOCSPACE_CHARACTERSET_SJIS
- 接続先文書空間の文字コード種別が UTF-8 の場合
DBR_DOCSPACE_CHARACTERSET_UTF8

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::GetLastDetailError メソッド

機能

詳細エラー情報の取得

形式

```
void GetLastDetailError(  
    ppDmaString_T    ppDetailErrorMessage)
```

詳細

エラーが発生した場合に、詳細なエラーメッセージを取得するメソッドです。

エラーになったメソッドの情報と、CdbCore::GetLastError メソッドで得た major_code および minor_code の情報に加えて、このメソッドをコールしてさらに詳しい情報を得ることで、アプリケーション開発でのデバッグ時や実行時のエラーに適切に対処できます。

このメソッドによって取得できる詳細なエラーメッセージについては、マニュアル「DocumentBroker Version 3 メッセージ」を参照してください。

このメソッドは CdbCore::GetLastError メソッドをコールしたあとで、必要に応じてコールしてください。エラーが発生していない場合にはコールしないでください。

なお、このメソッドによって取得できるメッセージは、環境変数 DBR_DETAIL_ERRORLOG を ON にしている場合に、環境変数 DBR_DETAIL_ERRORLOG_DIR で指定したパスのファイルに出力されるログメッセージと同じものです。

引数

ppDetailErrorMessage (出力)

詳細エラーメッセージが設定されます。該当するメッセージが存在しない場合は、NULL が設定されません。

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。

なお、ppDetailErrorMessage の領域は、dbrDelete 関数を使ってユーザプログラムで解放してください。

戻り値

なし

戻り値の詳細

なし

CdbSession::GetReferencePath メソッド

機能

コンテンツ格納先ベースパスの取得

形式

```
DmaBoolean GetReferencePath(
    ppDmaString_T      ppTargetContentPath)
```

詳細

リファレンスファイル管理機能を使用して、コンテンツを任意の領域に格納する場合、CdbSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを取得します。

この文字列領域は、オブジェクト内に確保されて、オブジェクトを削除するときに同時に削除されます。したがって、メソッドをコールしたユーザが削除する必要はありません。オブジェクトの削除後も、この値を使用したい場合は、内容をほかの領域にコピーしてください。

引数

ppTargetContentPath (出力)

リファレンスファイル管理機能でコンテンツの格納先として設定したベースパスへのポインタを格納する領域を指定します。NULL を指定した場合、DMARC_BAD_PARAMETER のエラーになります。値が設定されていない場合は、NULL が返却されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::GetUserInfo メソッド

機能

ユーザ情報の取得

形式

```
DmaBoolean GetUserInfo(
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*    pPropDefList,
    SDBR_PROPLIST**  ppPropList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;          // データ型
    DmaInteger32 lCardinality;    // 基本単位
    DmaInteger32 lCount;         // 要素数
    union {                       // プロパティの値
        pDmaBoolean pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32; // Integer32型
        pDmaId pId;               // ID型
        pDmapv ppObject;          // Object型
        ppDmaString_T ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP* pItem;
};
```

詳細

セッションが保持するユーザに関する情報を一括取得するメソッドです。

情報は次に示すプロパティとして取得できます。pPropDefList にプロパティの識別子を指定すると、ppPropList に値が設定されます。

なお、アクセス制御機能に対応していない文書空間でこのメソッドをコールするとエラーになります。

表 13-2 取得できるプロパティ

プロパティ名	プロパティ識別子	データ型	基本単位	説明
UserId	dbrProp_UserId	String	Scalar	ユーザ識別子
GroupCount	dbrProp_GroupCount	Integer32	Scalar	グループ数
GroupList	dbrProp_GroupList	String ¹	Enumeration	グループ識別子の配列
UserPrivilege	dbrProp_UserPrivilege	Integer32	Scalar	文書空間に対するユーザの特権 ²
UserPermission	dbrProp_UserPermission	Integer32	Scalar	ユーザ権限 ³

注 1

String 型で Enumeration 型のデータは、SDBR_PROP 構造体の lCount に要素数、共用体 uniValue の ppString に文字列ポインタ配列が設定されます。

注 2

ユーザの特権は、次に示す定数で返却されます。

表 13-3 ユーザ情報に設定された特権を表す定数

定数	意味
DBR_PRIV_SECURITY_ADMINISTRATOR	セキュリティ管理者である
DBR_PRIV_NONE	セキュリティ管理者ではない

注 3

ユーザ権限を表すパーミッションは次に示す定数の論理和で返却されます。

表 13-4 ユーザ権限を表すパーミッション定数

定数	意味
DBR_PERM_CREATE	オブジェクト作成権
DBR_PERM_PRIM_READ_PROPS	基本プロパティ参照権
DBR_PERM_PRIM_WRITE_PROPS	基本プロパティ更新権
DBR_PERM_PRIM_READ_CONTENTS	基本コンテンツ参照権
DBR_PERM_PRIM_WRITE_CONTENTS	基本コンテンツ更新権
DBR_PERM_PRIM_LINK	基本リンク権
DBR_PERM_PRIM_VERSION	基本バージョン管理権
DBR_PERM_PRIM_DELETE	基本オブジェクト削除権

引数

IPropDefListCount (入力)

pPropDefList に指定する配列の要素数を指定します。0 以下を指定するとエラーになります。

pPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppPropList (出力)

取得したプロパティ値を格納した構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。NULL を指定した場合は、エラーになります。

*ppPropList がポイントする領域は、ライブラリが新たに確保したものです。不要になった場合、この領域はメソッドをコールした側が dbrDelete 関数によって解放してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プ

ロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbSession::OrbComponentShutdown メソッド

機能

TPBroker の ORB コンポーネントの終了

形式

```
DmaBoolean CdbSession::OrbComponentShutdown()
```

詳細

TPBroker の ORB コンポーネントを終了するメソッドです。

このメソッドは、DocumentBroker のクラスライブラリを使用した Web アプリケーション (J2EE アプリケーションの場合) を終了するときに必ずコールする必要があります。

また、同様に Web アプリケーション (J2EE アプリケーションの場合) を開始するときに CdbSession::OrbComponentStart メソッドをコールする必要があります。

メソッド実行時の注意事項

- このメソッドと CdbSession::OrbComponentStart メソッドは、必ず対でコールしてください。対でコールしないと、Web アプリケーションの停止時に J2EE サーバが停止しない場合があります。
- このメソッドのコール回数が、CdbSession::OrbComponentStart メソッドのコール回数より多い場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_FAILED が返却されます。
- このメソッドは、マルチスレッド対応のクライアントアプリケーションを作成するときだけ使用できます。シングルスレッド対応のクライアントアプリケーションからコールされた場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_NOT_SUPPORTED が返却されます。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	シングルスレッド対応のクライアントアプリケーションから発行されました。	マルチスレッド対応のクライアントアプリケーションから発行してください。

13. CdbSession クラス (セッションを確立するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NETWORK_UNAVAILABLE	この操作を実行するために必要なネットワークが利用できません。	ネットワーク環境を確認してください。
ERR_DMA	DMARC_FAILED	メソッド発行順序が不正です。	メソッド発行順序を見直してください。

CdbSession::OrbComponentStart メソッド

機能

TPBroker の ORB コンポーネントの開始

形式

```
DmaBoolean CdbSession::OrbComponentStart ()
```

詳細

TPBroker の ORB コンポーネントを開始するメソッドです。

このメソッドは、DocumentBroker のクラスライブラリを使用した Web アプリケーション (J2EE アプリケーションの場合) を開始するときに必ずコールする必要があります。

また、同様に Web アプリケーション (J2EE アプリケーションの場合) を終了するときに CdbSession::OrbComponentShutdown メソッドをコールする必要があります。

メソッド実行時の注意事項

- このメソッドと CdbSession::OrbComponentShutdown メソッドは、必ず対でコールしてください。対でコールしないと、Web アプリケーションの停止時に J2EE サーバが停止しない場合があります。
- このメソッドのコール回数が、CdbSession::OrbComponentShutdown メソッドのコール回数より少ない場合は、CdbSession::OrbComponentShutdown メソッドコール時にエラーになります。このとき、major_code として ERR_DMA, minor_code として DMARC_FAILED が返却されます。
- このメソッドは、CdbSession::OrbComponentShutdown メソッドをコールしたあとは、実行できません。実行した場合、major_code として ERR_DMA, minor_code として DMARC_FAILED が返却されます。
- このメソッドは、マルチスレッド対応のクライアントアプリケーションを作成するときだけ使用できます。シングルスレッド対応のクライアントアプリケーションからコールされた場合はエラーになります。このとき、major_code として ERR_DMA, minor_code として DMARC_NOT_SUPPORTED が返却されます。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

13. CdbSession クラス (セッションを確立するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_SUPPORTED	シングルスレッド対応のクライアントアプリケーションから発行されました。	マルチスレッド対応のクライアントアプリケーションから発行してください。
ERR_DMA	DMARC_NETWORK_UNAVAILABLE	この操作を実行するために必要なネットワークが利用できません。	ネットワーク環境を確認してください。
ERR_DMA	DMARC_FAILED	メソッド発行順序が不正です。	メソッド発行順序を見直してください。

CdbSession::Rollback メソッド

機能

最後に CdbSession::Commit メソッドがコールされた状態まで戻す

形式

```
DmaBoolean CdbSession::Rollback()
```

詳細

最後に CdbSession::Commit メソッドをコールした時点以降の更新を、無効にするメソッドです。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CON NECT	セッションが接続されていませ ん。	セッションを接続してくださ い。
ERR_DMA	DMARC_CONSTRAINT_VI OLATED	制約違反の操作です。	この操作は実行できません。
ERR_DMA	DMARC_ILLEGAL_OPERA TION	DMA 規約で禁止されている操 作です。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しまし た。	処理を終了してください。

CdbSession::SetReferencePath メソッド

機能

コンテンツ格納先ベースパスの設定

形式

```
DmaBoolean SetReferencePath(
    pDmaString_T      pTargetContentPath)
```

詳細

リファレンスファイル管理機能を使用して、コンテンツを任意の領域に格納する場合に、コンテンツ格納先ベースパスを指定します。

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に対して、このメソッドでコンテンツ格納先ベースパスが設定されていない場合、エラーになります。

コンテンツ格納先ベースパスと SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath で指定するコンテンツ格納先パスは、結合して絶対パスとして使用されます。このとき、DocumentBroker がパスの区切り文字として "/" (半角) を設定します。ただし、ユーザがコンテンツ格納先ベースパスの末尾または pTargetPath で指定するコンテンツ格納先パスの先頭にパスの区切り文字を指定している場合は、パスの区切り文字は設定されません。

引数

pTargetContentPath (入力)

リファレンスファイル管理機能でコンテンツの格納先となるベースパスを指定します。

この引数は、DocumentBroker サーバが Windows の場合 UNC 形式のパスで指定することもできます。ただし、設定するコンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけを指定することはできません。

次に示す場合は、DMARC_BAD_PARAMETER のエラーになります。

- コンピュータ名または IP アドレスだけを指定した場合
- AIX の DocumentBroker サーバに対して UNC 形式で指定した場合

設定した値を削除する場合は、NULL を指定します。

この引数に指定した文字列長が、Windows の場合 260 バイト以上、AIX の場合 1023 バイト以上であると DMARC_BAD_PARAMETER のエラーになります。また、指定するパスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_OUTTRANS ACTION	トランザクション内でメソッドが呼ばれました。	CdbrSession::Begin() メソッドを呼び出す前に、この操作を実行してください。

14 CdbVariableArray クラス（可変長配列を扱うクラス）

同じ型の複数の要素から構成される可変長の配列を扱う機能を提供するクラスです。
主に、VariableArray 型のプロパティの値を参照，設定するときに使います。

CdbVariableArray クラスの詳細

CdbVariableArray::~CdbVariableArray メソッド

CdbVariableArray::Add メソッド

CdbVariableArray::CdbVariableArray メソッド

CdbVariableArray::Delete メソッド

CdbVariableArray::GetCount メソッド

CdbVariableArray::GetDataType メソッド

CdbVariableArray::GetValue メソッド

CdbVariableArray::Resize メソッド

CdbVariableArray::SetValue メソッド

CdbrVariableArray クラスの詳細

スーパークラス

なし

プロパティ一覧

なし

メソッド一覧

CdbrVariableArray クラスで導入されるメソッド

メソッド	機能
~CdbrVariableArray	デストラクタ
Add	要素の追加
CdbrVariableArray	コンストラクタ
Delete	要素の削除
GetCount	要素数の取得
GetDataType	要素の型の取得
GetValue	値の取得
Resize	要素数の変更
SetValue	値の設定

以降, CdbrVariableArray クラスで導入されるメソッドについて説明します。

CdbrVariableArray::~~CdbrVariableArray メソッド

機能

デストラクタ

形式

```
~CdbrVariableArray()
```

詳細

CdbrVariableArray オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrVariableArray::Add メソッド

機能

要素の追加

形式

```
//Boolean型の場合
DmaBoolean Add(DmaBoolean      bNewValue)
//ID型の場合
DmaBoolean Add(DmaId&          rNewValue)
//Integer32型の場合
DmaBoolean Add(DmaInteger32    lNewValue)
//String型の場合
DmaBoolean Add(pDmaString_T    pNewValue)
//複合データの場合
DmaBoolean Add(CdbrCompound&   rNewValue)
```

詳細

可変長配列の配列の末尾に要素を追加して、bNewValue、lNewValue、pNewValue または rNewValue に指定した値を設定するメソッドです。

引数

bNewValue、lNewValue、pNewValue、rNewValue (入力)

追加する要素に設定する値を指定します。指定した値の型と要素の型が一致しない場合、エラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

CdbrVariableArray::CdbrVariableArray メソッド

機能

コンストラクタ

形式

```
CdbrVariableArray(
    DmaInteger32 lDataType )
```

詳細

CdbrVariableArray オブジェクトのコンストラクタです。

lDataType に指定した型の要素を持つ可変長配列を作成します。指定した型は変更できません。

作成した直後の、配列の要素数は 0 です。配列の要素数は Resize メソッドで変更してください。

要素の型が複合データの場合、一つの要素に対して複数の型のデータを格納することができます。

可変長配列クラスは、主に VariableArray 型のプロパティの参照および設定に使用します。その場合に使用できる型は、DBR_DATATYPE_COMPOUND (複合データ) だけです。クラスライブラリでは、VariableArray 型のプロパティの要素を、複合データとして参照および設定します。

引数

lDataType (入力)

要素の型を指定します。指定できる値は次のとおりです。

- DMA_DATATYPE_BOOLEAN : Boolean 型
- DMA_DATATYPE_ID : ID 型
- DMA_DATATYPE_INTEGER32 : Integer32 型
- DMA_DATATYPE_STRING : String 型
- DBR_DATATYPE_COMPOUND : 複合データ

戻り値

なし

戻り値の詳細

なし

CdbrVariableArray::Delete メソッド

機能

要素の削除

形式

```
DmaBoolean Delete(  
    DmaInteger32 lIndex)
```

詳細

lIndex に指定した位置の要素を削除するメソッドです。指定位置以降の要素は、削除後、前に詰まります。

引数

lIndex (入力)

削除する要素の位置を、0 ~ (要素数 - 1) の値で指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

CdbVariableArray::GetCount メソッド

機能

要素数の取得

形式

```
DmaInteger32 GetCount()
```

詳細

要素の個数を取得するメソッドです。要素がまったくない場合および要素の個数が算出できない場合は、戻り値に 0 が返ります。

引数

なし

戻り値

配列の要素数が返ります。

戻り値の詳細

なし

CdbrVariableArray::GetDataType メソッド

機能

要素の型の取得

形式

```
DmaInteger32 GetDataType ()
```

詳細

要素の型を取得するメソッドです。なお、CdbrVariableArray クラスで作成した可変長配列のすべての要素の型は同じです。

引数

なし

戻り値

取得した要素の型として、次のどれかが返ります。

- DMA_DATATYPE_BOOLEAN : Boolean 型
- DMA_DATATYPE_ID : ID 型
- DMA_DATATYPE_INTEGER32 : Integer32 型
- DMA_DATATYPE_STRING : String 型
- DBR_DATATYPE_COMPOUND : 複合データ
- 0 : 未設定。

戻り値の詳細

なし

CdbrVariableArray::GetValue メソッド

機能

値の取得

形式

```
// Boolean型の場合
DmaBoolean GetValue(DmaIndex32 iIndex, pDmaBoolean pbRetVal)
// ID型の場合
DmaBoolean GetValue(DmaIndex32 iIndex, pDmaId pRetVal)
// Integer32型の場合
DmaBoolean GetValue(DmaIndex32 iIndex, pDmaInteger32 plRetVal)
// String型の場合
DmaBoolean GetValue(DmaIndex32 iIndex, ppDmaString_T ppRetVal)
// 複合データの場合
DmaBoolean GetValue(DmaIndex32 iIndex, CdbrCompound* pRetVal)
```

詳細

iIndex に指定した位置の要素の値を取得するメソッドです。指定した要素が存在し、かつ値が設定されていない場合は、戻り値に DMA_UNKNOWN が返ります。

引数

iIndex (入力)

取得する要素の位置を、0 ~ (要素数 - 1) の値で指定します。範囲外の値を指定した場合はエラーになります。

pbRetVal, pRetVal, plRetVal, ppRetVal (出力)

取得する値を格納する領域のアドレスを指定します。指定した領域の型と要素の型が異なる場合、エラーになります。

String 型の場合、使用後に dbrDelete 関数によって解放してください。

戻り値

DMA_TRUE 正常終了 (設定値あり)

DMA_UNKNOWN 正常終了 (設定値なし)

DMA_FALSE 異常終了

戻り値の詳細

なし

CdbrVariableArray::Resize メソッド

機能

要素数の変更

形式

```
DmaBoolean Resize(  
    DmaInteger32    lSize)
```

詳細

可変長配列の要素数を変更するメソッドです。

lSize に指定した要素数が現在の要素数より大きい場合、差分の要素が末尾に追加されます。追加された要素の値は空 (未設定) です。

指定した要素数が現在の要素数より小さい場合、差分の要素が末尾から削除されます。

0 未満の値を指定した場合はエラーになります。

要素数が増減しても、各要素のインデクスは変わりません。

引数

lSize (入力)

変更後の要素数を指定します。0 以上の値で指定します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

CdbrVariableArray::SetValue メソッド

機能

値の設定

形式

```
//Boolean型の場合
DmaBoolean SetValue(DmaIndex32 iIndex, DmaBoolean bNewValue)
//ID型の場合
DmaBoolean SetValue(DmaIndex32 iIndex, DmaId& rNewValue)
//Integer32型の場合
DmaBoolean SetValue(DmaIndex32 iIndex, DmaInteger32 lNewValue)
//String型の場合
DmaBoolean SetValue(DmaIndex32 iIndex, pDmaString_T pNewValue)
//複合データの場合
DmaBoolean SetValue(DmaIndex32 iIndex, CdbrCompound& rNewValue)
```

詳細

iIndex に指定した位置の要素に値を設定します。

引数

iIndex (入力)

設定する要素の位置を、0 ~ (要素数 - 1) の値で指定します。範囲外の値を指定した場合、エラーになります。

bNewValue, lNewValue, pNewValue, rNewValue (入力)

設定する値を指定します。指定した値の型と要素の型が一致しない場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

15 CdbVersionable クラス (バージョン管理機能を提供するクラス)

バージョン管理機能を提供するクラスです。
バージョン管理の対象になるオブジェクトを表す抽象クラスで、このクラス単独のインスタンスは作成しません。

CdbVersionable クラスの詳細

CdbVersionable::~CdbVersionable メソッド

CdbVersionable::BindPublicACL メソッド

CdbVersionable::CdbVersionable メソッド

CdbVersionable::DeleteVersion メソッド

CdbVersionable::GetPropertyValues メソッド

CdbVersionable::GetPropertyValuesAndLock メソッド

CdbVersionable::GetPublicACLList メソッド

CdbVersionable::GetPublicACLListAndLock メソッド

CdbVersionable::GetReservationStatus メソッド

CdbVersionable::GetReservationStatusAndLock メソッド

CdbVersionable::GetVersionList メソッド

CdbVersionable::GetVersionListAndLock メソッド

CdbVersionable::PutPropertyValues メソッド

CdbVersionable::UnbindPublicACL メソッド

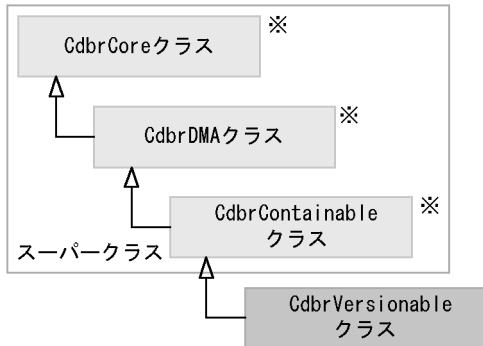
CdbVersionable::VersionCheckIn メソッド

CdbVersionable::VersionCheckOut メソッド

CdbVersionable::VersionRevoke メソッド

CdbrVersionable クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を示します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 15-1 CdbrVersionable クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContainersCountVT	dbrProp_ContainersCountVT	Integer32 型	
VersionsCount	dbrProp_VersionsCount	Integer32 型	
CurrentVersion	dbrProp_CurrentVersion	String 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 15-2 CdbrVersionable クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- 2
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用している場合)

- PrimaryGroupId
- ACL
- SACL
- OwnerPermission
- PrimaryGroupPermission
- EveryonePermission
- PublicACLCount
- PublicACLIds

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrVersionable クラスで導入されるプロパティ

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

- VersionsCount
オブジェクトが管理しているバージョンの個数を表します。CdbrVersionable::VersionCheckIn メソッドによって作成されます。まだチェックインされていないものは含みません。
- CurrentVersion
カレントバージョンのバージョン識別子を表します。
- ContainersCountVT
構成管理型の包含関係でこのオブジェクトを包含している CdbrConfiguredReferentialContainer オブジェクトおよび CdbrVersionTraceableContainer オブジェクトの個数を示します。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド

メソッド	機能
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrVersionable クラスで導入されるメソッド

メソッド	機能
~CdbrVersionable	デストラクタ
BindPublicACL	パブリック ACL のバインド (バージョン指定あり)
CdbrVersionable	コンストラクタ
DeleteVersion	バージョンの削除
GetPropertyValues	プロパティの取得 (バージョン指定あり)
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定 (バージョン指定あり)
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得 (バージョン指定あり)
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定 (バージョン指定あり)
GetReservationStatus	チェックアウト状態の取得
GetReservationStatusAndLock	チェックアウト状態の取得と同時に指定したロックの設定
GetVersionList	バージョン情報一覧の取得
GetVersionListAndLock	バージョン情報一覧の取得と同時に指定したロックの設定
PutPropertyValues	プロパティの設定 (バージョン指定あり)
UnbindPublicACL	パブリック ACL のアンバインド (バージョン指定あり)
VersionCheckIn	バージョンのチェックイン
VersionCheckOut	バージョンのチェックアウト
VersionRevoke	チェックアウトの取り消し

以降, CdbrVersionable クラスで導入されるメソッドについて説明します。

CdbrVersionable::~~CdbrVersionable メソッド

機能

デストラクタ

形式

```
virtual ~CdbrVersionable()
```

詳細

CdbrVersionable オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbVersionable::BindPublicACL メソッド

機能

パブリック ACL のバインド (バージョン指定あり)

形式

```
DmaBoolean BindPublicACL(
    pDmaString_T      pVersionId,
    DmaInteger32     lIdListCount,
    ppDmaString_T    ppIdList)
```

詳細

パブリック ACL をバインドするメソッドです。

pVersionId で指定したバージョンから ppIdList で指定したパブリック ACL をバインドします。ppIdList に複数のパブリック ACL を指定することで、一度に複数のパブリック ACL にバインドできます。

存在しない OIID を指定した場合、重複した OIID を指定した場合、すでにバインドされているパブリック ACL の OIID を指定した場合、またはバインド数の制限値を超えて指定した場合はエラーになります。

引数

pVersionId (入力)

バージョン識別子を指定します。

NULL を指定した場合、カレントバージョンが対象になります。

lIdListCount (入力)

バインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

バインドするパブリック ACL の OIID 文字列のポインタ配列を指定します。NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_PUBLICACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は、10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLICACL_ALREADY_BOUND	指定されたパブリック ACL はすでにバインドされています。	すでにバインドされているパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrVersionable::CdbrVersionable メソッド

機能

コンストラクタ

形式

`CdbrVersionable()`

詳細

`CdbrVersionable` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbVersionable::DeleteVersion メソッド

機能

バージョンの削除

形式

```
DmaBoolean DeleteVersion (
    pDmaString_T    pVersionId )
```

詳細

pVersionId に指定したバージョンを削除するメソッドです。

カレントバージョンを削除した場合は、指定したカレントバージョンを削除して、既存の確定されたバージョンのうち最も新しいバージョンをカレントバージョンとします。ただし、カレントバージョン以外に確定されたバージョンが存在しない場合、カレントバージョンの削除はできません。

このメソッドをコールすることによって、DMA オブジェクトの VersionDescription オブジェクトおよび該当するバージョンの DMA オブジェクトの dmaClass_Versionable クラスのサブクラスのオブジェクトが削除されます。

CdbReferentialContainer オブジェクトおよび CdbConfiguredReferentialContainer オブジェクトから参照されているオブジェクトのバージョンを削除する場合は、削除するバージョンを参照している DirectContainmentRelationship オブジェクト、ReferentialContainmentRelationship オブジェクトおよび VersionTraceableContainmentRelationship オブジェクトも削除されます。ただし、削除するバージョンが CdbConfiguredReferentialContainer オブジェクトから FLOATING モードで参照されている場合は、VersionTraceableContainmentRelationship オブジェクトは削除されません。VersionTraceableContainmentRelationship オブジェクトは削除後のカレントバージョンに変更されません。

なお、pVersionId に CdbVersionable::VersionCheckOut メソッドで取得した仮のバージョン識別子を指定した場合はエラーになります。チェックアウトした仮のバージョンを削除する場合は CdbVersionable::VersionRevoke メソッドを使用してください。

リファレンスファイル文書の場合

バージョンを削除したあと、カレントとなるバージョンのコンテンツが存在しているか確認します。カレントバージョンのコンテンツが存在しない場合、ERR_FILE_REFERENCE_CURRENT_CONTENT_NOTFOUND のエラーになります。また、CdbSession::SetReferencePath メソッドで、コンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

引数

pVersionId (入力)

削除するバージョンのバージョン識別子を指定します。確定されたバージョンだけ指定できます。NULL を指定した場合は、カレントバージョンが削除されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新 (カレントバージョンを削除した場合は一つ前のバージョン) から順に確認し、コンテンツが存在しないバージョンを削除してください。
ERR_DBR	ERR_FILE_REFERENCE_MISMATCH_STATUS	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生し、オブジェクトとコンテンツが不整合な状態となりました。	再度実行してバージョンを削除してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_LAST_VERSION	指定されたバージョンは唯一のバージョンであるため、削除できません。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionable::GetPropertyValues メソッド

機能

プロパティの取得 (バージョン指定あり)

形式

```
// 全バージョン共有のプロパティを対象にする場合
DmaBoolean GetPropertyValues(
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_PROPLIST**      ppPropList )

// バージョンをバージョン識別子で指定する場合
DmaBoolean GetPropertyValues(
    pDmaString_T          pVersionId,
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_PROPLIST**      ppPropList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {
        pDmaBoolean   pbBoolean;    // Boolean型
        pDmaInteger32 plInteger32;   // Integer32型
        pDmaId        pId;          // ID型
        pDmapv         ppObject;     // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};
```

詳細

オブジェクトのプロパティの値を取得するメソッドです。

バージョン識別子を指定しない場合、DMA オブジェクトの ConfigurationHistory オブジェクトに設定された、オブジェクト全体に対するプロパティが対象になります。

バージョン識別子を指定した場合は、バージョンごとの個別のプロパティが対象になります。バージョン識別子として NULL を指定した場合は、カレントバージョンが対象になります。

引数

IPropDefListCount (入力)

pPropDefList に指定した配列の要素数を指定します。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

取得するプロパティの定義情報である, SDBR_PROPDEF 構造体の配列を指定します。なお, lPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると, エラーになります。

SDBR_PROPDEF 構造体に指定する内容については, 「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppPropList (出力)

取得したプロパティリスト構造体である, SDBR_PROPLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppPropList がポイントする領域は, ライブラリが新たに確保したものです。この領域はメソッドをコールした側が dbrDelete 関数を使用して解放してください。なお, プロパティがまったく取得されなかった場合は, NULL が設定されます。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbrVersionable::GetVersionList メソッド, CdbrVersionable::GetVersionListAndLock メソッドまたは CdbrVersionable::VersionCheckOut メソッドで取得します。NULL を指定した場合は, カレントバージョンのプロパティを取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて, 再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して, 障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて, 再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては, 一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから, このメソッドをコールしてください。

15. CdbVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_VALUE_NOT_SET	要求されたプロパティは値を保持していません。	この操作は実行できません。

CdbVersionable::GetPropertyValuesAndLock メソッド

機能

プロパティの取得と同時に指定したロックの設定 (バージョン指定あり)

形式

```
// 全バージョン共有のプロパティを対象にする場合
DmaBoolean GetPropertyValuesAndLock (
DmaInteger32          lPropDefListCount,
SDBR_PROPDEF*        pPropDefList,
SDBR_PROPLIST**      ppPropList,
DmaInteger32          lLockType = DMA_LOCK_WRITE)

// バージョンをバージョン識別子で指定する場合
DmaBoolean GetPropertyValuesAndLock (
pDmaString_T          pVersionId,
DmaInteger32          lPropDefListCount,
SDBR_PROPDEF*        pPropDefList,
SDBR_PROPLIST**      ppPropList,
DmaInteger32          lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;           // 要素数
    union {
        pDmaBoolean    pbBoolean;   // Boolean型
        pDmaInteger32  plInteger32;  // Integer32型
        pDmaId         pId;          // ID型
        pDmapv         ppObject;     // Object型
        ppDmaString_T  ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};
```

詳細

ロックタイプを明示的に指定して、オブジェクトのプロパティの値を取得するメソッドです。

バージョン識別子を指定しない場合、DMA オブジェクトの ConfigurationHistory オブジェクトに設定された、オブジェクト全体に対するプロパティが対象になります。

バージョン識別子を指定した場合は、バージョンごとの個別のプロパティが対象になります。バージョン識別子として NULL を指定した場合は、カレントバージョンが対象になります。

引数

IPropDefListCount (入力)

pPropDefList に指定した配列の要素数を指定します。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppPropList (出力)

取得したプロパティリスト構造体である、SDBR_PROPLIST 構造体へのポインタが設定されます。ポインタを格納する領域を指定してください。*ppPropList がポイントする領域は、ライブラリが新たに確保したものです。この領域はメソッドをコールした側が dbrDelete 関数を使用して解放してください。なお、プロパティがまったく取得されなかった場合は、NULL が設定されます。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定される内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。NULL を指定した場合は、カレントバージョンのプロパティを取得します。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

15. CdbvVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

15. CdbVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_VALUE_NOT_SET	要求されたプロパティは値を保持していません。	この操作は実行できません。

CdbrVersionable::GetPublicACLList メソッド

機能

バインドしているパブリック ACL 一覧情報の取得 (バージョン指定あり)

形式

```
DmaBoolean GetPublicACLList(
    pDmaString_T      pVersionId,
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*     pPropDefList,
    SDBR_OBJLIST**    ppObjList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*   pItem;        //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T      pOIID;      // オブジェクトのOIID
    DmaInteger32      lType;      // オブジェクトの種別
    SDBR_PROPLIST     PropList;   // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32      lCount;
    SDBR_OBJINFO*     pItem;
};
```

詳細

オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

pVersionId で指定したバージョンがバインドしているパブリック ACL の一覧情報が ppObjList に設定されます。

引数

pVersionId (入力)

バージョン識別子を指定します。

NULL を指定した場合、カレントバージョンが対象になります。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOIID にはパブリック ACL の OIID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionable::GetPublicACLListAndLock メソッド

機能

バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定 (バージョン指定あり)

形式

```
DmaBoolean GetPublicACLListAndLock(
    pDmaString_T      pVersionId,
    DmaInteger32      lPropDefListCount,
    SDBR_PROPDEF*     pPropDefList,
    SDBR_OBJLIST**    ppObjList,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*   pItem;        //プロパティ情報
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T      pOIID;      // オブジェクトのOIID
    DmaInteger32      lType;      // オブジェクトの種別
    SDBR_PROPLIST     PropList;   // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32 lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

ロックタイプを明示的に指定して、オブジェクトがバインドしているパブリック ACL の一覧情報を取得するメソッドです。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

pVersionId で指定したバージョンがバインドしているパブリック ACL の一覧情報が ppObjList に設定されます。

引数

pVersionId (入力)

バージョン識別子を指定します。

NULL を指定した場合、カレントバージョンが対象になります。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。

プロパティを取得しない場合は 0 を指定してください。負の値を指定した場合はエラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。NULL は IPropDefListCount が 0 の場合に指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。NULL を指定した場合はエラーになります。

*ppObjList がポイントするメモリは、クラスライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体の pOID にはパブリック ACL の OID が設定されます。

また、SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_IN_TRANSACTION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrVersionable::GetReservationStatus メソッド

機能

チェックアウト状態の取得

形式

```
DmaBoolean GetReservationStatus (
    pDmaBoolean          pbReserved,
    ppDmaString_T       ppReservedVerId,
    ppDmaString_T       ppReservedUser )
```

詳細

オブジェクトがチェックアウトされているかどうかの情報を取得するメソッドです。チェックアウトされている場合はそのバージョンの DMA オブジェクトの VersionSeries オブジェクトが保持している仮のバージョン識別子と、チェックアウトしているユーザ名を取得します。

引数

pbReserved (出力)

チェックアウトされている状態であれば DMA_TRUE が設定されます。

チェックアウトされていない場合は DMA_FALSE が設定されます。

ppReservedVerId (出力)

仮のバージョン識別子へのポインタが設定されます。

値を格納した領域は、使用後に明示的に dbrDelete 関数を使用して解放してください。

ppReservedUser (出力)

チェックアウトしているユーザ名を表す文字列へのポインタが設定されます。ユーザ名は、CdbrSession::Connect メソッドの引数として指定された文字列です。チェックアウトされていない場合またはチェックアウトしたユーザ名が記録されていない場合は、NULL が設定されます。

値を格納した領域は、使用後に明示的に dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを再接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrVersionable::GetReservationStatusAndLock メソッド

機能

チェックアウト状態の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetReservationStatusAndLock (
    pDmaBoolean                pbReserved,
    ppDmaString_T              ppReservedVerId,
    ppDmaString_T              ppReservedUser,
    DmaInteger32                lLockType = DMA_LOCK_WRITE )
```

詳細

ロックタイプを明示的に指定して、オブジェクトがチェックアウトされているかどうかの情報を取得するメソッドです。チェックアウトされている場合はそのバージョンの DMA オブジェクトの VersionSeries オブジェクトが保持している仮のバージョン識別子と、チェックアウトしているユーザ名を取得します。

引数

pbReserved (出力)

チェックアウトされている状態であれば DMA_TRUE が設定されます。

チェックアウトされていない場合は DMA_FALSE が設定されます。

ppReservedVerId (出力)

仮のバージョン識別子へのポインタが設定されます。

値を格納した領域は、使用後に明示的に dbrDelete 関数を使用して解放してください。

ppReservedUser (出力)

チェックアウトしているユーザ名を表す文字列へのポインタが設定されます。ユーザ名は、CdbrSession::Connect メソッドの引数として指定された文字列です。チェックアウトされていない場合またはチェックアウトしたユーザ名が記録されていない場合は、NULL が設定されます。

値を格納した領域は、使用後に明示的に dbrDelete 関数を使用して解放してください。

lLockType (入力)

ロックの種別を設定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INV ALID	クラスライブラリのオブジェク トは無効です。	有効なクラスライブラリのオブ ジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除され ています。	正しいオブジェクトを指定して ください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールし てから、このメソッドをコール してください。
ERR_DBR	ERR_SESSION_NOT_CON NECT	セッションが接続されていませ ん。	セッションを接続してくださ い。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセ ス権がありません。	正しいアクセス権を設定してく ださい。
ERR_DMA	DMARC_BAD_PARAMETE R	不正な引数が入力されました。	正しい引数を指定してくださ い。
ERR_DMA	DMARC_LOST_CONNECTI ON	現在のオブジェクトからデータ ベースへの論理的な接続が失わ れました。操作は完了できませ ん。	文書空間に再接続してくださ い。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。ま たはメモリを大量に消費してい るアプリケーションを停止させ て、処理を再実行してください。 い。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりませ ん。	正しいオブジェクトを設定して ください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しまし た。	処理を終了してください。

15. CdbVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

CdbVersionable::GetVersionList メソッド

機能

バージョン情報一覧の取得

形式

```
DmaBoolean GetVersionList (
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lOrdered = DBR_ORDER_NONE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;           // 要素数
    union {
        pDmaBoolean    pbBoolean;    // Boolean型
        pDmaInteger32  plInteger32;   // Integer32型
        pDmaId         pId;           // ID型
        pDmapv         ppObject;      // Object型
        ppDmaString_T  ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;            // オブジェクトのOIID
    DmaInteger32   lType;            // オブジェクトの種類
    SDBR_PROPLIST  PropList;        // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32   lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

このオブジェクトが包含している、すでに確定したバージョン情報の一覧を取得するメソッドです。取得するバージョン情報を、バージョンが作成された順序によってソートして取得できます。また、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。ただし、再度取得する前に文書の削

除処理をした場合、残りが正しく取得されないことがあります。

要素の個数に変更がなく、同じオブジェクトに対してこのメソッドをコールした場合も、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力/出力)

リストの取得方法を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得したものの残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてこのメソッドをコールする場合、または前回のコールですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されず。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 15-3 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		
lOrdered	×	

(凡例)

: 有効

×: 無効

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合も、プロパティを取得しません。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパ

ティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報が設定されます。バージョン情報の一覧は、オブジェクト情報リスト構造体である、SDBR_OBJINFO 構造体の配列として格納されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリ解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体メンバの pOID には、該当するオブジェクトのバージョン識別子が設定されます。また、lType に相当する情報が設定されても、その内容は保証できません。

なお、バージョン管理されるオブジェクト (dmaClass_Versionable クラスのサブクラスのオブジェクト) が引数 pPropDefList で指定したプロパティを持つ場合に、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IOrdered (入力)

バージョン情報を取得する順序を指定します。順序は、バージョンの順序値に依存します。次のどれかを指定してください。なお、省略した場合は、取得するバージョンに順序性はありません。

- DBR_ORDER_NONE
取得するバージョンの順序を指定しません。
- DBR_ORDER_ASC
取得するバージョン情報を最も古いバージョンから順に取得します。
- DBR_ORDER_DESC
取得するバージョン情報を最も新しいバージョンから順に取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionable::GetVersionListAndLock メソッド

機能

バージョン情報一覧の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetVersionListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE,
    DmaInteger32         lOrdered = DBR_ORDER_NONE )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;           // 要素数
    union {
        pDmaBoolean    pbBoolean;    // Boolean型
        pDmaInteger32  plInteger32;   // Integer32型
        pDmaId         pId;           // ID型
        pDmapv         ppObject;      // Object型
        ppDmaString_T  ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;            // オブジェクトのOIID
    DmaInteger32   lType;            // オブジェクトの種別
    SDBR_PROPLIST  PropList;         // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32   lCount;
    SDBR_OBJINFO* pItem;
};
```

詳細

ロックタイプを明示的に指定して、このオブジェクトが包含している、すでに確定したバージョン情報の一覧を取得するメソッドです。取得するバージョン情報を、バージョンが作成された順序によってソートして取得できます。また、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。

初回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。ただし、再度取得する前に文書の削除処理をした場合は、残りが正しく取得されないことがあります。

複数回に分けて情報を取得する場合、取得される内容はすべて初回にメソッドをコールした時点での状態を示しています。したがって、1回目のメソッドのコール以降に変更した要素数やプロパティ値は、2回目には反映されません。

要素の個数に変更がなく、同じオブジェクトに対してこのメソッドをコールした場合も、取得される情報の順序は常に異なる可能性があります。

使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得したものの次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得したものの残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてこのメソッドをコールする場合、または前回のコールですべて取得した場合は、すべて DMA_FALSE が指定されたものとして処理されず。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 15-4 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		
lLockType	×	

(凡例)

: 有効

x : 無効

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合は、プロパティを取得しません。負の数を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、lPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、プロパティを取得しません。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報へのポインタが設定されます。バージョン情報の一覧は、オブジェクト情報リスト構造体である、SDBR_OBJINFO 構造体として格納されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリ解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。ただし、SDBR_OBJINFO 構造体メンバの pOID にはバージョン管理されるオブジェクトのバージョン識別子が設定されます。また、lType に相当する情報が設定されても、その内容は保証できません。

なお、バージョン管理されるオブジェクト (dmaClass_Versionable クラスのサブクラスのオブジェクト) が引数 pPropDefList で指定したプロパティを持つ場合に、プロパティ情報が設定される

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE

排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

IOrdered (入力)

バージョン情報を取得する順序を指定します。順序は、バージョンの順序値に依存します。次のどれかを指定してください。なお、省略した場合は、取得するバージョンに順序性はありません。

- DBR_ORDER_NONE
取得するバージョンの順序を指定しません。
- DBR_ORDER_ASC
取得するバージョン情報を最も古いバージョンから順に取得します。
- DBR_ORDER_DESC
取得するバージョン情報を最も新しいバージョンから順に取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionable::PutPropertyValues メソッド

機能

プロパティの設定 (バージョン指定あり)

形式

```
// 全バージョン共有のプロパティを対象にする場合
DmaBoolean PutPropertyValues(
SDBR_PROPLIST*      pPropList )

// バージョンをバージョン識別子で指定する場合
DmaBoolean PutPropertyValues(
pDmaString_T        pVersionId,
SDBR_PROPLIST*      pPropList )

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           // プロパティ識別子
    DmaInteger32       lType;           // データ型
    DmaInteger32       lCardinality;    // 基本単位
    DmaInteger32       lCount;         // 要素数
    union {                // プロパティの値
        pDmaBoolean     pbBoolean;     // Boolean型
        pDmaInteger32   plInteger32;   // Integer32型
        pDmaId          pId;           // ID型
        pDmapv          ppObject;      // Object型
        ppDmaString_T   ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*         pItem;
};
```

詳細

プロパティに値を設定するメソッドです。

バージョン識別子を指定しない場合、DMA オブジェクトの ConfigurationHistory オブジェクトに設定された、オブジェクト全体に対するプロパティが対象になります。

CdbVersionable::VersionCheckOut メソッドで取得した仮のバージョン識別子を指定した場合は、チェックアウトしたバージョンの dmaClass_Versionable クラスのサブクラスのオブジェクトのプロパティが対象になります。

バージョン識別子として NULL を指定した場合は、カレントバージョンが対象になります。

引数

pPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。SDBR_PROPLIST 構造体および SDBR_PROP 構造体に指定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

同じプロパティ識別子を指定したプロパティ構造体を複数指定した場合は、エラーになります。

また、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。

バージョン識別子は CdbrVersionable::VersionCheckOut メソッドで取得した仮のバージョン識別子も指定できます。NULL を指定した場合、カレントバージョンのプロパティが対象になります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbrCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbVersionable::UnbindPublicACL メソッド

機能

パブリック ACL のアンバインド (バージョン指定あり)

形式

```
// 指定したパブリックACLをアンバインドする場合
DmaBoolean UnbindPublicACL(
    pDmaString_T      pVersionId,
    DmaInteger32     lIdListCount,
    ppDmaString_T    ppIdList)

// すべてのパブリックACLをアンバインドする場合
DmaBoolean UnbindPublicACL(
    pDmaString_T      pVersionId )
```

詳細

指定したバージョンのオブジェクトにバインドしているパブリック ACL をアンバインドするメソッドです。

複数のパブリック ACL を一度にアンバインドしたり、すべてのパブリック ACL を一度にアンバインドしたりできます。

指定したパブリック ACL の OIID が存在しない場合、指定したパブリック ACL の OIID がバインドされていない場合、または重複した OIID を指定した場合はエラーになります。

引数

pVersionId (入力)

バージョン識別子を指定します。

NULL を指定した場合、カレントバージョンが対象になります。

lIdListCount (入力)

アンバインドするパブリック ACL の個数を指定します。0 以下の数を指定するとエラーになります。

ppIdList (入力)

アンバインドするパブリック ACL の OIID のポインタ配列を指定します。

NULL を指定するとエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUSES	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドで OIID を設定するか、ConnectObject メソッドでオブジェクトを接続してください。
ERR_DBR	ERR_PUBLICACL_NOT_BOUND	指定されたパブリック ACL はバインドされていません。	バインドされているパブリック ACL を指定して処理を再実行してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

15. CdbVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRAN SACTION	トランザクション外でメソッドが 呼ばれました。	Begin メソッドの呼び出し後に、こ の操作を実行してください。

CdbrVersionable::VersionCheckIn メソッド

機能

バージョンのチェックイン

形式

DmaBoolean VersionCheckIn ()

詳細

チェックアウトによって予約した仮のバージョンの追加を確定するメソッドです。

DMA オブジェクトとしては、Reservation オブジェクトを介して間接的に VersionSeries オブジェクトと関連づけられていた dmaClass_Versionable クラスのサブクラスのオブジェクトが、直接 VersionSeries オブジェクトに関連づけられます。

リファレンスファイル文書の場合

チェックインし、カレントとなるバージョンのコンテンツが存在するかを確認します。カレントバージョンのコンテンツが存在しない場合、ERR_FILE_REFERENCE_CURRENT_CONTENT_NOTFOUND のエラーになります。

また、CdbrSession::SetReferencePath メソッドでコンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行して ください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行して ください。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INV ALID	クラスライブラリのオブジェク トは無効です。	有効なクラスライブラリのオブ ジェクトを指定してください。

15. CdbrVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	CdbrVersionable::VersionRevoke メソッドを実行し、VersionCheckOut を実行してください。必要に応じてコンテンツを更新してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	チェックアウトされています。	チェックアウト状態を確認してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrVersionable::VersionCheckOut メソッド

機能

バージョンのチェックアウト

形式

```
DmaBoolean VersionCheckOut (
    ppDmaString_T ppReservedVerId)
```

詳細

次バージョンの追加を予約して、カレントバージョンのコピーを作成するメソッドです。DMA オブジェクトとしては、新たに Reservation オブジェクトと dmaClass_Versionable クラスのサブクラスのオブジェクトが作成されます。このオブジェクトは Reservation オブジェクトを介して、間接的に VersionSeries オブジェクトに関連づけられます。

CdbrVersionable::VersionRevoke メソッドでバージョンの追加をキャンセルすると、チェックアウト以降の処理は取り消されます。

リファレンスファイル文書の場合

カレントバージョンのコンテンツの存在を確認します。カレントバージョンのコンテンツが存在しない場合、ERR_FILE_REFERENCE_CURRENT_CONTENT_NOTFOUND のエラーになります。

また、CdbrSession::SetReferencePath メソッドで、コンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

なお、リファレンスファイル管理機能を使用している場合に、major_code が ERR_DB 以外のエラーのときは、ロールバック実行前に CdbrVersionable::VersionRevoke メソッドをコールしたあと、CdbrSession::Rollback メソッドをコールしてください。これによって、不要なコンテンツが残ることを防ぎます。

このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定して登録した文書に対し、ReferenceStorageMode エントリに Divide を指定してこのメソッドを実行する場合、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定して登録した文書に対し、ReferenceStorageMode エントリに Origin を指定してこのメソッドを実行する場合、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)

- pEntry で指定したファイル名

引数

ppReservedVerId (出力)

仮のバージョン識別子へのポインタが設定されます。値を格納した領域は、使用後にユーザプログラムで、dbrDelete 関数を使用して明示的に解放してください。この値はチェックイン後に割り当てられる正式なバージョン識別子とは異なるので注意してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新から順に確認し、コンテンツが存在しないバージョンを削除してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_RESERVATION_EXISTS	VersionSeries は Reservation をすでに保持しています。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbrVersionable::VersionRevoke メソッド

機能

チェックアウトの取り消し

形式

```
DmaBoolean VersionRevoke ()
```

詳細

バージョンのチェックアウト操作以降の処理を取り消します。

DMA オブジェクトとしては、Reservation オブジェクトを介して間接的に VersionSeries オブジェクトと関連づけられていた dmaClass_Versionable クラスのサブクラスのオブジェクトが破棄されて、Reservation オブジェクトも破棄されます。

チェックアウト状態でない場合にこのメソッドをコールすると、エラーになります。

リファレンスファイル文書の場合

カレントバージョンのコンテンツの存在を確認します。カレントバージョンのコンテンツが存在しない場合、ERR_FILE_REFERENCE_CURRENT_CONTENT_NOTFOUND のエラーになります。
また、CdbrSession::SetReferencePath メソッドで、コンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

引数

なし

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新から順に確認し、コンテンツが存在しないバージョンを削除してください。
ERR_DBR	ERR_FILE_REFERENCE_MISMATCH_STATUS	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生し、オブジェクトとコンテンツが不整合な状態となりました。	再度実行してチェックアウトを取り消してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	チェックアウトされていません。	チェックアウト状態を確認してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

15. CdbVersionable クラス (バージョン管理機能を提供するクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

16 CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

DocumentBroker で扱う文書のうち、バージョン管理をする文書を作成および管理する機能を提供するクラスです。

CdbVersionableDocument クラスの詳細

CdbVersionableDocument::~CdbVersionableDocument メソッド

CdbVersionableDocument::AddRendition メソッド

CdbVersionableDocument::CdbVersionableDocument メソッド

CdbVersionableDocument::ChangeMasterRendition メソッド

CdbVersionableDocument::ConvertContentType メソッド

CdbVersionableDocument::CreateIndex メソッド

CdbVersionableDocument::CreateObject メソッド

CdbVersionableDocument::CreateRelation メソッド

CdbVersionableDocument::DeleteRendition メソッド

CdbVersionableDocument::GetContent メソッド

CdbVersionableDocument::GetContentAndLock メソッド

CdbVersionableDocument::GetContentInfo メソッド

CdbVersionableDocument::GetRelationList メソッド

CdbVersionableDocument::GetRelationListAndLock メソッド

CdbVersionableDocument::GetRenditionList メソッド

CdbVersionableDocument::GetRenditionListAndLock メソッド

CdbVersionableDocument::PutRelationPropertyValues メソッド

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

CdbVersionableDocument::PutRenditionPropertyValues メソッド

CdbVersionableDocument::RemoveIndex メソッド

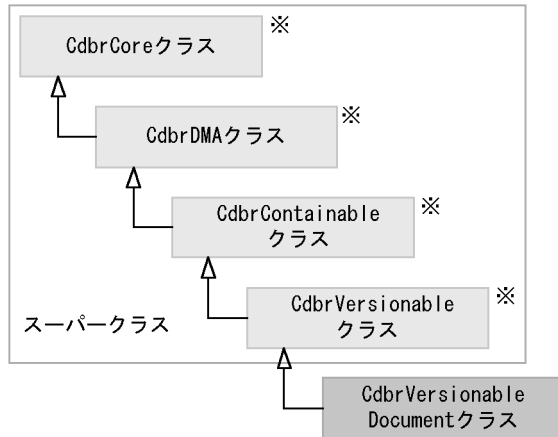
CdbVersionableDocument::RemoveRelation メソッド

CdbVersionableDocument::UpdateContent メソッド

CdbVersionableDocument::UpdateContentAndRenditionType メソッド

CdbrVersionableDocument クラスの詳細

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 16-1 CdbrVersionableDocument クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContainersCountVT	dbrProp_ContainersCountVT	Integer32 型	
VersionsCount	dbrProp_VersionsCount	Integer32 型	
CurrentVersion	dbrProp_CurrentVersion	String 型	
RenditionType	dbrProp_RenditionType	String 型	
RetrievalName	dbrProp_RetrievalName	String 型	-
HeadRelationsCount	dbrProp_HeadRelationsCount	Integer32 型	
TailRelationsCount	dbrProp_TailRelationsCount	Integer32 型	
ReferenceType	dbrProp_ReferenceType	Integer32 型	
ContentLocation	dbrProp_ContentLocation	String 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 16-2 CdbrVersionableDocument クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- ²

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- 3
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- 3
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID
オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティについては、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- PrimaryGroupId
- ACL
- SACL

- OwnerPermission
- PrimaryGroupPermission
- EveryonePermission
- PublicACLCount
- PublicACLIds

これらのプロパティについては、「CdbContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbVersionable クラスから継承されるプロパティ

- VersionsCount
- CurrentVersion
- ContainersCountVT

これらのプロパティの詳細については、「CdbVersionable クラスの詳細」のプロパティの詳細を参照してください。

CdbVersionableDocument クラスで導入されるプロパティ

- RenditionType
カレントバージョンの RenditionType を表す文字列を表します。
- RetrievalName
カレントバージョンに登録されているファイルのファイル名を表す文字列を表します。CreateObject メソッドまたは UpdateContent メソッドでファイルを登録・更新する場合、指定された URL からパス名を除いた文字列が設定されます。例えば、UNIX の場合に URL が "file:///tmp/sample.doc" のファイルを登録すると、文字列 "sample.doc" が設定されます。Windows の場合に URL が "file:///c:/temp/sample.doc" のファイルを登録すると、文字列 "sample.doc" が設定されます。ファイルが未登録の場合、長さ 0 の文字列が設定されます。値の変更も可能です。
- HeadRelationsCount
このオブジェクトが文書間リレーションで関連づけしているリレーション先文書の数を表します。
- TailRelationsCount
このオブジェクトが文書間リレーションでほかの文書からリレーション先として関連づけされている数を表します。

なおこのほかに、CdbVersionableDocument クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA のクラスの dmaClass_ConfigurationHistory クラスと dmaClass_DocVersion クラス、およびこれらのサブクラスのプロパティを参照、設定できます。DMA のクラスのプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbCore クラスから継承されるメソッド

メソッド	機能
~CdbCore	デストラクタ
CdbCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

16. CdbrVersionableDocument クラス (バージョン付き文書を扱うクラス)

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrVersionable クラスから継承されるメソッド

メソッド	機能
~CdbrVersionable	デストラクタ
BindPublicACL	パブリック ACL のバインド (バージョン指定あり)
CdbrVersionable	コンストラクタ
DeleteVersion	バージョンの削除
GetPropertyValues	プロパティの取得 (バージョン指定あり)

メソッド	機能
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定 (バージョン指定あり)
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得 (バージョン指定あり)
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定 (バージョン指定あり)
GetReservationStatus	チェックアウト状態の取得
GetReservationStatusAndLock	チェックアウト状態の取得と同時に指定したロックの設定
GetVersionList	バージョン情報一覧の取得
GetVersionListAndLock	バージョン情報一覧の取得と同時に指定したロックの設定
PutPropertyValues	プロパティの設定 (バージョン指定あり)
UnbindPublicACL	パブリック ACL のアンバインド (バージョン指定あり)
VersionCheckIn	バージョンのチェックイン
VersionCheckOut	バージョンのチェックアウト
VersionRevoke	チェックアウトの取り消し

これらのメソッドの詳細については、「15. CdbVersionable クラス (バージョン管理機能を提供するクラス)」を参照してください。

CdbVersionableDocument クラスで導入されるメソッド

メソッド	機能
~CdbVersionableDocument	デストラクタ
AddRendition	レンディションの追加
CdbVersionableDocument	コンストラクタ
ChangeMasterRendition	マスタレンディションへの変更
ConvertContentType	コンテンツ種別の変換
CreateIndex	全文検索インデクスの作成
CreateObject	オブジェクトの新規作成
CreateRelation	文書間リレーションの設定
DeleteRendition	レンディションの削除
GetContent	ファイルの取得
GetContentAndLock	ファイルの取得と同時に指定したロックの設定
GetContentInfo	マルチファイル文書の情報の取得
GetRelationList	文書のリレーション情報の一覧取得
GetRelationListAndLock	文書のリレーション情報の一覧取得と同時に指定したロックの設定
GetRenditionList	レンディション一覧の取得
GetRenditionListAndLock	レンディション一覧の取得と同時に指定したロックの設定
PutRelationPropertyValues	文書間リレーションのプロパティの設定
PutRenditionPropertyValues	レンディションのプロパティの設定
RemoveIndex	全文検索インデクスの削除

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

メソッド	機能
RemoveRelation	文書間リレーションの解除
UpdateContent	ファイルの更新
UpdateContentAndRenditionType	ファイルの更新および RenditionType の変更

以降, CdbVersionableDocument クラスで導入されるメソッドについて説明します。

CdbVersionableDocument::~CdbVersionableDocument メソッド

機能

デストラクタ

形式

```
virtual ~CdbVersionableDocument()
```

詳細

CdbVersionableDocument オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbVersionableDocument::AddRendition メソッド

機能

レンディションの追加

形式

```
// シングルファイル文書であるコンテンツを
// レンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pVersionId,
    pDmaString_T          pFilePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList )

// シングルファイル文書であるコンテンツを
// レンディション変換機能を使って登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pVersionId,
    pDmaString_T          pFilePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList,
    DmaInteger32          lConvertType)

// リファレンスファイル文書であるコンテンツをレンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pVersionId,
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList)

// File Link文書であるコンテンツをレンディションの追加と同時に登録する場合
DmaBoolean AddRendition(
    pDmaString_T          pVersionId,
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T          pRenditionType,
    SDBR_PROPLIST*       pPropList)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           // プロパティ識別子
    DmaInteger32         lType;           // データ型
    DmaInteger32         lCardinality;    // 基本単位
    DmaInteger32         lCount;         // 要素数
    union {
        pDmaBoolean      pbBoolean;     // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;           // ID型
        pDmapv           ppObject;      // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32         lContentOperateMode; // 操作モード
    pDmaString_T         pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T         pTargetPath;        // コンテンツ格納先パス
    pDmaString_T         pDeleteRootPath;    // 削除ディレクトリのルートパス
};
```

```
// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T    pFilePath;           // 登録ファイルまたはディレクトリのURL
    pDmaString_T    pFamName;           // FAM名
    pDmaString_T    pFamFilePath;       // コンテンツ保管先パス
    DmaInteger32    lContentDeleteMode; // コンテンツ削除モード
    DmaInteger32    lFileNamingMode;    // コンテンツ名自動生成モード
};
```

詳細

バージョン付き文書の指定されたバージョンに、指定されたレンディションとコンテンツを追加するメソッドです。レンディション変換機能を使ってコンテンツを登録する場合、コンテンツが空のレンディションを追加できます。ただし、レンディション変換機能を使うには、DocumentBroker Rendering Option が必要です。

このメソッドを使うことで、レンディションを複数持つ、マルチレンディション文書にすることができます。

なお、レンディションを複数持つマルチレンディション文書の場合、すべてのレンディションに空でない RenditionType が設定されている必要があります。つまり、RenditionType が NULL または空文字列 ("") のレンディションは存在できません。したがって、マスタレンディションの RenditionType が設定されていない (NULL または空文字列 ("") の場合) か、このメソッドの引数 pRenditionType が NULL または空文字列 ("") の場合、このメソッドはエラーになります。

このメソッドは、マルチファイルを管理するバージョン付き文書に対しては実行できません。

リファレンスファイル文書であるコンテンツを登録する場合

登録するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書であるコンテンツを登録する場合、CdbSession::SetReferencePath メソッドでコンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合は、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

File Link 文書であるコンテンツを登録する場合

登録するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、登録するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定したディレクトリ下のファイル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登録できます。また、すでに HiRDB File Link で管理されているデータをコンテンツとして登録することもできます。

File Link 文書であるコンテンツを登録する場合、このメソッドで、ERR_FILELINK_FAILED のエラーが発生したときは、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合、カレントバージョンがレンディション追加の対象になります。

pFilePath (入力)

追加するサブレンディションに登録するファイルのパス名を URL 形式で指定します。

例えば、レンディションの追加と同時にコンテンツを登録する場合で、UNIX の場合にファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合にファイル "c:\\temp\\sample.doc" を登録するには、"file:///c:\\temp\\sample.doc" と指定します。有効なプロトコルは "file:" だけです。"file:" 以外のプロトコルを指定した場合、指定を省略した場合または NULL を指定した場合、エラーになります。

なお、レンディション変換機能を使ってコンテンツを登録する場合 (IConvertType に DBR_CONVERT_TYPE_BATCH を指定した場合) だけ、pFilePath に NULL を指定できます。

pRenditionType (入力)

追加するサブレンディションの RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。NULL または空文字列 ("") は指定できません。

レンディションを追加しようとする文書に、この引数で指定する RenditionType と同じレンディションがすでにある場合、エラーになります。

pPropList (入力)

追加するレンディションに設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。プロパティを設定しない場合は NULL を指定してください。

なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

このメソッドで値を設定できるプロパティのプロパティ識別子を次に示します。

- dbrProp_RetrievalName

IConvertType (入力)

レンディション変換の方法を指定します。次に示す値のどちらかを指定してください。

- DBR_CONVERT_TYPE_USER
レンディション変換機能を使用しないでコンテンツを登録する場合に指定します。この場合、コンテンツとして登録するファイルはユーザ自身で用意し、pFilePath にそのパス名を指定します。
- DBR_CONVERT_TYPE_BATCH
このメソッドの実行後に DocumentBroker Rendering Option が提供するコマンドでレンディション変換を実行し、変換されたファイルをコンテンツとして自動的に登録する場合に指定します。
なお、レンディション変換機能を使用しない場合は、DBR_CONVERT_TYPE_USER を指定してください。
また、マスタレンディションがシングルファイル文書の場合だけ、DBR_CONVERT_TYPE_BATCH を指定してレンディションの追加を行うことができます。マスタレンディションがシングルファイル文書でない文書に対して、DBR_CONVERT_TYPE_BATCH を指定した場合は、DocumentBroker Rendering Option によるレンディション変換で、major_code として ERR_DBR、minor_code として ERR_CONTENTTYPE_MISMATCH のエラーになります。

ReferencePath (入力)

リファレンスファイル文書を登録するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。SDBR_REFERENCE_PATHINFO 構造体のメンバ lContentOperateMode に DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定し、pEntry に追加するサブレンディションに登録するファイルのパス名を URL 形式で指定します。

また、SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath に、コンテンツ格納先パスを、CdbSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを基点とする相対パスで指定します。DocumentBroker サーバが Windows の場合 UNC 形式のパスで指定することもできます。

なお、lContentOperateMode に DBR_OPERATEMODE_NONE を指定した場合、コンテンツを持たないオブジェクトが作成されます。

このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023
バイト (AIX の場合) 以上のときにこのメソッドを実行すると、

ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023
バイト (AIX の場合) 以上のときにこのメソッドを実行すると、

ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書を登録する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15)
SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を登録するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には、DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5)

SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体) を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_LIMITNUM_OF_RENDITIONS	保持できるレンディション数の上限に達しています。	この操作は実行できません。
ERR_DBR	ERR_MASTERRENDTYPE_NOT_SET	マスターレンディションの RenditionType に値が設定されていません。	この操作は実行できません。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_RENDITIONTYPE_DUPLICATED	RenditionType が重複しています。	重複しない RenditionType を指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND		要求された要素が見つかりません。 正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

16. CdbrVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbVersionableDocument::CdbVersionableDocument メソッド

機能

コンストラクタ

形式

`CdbVersionableDocument()`

詳細

`CdbVersionableDocument` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbVersionableDocument::ChangeMasterRendition メソッド

機能

マスタレンディションへの変更

形式

```
DmaBoolean ChangeMasterRendition(
    pDmaString_T      pVersionId,
    pDmaString_T      pRenditionType )
```

詳細

マルチレンディション文書の指定されたバージョンの、指定されたレンディションをマスタレンディションに変更します。変更した場合、既存のマスタレンディションはサブレンディションになります。

なお、メソッド実行時の RenditionStatus プロパティの状態フラグの値によって、このメソッドの実行可否に関して次に示す制限があります。

メソッド実行時の RenditionStatus プロパティの状態フラグの値	ChangeMasterRendition メソッドの実行可否
DBR_RENDSTATUS_SUBREND_EXIST	実行できる
DBR_RENDSTATUS_NO_SUBREND	実行できない
DBR_RENDSTATUS_MASTERREND_UPDATE	変換フラグが DBR_RENDSTATUS_CONVERT_NOTREQUIRED であれば実行できる

このメソッドは、マルチファイル文書に対して実行できません。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合、カレントバージョンがマスタレンディション変更の対象になります。

pRenditionType (入力)

対象となるバージョンのサブレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象バージョンのマスタレンディションの場合、または対象バージョンのサブレンディションとして存在しない場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENT_IS_EMPTY	コンテンツが変換処理されてなくて空です。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_IS_MASTERRENDITION	指定された RenditionType はマスタレンディションです。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_RENDITION_NOT_CONVERTED	コンテンツが更新処理されていません。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrVersionableDocument::ConvertContentType メソッド

機能

コンテンツ種別の変換

形式

```
// リファレンスファイル文書に変換する場合
DmaBoolean ConvertContentType(
    DmaInteger32                lExecuteMode,
    DmaInteger32                lSourceScope,
    DmaBoolean                  bIsChangeMaster,
    pDmaString_T                pVersionId,
    pDmaString_T                pRenditionType,
    DmaInteger32                lSourceContentType,
    SDBR_RENDITION_COMMENTINFO& CommentInfo,
    SDBR_REFERENCE_PATHINFO&   ReferencePath )

// シングルファイル文書に変換する場合
DmaBoolean ConvertContentType(
    DmaInteger32                lExecuteMode,
    DmaInteger32                lSourceScope,
    DmaBoolean                  bIsChangeMaster,
    pDmaString_T                pVersionId,
    pDmaString_T                pRenditionType,
    DmaInteger32                lSourceContentType,
    SDBR_RENDITION_COMMENTINFO& CommentInfo )

// レン디션コメント情報構造体
struct SDBR_RENDITION_COMMENTINFO {
    DmaBoolean    bIsInvestSourceComment; // コメントの付与フラグ
    DmaInteger32  lInvestMode;           // コメントの付与方式
    pDmaString_T  pRenditionComment;    // レン디션コメント
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32  lContentOperateMode; // 操作モード
    pDmaString_T  pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T  pTargetPath;        // コンテンツ格納先パス
    pDmaString_T  pDeleteRootPath;    // 削除ディレクトリのルートパス
};
```

詳細

コンテンツの格納先 (コンテンツ種別) を変換するメソッドです。

このメソッドを使用することでコンテンツの格納先を変換したレンディションが追加されます。

このメソッドを実行する前にコンテンツ保管先ベースパスを CdbrSession::SetReferencePath メソッドで設定しておく必要があります。設定されていない場合は、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

lSourceScope, pRenditionType, および lSourceContentType で指定した変換対象のコンテンツが存在しない場合は、ERR_CONVERT_CONTENT_TARGET_NOTFOUND のエラーになります。

このメソッドで変換可能なコンテンツ種別の組み合わせを次に示します。

表 16-3 ConvertContentType メソッドで変換できるコンテンツ種別

変換前のコンテンツ種別 (ISourceContentType で指定する値)	変換後のコンテンツ種別 (メソッドの形式)
シングルファイル文書 (DBR_CONTENTTYPE_CONTENT)	リファレンスファイル文書 (リファレンスファイル文書に変換する場合の形式)
リファレンスファイル文書 (DBR_CONTENTTYPE_REFERENCE)	シングルファイル文書 (シングルファイル文書に変換する場合の形式)

シングルファイル文書からリファレンスファイル文書に変換する場合

コンテンツが登録されていないシングルファイル文書をリファレンスファイル文書に変換すると、リファレンスファイル文書の edmProp_ReferenceType には、DBR_REFERENCETYPE_NONE が設定され、コンテンツは登録されません。

また、このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

リファレンスファイル文書からシングルファイル文書に変換する場合

リファレンスファイル文書の ContentLocation プロパティと CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパスを合わせた文字列長 (パスの区切り文字も含む) が、Windows の場合 260 バイト以上、AIX の場合 1023 バイト以上であると ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

引数

IExecuteMode (入力)

変換モードを指定します。

この引数には DBR_CONVERT_MODE_VERBOSE を指定します。

ISourceScope (入力)

変換対象とするレンディションの範囲を指定します。

- `DBR_CONVERT_SOURCE_ALL`
すべてのレンディションを対象とします。
- `DBR_CONVERT_SOURCE_SEPARATE`
単一のレンディションを対象とします。変換するレンディションのレンディションタイプは、`pRenditionType` で指定します。

`blsChangeMaster` (入力)

マスタレンディションのコンテンツを変換する場合に、変換して追加されるレンディションをマスタレンディションにするかしないかを指定します。

- `DMA_TRUE`
変換して追加されるレンディションをマスタレンディションに変更します。
- `DMA_FALSE`
マスタレンディションの変更は行いません。

`pRenditionType` (入力)

`lSourceScope` に `DBR_CONVERT_SOURCE_SEPARATE` を指定した場合は、変換の対象とするレンディションのレンディションタイプを表す文字列を指定します。指定したレンディションのコンテンツが変換されます。この引数で指定したレンディションタイプが、対象文書のレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は `NULL` を使用することもできます。

空文字列 ("") は指定できません。

ただし、`lSourceScope` に `DBR_CONVERT_SOURCE_ALL` が指定されている場合は、`pRenditionType` への値の指定は無効となります。

`pVersionId` (入力)

対象となるバージョンのバージョン識別子を指定します。これは `CdbVersionable::GetVersionList` メソッド、`CdbVersionable::GetVersionListAndLock` メソッドまたは `CdbVersionable::VersionCheckOut` メソッドで取得します。省略した場合または `NULL` を指定した場合は、カレントバージョンのコンテンツ種別を変換します。

`lSourceContentType` (入力)

変換対象とするコンテンツ種別を指定します。

- `DBR_CONTENTTYPE_CONTENT`
シングルファイル文書のコンテンツを対象とします。リファレンスファイル文書に変換する形式のメソッドの場合に指定します。メソッドの形式が一致しない場合、`DMARC_BAD_PARAMETER` のエラーになります。
- `DBR_CONTENTTYPE_REFERENCE`
リファレンスファイル文書のコンテンツを対象とします。シングルファイル文書に変換する形式のメソッドの場合に指定します。メソッドの形式が一致しない場合、`DMARC_BAD_PARAMETER` のエラーになります。
- `DBR_CONTENTTYPE_OTHER`
コンテンツの種別に関係なく変換を行います。ただし、表 17-3 の `ConvertContentType` メソッドで変換できるコンテンツ種別に示す組み合わせを対象とします。変換対象外のコンテンツ種別は無視されます。

CommentInfo (入力)

サブレンディションに付与するコメントの情報を SDBR_RENDITION_COMMENTINFO 構造体で指定します。

SDBR_RENDITION_COMMENTINFO 構造体については、「付録 C.1(18) SDBR_RENDITION_COMMENTINFO 構造体 (レンディションコメント情報構造体)」を参照してください。

ReferencePath (入力)

リファレンスファイル文書に変換するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書に変換するには、SDBR_REFERENCE_PATHINFO 構造体のメンバ lContentOperateMode に DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定します。

また、SDBR_REFERENCE_PATHINFO 構造体の pTargetPath に、コンテンツ格納先パスを、CdbSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを基点とした相対パスで指定します。

pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書に変換する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry、および pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OC CURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_LIMITNUM_OF_RENDITIONS	保持できるレンディション数の上限に達しています。	この操作は実行できません。
ERR_DBR	ERR_MASTERRENDTYPE_NOT_SET	マスタレンディションタイプ値が設定されていません。	マスタレンディションタイプ値を設定してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_RENDITIONTYPE_DUPLICATED	レンディションタイプが重複しています。	正しいレンディションタイプを設定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	次の要因が考えられます。 <ul style="list-style-type: none"> セッションが接続されていません。 EDMRefresher コマンドが実行中です。 	要因に応じて、次のどちらかの対策を実施してください。 <ul style="list-style-type: none"> セッションを接続してください。 EDMRefresher コマンドが終了したあと、再度実行してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_CONVERT_CONTENT_TARGET_NOTFOUND	変換対象のコンテンツは存在しません。	オブジェクトのコンテンツ種別または指定した変換対象とするコンテンツ種別 (ISourceContentType) を見直してください。
ERR_DBR	ERR_CONVERT_CONTENT_NOT_RETRIEVALNAME	変換対象コンテンツの dmaProp_RetrievalName プロパティの値が設定されていない (値が NULL または、空文字列 ("")) のため、変換処理を実行できません。	dmaProp_RetrievalName プロパティの値を設定してメソッドを実行してください。

CdbVersionableDocument::CreateIndex メソッド

機能

全文検索インデクスの作成

形式

```
DmaBoolean CreateIndex(
    const pDmaString_T      pFilePath,
    const pDmaString_T      pVersionId = NULL )
```

詳細

指定したパス名のファイルから、全文検索インデクスを作成するメソッドです。パス名に NULL を指定した場合は、登録されている文書 (マルチレンディション文書の場合はマスタレンディション) に含まれているコンテンツから全文検索インデクスが作成されます。ただし、マルチファイル文書、リファレンスファイル文書または File Link 文書の場合、NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

既存の全文検索インデクスを、別のファイルで更新する場合も、このメソッドを実行してください。これによって、インデクスの遅延更新が実現できます。

このメソッドは、全文検索機能付き文書クラスから作成されている DocVersion オブジェクトを構成要素とした CdbVersionableDocument オブジェクトにだけ実行できます。それ以外のオブジェクトを対象に実行した場合は、エラーになります。

また、インデクスを作成できるのは、テキストファイルとして作成されたファイルだけです。それ以外の RenditionType の文書を指定してこのメソッドを実行した場合の動作は保証されません。場合によっては、データベースがロールバックする可能性があります。この場合はデータベースエラーが出力されますので、エラーコードを参照して、データを再登録してください。

なお、登録されている文書とインデクスを作成するためのファイルの対応づけは、上位で行います。

引数

pFilePath (入力)

全文検索インデクスを作成するファイルのパス名を設定します。登録されている文書 (マルチレンディション文書の場合はマスタレンディション) に含まれるコンテンツからインデクスを作成する場合には NULL を設定します。それ以外のファイルからインデクスを作成する場合には、ファイルのパス名を URL 形式で設定します。例えば、UNIX の場合、全文検索インデクスをファイル "/tmp/sample.txt" から作成するには、"file:///tmp/sample.txt" と指定します。Windows の場合、全文検索インデクスをファイル "c:¥temp¥sample.txt" から作成するには、"file:///c:¥temp¥sample.txt" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合、カレントバージョンが対象になります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbrVersionableDocument::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
// シングルファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    pDmaString_T      pFilePath,
    pDmaString_T      pRenditionType,
    ppDmaString_T     ppOIID,
    DmaInteger32      lCreateIndexFlg = DBR_CREATE_INDEX)

// マルチファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_PATHLIST&    PathList,
    pDmaString_T      pRenditionType,
    ppDmaString_T     ppOIID)

// リファレンスファイル文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T      pRenditionType,
    ppDmaString_T     ppOIID)

// File Link文書の場合
DmaBoolean CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAININFO*   pDMAInfoList,
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T      pRenditionType,
    ppDmaString_T     ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           // プロパティ識別子
    DmaInteger32       lType;           // データ型
    DmaInteger32       lCardinality;    // 基本単位
    DmaInteger32       lCount;         // 要素数
    union {
        pDmaBoolean    pbBoolean;      // Boolean型
        pDmaInteger32  plInteger32;    // Integer32型
        pDmaId         pId;           // ID型
        pDmapv         ppObject;      // Object型
        ppDmaString_T  ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*        pItem;
};

// DMAオブジェクト生成用の構造体
```

```

struct SDBR_DMMAINFO {
    DmaId          ClassId;
    SDBR_PROPLIST  PropList;
};

// ファイルパス情報構造体
struct SDBR_PATHINFO {
    pDmaString_T   pFilePath;          // ファイルのURL
    pDmaString_T   pComponentType;    // ファイルを識別する文字列
};

// ファイルパス情報リスト構造体
struct SDBR_PATHLIST {
    DmaInteger32   lCount;             // 登録ファイル数
    SDBR_PATHINFO* pItem;            // ファイルパス情報構造体のポインタ
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32   lContentOperateMode; // 操作モード
    pDmaString_T   pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T   pTargetPath;       // コンテント格納先パス
    pDmaString_T   pDeleteRootPath;   // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T   pFilePath;          // 登録ファイルまたはディレクトリのURL
    pDmaString_T   pFamName;          // FAM名
    pDmaString_T   pFamFilePath;      // コンテント保管先パス
    DmaInteger32   lContentDeleteMode; // コンテント削除モード
    DmaInteger32   lFileNamingMode;   // コンテント名自動生成モード
};

```

詳細

オブジェクトを新規に作成するメソッドです。

SetOID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更されずに、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

Windows の場合、次のことに注意してください。

- GetContentAndLock メソッドによるコンテンツの取得時、大文字と小文字を区別しないと同一になるファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。
- GetContentAndLock メソッドによるコンテンツの取得時、ロングファイル名とショートファイル名の関係で同一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合

作成するオブジェクトが次の条件に当てはまる場合に、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデックスを作成できます。

- DMA オブジェクト生成用の構造体である、SDBR_DMMAINFO 構造体の ClassId に、dmaClass_DocVersion クラスのサブクラスとして全文検索機能付き文書クラスが指定されている。
- RenditionType がテキスト型 (MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

ファイルの登録が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合にパス名が "file:///tmp/sample.doc" のファイルを登録すると、RetrievalName プロパティには "sample.doc" が設定されます。Windows の場合にパス名が "file:///c:¥temp¥sample.doc" のファイルを登録すると、RetrievalName プロパティには "sample.doc" が設定されます。

UNIX の場合に、CdbrVersionableDocument オブジェクトを新規に作成して接続する例を次に示します。この例では、DMA オブジェクトの ConfigurationHistory オブジェクトの基になるクラスの識別子「ClassID_ConfHist」および DMA オブジェクトの DocVersion オブジェクトの基になるクラスの識別子「ClassID_DocVer」を指定して、CdbrVersionableDocument オブジェクトを新規に作成します。そして、ファイル /tmp/sample.txt を、RenditionType"MIME::text/plain" で登録して、そのポインタを取得します。なお、Windows の場合は、ファイル c:¥temp¥sample.txt を登録する例として、例中の "file:///tmp/sample.txt" を "file:///c:¥temp¥sample.txt" と読み替えてください。

CdbrVersionableDocument オブジェクトを新規に作成して接続する例

```
// 変数宣言
CdbrVersionableDocument DbrVerDoc;
SDBR_DMAINFO             DMAClassList [2];
pDmaString                pOIID;

// DMAクラス情報の作成
DMAClassList[0].ClassId = ClassID_ConfHist;
DMAClassList[0].PropList.pItem = NULL;
DMAClassList[0].PropList.lCount = 0;
DMAClassList[1].ClassId = ClassID_DocVer;
DMAClassList[1].PropList.pItem = NULL;
DMAClassList[1].PropList.lCount = 0;
// オブジェクトの新規作成
DbrVerDoc.CreateObject(pDbrSession,
    2, DMAClassList,
    "file:///tmp/sample.txt", "MIME::text/plain", &pOIID,
    DBR_CREATE_INDEX);
```

マルチファイル文書の場合

登録するコンテンツの情報として、SDBR_PATHLIST 構造体を指定します。

マルチファイル文書を作成する場合は、次のことに注意してください。

- マルチファイル文書では、テキストファイルの部分が特定できないため、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、マルチファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- dbrProp_RetrievalName プロパティに値を設定する場合は、SDBR_DMAINFO 構造体のメンバ PropList に設定された SDBR_PROPLIST 構造体に設定してください。SDBR_PROPLIST 構造体に dbrProp_RetrievalName を設定していない場合、データベースには NULL 値が格納されます。
- SDBR_PATHLIST 構造体の lCount に指定したファイル数より、SDBR_PATHINFO の配列の個数が多い場合は、lCount に指定したファイル数分のファイルが登録されます。
- 同じファイル名を持つコンテンツを格納しようとした場合は、ERR_RETRIEVALNAME_DUPLICATED のエラーになります。なお、ファイル名は、英字の大文字と小文字を区別して比較されます。
- SDBR_PATHLIST 構造体の lCount が 1 の場合で、コンテンツを格納するオブジェクトとして dmaClass_ContentTransfer オブジェクトを作成するときは、マルチファイル管理機能を使用しない場合の形式で、メソッドを実行してください。

リファレンスファイル文書の場合

登録するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。

リファレンスファイル文書を作成する場合は、次のことに注意してください。

- リファレンスファイル文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、リファレンスファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツに登録するファイルの種類を表すレンディションタイプは、登録したファイルの拡張子からは設定されません。レンディションタイプは、CreateObject メソッドの引数として明示的に指定してください。この引数に指定した値が、レンディションごとのファイルの種類を表すプロパティとして設定されます。

File Link 文書の場合

登録するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、登録するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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登録できます。また、すでに HiRDB File Link で管理されているデータをコンテンツとして登録することもできます。

File Link 文書を作成する場合は、次のことに注意してください。

- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツに登録するファイルの種類を表すレンディションタイプは、登録したファイルの拡張子からは設定されません。レンディションタイプは、CreateObject メソッドの引数として明示的に指定してください。この引数に指定した値が、レンディションごとのファイルの種類を表すプロパティとして設定されます。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。

1 を指定した場合は、edmClass_VersionTracedComponentDocVersion クラスを使用して DocVersion オブジェクトが作成されます。

0 を指定した場合は、pDMAInfoList の内容は無視され、edmClass_VersionTracedComponentDocVersion クラスを使用して DocVersion オブジェクトが作成されます。また、dmaClass_ConfigurationHistory クラスを使用して ConfigurationHistory オブジェクトが作成されます。

負の数を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報である、SDBR_DMAINFORM 構

造体を格納した構造体の配列を指定します。

DMA オブジェクトの情報として pDMAInfoList[0] には dmaClass_ConfigurationHistory クラスまたはそのサブクラスの情報, pDMAInfoList[1] には dmaClass_DocVersion クラスまたはそのサブクラスの情報を指定してください。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMMAINFO 構造体については、「付録 C.1(3) SDBR_DMMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA, minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

PathList (入力)

登録するファイルのファイルパス情報が設定された SDBR_PATHLIST 構造体を指定します。ファイルパス情報リスト構造体については、「付録 C.1(11) SDBR_PATHLIST 構造体 (ファイルパス情報リスト構造体)」を参照してください。

pFilePath (入力)

登録するファイルのパス名を指定します。パス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" を登録するには、"file:///c:\temp\sample.doc" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合または指定を省略した場合はエラーになります。NULL を指定した場合、ファイルは登録されません。

pRenditionType (入力)

RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。

NULL を指定した場合、ファイルの拡張子から RenditionType を決定する処理はされません。したがって、RenditionType には何も設定されません。

ppOID (出力)

作成した CdbrVersionableDocument オブジェクトの OID へのポインタが設定されます。ポインタを格納する領域を指定してください。なお、*ppOID がポイントするメモリは、ユーザプログラムでは解放しないでください。

ICreateIndexFlg (入力)

全文検索インデクスを作成するかどうかを指定します。次のどちらかを指定してください。省略した場合は、DBR_CREATE_INDEX を指定したものとして処理されます。

- DBR_CREATE_INDEX
全文検索インデクスを作成します。
- DBR_NOT_CREATE_INDEX
全文検索インデクスを作成しません。

なお、SDBR_DMAININFO 構造体の ClassId に全文検索機能付きクラス以外のクラスを指定して作成した場合またはファイルの RenditionType がテキスト型でない場合は、ICreateIndexFlg の値は無視されます。

ReferencePath (入力)

リファレンスファイル文書を登録するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。リファレンスファイル文書を登録するには、SDBR_REFERENCE_PATHINFO 構造体のメンバ lContentOperateMode に DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定し、pEntry に登録するファイルのパス名を URL 形式で指定します。また、SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath に、コンテンツ格納先パスを、CdbrSession::SetReferencePath メソッドで設定したコンテンツ格納先ベースパスを基点とする相対パスで指定します。DocumentBroker サーバが Windows の場合 UNC 形式のパスで指定することもできます。

なお、lContentOperateMode に DBR_OPERATEMODE_NONE を指定した場合、コンテンツがないオブジェクトが作成されます。

このメソッドを実行する場合、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合
次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときにこのメソッドを実行すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書を登録する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を登録するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には、登録コンテンツがないオブジェクトが作成されます。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA		CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_BAD_CONTENTSCOUNT	登録しようとするファイルの数が多過ぎます。	登録するファイルの数を見直してください。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	• 詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	• 詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACE として正しい CdbrCompound オブジェクトを指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLICACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1 文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1 文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1 文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能で、コンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> • ネットワーク環境を確認してください。 • _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 • 時間を置いて、再度実行してください。 • ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbrVersionableDocument::CreateRelation メソッド

機能

文書間リレーションの設定

形式

```
// バージョンなし文書と関連づける場合
DmaBoolean CreateRelation(
    CdbrDocument*      pDbrDocument,
    SDBR_PROPLIST*    pRelPropList,
    ppDmaString_T     ppRelId)

// バージョン付き文書と関連づける場合
DmaBoolean CreateRelation(
    CdbrVersionableDocument* pDbrVersionableDocument,
    SDBR_PROPLIST*          pRelPropList,
    ppDmaString_T           ppRelId)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality; // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;        // ID型
        pDmapv           ppObject;   // Object型
        ppDmaString_T    ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};
```

詳細

文書間のリレーションを設定するメソッドです。リレーション先には文書系のオブジェクトを指定します。指定できる文書系のオブジェクトを次に示します。

- CdbrDocument オブジェクト (バージョンなし文書)
- CdbrVersionableDocument オブジェクト (バージョン付き文書)

なお、リレーションには、ユーザ設定のプロパティを設定できます。

引数

pDbrDocument (入力)

このオブジェクトと関連づける CdbrDocument オブジェクト (バージョンなし文書) を指定します。

pDbrVersionableDocument (入力)

このオブジェクトと関連づける CdbrVersionableDocument オブジェクト (バージョン付き文書) を指定します。

pRelPropList (入力)

リレーションに設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。プロパティを設定しない場合は、NULL を指定してください。

なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppRelId (出力)

リレーション識別子が設定されます。リレーション識別子が格納される領域を指定してください。

このリレーション識別子は、文書間リレーションを解除するための RemoveRelation メソッドや、リレーションにプロパティを設定するための PutRelationPropertyValues メソッドで使用します。なお、*ppRelId がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionableDocument::DeleteRendition メソッド

機能

レンディションの削除

形式

```
DmaBoolean DeleteRendition(
    pDmaString_T          pVersionId,
    pDmaString_T          pRenditionType )
```

詳細

マルチレンディション文書の指定されたサブレンディションとそのコンテンツを削除します。

リファレンスファイル文書であるレンディションとそのコンテンツを削除する場合、CdbSession::SetReferencePath メソッドでコンテンツ格納先ベースパスを設定しておく必要があります。設定されていない場合、ERR_TARGETCONTENTPATH_NOT_SET のエラーになります。

このメソッドは、マルチファイルを管理するバージョン付き文書に対しては実行できません。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合、カレントバージョンがレンディション削除の対象になります。

pRenditionType (入力)

削除するサブレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が対象バージョンのマスタレンディションの場合、または対象バージョンのサブレンディションとして存在しない場合はエラーになります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

16. CdbvVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_IS_MASTERRENDITION	指定されたレンディションはマスターレンディションです。	この操作は実行できません。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	この操作は実行できません。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbVersionableDocument::GetContent メソッド

機能

ファイルの取得

形式

```
// マスタレンディションのファイルを取得する場合
DmaBoolean GetContent(
    const pDmaString_T      pFilePath,
    const pDmaString_T      pVersionId = NULL)

// RenditionTypeを指定してファイルを取得する場合
DmaBoolean GetContent(
    const pDmaString_T      pFilePath,
    const pDmaString_T      pVersionId,
    pDmaString_T            pRenditionType )
```

詳細

文書空間に登録されているファイルを、指定したパスに複写するメソッドです。

マルチレンディション文書の場合、pRenditionType を指定することで、対応する RenditionType を持つレンディションのファイルが取得できます。

UNIX の場合に、カレントバージョンとして登録されているファイルを /tmp/sample.doc に複写して取得する例を次に示します。Windows の場合は、c:¥temp¥sample.doc に複写して取得する例として、例中の "file:///tmp/sample.doc" を "file:///c:¥temp¥sample.doc" と読み替えてください。

ファイルを取得する例

```
pDbrVerDoc->GetContent("file:///tmp/sample.doc");
```

引数

pFilePath (入力)

取得したファイルを複写する複写先のパス名を指定します。

ファイルのパス名は URL 形式で設定します。有効なプロトコルは "file:" だけです。例えば、UNIX の場合、ファイル "/tmp/sample.doc" に複写するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:¥temp¥sample.doc" に複写するには、"file:///c:¥temp¥sample.doc" と指定します。

指定したパスのファイルが存在する場合は上書きします。ファイル名は、RenditionType プロパティまたは RetrievalName プロパティを参考にして、メソッドをコールする側が任意に決定してください。

指定したパスのファイルが存在しない場合、ディレクトリが存在し、かつファイルの作成が可能ならば新規にファイルを作成します。それ以外の場合はエラーになります。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。この識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。省略した場合は NULL を指定した場合は、カレントバージョンのファイルを取得します。

pRenditionType (入力)

RenditionType を指定してファイルを取得する場合、取得するレンディションの RenditionType を表す文

字列を指定します。

この引数で指定した RenditionType が、対象バージョンのレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は NULL を指定するか、「マスタレンディションのファイルを取得する場合」の形式を使用することもできます。

空文字列 ("") は指定できません。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbVersionableDocument::GetContentAndLock メソッド

機能

ファイルの取得と同時に指定したロックの設定

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの取得)
DmaBoolean GetContentAndLock(
    const pDmaString_T      pFilePath,
    const pDmaString_T      pVersionId = NULL,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの取得)
DmaBoolean GetContentAndLock(
    const pDmaString_T      pFilePath,
    const pDmaString_T      pVersionId,
    pDmaString_T            pRenditionType,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// マルチファイル文書の場合
DmaBoolean GetContentAndLock(
    pDmaString_T            pDirPath,
    SDBR_CONTENTLIST&     ContentList,
    pDmaString_T            pVersionId,
    pDmaString_T            pRenditionType,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// リファレンスファイル文書の場合
DmaBoolean GetContentAndLock(
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T            pVersionId,
    pDmaString_T            pRenditionType,
    pDmaInteger32          plReferenceType,
    ppDmaString_T          ppContentLocation,
    DmaInteger32            lLockType=DMA_LOCK_WRITE)

// File Link文書の場合
DmaBoolean GetContentAndLock(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    const pDmaString_T      pVersionId,
    pDmaString_T            pRenditionType,
    DmaBinaryValue**       ppbvFileHandle,
    DmaInteger32            lLockType = DMA_LOCK_WRITE)

// コンテント情報構造体
struct SDBR_CONTENTINFO {
    pDmaString_T      pComponentType; // ファイルを識別する文字列
    pDmaString_T      pRetrievalName; // ファイル名
    DmaUInteger32     lContentSize;   // ファイルのサイズ
};

// コンテント情報リスト構造体
struct SDBR_CONTENTLIST {
    DmaInteger32      lCount;          // 取得ファイル数
    SDBR_CONTENTINFO* pItem;         // コンテント情報構造体の
                                        // ポインタ
    DmaInteger32      lValidMode;     // 有効モード
    DmaBoolean        bIsMultiContents; // マルチファイル識別
};

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
```

```

    DmaInteger32  lContentOperateMode;    // 操作モード
    pDmaString_T  pEntry;                // ファイルのパスおよびロケーション
    pDmaString_T  pTargetPath;          // コンテンツ格納先パス
    oDmaString_T  pDeleteRootPath;      // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T  pFilePath; // 取得先のファイルまたはディレクトリのURL
    pDmaString_T  pFamName;
    pDmaString_T  pFamFilePath;
    DmaInteger32  lContentDeleteMode;
    DmaInteger32  lFileNamingMode;
};

```

詳細

ロックタイプを明示的に指定して、文書空間に登録されているファイルを、指定したパスに複写するメソッドです。マルチレンディション文書の場合、pRenditionType を指定することで、対応する RenditionType を持つレンディションのファイルが取得できます。

Windows の場合、次のことに注意してください。

- 大文字と小文字を区別しないと同一になるファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。
- ロングファイル名とショートファイル名の関係で同一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合

取得したコンテンツが引数 pFilePath に指定したファイルに複写されます。マルチレンディションのサブレンディションのコンテンツを取得する場合は、引数 pRenditionType に対象にするレンディションを指定します。

UNIX の場合に、カレントバージョンとして登録されているファイルを /tmp/sample.doc に複写して取得する例を次に示します。Windows の場合は、c:\temp\sample.doc に複写して取得する例として、例中の "file:///tmp/sample.doc" を "file:///c:\temp\sample.doc" と読み替えてください。なお、この例で取得できるのは、マスタレンディションのコンテンツです。

ファイルを取得する例

```
pDbrVerDoc->GetContentAndLock("file:///tmp/sample.doc", DMA_LOCK_READ);
```

マルチファイル文書の場合

引数 ContentList で指定した複数のファイルをコンテンツとして取得します。取得したコンテンツは、引数 pDirPath で指定したディレクトリに出力されます。ファイル名は、SDBR_CONTENTINFO 構造体に設定している pRetrievalName の値になります。

マルチファイル文書の場合、次のことに注意してください。

- GetContentInfo メソッドで取得した SDBR_CONTENTLIST 構造体を使用して、このメソッドを実行する場合、IValidMode は変更しないでください。
- この形式の GetContent メソッドを、マルチファイル文書ではないバージョン付き文書に対して実行すると、ERR_NOT_MULTI_CONTENTS のエラーになります。

リファレンスファイル文書の場合

取得したコンテンツが、引数 ReferencePath の SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry に指定したファイルに出力されます。

リファレンスファイル文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentReference オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_REFERENCE の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

取得したコンテンツが、引数 FileLinkPath の SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に指定したファイルまたはディレクトリに出力されます。

コンテンツとしてディレクトリが登録されている文書のレンディションに対してこのメソッドを実行すると、引数に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に指定した名前のディレクトリが作成され、その下に登録されているディレクトリおよびサブディレクトリ下のすべてのファイルが取得されます。コンテンツとしてディレクトリが登録されている文書のレンディションとは、AddRendition メソッド、CreateObject メソッド、UpdateContent メソッドまたは UpdateContentAndRenditionType メソッドの引数 FileLinkPath に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に、ディレクトリを指定して登録または更新した文書のレンディションのことです。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- コンテンツとして空のディレクトリが登録されているレンディションに対してこのメソッドを実行すると、ERR_CONTENT_NOT_REGISTERED のエラーになります。コンテンツとして空のディレクトリが登録されているレンディションとは、AddRendition メソッド、CreateObject メソッド、UpdateContent メソッドまたは UpdateContentAndRenditionType メソッドの引数 FileLinkPath に指定した SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath に、空のディレクトリを指定して作成または更新した文書のレンディションのことです。
- 引数に指定する SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath で指定したファイル名のファイル、またはディレクトリパスのディレクトリ下に取得しようとするファイルがすでに存在する場合、ERR_FILELINK_FAILED のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。

引数

pDirPath (入力)

出力されるファイルの格納先ディレクトリを、URL 形式で指定します。

指定したパスのディレクトリが存在し、かつそのディレクトリへの実行権限、書き込み権限があれば、取得したファイルを格納します。それ以外の場合、エラーになります。

ContentList (入力)

取得するコンテンツの情報を設定した SDBR_CONTENTLIST 構造体を指定します。

SDBR_CONTENTLIST 構造体については、「付録 C.1(2) SDBR_CONTENTLIST 構造体 (コンテンツ情報リスト構造体)」を参照してください。

pFilePath (入力)

取得したファイルを複写する複写先のパス名を指定します。

ファイルのパス名は URL 形式で指定します。有効なプロトコルは "file:" だけです。例えば、UNIX の場合、ファイル "/tmp/sample.doc" に複写するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" に複写するには、"file:///c:\temp\sample.doc" と指定します。

指定したパスのファイルが存在する場合は上書きします。ファイル名は、RenditionType プロパティまたは RetrievalName プロパティを参考にして、メソッドをコールする側が任意に決定してください。

指定したパスのファイルが存在しない場合、ディレクトリが存在し、かつファイルの作成が可能ならば新規にファイルを作成します。それ以外の場合はエラーになります。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。これは CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。省略した場合または NULL を指定した場合は、カレントバージョンのファイルを取得します。

pRenditionType (入力)

RenditionType を指定してファイルを取得する場合、取得するレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象バージョンのレンディションとして存在しない場合はエラーになります。

マスタレンディションの場合は NULL を指定するか、「マスタレンディションのファイルを取得する場合」の形式を使用することもできます。なお、空文字列 ("") は指定できません。

マルチファイル文書の場合、NULL または「シングルファイル文書の場合 (マスタレンディションのファイルの取得)」の形式を指定してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

ReferencePath (入力)

リファレンスファイル文書を取得するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。ContentLocation プロパティと CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパスを合わせた文字列長 (パスの区切り文字も含む) が、Windows の場合 260 バイト以上、AIX の場合は 1023 バイト以上であると ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

リファレンスファイル文書を取得する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ

pTargetPath および pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

plReferenceType (出力)

コンテンツのリファレンス種別 (ReferenceType) を取得する場合、リファレンス種別を格納する領域のポインタを指定します。NULL を指定した場合、リファレンス種別は返却されません。

ppContentLocation (出力)

コンテンツロケーション (ContentLocation プロパティ) を取得する場合、コンテンツロケーションへのポインタを格納する領域を指定します。NULL を指定した場合、コンテンツロケーションは返却されません。取得した領域は、dbrDelete 関数で解放する必要があります。dbrDelete 関数の詳細については、「20. 関数」の「dbrDelete 関数」を参照してください。

FileLinkPath (出力)

SDBR_FILELINK_PATHINFO 構造体の pFilePath メンバだけが有効となります。そのほかのメンバの設定値は無視されます。pFilePath に、コンテンツの取得先となるクライアントのファイルパスまたはディレクトリパスを指定してください。

SDBR_FILELINK_PATHINFO 構造体の指定内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

ppbvFileHandle (出力)

ppbvFileHandle にファイルハンドル情報が返却されます。この引数には NULL を指定してください。取得した領域は、dbrDelete 関数で解放する必要があります。dbrDelete 関数の詳細については、「20. 関数」の「dbrDelete 関数」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_COMPONENTTYPE_DUPLICATED	ComponentType が重複して指定されています。	ComponentType が重複しないように指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_CONTENT_NOT_REGISTERED	コンテンツが登録されていません。	コンテンツを登録してください。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_NOT_MULTI_CONTENTS	マルチファイル文書ではありません。	マルチファイル管理機能を使用しない場合の形式で実行してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_REFERENCETYPE_MISMATCH	オブジェクト種別が一致していません。	オブジェクト種別または指定した操作モード (IContentOperateMode) を見直してください。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザー数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbVersionableDocument::GetContentInfo メソッド

機能

マルチファイル文書の情報の取得

形式

```
DmaBoolean GetContentInfo (
    pDmaString_T          pVersionId,
    pDmaString_T          pRenditionType,
    SDBR_CONTENTLIST**   ppContentList,
    DmaInteger32          lLockType = DMA_LOCK_READ)

//コンテンツ情報構造体
struct SDBR_CONTENTINFO {
    pDmaString_T          pComponentType; // ファイルを識別する文字列
    pDmaString_T          pRetrievalName; // ファイル名
    DmaUInteger32         lContentSize; // ファイルのサイズ
};

//コンテンツ情報リスト構造体
struct SDBR_CONTENTLIST {
    DmaInteger32          lCount; // 取得ファイル数
    SDBR_CONTENTINFO*    pItem; // コンテンツ情報構造体の
                                // ポインタ
    DmaInteger32          lValidMode; // 有効モード
    DmaBoolean            bIsMultiContents; // マルチファイル識別
};
```

詳細

指定したレンディションのコンテンツの情報を取得します。取得した情報は、引数 ppContentList で指定した SDBR_CONTENTLIST 構造体の領域に格納されます。返却された SDBR_CONTENTLIST 構造体のメンバに設定される値については、「付録 C.1(2) SDBR_CONTENTLIST 構造体 (コンテンツ情報リスト構造体)」を参照してください。

この ppContentList に返却された SDBR_CONTENTLIST 構造体は、直接 GetContentAndLock メソッドの引数として指定できます。ただし、返却された SDBR_CONTENTLIST 構造体の lCount が 0 でないことを確認してから、GetContentAndLock メソッドの引数として指定してください。lCount が 0 である SDBR_CONTENTLIST 構造体を GetContentAndLock メソッドの引数として指定すると、DMARC_BAD_PARAMETER のエラーになります。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合、カレントバージョンがマスタレンディション変更の対象になります。

pRenditionType (入力)

レンディションの RenditionType を表す文字列を指定します。指定したレンディションのコンテンツの情報が取得されます。

NULL を指定した場合、マスタレンディションが処理対象になります。

ppContentList (出力)

コンテンツ情報リスト構造体のポインタを格納する領域を指定します。このポインタが示す領域はクラスライブラリが確保するため、使用後はこのポインタを dbrDelete 関数で解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrVersionableDocument::GetRelationList メソッド

機能

文書のリレーション情報の一覧取得

形式

```
DmaBoolean GetRelationList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lRelPropDefListCount,
    SDBR_PROPDEF*       pRelPropDefList,
    DmaInteger32         lRelationListCount,
    SDBR_RELATIONLIST** ppRelationList,
    DmaInteger32         lRelationendType,
    DmaInteger32         lRelationendStatus)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32   lType;           //データ型
    DmaInteger32   lCardinality;    //基本単位
    DmaInteger32   lCount;         //値の個数
    union {
        pDmaBoolean pbBoolean;     //Boolean型
        pDmaInteger32 plInteger32; //Integer32型
        pDmaId       pId;          //ID型
        pDmapv       ppObject;     //Object型
        ppDmaString_T ppString;    //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;          //プロパティ情報の個数
    SDBR_PROP*     pItem;         //プロパティ情報
};

// リレーションリスト構造体
struct SDBR_RELATIONLIST {
    DmaInteger32   lCount; //リレーションしている数
    SDBR_RELATIONINFO* pItem; //リレーション情報構造体へのポインタ
};

// リレーション情報構造体
struct SDBR_RELATIONINFO {
    pDmaString_T   pOIID;          //文書オブジェクトのOIID
    DmaInteger32   lType;          //オブジェクトの種別
    pDmaString_T   pRelId;        //リレーション識別子
    SDBR_PROPLIST  PropList;      //オブジェクトのプロパティリスト
    SDBR_PROPLIST  RelPropList;   //リレーションのプロパティリスト
};
```

詳細

文書間リレーションで関連づけられている文書の一覧情報を取得するメソッドです。

引数 lRelationendType に DBR_RELATIONEND_HEAD を設定することによって、このオブジェクトに

関連づけられているリレーション先文書の一覧情報を取得できます。また、lRelationendType に DBR_RELATIONEND_TAIL を設定することによって、このオブジェクトをリレーション先に持つリレーション元文書の一覧情報を取得できます。

初回取得時は、引数 pbContinue に DMA_FALSE を設定してメソッドをコールします。1回のメソッドのコールで一覧の取得が完了しなかった場合、引数 pbContinue に DMA_TRUE が設定されます。残りの要素は、引数 pbContinue に DMA_TRUE を設定して取得します。引数 pbContinue が DMA_FALSE になるまで、繰り返してこのメソッドをコールしてください。

リレーション先のオブジェクトが削除されていて、RemoveRelation メソッドで文書間リレーションが解除されないままの場合、リレーション情報は残っている状態になります。この場合も、引数 lRelationendStatus に DBR_RELATIONEND_STATUS_NOT_EXIST または DBR_RELATIONEND_STATUS_ALL を設定することによって、削除されたオブジェクトのリレーション情報を取得できます。オブジェクトが削除されていてリレーション情報だけが残っている場合、リレーション情報構造体である、SDBR_RELATIONINFO 構造体の内容は、pOIID = NULL、lType = DBR_CLASS_UNKNOWN、PropList.lCount = 0、PropList.pItem = NULL になり、pRelId と RelPropList だけに正当な値が設定されます。

なお、このメソッドで取得したリレーション識別子は、トランザクションが終了するまで有効とは限りません。ほかのトランザクションから削除されたり、リレーションのプロパティを変更されたりする可能性があります。

引数

pbContinue (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得した要素の残りを破棄して、新たに先頭から取得します。

なお、オブジェクトの生成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、DMA_FALSE が指定されたものとして動作します。

前回の読み出しと異なる引数を持つメソッドがコールされた場合は、pbContinue の値を無視し、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- DMA_TRUE
継続可能 (まだ取得していない残りの要素があります)
- DMA_FALSE
継続不可 (すべての要素を取得しました)

lClassType (入力)

指定されたクラス識別子の要素だけを取得します。ただし、関連づけの対象オブジェクトがすでに削除されていて、リレーションだけが残っているリレーション情報については lClassType の指定とは無関係にすべて取得されます。

指定できるクラス識別子を次に示します。

クラス識別子	クラスの種別
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件とした取得もできます。例えば、CdbrDocument クラスと CdbrVersionableDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONABLEDOCUMENT
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IPropDefListCount が 0 の場合だけ指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの個数 (引数 pRelPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pRelPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IRelPropDefListCount が 0 の場合だけ指定できます。IRelPropDefListCount が正の整数のときに pRelPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelationListCount (入力)

取得する文書間リレーション情報の数を 1 以上の値で指定します。0 以下の値を指定するとエラーになります。

ppRelationList (出力)

取得した文書間リレーション情報一覧へのポインタである、SDBR_RELATIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。**ppRelationList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してく

ださい。なお、**ppRelationList を指定してメモリを解放すると、構造体のメンバ (**ppRelationList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RELATIONLIST 構造体および SDBR_RELATIONINFO 構造体に設定される内容については、「付録 C.1(17) SDBR_RELATIONLIST 構造体 (リレーションリスト構造体)」および「付録 C.1(16) SDBR_RELATIONINFO 構造体 (リレーション情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IRelationendType (入力)

このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得するか、このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得するかを指定します。

次に示す値のどちらかを指定してください。

- DBR_RELATIONEND_HEAD
このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得する場合に指定します。
- DBR_RELATIONEND_TAIL
このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得する場合に指定します。

IRelationendStatus (入力)

関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得するか、すでに関連づけの対象オブジェクトが削除されていて、リレーションだけ残っているリレーション情報の一覧を取得するかを指定します。

次に示す値のどれかを指定してください。

- DBR_RELATIONEND_STATUS_EXIST
関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得する場合に指定します。
- DBR_RELATIONEND_STATUS_NOT_EXIST
すでに関連づけの対象オブジェクトが削除されていて、リレーションだけが残っているリレーション情報の一覧を取得する場合に指定します。
この場合、IClassType でのクラス種別の指定は無効になります。つまり、IClassType でクラス種別を指定しているかどうかに関係なく、関連づけの対象オブジェクトが削除されているリレーション情報はすべて取得されます。
- DBR_RELATIONEND_STATUS_ALL
すべてのリレーション情報の一覧を取得する場合に指定します。
この場合、関連づけの対象オブジェクトが存在しているリレーションに関しては、IClassType でのクラス種別の指定が有効です。ただし、関連づけの対象オブジェクトが削除されているリレーション情報は、IClassType でのクラス種別の指定とは無関係にすべて取得されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionableDocument::GetRelationListAndLock メソッド

機能

文書のリレーション情報の一覧取得と同時に指定したロックの設定

形式

```
DmaBoolean GetRelationListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lRelPropDefListCount,
    SDBR_PROPDEF*       pRelPropDefList,
    DmaInteger32         lRelationListCount,
    SDBR_RELATIONLIST** ppRelationList,
    DmaInteger32         lRelationendType,
    DmaInteger32         lRelationendStatus,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          //プロパティ識別子
    DmaInteger32  lType;           //データ型
    DmaInteger32  lCardinality;    //基本単位
    DmaInteger32  lCount;         //値の個数
    union {
        pDmaBoolean pbBoolean;    //Boolean型
        pDmaInteger32 plInteger32; //Integer32型
        pDmaId       pId;         //ID型
        pDmapv       ppObject;    //Object型
        ppDmaString_T ppString;   //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;          //プロパティ情報の個数
    SDBR_PROP*   pItem;          //プロパティ情報
};

// リレーションリスト構造体
struct SDBR_RELATIONLIST {
    DmaInteger32  lCount; //リレーションしている数
    SDBR_RELATIONINFO* pItem; //リレーション情報構造体へのポインタ
};

// リレーション情報構造体
struct SDBR_RELATIONINFO {
    pDmaString_T pOIID;          //文書オブジェクトのOIID
    DmaInteger32 lType;          //オブジェクトの種別
    pDmaString_T pRelId;        //リレーション識別子
    SDBR_PROPLIST PropList;     //オブジェクトのプロパティリスト
    SDBR_PROPLIST RelPropList; //リレーションのプロパティリスト
};
```

詳細

ロックタイプを明示的に指定して、文書間リレーションで関連づけられている文書の一覧情報を取得するメソッドです。

引数 `lRelationendType` に `DBR_RELATIONEND_HEAD` を設定することによって、このオブジェクトに関連づけられているリレーション先文書の一覧情報を取得します。また、`lRelationendType` に `DBR_RELATIONEND_TAIL` を設定することによって、このオブジェクトをリレーション先に持つリレーション元文書の一覧情報を取得できます。

初回取得時は、引数 `pbContinue` に `DMA_FALSE` を設定してメソッドをコールします。1回のメソッドのコールで一覧の取得が完了しなかった場合、引数 `pbContinue` に `DMA_TRUE` が設定されます。残りの要素は、引数 `pbContinue` に `DMA_TRUE` を設定して取得します。引数 `pbContinue` が `DMA_FALSE` になるまで、繰り返してこのメソッドをコールしてください。

リレーション先のオブジェクトが削除されていて、`RemoveRelation` メソッドで文書間リレーションが解除されないままの場合、リレーション情報は残っている状態になります。この場合も、引数 `lRelationendStatus` に `DBR_RELATIONEND_STATUS_NOT_EXIST` または `DBR_RELATIONEND_STATUS_ALL` を設定することによって、削除されたオブジェクトのリレーション情報を取得できます。オブジェクトが削除されていてリレーション情報だけが残っている場合、リレーション情報構造体である、`SDBR_RELATIONINFO` 構造体の内容は、`pOIID = NULL`、`lType = DBR_CLASS_UNKNOWN`、`PropList.lCount = 0`、`PropList.pItem = NULL` になり、`pRelId` と `RelPropList` だけに正当な値が設定されます。

なお、このメソッドで取得したリレーション識別子は、トランザクションが終了するまで有効とは限りません。ほかのトランザクションから削除されたり、リレーションのプロパティを変更されたりする可能性があります。

引数

`pbContinue` (入力/出力)

リストの取得方法を示す値を格納する領域を指定します。

`DMA_TRUE` を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。`DMA_FALSE` または `DMA_UNKNOWN` を指定すると、前回取得した要素の残りを破棄して、新たに先頭から取得します。

なお、オブジェクトの生成後初めてメソッドをコールした場合、または前回の読み出しですべて取得した場合は、`DMA_FALSE` が指定されたものとして動作します。

前回の読み出しと異なる引数を持つメソッドがコールされた場合は、`pbContinue` の値無視し、前回取得の残りを破棄して新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す次の値が設定されます。

- `DMA_TRUE`
継続可能 (まだ取得していない残りの要素があります)
- `DMA_FALSE`
継続不可 (すべての要素を取得しました)

`lClassType` (入力)

指定されたクラス識別子の要素だけを取得します。ただし、関連づけの対象オブジェクトがすでに削除されていて、リレーションだけが残っているリレーション情報については `lClassType` の指定とは無関係にす

べて取得されます。

指定できるクラス識別子を次に示します。

クラス識別子	クラスの種別
DBR_CLASS_VERSIONABLEDOCUMENT	CdbVersionableDocument クラス
DBR_CLASS_DOCUMENT	CdbDocument クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件とした取得もできます。例えば、CdbDocument クラスと CdbVersionableDocument クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONABLEDOCUMENT
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

IPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの個数 (引数 pPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IPropDefListCount が 0 の場合だけ指定できます。IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelPropDefListCount (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの個数 (引数 pRelPropDefList に指定する配列の要素数) を 0 以上の値で指定します。プロパティを取得しない場合は 0 を指定します。負の値を指定した場合は、エラーになります。

pRelPropDefList (入力)

文書間リレーション情報の一覧と同時に取得するリレーションのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。NULL は、IRelPropDefListCount が 0 の場合だけ指定できます。IRelPropDefListCount が正の整数のときに pRelPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IRelationListCount (入力)

取得する文書間リレーション情報の数を 1 以上の値で指定します。0 以下の値を指定するとエラーになります。

ppRelationList (出力)

取得した文書間リレーション情報一覧へのポインタである, SDBR_RELATIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。**ppRelationList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、**ppRelationList を指定してメモリを解放すると、構造体のメンバ (**ppRelationList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RELATIONLIST 構造体および SDBR_RELATIONINFO 構造体に設定される内容については、「付録 C.1(17) SDBR_RELATIONLIST 構造体 (リレーションリスト構造体)」および「付録 C.1(16) SDBR_RELATIONINFO 構造体 (リレーション情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IRelationendType (入力)

このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得するか、このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得するかを指定します。

次に示す値のどちらかを指定してください。

- DBR_RELATIONEND_HEAD
このオブジェクトが文書間リレーションに関連づけているリレーション先オブジェクトの一覧を取得する場合に指定します。
- DBR_RELATIONEND_TAIL
このオブジェクトが文書間リレーションに関連づけられているリレーション元オブジェクトの一覧を取得する場合に指定します。

IRelationendStatus (入力)

関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得するか、すでに関連づけの対象オブジェクトが削除されていて、リレーションだけ残っているリレーション情報の一覧を取得するかを指定します。

次に示す値のどれかを指定してください。

- DBR_RELATIONEND_STATUS_EXIST
関連づけの対象オブジェクトが存在しているリレーション情報の一覧を取得する場合に指定します。
- DBR_RELATIONEND_STATUS_NOT_EXIST
すでに関連づけの対象オブジェクトが削除されていて、リレーションだけが残っているリレーション情報の一覧を取得する場合に指定します。
この場合、IClassType でのクラス種別の指定は無効になります。つまり、IClassType でクラス種別を指定しているかどうかに関係なく、関連づけの対象オブジェクトが削除されているリレーション情報はすべて取得されます。
- DBR_RELATIONEND_STATUS_ALL
すべてのリレーション情報の一覧を取得する場合に指定します。
この場合、関連づけの対象オブジェクトが存在しているリレーションに関しては、IClassType でのクラス種別の指定が有効です。ただし、関連づけの対象オブジェクトが削除されているリレーション情報は、IClassType でのクラス種別の指定とは無関係にすべて取得されます。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionableDocument::GetRenditionList メソッド

機能

レンディション一覧の取得

形式

```
DmaBoolean GetRenditionList(
    pDmaString_T          pVersionId,
    DmaInteger32          lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_RENDITIONLIST** ppRenditionList )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;        //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*   pItem;        //プロパティ情報
};

// レンディションリスト構造体
struct SDBR_RENDITIONLIST {
    DmaInteger32 lCount;        //レンディション情報の数
    SDBR_RENDITIONINFO* pItem; //レンディション情報構造体
};

// レンディション情報構造体
struct SDBR_RENDITIONINFO {
    pDmaString_T pRenditionType; //RenditionTypeを表す文字列
    SDBR_PROPLIST PropList;      //プロパティリスト構造体
};
```

詳細

文書の指定したバージョンに登録されているレンディションの一覧情報とそのプロパティを取得するメソッドです。

マルチレンディション文書の場合、マスタレンディションの情報は、レンディション情報構造体である、SDBR_RENDITIONINFO 構造体の配列要素の 1 番目に格納され、サブレンディションの情報は 2 番目以降に格納されます。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド, CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合, カレントバージョンがレンディション一覧の取得対象になります。

IPropDefListCount (入力)

レンディションの一覧と同時に取得するプロパティの数 (pPropDefList に指定する配列の要素数) を指定します。0 を指定した場合は, プロパティは取得されません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

レンディションの一覧と同時に取得するプロパティの定義情報である, SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定してください。IPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName
レンディションの RetrievalName を取得する場合に指定します。
- dbrProp_RenditionStatus
レンディションの RenditionStatus を取得する場合に指定します。なお, 設定される RenditionStatus の値については, 「CdbDocument クラスの詳細」の「RenditionStatus プロパティの値」を参照してください。
- dbrProp_ContentType
レンディションのコンテンツ種別を取得する場合に指定します。なお, 設定される ContentType の値については, 「CdbDocument クラスの詳細」の「ContentType プロパティの値とコンテンツ種別の対応」を参照してください。

SDBR_PROPDEF 構造体については, 「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppRenditionList (出力)

取得したレンディション一覧へのポインタである, SDBR_RENDITIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。*ppRenditionList がポイントするメモリは, ライブラリが新たに確保したものです。このメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。なお, *ppRenditionList を指定してメモリを解放すると, 構造体のメンバ (*ppRenditionList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RENDITIONLIST 構造体および SDBR_RENDITIONINFO 構造体に設定される内容については, 「付録 C.1(20) SDBR_RENDITIONLIST 構造体 (レンディションリスト構造体)」および「付録 C.1(19) SDBR_RENDITIONINFO 構造体 (レンディション情報構造体)」を参照してください。

また, プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数 のメソッド処理要求が出されて います。	一つのセッションに対しては、 一度に一つの処理要求しか出さ ないように見直してください。
ERR_DBR	ERR_NONEXISTS_VERSIO N	指定されたバージョンは存在し ません。	正しいバージョンを指定してく ださい。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールし てから、このメソッドをコール してください。
ERR_DBR	ERR_OBJECT_NOT_CONN ECT	オブジェクトは接続されていま せん。	オブジェクトを接続してくださ い。
ERR_DBR	ERR_SESSION_NOT_CON NECT	セッションが接続されていませ ん。	セッションを接続してくださ い。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してくださ い。
ERR_DMA	DMARC_BAD_PARAMETE R	不正な引数が入力されました。	正しい引数を指定してくださ い。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正で す。	正しいプロパティ識別子を指定 してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。ま たはメモリを大量に消費してい るアプリケーションを停止させ て、処理を再実行してください。 い。
ERR_DMA	DMARC_NO_RESERVATIO N	バージョン識別子が予約されて いません。	チェックイン後に割り当てられ る正式なバージョン識別子を、 入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりませ ん。	正しいオブジェクトを設定して ください。
ERR_DMA	DMARC_NOT_SUPPORTE D	このセッションまたはオブジェ クトまたはクラスではサポート されていないメソッドです。	この操作は実行できません。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbVersionableDocument::GetRenditionListAndLock メソッド

機能

レンディション一覧の取得と同時に指定したロックの設定

形式

```
DmaBoolean GetRenditionListAndLock(
    pDmaString_T          pVersionId,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*        pPropDefList,
    SDBR_RENDITIONLIST** ppRenditionList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE )

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId      PropId;          //プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;         //データ型
    DmaInteger32 lCardinality;  //基本単位
    DmaInteger32 lCount;       //値の個数
    union {
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;        //プロパティ情報の個数
    SDBR_PROP*   pItem;        //プロパティ情報
};

// レンディションリスト構造体
struct SDBR_RENDITIONLIST {
    DmaInteger32 lCount;        //レンディション情報の数
    SDBR_RENDITIONINFO* pItem; //レンディション情報構造体
};

// レンディション情報構造体
struct SDBR_RENDITIONINFO {
    pDmaString_T pRenditionType; //RenditionTypeを表す文字列
    SDBR_PROPLIST PropList;      //プロパティリスト構造体
};
```

詳細

ロックタイプを明示的に指定して、文書の指定したバージョンに登録されているレンディションの一覧情報とそのプロパティを取得するメソッドです。

マルチレンディション文書の場合、マスタレンディションの情報は、レンディション情報構造体である、SDBR_RENDITIONINFO 構造体の配列要素の 1 番目に格納され、サブレンディションの情報は 2 番目以降に格納されます。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド, CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合, カレントバージョンがレンディション一覧の取得対象になります。

IPropDefListCount (入力)

レンディションの一覧と同時に取得するプロパティの数 (pPropDefList に指定する配列の要素数) を指定します。0 を指定した場合は, プロパティは取得されません。負の値を指定した場合はエラーになります。

pPropDefList (入力)

レンディションの一覧と同時に取得するプロパティの定義情報である, SDBR_PROPDEF 構造体を指定します。プロパティを取得しない場合は NULL を指定してください。IPropDefListCount に正の値を指定して pPropDefList に NULL を指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName
レンディションの RetrievalName を取得する場合に指定します。
- dbrProp_RenditionStatus
レンディションの RenditionStatus を取得する場合に指定します。なお, 設定される RenditionStatus の値については, 「CdbDocument クラスの詳細」の「RenditionStatus プロパティの値」を参照してください。
- dbrProp_ContentType
レンディションのコンテンツ種別を取得する場合に指定します。なお, 設定される ContentType の値については, 「CdbDocument クラスの詳細」の「ContentType プロパティの値とコンテンツ種別の対応」を参照してください。

SDBR_PROPDEF 構造体については, 「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppRenditionList (出力)

取得したレンディション一覧へのポインタである, SDBR_RENDITIONLIST 構造体が設定されます。ポインタを格納する領域を指定してください。*ppRenditionList がポイントするメモリは, ライブラリが新たに確保したものです。このメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。なお, *ppRenditionList を指定してメモリを解放すると, 構造体のメンバ (*ppRenditionList)->pItem およびその先に確保されているメモリはすべて解放されます。したがってメソッドをコールした側がこれらのメンバのメモリを個別に解放する必要はありません。

SDBR_RENDITIONLIST 構造体および SDBR_RENDITIONINFO 構造体に設定される内容については, 「付録 C.1(20) SDBR_RENDITIONLIST 構造体 (レンディションリスト構造体)」および「付録 C.1(19) SDBR_RENDITIONINFO 構造体 (レンディション情報構造体)」を参照してください。

また, プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ILockType (入力)

ロックの種別を指定します。次のどちらかの値を指定してください。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbVersionableDocument::PutRelationPropertyValues メソッド

機能

文書間リレーションのプロパティの設定

形式

```
DmaBoolean PutRelationPropertyValues (
    pDmaString_T          pRelId,
    SDBR_PROPLIST*       pRelPropList)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           //プロパティ識別子
    DmaInteger32         lType;           //データ型
    DmaInteger32         lCardinality;    //基本単位
    DmaInteger32         lCount;         //値の個数
    union {
        pDmaBoolean      pbBoolean;      //Boolean型
        pDmaInteger32    plInteger32;     //Integer32型
        pDmaId           pId;            //ID型
        pDmapv           ppObject;       //Object型
        ppDmaString_T    ppString;       //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;         //プロパティ情報の個数
    SDBR_PROP*          pItem;         //プロパティ情報
};
```

詳細

リレーション識別子を指定して、文書間リレーションのプロパティを設定するメソッドです。

引数

pRelId (入力)

プロパティを設定する文書間リレーションのリレーション識別子を指定します。

指定するリレーション識別子は、CreateRelation メソッドの引数 ppRelId で返却される値です。また、GetRelationList メソッドまたは GetRelationListAndLock メソッドを使用しても取得できます。

pRelPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。このメソッドで値を設定できる文書間リレーションのプロパティは、ユーザ設定のプロパティだけです。

NULL を指定するとエラーになります。なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

16. CdbVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionableDocument::PutRenditionPropertyValues メソッド

機能

レンディションのプロパティの設定

形式

```
DmaBoolean PutRenditionPropertyValues(
    pDmaString_T      pVersionId,
    pDmaString_T      pRenditionType,
    SDBR_PROPLIST*    pPropList )

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;          //プロパティ識別子
    DmaInteger32 lType;        //データ型
    DmaInteger32 lCardinality; //基本単位
    DmaInteger32 lCount;       //値の個数
    union {                    //プロパティの値
        pDmaBoolean      pbBoolean;    //Boolean型
        pDmaInteger32    plInteger32;   //Integer32型
        pDmaId           pId;          //ID型
        pDmapv           ppObject;     //Object型
        ppDmaString_T    ppString;     //String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;          //プロパティ情報の個数
    SDBR_PROP*   pItem;         //プロパティ情報
};
```

詳細

サブレンディションのプロパティを設定するメソッドです。

マルチレンディション文書のサブレンディションのプロパティ (RenditionType 以外) を設定, 変更する場合, このメソッドを使用します。

なお, RenditionType を変更する場合は,

CdbVersionableDocument::UpdateContentAndRenditionType メソッドを使用してください。

引数

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。バージョン識別子は CdbVersionable::GetVersionList メソッド, CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合, カレントバージョンがレンディションのプロパティ設定の対象になります。

pRenditionType (入力)

プロパティを設定するレンディションの RenditionType を表す文字列を指定します。この引数で指定した RenditionType が対象バージョンのレンディションとして存在しない場合はエラーになります。NULL を指定した場合, マスタレンディションが処理対象になります。空文字列 ("") は指定できません。

pPropList (入力)

設定するプロパティを格納したプロパティリスト構造体である、SDBR_PROPLIST 構造体を指定します。NULL を指定するとエラーになります。

なお、SDBR_PROPLIST 構造体に、同じプロパティ識別子を指定した SDBR_PROP 構造体を複数指定した場合はエラーになります。

指定できるプロパティ識別子を次に示します。

- dbrProp_RetrievalName
RetrievalName プロパティを設定する場合に指定します。
- dbrProp_RenditionStatus
RenditionStatus プロパティを設定する場合に指定します。なお、RenditionStatus プロパティの値としてユーザが設定できるのは上位 2 バイトの変換フラグだけで、下位 2 バイトの状態フラグは設定できません。ユーザが設定できる変換フラグの値を次に示します。

DBR_RENDSTATUS_CONVERT_REQUIRED

レンディション変換機能の対象になる状態、つまり、レンディション変換機能によって、マスタレンディションの内容から、コンテンツを自動更新できる状態です。DocumentBroker Rendering Option が提供するコマンドを実行することで、コンテンツが自動更新されます。サブレンディションのコンテンツ登録後にマスタレンディションを更新した場合のような、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のときに、レンディション変換機能を使ってサブレンディションを更新したい場合に指定します。ただし、レンディション変換機能を使うには、DocumentBroker Rendering Option が必要です。

DBR_RENDSTATUS_CONVERT_NOTREQUIRED

レンディション変換機能の対象にならない状態です。DocumentBroker Rendering Option が提供するコマンドを実行しても、コンテンツは更新されません。ただし、変換フラグの値の設定に関しては、メソッド実行時の RenditionStatus プロパティの値によって、次に示す制限があります。

メソッド実行時の RenditionStatus プロパティの値	変換フラグの値の設定に関する制限の内容
状態フラグが DBR_RENDSTATUS_NO_SUBREND の場合	DBR_RENDSTATUS_CONVERT_NOTREQUIRED を設定できません。
変換フラグが DBR_RENDSTATUS_CONVERT_ERROR の場合	DBR_RENDSTATUS_CONVERT_REQUIRED または DBR_RENDSTATUS_CONVERT_NOTREQUIRED に変更できません。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体に設定する内容については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定されたレンディションは存在しません。	正しい RenditionType を指定してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrVersionableDocument::RemoveIndex メソッド

機能

全文検索インデックスの削除

形式

```
DmaBoolean      RemoveIndex(
    const pDmaString_T pVersionId = NULL)
```

詳細

CdbrVersionableDocument オブジェクトに作成されている全文検索インデックスだけを削除します。インデックスが作成されていない CdbrVersionableDocument オブジェクトに対してこのメソッドをコールした場合、何もしないで正常終了します。ただし、全文検索機能付き文書クラス以外のクラスから作成された DMA オブジェクトの DocVersion オブジェクトを構成要素とする CdbrDocument オブジェクトを対象にこのメソッドを実行した場合は、エラーになります。

引数

pVersionId (入力)

対象になるバージョン識別子を指定します。バージョン識別子は、CdbrVersionable::GetVersionList メソッド CdbrVersionable::GetVersionListAndLock メソッドまたは CdbrVersionable::VersionCheckOut メソッドによって取得します。NULL を指定した場合は、カレントバージョンが対象になります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrVersionableDocument::RemoveRelation メソッド

機能

文書間リレーションの解除

形式

```
DmaBoolean RemoveRelation(
    DmaInteger32      lRelIdCount,
    ppDmaString_T    ppRelIds)
```

詳細

リレーションで結びつけられた文書間の関連づけを解除するメソッドです。

引数 ppRelIds のポインタ配列で指定された文書間のリレーションをすべて解除します。

リレーション識別子で示されるリレーション先のオブジェクトを削除しても、このメソッドで文書間リレーションを解除しないかぎり、リレーション情報は削除されないまま残ります。不要なリレーション情報を削除する場合にも、このメソッドをコールしてください。

引数

lRelIdCount (入力)

解除する文書間リレーションの個数を指定します。0 以下の値を指定した場合はエラーになります。0 より大きい値または次に示す値を指定してください。

- DBR_RELATION_ALL
このオブジェクトがリレーション元である文書間リレーションをすべて解除します。
- DBR_RELATION_END_NOT_EXIST
リレーション先のオブジェクトが削除されているリレーション情報をすべて削除します。

ppRelIds (入力)

解除する文書間リレーションのリレーション識別子のポインタ配列を指定します。配列の要素で示される文書間リレーションが解除 (リレーション情報が削除) されます。

lRelIdCount が 1 以上の場合に、NULL を指定するとエラーになります。

指定するリレーション識別子は、CreateRelation メソッドの引数 ppRelId で返却される値です。また、GetRelationList メソッドまたは GetRelationListAndLock メソッドを使用しても取得できます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトまたはクラスではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionableDocument::UpdateContent メソッド

機能

ファイルの更新

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの更新)
DmaBoolean UpdateContent(
    const pDmaString_T    pFilePath,
    const pDmaString_T    pVersionId = NULL,
    DmaInteger32          lCreateIndexFlg = DBR_CREATE_INDEX)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの更新)
DmaBoolean UpdateContent(
    const pDmaString_T    pFilePath,
    pDmaString_T          pRenditionType,
    const pDmaString_T    pVersionId,
    DmaInteger32          lCreateIndexFlg)

// マルチファイル文書の場合
DmaBoolean UpdateContent(
    SDBR_PATHLIST&        PathList,
    pDmaString_T          pRenditionType,
    const pDmaString_T    pVersionId)

// File Link文書の場合
DmaBoolean UpdateContent(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T          pRenditionType,
    const pDmaString_T    pVersionId)

// ファイルパス情報構造体
struct SDBR_PATHINFO {
    pDmaString_T    pFilePath;           // ファイルのURL
    pDmaString_T    pComponentType;     // ファイルを識別する文字列
};

// ファイルパス情報リスト構造体
struct SDBR_PATHLIST {
    DmaInteger32    lCount;              // 登録ファイル数
    SDBR_PATHINFO* pItem;              // ファイルパス情報構造体のポインタ
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T    pFilePath;           // 登録ファイルまたはディレクトリのURL
    pDmaString_T    pFamName;           // FAM名
    pDmaString_T    pFamFilePath;       // コンテンツ保管先パス
    DmaInteger32    lContentDeleteMode; // コンテンツ削除モード
    DmaInteger32    lFileNamingMode;    // コンテンツ名自動
                                           // 生成モード
};
```

詳細

登録されている CdbVersionableDocument オブジェクトの指定したバージョンの内容を、指定したパスのファイルで更新するメソッドです。

このメソッドは、リファレンスファイルを管理する文書に対しては実行できません。

Windows の場合、次のことに注意してください。

- GetContentAndLock メソッドによるコンテンツの取得時、大文字と小文字を区別しないと同一になる

ファイル名がある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAA.txt」は、あとから取得された「aaa.txt」によって上書きされます。

- GetContentAndLock メソッドによるコンテンツの取得時、ロングファイル名とショートファイル名の関係で同一のファイル名とみなされるファイルがある場合、先に取得したファイルが上書きされます。例えば、先に取得された「AAAAAAAAA.txt」は、あとから取得された「AAAAAAA~1.txt」によって上書きされます。

シングルファイル文書の場合 (マスタレンディションのファイルの更新)

シングルファイル文書 (マルチレンディション文書の場合はマスタレンディションであるシングルファイル文書) を更新する場合は、更新に使用するファイルのパス、対象バージョンおよび全文検索インデクスを作成するかどうかを指定します。

内容を更新する CdbVersionableDocument オブジェクト (マルチレンディション文書の場合はマスタレンディション) が次の条件に当てはまる場合に、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデクスを作成できます。

- 全文検索機能付き文書クラスを基に作成された DocVersion オブジェクトを構成要素とする CdbVersionableDocument オブジェクトである。
- RenditionType がテキスト型 (MIME::text/ で始まる MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

更新が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合、"/tmp/sample.doc" で更新すると、"sample.doc" が設定されます。Windows の場合、"c:¥temp¥sample.doc" で更新すると、"sample.doc" が設定されます。

UNIX の場合に、チェックアウトした文書に登録されたコンテンツの内容をファイル "/tmp/sample.doc" で更新する例を次に示します。Windows の場合は、ファイル "c:¥temp¥sample.doc" で更新する例として、例中の "file:///tmp/sample.doc" を "file:///c:¥temp¥sample.doc" と読み替えてください。なお、この例で更新できるのは、マスタレンディションのコンテンツです。

チェックアウトした文書の内容 (コンテンツ) を更新する例

```
pDmaString_T pReservedVerId;
pDbrVerDoc->VersionCheckOut (&pReservedVerId);
pDbrVerDoc->UpdateContent ("file:///tmp/sample.doc" ,
                          pReservedVerId, DBR_CREATE_INDEX);
pDbrVerDoc->VersionCheckIn ();
dbrDelete (pReservedVerId);
```

シングルファイル文書の場合 (RenditionType を指定したファイルの更新)

マルチレンディション文書のサブレンディションであるシングルファイル文書を更新する場合は、更新に使用するファイルのパス、対象バージョンのほかに、更新対象のサブレンディションを pRenditionType で指定します。なお、サブレンディションの内容を更新する場合、全文検索インデクスは作成できません。

マルチファイル文書の場合

マルチファイル文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_PATHLIST 構造体を指定します。

マルチファイル文書の場合、次のことに注意してください。

- CdbVersionableDocument クラスに格納した複数のファイルを、SDBR_PATHINFO 構造体に設定したファイルパスのファイルで、一括して更新します。SDBR_PATHINFO 構造体は、引数 PathList で指定する SDBR_PATHINFO 構造体に設定します。なお、特定のコンテンツだけを更新することはできません。また、更新後は、更新前のファイルは使用できません。
- 更新前のファイルを格納した BLOB のデータが書き換えられるため、BLOB のデータの長さは、更新後のファイルサイズの合計になります。

- この形式の UpdateContent を、マルチファイル文書ではないバージョン付き文書に対して実行すると、ERR_NOT_MULTI_CONTENTS のエラーになります。
- マルチファイル文書では、テキストファイルの部分が特定できないため、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、マルチファイル文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

File Link 文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、更新に使用するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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でコンテンツを更新できます。また、すでに HiRDB File Link で管理されているデータでコンテンツを更新することもできます。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効です。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。
- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

引数

PathList (入力)

登録するファイルのファイルパス情報が設定された SDBR_PATHLIST 構造体を指定します。SDBR_PATHLIST 構造体については、「付録 C.1(11) SDBR_PATHLIST 構造体 (ファイルパス情報リスト構造体)」を参照してください。

pFilePath (入力)

登録するファイルのパス名を指定します。ファイルのパス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" を登録するには、"file:///c:\temp\sample.doc" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

なお、ここで指定するパスは、GetContent メソッドで取得したファイルである必要はありません。ただし、RenditionType が異なるファイルは指定できません。

pRenditionType (入力)

RenditionType を指定してレンディションの内容を更新する場合、内容を更新するレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象バージョンのサブレンディションとして存在しない場合はエラーになります。

マスタレンディションの内容を更新する場合は NULL を指定するか、「マスタレンディションのファイルを更新する場合」の形式を使用することもできます。なお、空文字列 ("") は指定できません。

マルチファイル文書の場合、NULL または「シングルファイル文書の場合 (マスタレンディションのファイルの更新)」の形式を指定してください。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。この識別子は、CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。省略した場合または NULL を指定した場合は、カレントバージョンが更新対象になります。

ICreateIndexFlg (入力)

全文検索インデックスを作成するかどうかを指定します。

次のどちらかを指定してください。この値を省略した場合は、DBR_CREATE_INDEX が指定されたものとして処理されます。

- DBR_CREATE_INDEX
pFilePath に指定したパス名のファイルから全文検索インデックスを作成します。
- DBR_NOT_CREATE_INDEX
全文検索インデックスを作成しません。

なお、全文検索機能付きクラス以外から作成された DocVersion オブジェクトを指定した場合およびファイルの RenditionType がテキスト型でない場合は、ICreateIndexFlg の値は無視されます。

また、マルチレンディション文書の場合、ICreateIndexFlg はマスタレンディションの内容を更新する場合だけ有効です。

FileLinkPath (入力)

File Link 文書を登録するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には、DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体に設定する内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。

16. CdbvVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。 または、マルチファイル文書で指定したファイルのサイズの合計が、BLOB の定義長を超えました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_BAD_CONTENTSCOUNT	登録しようとするファイルの数が多過ぎます。	登録するファイルの数を見直してください。
ERR_DBR	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。 一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_NOT_MULTI_CONTENTS	マルチファイル文書ではありません。	マルチファイル管理機能を使用しない場合の形式で実行してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

CdbVersionableDocument::UpdateContentAndRendition Type メソッド

機能

ファイルの更新および RenditionType の変更

形式

```
// シングルファイル文書の場合
// (マスタレンディションのファイルの更新)
DmaBoolean UpdateContentAndRenditionType(
    const pDmaString_T    pFilePath,
    pDmaString_T          pRenditionType,
    const pDmaString_T    pVersionId = NULL,
    DmaInteger32          lCreateIndexFlg = DBR_CREATE_INDEX)

// シングルファイル文書の場合
// (RenditionTypeを指定したファイルの更新)
DmaBoolean UpdateContentAndRenditionType(
    const pDmaString_T    pFilePath,
    pDmaString_T          pBeforeRenditionType,
    pDmaString_T          pAfterRenditionType,
    const pDmaString_T    pVersionId,
    DmaInteger32          lCreateIndexFlg)

// リファレンスファイル文書の場合
DmaBoolean UpdateContentAndRenditionType(
    SDBR_REFERENCE_PATHINFO& ReferencePath,
    pDmaString_T              pBeforeRenditionType,
    pDmaString_T              pAfterRenditionType,
    pDmaString_T              pVersionId)

// File Link文書の場合
DmaBoolean UpdateContentAndRenditionType(
    SDBR_FILELINK_PATHINFO& FileLinkPath,
    pDmaString_T              pBeforeRenditionType,
    pDmaString_T              pAfterRenditionType,
    const pDmaString_T        pVersionId)

// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32    lContentOperateMode;    // 操作モード
    pDmaString_T    pEntry;                 // ファイルのパスおよびロケーション
    pDmaString_T    pTargetPath;           // コンテンツ格納先パス
    pDmaString_T    pDeleteRootPath;      // 削除ディレクトリのルートパス
};

// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T    pFilePath;              // 登録ファイルまたはディレクトリのURL
    pDmaString_T    pFamName;              // FAM名
    pDmaString_T    pFamFilePath;          // コンテンツ保管先パス
    DmaInteger32    lContentDeleteMode;    // コンテンツ削除モード
    DmaInteger32    lFileNamingMode;       // コンテンツ名自動
                                                    // 生成モード
};
```

詳細

登録されている CdbVersionableDocument オブジェクトの指定したバージョンの内容を、指定したパスのファイルで更新するメソッドです。更新するときには、RenditionType も変更できます。

シングルファイル文書の場合 (マスタレンディションのファイルの更新)

シングルファイル文書 (マルチレンディション文書の場合はマスタレンディションであるシングル

ファイル文書)を更新する場合は、更新に使用するファイルのパス、対象バージョンおよび全文検索インデックスを作成するかどうかを指定します。RenditionType を変更する場合は、pRenditionType に変更後の RenditionType を指定します。

内容を更新する CdbVersionableDocument オブジェクト (マルチレンディション文書の場合はマスタレンディション) が次の条件に当てはまる場合に、lCreateIndexFlg に DBR_CREATE_INDEX を指定すると、全文検索インデックスを作成できます。

- 全文検索機能付き文書クラスを基に作成された DocVersion オブジェクトを構成要素とする CdbDocument オブジェクトである。
- RenditionType がテキスト型 (MIME::text/ で始まる MIME::text/plain, MIME::text/sgml または MIME::text/html) である。

更新が正常に完了すると、パス名を除いたファイル名が RetrievalName プロパティに設定されます。例えば、UNIX の場合に、"/tmp/sample.doc" で更新すると、"sample.doc" が設定されます。Windows の場合に、"c:\temp\sample.doc" で更新すると、"sample.doc" が設定されます。UNIX の場合に、チェックアウトした文書に登録されたコンテンツの内容をファイル "/tmp/sample.html" で更新する例を次に示します。Windows の場合は、ファイル "c:\temp\sample.html" で更新する例として、例中の "file:///tmp/sample.html" を "file:///c:\temp\sample.html" と読み替えてください。

チェックアウトした文書の内容 (コンテンツ) を更新して RenditionType を変更する例

```
pDmaString_T pReservedVerId;
pDbrVerDoc->VersionCheckOut (&pReservedVerId);
pDbrVerDoc->UpdateContentAndRenditionType (
    "file:///tmp/sample.html" ,pReservedVerId,
    "MIME::text/html",DBR_CREATE_INDEX);
pDbrVerDoc->VersionCheckIn ();
dbrDelete (pReservedVerId);
```

シングルファイル文書の場合 (RenditionType を指定したファイルの更新)

マルチレンディション文書のサブレンディションであるシングルファイル文書を更新する場合は、更新に使用するファイルのパス、対象バージョンのほかに、更新対象のサブレンディションを pBeforeRenditionType で指定します。RenditionType を変更する場合は、pAfterRenditionType に変更後の RenditionType を指定します。なお、サブレンディションの内容を更新する場合、全文検索インデックスは作成できません。

リファレンスファイル文書の場合

リファレンスファイル文書を更新する場合は、更新に使用するコンテンツの情報として、SDBR_REFERENCE_PATHINFO 構造体を指定します。

リファレンスファイル文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentReference オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_REFERENCE の場合) にだけ有効となります。そのほかのコンテンツ種別を持つレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。

File Link 文書の場合

File Link 文書を更新する場合は、更新に使用するコンテンツの情報として、

SDBR_FILELINK_PATHINFO 構造体を指定します。この構造体には、更新に使用するコンテンツとして、ファイルまたはディレクトリを指定できます。ディレクトリを指定した場合、指定した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でコンテンツを更新できます。また、すでに HiRDB File Link で管理されているデータでコンテンツを更新することもできます。

File Link 文書の場合、次のことに注意してください。

- このメソッドは、レンディションのコンテンツ種別が ContentFileLink オブジェクトの場合 (dbrProp_ContentType プロパティの値が DBR_CONTENTTYPE_FILELINK の場合) にだけ有効です。そのほかのコンテンツ種別を持つ文書のレンディションに対して実行すると、ERR_CONTENTTYPE_MISMATCH のエラーになります。
- このメソッドで、ERR_FILELINK_FAILED のエラーが発生した場合は、詳細メッセージに HiRDB File Link が出力するエラー情報が出力されます。HiRDB File Link が出力するエラーを参照して、対処してください。
- File Link 文書では、全文検索用インデクスは作成されません。全文検索インデクスを作成する場合、テキスト部分を抽出したファイルを、CreateIndex メソッドの引数に指定して実行してください。なお、File Link 文書の場合、CreateIndex メソッドの引数に NULL を指定すると ERR_CONTENTTYPE_MISMATCH のエラーになります。

引数

pFilePath (入力)

更新に使用するファイルのパス名を指定します。ファイルのパス名は URL 形式で設定します。例えば、UNIX の場合、ファイル "/tmp/sample.doc" を登録するには、"file:///tmp/sample.doc" と指定します。Windows の場合、ファイル "c:\temp\sample.doc" を登録するには、"file:///c:/temp/sample.doc" と指定します。有効なプロトコルは "file:" だけです。これ以外のプロトコルを指定した場合はエラーになります。

なお、ここで指定するパスは、GetContent メソッドで取得したファイルである必要はありません。ただし、RenditionType が異なるファイルは指定できません。

pRenditionType (入力)

文書 (マルチレンディション文書の場合はマスタレンディション) の内容を更新すると同時にその RenditionType を変更する場合、変更後の RenditionType を表す文字列を指定します。例えば HTML 文書の場合は、"MIME::text/html" などとします。NULL を指定した場合、RenditionType は変更されません。また、pFilePath に指定したファイルの拡張子から DocumentBroker が RenditionType を決定することはありません。

なお、マルチレンディション文書 (レンディションが複数ある文書) の場合、空文字列 ("") は指定できません。

pBeforeRenditionType (入力)

RenditionType を指定してレンディションの内容を更新する場合、更新対象のレンディションの RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象バージョンのサブレンディションとして存在しない場合はエラーになります。

マスタレンディションの内容を更新する場合は NULL を指定するか、「マスタレンディションのファイルを更新する場合」の形式を使用することもできます。

pAfterRenditionType (入力)

RenditionType を指定してレンディションの内容を更新すると同時にその RenditionType を変更する場合、変更後の RenditionType を表す文字列を指定します。

この引数で指定した RenditionType が、対象バージョンのサブレンディションとしてすでに登録されている場合はエラーになります。ただし、pBeforeRenditionType 値と pAfterRenditionType 値とが同一値で

あってもエラーになりません。NULL を指定した場合は、サブレンディションの RenditionType は変更されません。また、pFilePath または ReferencePath の pEntry メンバに指定したファイルの拡張子から DocumentBroker が RenditionType を決定することはありません。

なお、空文字列 ("") は指定できません。

pVersionId (入力)

対象となるバージョンのバージョン識別子を指定します。この識別子は、CdbVersionable::GetVersionList メソッド、CdbVersionable::GetVersionListAndLock メソッドまたは CdbVersionable::VersionCheckOut メソッドで取得します。省略した場合または NULL を指定した場合は、カレントバージョンが更新対象になります。

ICreateIndexFlg (入力)

全文検索インデックスを作成するかどうかを指定します。次のどちらかを指定してください。この値を省略した場合は、DBR_CREATE_INDEX が指定されたものとして処理されます。

- DBR_CREATE_INDEX
pFilePath に指定したパス名のファイルから全文検索インデックスが作成されます。
- DBR_NOT_CREATE_INDEX
全文検索インデックスが作成されません。

なお、CdbDocument オブジェクトの構成要素が全文検索機能付きクラス以外から作成された DocVersion オブジェクトである場合、およびファイルの RenditionType がテキスト型でない場合は、ICreateIndexFlg の値は無視されます。

また、マルチレンディション文書の場合、ICreateIndexFlg はマスタレンディションの内容を更新する場合だけ有効です。

ReferencePath (入力)

リファレンスファイル文書を更新するために必要な情報を格納した、SDBR_REFERENCE_PATHINFO 構造体を指定します。

コンテンツを持たないオブジェクトを更新する場合

SDBR_REFERENCE_PATHINFO 構造体のメンバ pTargetPath に、コンテンツ格納先パスを、コンテンツ格納先ベースパスを基点とした相対パスで指定します。ContentLocation プロパティに格納されるパスは、次のように条件によって異なります。

- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合 ContentLocation プロパティには、pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ、および pEntry で指定したファイル名が連結されて格納されます。
- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合 ContentLocation プロパティには、文書登録時に pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ、DocumentBroker が指定するファイル名が連結されて格納されます。

コンテンツを持つオブジェクトを更新する場合

ContentLocation プロパティに格納されるパスは、次のように条件によって異なります。

- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合 ContentLocation プロパティには、更新前の ContentLocation プロパティのファイル名を除くパスに、SDBR_REFERENCE_PATHINFO 構造体のメンバ pEntry で指定したファイル名が連結され

て格納されます。

- DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合 ContentLocation プロパティには、文書登録時に pTargetPath で指定したパス、DocumentBroker がコンテンツを管理するためのディレクトリ、DocumentBroker が指定するファイル名が連結されて格納されます。

なお、次に示すことに注意してください。

DocumentSpace 構成定義の ReferenceStorageMode エントリに Origin を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツを持たないオブジェクトにコンテンツを登録すると、コンテンツロケーションの指定で ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

また、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツ登録済みのオブジェクトのコンテンツを更新すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (35 バイト)
- pEntry で指定したファイル名

DocumentSpace 構成定義の ReferenceStorageMode エントリに Divide を指定した場合

次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツを持たないオブジェクトにコンテンツを登録すると、コンテンツロケーションの指定で ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したコンテンツ格納先パス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

また、次に示す文字列の合計長 (パスの区切り文字も含む) が、260 バイト (Windows の場合) または 1023 バイト (AIX の場合) 以上のときに、コンテンツ登録済みのオブジェクトのコンテンツを更新すると、ERR_FILE_REFERENCE_OPERATION_FAILED のエラーになります。

- CdbrSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパス
- 文書登録時に pTargetPath で指定したパス
- DocumentBroker がコンテンツを管理するためのディレクトリ (6 バイト)
- DocumentBroker が指定するファイル名 (最大 59 バイト)

pTargetPath で指定するコンテンツ格納先パスの中に、カレントディレクトリの一つ上位のディレクトリを示す ".." は指定できません。指定した場合、DMARC_BAD_PARAMETER のエラーになります。

リファレンスファイル文書を更新する場合、SDBR_REFERENCE_PATHINFO 構造体のメンバ pDeleteRootPath への値の指定は、無効になります。

SDBR_REFERENCE_PATHINFO 構造体に設定する内容については、「付録 C.1(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)」を参照してください。

FileLinkPath (入力)

File Link 文書を更新するために必要な情報を格納した、SDBR_FILELINK_PATHINFO 構造体を指定します。

なお、SDBR_FILELINK_PATHINFO 構造体のメンバ pFilePath とメンバ pFamFilePath の両方に NULL を設定した場合には、DMARC_BAD_PARAMETER のエラーになります。

SDBR_FILELINK_PATHINFO 構造体の指定内容については、「付録 C.1(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定され ます。	CORBA でエラーが発生しまし た。	ユーザプログラムでは対処でき ません。エラー情報として文字 列の値を取得して、ユーザプロ グラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCC URRED	データベースでデッドロックが 発生しました。	時間を置いて、再度実行してく ださい。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しま した。	syslog を参照して、障害の原 因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされてい ます。	時間を置いて、再度実行してく ださい。
ERR_DBR	ERR_ACCESS_NOT_PERM ITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INV ALID	クラスライブラリのオブジェク トは無効です。	有効なクラスライブラリのオブ ジェクトを指定してください。
ERR_DBR	ERR_CONTENTTYPE_MIS MATCH	指定したレンディションタイプ のコンテンツに対して実行でき ません。	正しい文書オブジェクトを指定 してメソッドを実行してくださ い。
ERR_DBR	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリ がロードできません。	詳細メッセージを基に対処して ください。
ERR_DBR	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発 生しました。	詳細メッセージを基に対処して ください。
ERR_DBR	ERR_FILE_REFERENCE_ MISMATCH_STATUS	リファレンスファイル管理機能 で扱うサーバマシンでのコンテ ントの操作でエラーが発生し、 オブジェクトとコンテンツが不 整合な状態となりました。	文書を削除したあと、更新後の コンテンツを使用して文書を作 成してください。
ERR_DBR	ERR_FILE_REFERENCE_ OPERATION_FAILED	リファレンスファイル管理機能 で扱うサーバマシンでのコンテ ントの操作でエラーが発生しま した。	詳細メッセージを基に対処して ください。

16. CdbrVersionableDocument クラス (バージョン付き文書を扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
ERR_DBR	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_REFERENCETYPE_MISMATCH	オブジェクト種別が一致していません。	オブジェクト種別または指定した操作モード (IContentOperateMode) を見直してください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DBR	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
ERR_DMA	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減ったあと、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
ERR_DMA	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
ERR_DMA	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。

17

CdbrVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

コンテナを使用して文書やコンテナのバージョン構成を管理する機能を提供するクラスです。また、バージョン管理しているコンテナ (CdbrConfiguratedReferentialContainer クラスのオブジェクト) の、個々のバージョンに当たるコンテナを操作する機能を提供します。

CdbrVersionTraceableContainer クラスの詳細

CdbrVersionTraceableContainer::~CdbrVersionTraceableContainer メソッド

CdbrVersionTraceableContainer::CdbrVersionTraceableContainer メソッド

CdbrVersionTraceableContainer::CreateObject メソッド

CdbrVersionTraceableContainer::GetVersionableList メソッド

CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド

CdbrVersionTraceableContainer::GetVTContaineelList メソッド

CdbrVersionTraceableContainer::GetVTContaineelListAndLock メソッド

CdbrVersionTraceableContainer::LinkVTFix メソッド

CdbrVersionTraceableContainer::LinkVTFixAndLock メソッド

CdbrVersionTraceableContainer::LinkVTFloat メソッド

CdbrVersionTraceableContainer::LinkVTFloatAndLock メソッド

CdbrVersionTraceableContainer::SetVTFix メソッド

CdbrVersionTraceableContainer::SetVTFixAndLock メソッド

CdbrVersionTraceableContainer::SetVTFloat メソッド

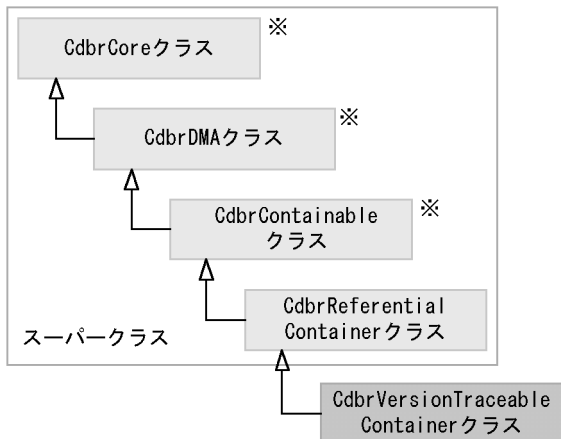
CdbrVersionTraceableContainer::SetVTFloatAndLock メソッド

CdbrVersionTraceableContainer::UnlinkVT メソッド

CdbVersionTraceableContainer::UnlinkVTAndLock メソッド

CdbVersionTraceableContainer クラスの詳細

スーパークラス



(凡例)

——▶ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

プロパティの一覧を次に示します。

表 17-1 CdbVersionTraceableContainer クラスのプロパティ

プロパティ名	プロパティ識別子	データ型	読み取り専用
OIID	dmaProp_OIID	String 型	
ParentCount	dbrProp_ParentCount	Integer32 型	
ContainersCount	dbrProp_ContainersCount	Integer32 型	
ContainersCountVT	dbrProp_ContainersCountVT	Integer32 型	
ChildrenCount	dbrProp_ChildrenCount	Integer32 型	
ContaineesCount	dbrProp_ContaineesCount	Integer32 型	
ContaineesCountVT	dbrProp_ContaineesCountVT	Integer32 型	

アクセス制御機能を使用している場合は、次のクラスライブラリの固有のプロパティが追加されます。

表 17-2 CdbVersionTraceableContainer クラスのプロパティ (アクセス制御機能を使用する場合に追加されるクラスライブラリ固有のプロパティ)

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
OwnerId	dbrProp_OwnerId	String 型	- ²
UserPermission	dbrProp_UserPermission	Integer32 型	
PrimaryGroupId	dbrProp_PrimaryGroupId	String 型	- ³
ACL	dbrProp_ACL	Object 型 (VariableArray 型)	- ³

プロパティ名	プロパティ識別子	データ型 ¹	読み取り専用
SACL	dbrProp_SACL	Object 型 (VariableArray 型)	- 2
OwnerPermission	dbrProp_OwnerPermission	Integer32 型	- 3
PrimaryGroupPermission	dbrProp_PrimaryGroupPermission	Integer32 型	- 3
EveryonePermission	dbrProp_EveryonePermission	Integer32 型	- 3
PublicACLCount	dbrProp_PublicACLCount	Integer32 型	
PublicACLIds	dbrProp_PublicACLIds	Object 型 (VariableArray 型)	- 3

注 1

基本単位が Scalar 型以外の場合は、() に基本単位を示します。

注 2

変更は、そのオブジェクトの所有者およびセキュリティ管理者だけが実行できます。

注 3

変更は、そのオブジェクトの所有者、セキュリティ ACL でアクセス制御情報変更権を与えられたユーザ、およびセキュリティ管理者だけが実行できます。

プロパティ詳細

DMA で規定されているプロパティ

- OIID

オブジェクトを識別する GUID 文字列です。実体は DMA オブジェクトの OIID です。

CdbrDMA クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- OwnerId
- UserPermission

これらのプロパティの詳細については、「CdbrDMA クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ

- ParentCount
- ContainersCount

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrContainable クラスから継承されるプロパティ (アクセス制御機能を使用する場合)

- PrimaryGroupId
- ACL
- SACL
- OwnerPermission
- PrimaryGroupPermission
- EveryonePermission
- PublicACLCount

- PublicACLIds

これらのプロパティの詳細については、「CdbrContainable クラスの詳細」のプロパティ詳細を参照してください。

CdbrReferentialContainer クラスから継承されるプロパティ

- ChildrenCount
- ContaineesCount

これらのプロパティの詳細については、「CdbrReferentialContainer クラスの詳細」のプロパティ詳細を参照してください。

CdbrVersionTraceableContainer クラスで導入されるプロパティ

- ContainersCountVT
このオブジェクトを構成要素として管理しているオブジェクトの個数を表します。
- ContaineesCountVT
構成管理の対象となるオブジェクト (CdbrVersionableDocument オブジェクトおよび CdbrConfiguredReferentialContainer オブジェクト) の個数を表します。

なおこのほかに、CdbrVersionTraceableContainer クラスでは、構成要素である DMA オブジェクトを作成する基になる DMA のクラスの edmClass_ContainerVersion クラスのプロパティが使用できます。DMA のクラスのプロパティについては、「付録 B DMA オブジェクトのプロパティ」を参照してください。

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrDMA クラスから継承されるメソッド

メソッド	機能
~CdbrDMA	デストラクタ
CdbrDMA	コンストラクタ
ConnectObject	オブジェクトを指定したモードでロックして接続
GetOIID	OIID の取得
GetPropertyValues	プロパティの取得
GetPropertyValuesAndLock	プロパティの取得と同時に指定したロックの設定
GetType	オブジェクトの種別の取得
PutPropertyValues	プロパティの設定

17. CdbrVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

メソッド	機能
ReleaseObject	オブジェクトとの接続の解除
RemoveObject	オブジェクトの削除
SetOIID	OIID の設定

これらのメソッドの詳細については、「7. CdbrDMA クラス (OIID の設定やプロパティを扱うクラス)」を参照してください。

CdbrContainable クラスから継承されるメソッド

メソッド	機能
~CdbrContainable	デストラクタ
BindPublicACL	パブリック ACL のバインド
CdbrContainable	コンストラクタ
GetContainerList	コンテナの一覧取得
GetContainerListAndLock	コンテナの一覧取得と同時に指定したロックの設定
GetPublicACLList	バインドしているパブリック ACL 一覧情報の取得
GetPublicACLListAndLock	バインドしているパブリック ACL 一覧情報の取得と同時に指定したロックの設定
UnbindPublicACL	パブリック ACL のアンバインド

これらのメソッドの詳細については、「5. CdbrContainable クラス (包含要素を扱うクラス)」を参照してください。

CdbrReferentialContainer クラスから継承されるメソッド

メソッド	機能
~CdbrReferentialContainer	デストラクタ
CdbrReferentialContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetContainableList	包括要素の一覧取得
GetContainableListAndLock	包括要素の一覧取得と同時に指定したロックの設定
GetLinkPropertyValues	リンクのプロパティの取得
GetLinkPropertyValuesAndLock	リンクのプロパティの取得と同時に指定したロックの設定
Link	関連づけの設定
LinkAndLock	関連づけの設定と同時に指定したロックの設定
PutLinkPropertyValues	リンクのプロパティの設定
Unlink	関連づけの解除
UnlinkAndLock	関連づけの解除と同時に指定したロックの設定

これらのメソッドの詳細については、「12. CdbrReferentialContainer クラス (バージョンなしコンテナを扱うクラス)」を参照してください。

CdbrVersionTraceableContainer クラスで導入されるメソッド

メソッド	機能
~CdbrVersionTraceableContainer	デストラクタ
CdbrVersionTraceableContainer	コンストラクタ
CreateObject	オブジェクトの新規作成
GetVersionableList	バージョン管理情報一覧の取得
GetVersionableListAndLock	バージョン管理情報一覧の取得と同時に指定したロックの設定
GetVTContaineeList	構成管理しているオブジェクトの一覧取得
GetVTContaineeListAndLock	構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定
LinkVTFix	構成管理するオブジェクトの FIX モードでの関連づけ
LinkVTFixAndLock	構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定
LinkVTFloat	構成管理するオブジェクトの FLOATING モードでの関連づけ
LinkVTFloatAndLock	構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定
SetVTFix	構成要素のバージョンの固定
SetVTFixAndLock	構成要素のバージョンの固定と同時に指定したロックの設定
SetVTFloat	構成要素のバージョン固定の解除
SetVTFloatAndLock	構成要素のバージョン固定の解除と同時に指定したロックの設定
UnlinkVT	構成管理しているオブジェクトとの関連づけの解除
UnlinkVTAndLock	構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

以降, CdbrVersionTraceableContainer クラスのメソッドについて説明します。

CdbVersionTraceableContainer::~~CdbVersionTraceableContainer メソッド

機能

デストラクタ

形式

```
virtual ~CdbVersionTraceableContainer()
```

詳細

CdbVersionTraceableContainer オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbVersionTraceableContainer::CdbVersionTraceableContainer メソッド

機能

コンストラクタ

形式

`CdbVersionTraceableContainer()`

詳細

`CdbVersionTraceableContainer` オブジェクトのコンストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrVersionTraceableContainer::CreateObject メソッド

機能

オブジェクトの新規作成

形式

```
CdbrVersionTraceableContainer::CreateObject (
    CdbrSession*      pDbrSession,
    DmaInteger32      lDMAInfoListCount,
    SDBR_DMAINFO*    pDMAInfoList,
    ppDmaString_T     ppOIID)

// プロパティ構造体
struct SDBR_PROP {
    DmaId      PropId;           // プロパティ識別子
    DmaInteger32 lType;         // データ型
    DmaInteger32 lCardinality;  // 基本単位
    DmaInteger32 lCount;       // 要素数
    union {                    // プロパティの値
        pDmaBoolean      pbBoolean; // Boolean型
        pDmaInteger32    plInteger32; // Integer32型
        pDmaId           pId;       // ID型
        pDmapv           ppObject;  // Object型
        ppDmaString_T    ppString;  // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32 lCount;
    SDBR_PROP*  pItem;
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAINFO {
    DmaId      ClassId;
    SDBR_PROPLIST PropList;
};
```

詳細

オブジェクトを新規に作成するメソッドです。

SetOIID メソッドもしくは ConnectObject メソッドによってオブジェクトとすでに接続されている場合、またはこのメソッドがすでにコールされている場合は、既存の接続は変更しないで、このメソッドのコールがエラーになります。接続を終了するには、ReleaseObject メソッドを実行してください。

引数

pDbrSession (入力)

CdbrSession オブジェクトを指定します。

IDMAInfoListCount (入力)

pDMAInfoList で指定した配列の個数を指定します。負の値を指定した場合は、エラーになります。

pDMAInfoList (入力)

オブジェクトの構成要素として新規に作成する DMA オブジェクトの情報を格納した構造体である、

SDBR_DMAINFO 構造体を指定します。DMA オブジェクトの情報として SDBR_DMAINFO 構造体には、edmClass_ContainerVersion クラスまたはそのサブクラスの情報を指定してください。NULL を指定した場合は、edmClass_ContainerVersion クラスを使用して DMA オブジェクトが作成されます。

IDMAInfoListCount が正の整数のときに pDMAInfoList に NULL を指定すると、エラーになります。

SDBR_DMAINFO 構造体については、「付録 C.1(3) SDBR_DMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

プロパティ情報を指定する SDBR_PROPLIST 構造体および SDBR_PROP については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお、VariableArray 型プロパティを設定するとき、次の場合はエラーになります。このとき、major_code として ERR_DMA、minor_code として DMARC_BAD_VALUE が返却されます。

- DocumentBroker サーバのメタ情報ファイル (edms.ini) の dmaProp_MaximumElements エントリの値に 2 以上の値を設定しているときに、このエントリの値を超えて VariableArray 型プロパティの要素を指定した場合
- DocumentSpace 構成定義ファイル docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義があるときに、VariableArray 型プロパティの要素の値として、範囲外の値や最大値を超えた値を指定した場合

それぞれのデータ型に対応する値の範囲については、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。

ppOID (出力)

作成した CdbVersionTraceableContainer オブジェクトの OIID へのポインタが設定されます。なお、*ppOID がポイントするメモリは、ユーザプログラムでは解放しないでください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。
ERR_DBR	ERR_INVALID_ACE	不正なアクセス制御エレメント (ACE) が指定されました。	ACEとして正しい CdbCompound オブジェクトを指定してください。
ERR_DBR	ERR_INVALID_ACE_COUNT	設定するアクセス制御エレメント (ACE) の個数が制限値を超えています。	ACEは64個以内で指定してください。
ERR_DBR	ERR_INVALID_PUBLIC_ACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACLは10個以内で指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常254文字ですが、DocumentBrokerの環境設定時に512文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常254文字ですが、DocumentBrokerの環境設定時に512文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
ERR_DBR	ERR_NONEXISTS_PUBLIC_ACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
ERR_DBR	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	次のどれかの要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	次のどれかの方法で対処してください。 1. プロパティやリストの値の範囲を見直してください。 2. VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 3. VariableArray 型プロパティの要素の値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切でない値を指定しています。	正しい型を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::GetVersionableList メソッド

機能

バージョン管理情報一覧の取得

形式

```
// バージョン識別子なしで取得する場合
DmaBoolean GetVersionableList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList)

// バージョン識別子付きで取得する場合
DmaBoolean GetVersionableList(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VROBJLIST**   ppVrObjList)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;           // 要素数
    union {                // プロパティの値
        pDmaBoolean   pbBoolean;     // Boolean型
        pDmaInteger32 plInteger32;    // Integer32型
        pDmaId        pId;           // ID型
        pDmapv        ppObject;      // Object型
        ppDmaString_T ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*    pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;           // オブジェクトのOIID
    DmaInteger32   lType;           // オブジェクトの種別
    SDBR_PROPLIST  PropList;        // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32   lCount;
    SDBR_OBJINFO* pItem;
};
```



```
// バージョン識別子付きオブジェクト情報構造体
struct SDBR_VROBJINFO {
    pDmaString_T      pOIID;           // オブジェクトのOIID
    DmaInteger32      lType;          // オブジェクトの種別
    pDmaString_T      pVersionId;     // バージョン識別子
    SDBR_PROPLIST     PropList;       // プロパティリスト構造体
};

// バージョン識別子付きオブジェクトリスト構造体
struct SDBR_VROBJLIST {
    DmaInteger32      lCount;
    SDBR_VROBJINFO*  pItem;
};
```

詳細

CdbVersionTraceableContainer オブジェクトを 1 バージョンとして包含している

CdbConfiguratedReferentialContainer オブジェクトの一覧を取得するメソッドです。このオブジェクトを表すバージョン識別子も同時に取得できます。その際、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。ただし、このオブジェクトが仮のバージョンに該当するオブジェクトの場合は、CdbConfiguratedReferentialContainer オブジェクトを取得できません。

1 回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて、1 回目のメソッドのコール時点の状態に依存します。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、DMA_FALSE を指定したものととして処理されます。

また、前回コールしたときと引数が異なるメソッドがコールされた場合は、pbContinue の値を無視して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 17-3 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppObjList		

(凡例)

○ : 有効

× : 無効

lPropDefListCount (入力)

pPropDefList の要素数を指定します。

0 を指定した場合、pPropDefList の内容は無視され、プロパティは取得されません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。lPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクトリスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリ解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側は、これらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppVrObjList (出力)

取得した情報を格納するバージョン識別子付きオブジェクトリスト構造体である、SDBR_VROBJLIST 構

造体へのポインタが設定されます。*ppVrObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで、dbrDelete 関数を使用して解放してください。なお、*ppVrObjList を指定したメモリ解放によって、構造体のメンバ (*ppVrObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側は、これらのメンバのメモリを個別に解放する必要はありません。

SDBR_VROBJLIST 構造体および SDBR_VROBJINFO 構造体に設定される内容については、「付録 C.1(22) SDBR_VROBJLIST 構造体 (バージョン識別子付きオブジェクトリスト構造体)」および「付録 C.1(21) SDBR_VROBJINFO 構造体 (バージョン識別子付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::GetVersionableListAndLock メソッド

機能

バージョン管理情報一覧の取得と同時に指定したロックの設定

形式

```
// バージョン識別子なしで取得する場合
DmaBoolean GetVersionableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_OBJLIST**      ppObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// バージョン識別子付きで取得する場合
DmaBoolean GetVersionableListAndLock(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VROBJLIST**    ppVrObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;           // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;            // データ型
    DmaInteger32   lCardinality;     // 基本単位
    DmaInteger32   lCount;           // 要素数
    union {
        pDmaBoolean pbBoolean;      // Boolean型
        pDmaInteger32 plInteger32;  // Integer32型
        pDmaId       pId;           // ID型
        pDmapv        ppObject;     // Object型
        ppDmaString_T ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32   lCount;
    SDBR_PROP*     pItem;
};

// オブジェクト情報構造体
struct SDBR_OBJINFO {
    pDmaString_T   pOIID;           // オブジェクトのOIID
    DmaInteger32   lType;           // オブジェクトの種別
    SDBR_PROPLIST  PropList;       // プロパティリスト構造体
};

// オブジェクトリスト構造体
struct SDBR_OBJLIST {
    DmaInteger32   lCount;
};
```

```

        SDBR_OBJINFO*      pItem;
    };

    // バージョン識別子付きオブジェクト情報構造体
    struct SDBR_VROBJINFO {
        pDmaString_T      pOIID;          // オブジェクトのOIID
        DmaInteger32      lType;          // オブジェクトの種別
        pDmaString_T      pVersionId;     // バージョン識別子
        SDBR_PROPLIST     PropList;       // プロパティリスト構造体
    };

    // バージョン識別子付きオブジェクトリスト構造体
    struct SDBR_VROBJLIST {
        DmaInteger32      lCount;
        SDBR_VROBJINFO*  pItem;
    };

```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトを包含している CdbConfiguratedReferentialContainer オブジェクトの一覧を取得するメソッドです。このオブジェクトを表すバージョン識別子も同時に取得できます。また、pPropDefList で指定したプロパティを持つオブジェクトについては、そのプロパティ値も取得します。ただし、このオブジェクトが仮のバージョンに該当するオブジェクトの場合は、CdbConfiguratedReferentialContainer オブジェクトを取得できません。

1 回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて、1 回目のメソッドのコール時点の状態に依存します。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

このメソッドの使用方法については、「5. CdbContainable クラス (包含要素を扱うクラス)」の CdbContainable::GetContainerList メソッドを参照してください。

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールした場合または前回の読み出しですべて取得した場合は、DMA_FALSE を指定したものと処理されます。

また、前回実行したときと引数が異なるメソッドが実行された場合は、pbContinue の値を無視して、新たに先頭から取得します。

正常終了すると、メソッドを継続してコールする必要があるかどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合は DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 17-4 ほかの引数の扱われ方 (SDBR_OBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IPropDefListCount	×	
pPropDefList	×	
IObjListCount		
ppObjList		
ppVrObjList		
ILockType	×	

(凡例)

○ : 有効

× : 無効

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合、pPropDefList の内容は無視され、プロパティの取得されません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。

プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppObjList (出力)

取得した情報を格納するオブジェクト情報リスト構造体である、SDBR_OBJLIST 構造体へのポインタが設定されます。*ppObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppObjList を指定したメモリ解放によって、構造体のメンバ (*ppObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側は、これらのメンバのメモリを個別に解放する必要はありません。

SDBR_OBJLIST 構造体および SDBR_OBJINFO 構造体に設定される内容については、「付録 C.1(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)」および「付録 C.1(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppVrObjList (出力)

取得した情報を格納するバージョン識別子付きオブジェクト情報リスト構造体である、

SDBR_VROBJLIST 構造体へのポインタが設定されます。*ppVrObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。なお、*ppVrObjList を指定したメモリ解放によって、構造体のメンバ (*ppVrObjList)->pItem およびその先に確保されているメモリはすべて解放されます。したがって、メソッドをコールした側は、これらのメンバのメモリを個別に解放する必要はありません。

SDBR_VROBJLIST 構造体および SDBR_VROBJINFO 構造体に設定される内容については、「付録 C.1(22) SDBR_VROBJLIST 構造体 (バージョン識別子付きオブジェクトリスト構造体)」および「付録 C.1(21) SDBR_VROBJINFO 構造体 (バージョン識別子付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lLockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTIION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::GetVTContaineelist メソッド

機能

構成管理しているオブジェクトの一覧取得

形式

//要素一覧を取得する場合

```
DmaBoolean CdbVersionTraceableContainer:: GetVTContaineelist(
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTOBJLIST**    ppVTObjList)
```

//種類別の要素一覧とリンク情報を取得する場合

```
DmaBoolean CdbVersionTraceableContainer:: GetVTContaineelist(
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTLINKEDOBJLIST** ppVTLinkedObjList)
```

// プロパティ定義情報構造体

```
struct SDBR_PROPDEF {
    DmaId          PropId;          // プロパティ識別子
};
```

// プロパティ構造体

```
struct SDBR_PROP {
    DmaId          PropId;          // プロパティ識別子
    DmaInteger32  lType;           // データ型
    DmaInteger32  lCardinality;    // 基本単位
    DmaInteger32  lCount;         // 要素数
    union {
        pDmaBoolean    pbBoolean; // Boolean型
        pDmaInteger32  plInteger32; // Integer32型
        pDmaId         pId;        // ID型
        pDmapv         ppObject;    // Object型
        ppDmaString_T  ppString;   // String型
    } uniValue;
};
```

// プロパティリスト構造体

```
struct SDBR_PROPLIST {
    DmaInteger32  lCount;          // プロパティ情報の個数
    SDBR_PROP*   pItem;          // プロパティ情報
};
```

// 構成管理オブジェクトリスト構造体

```
struct SDBR_VTOBJLIST{
    DmaInteger32  lCount;          // 構成管理オブジェクト情報の個数
    SDBR_VTOBJINFO* pItem;       // 構成管理オブジェクト情報
};
```

// 構成管理オブジェクト情報構造体

```
struct SDBR_VTOBJINFO{
    pDmaString_T  pOIID;          // オブジェクトのOIID
};
```

```

    pDmaString_T    pVerOIID;    // バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T    pLinkID;     // オブジェクトのリンク識別子
    DmaInteger32    lTraceMode;  // 構成管理モード
    DmaInteger32    lType;       // オブジェクトの種別
    SDBR_PROPLIST   PropList;    // プロパティ情報のリスト
};

// 構成管理リンク付きオブジェクトリスト構造体
struct SDBR_VTLINKEDOBJLIST{
    DmaInteger32    lCount;      // リンク付きオブジェクト情報の個数
    SDBR_VTLINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

// 構成管理リンク付きオブジェクト情報構造体
struct SDBR_VTLINKEDOBJINFO{
    pDmaString_T    pOIID;       // オブジェクトのOIID
    pDmaString_T    pVerOIID;   // バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T    pLinkID;    // リンク識別子
    DmaInteger32    lTraceMode;  // 構成管理モード
    DmaInteger32    lType;       // オブジェクトの種別
    SDBR_PROPLIST   PropList;    // プロパティリスト
    SDBR_PROPLIST   LinkPropList; // リンクのプロパティリスト
};

```

詳細

CdbVersionTraceableContainer オブジェクトが構成管理しているオブジェクトの一覧を取得するメソッドです。pPropDefList でプロパティを指定した場合は、指しているバージョン (構成管理モードが FLOATING モードの場合は最新のバージョン, FIX モードの場合は固定のバージョン) のプロパティも取得します。また, pLinkPropDefList で指定したユーザ定義プロパティを持つリンクについては, そのプロパティの値も取得します。なお, リンクのプロパティとは, DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトのプロパティです。

1 回のメソッドのコールですべての要素の取得が完了しなかった場合, *pbContinue に DMA_TRUE が設定されるので, 再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合, 取得される内容はすべて, 1 回目のメソッドのコール時点の状態に依存します。したがって, 1 回目のメソッドのコール以降に変更した要素数やプロパティ値は, 2 回目以降のコールに反映されません。

また, 要素の個数に変更がなく, 同じオブジェクトに対するメソッドのコールであっても, 取得される情報の順序は常に異なる可能性があります。

次に, 構成管理している CdbVersionableDocument オブジェクトの一覧を 10 個ずつ取得して, その OIID とバージョンが固定されているかどうかを出力する例を示します。なお, エラー処理については省略しています。

構成管理している CdbVersionableDocument オブジェクトの一覧を取得する例

```

SDBR_VTOBJLIST*   pVTObjList = NULL;
DmaInteger32      lLoop;
DmaBoolean        bContinue = DMA_UNKNOWN;
// すべての要素を取得します。
while (bContinue != DMA_FALSE){
    //情報の取得
    pDbrCfgRfct->GetVTContaineerList(&bContinue,
        NULL, 0, NULL, 10, &pVTObjList);
    //取得数だけループして,
    //OIIDとバージョンが固定されているかどうかを出力します。
    for (lLoop = 0; lLoop < pVTObjList->lCount; lLoop++) {

```

```

printf("OIID=%s¥n", pVTObjList->pItem[lLoop].pOIID);
if (pVTObjList->pItem[lLoop].lTraceMode == DBR_MODE_FIX)
{
    printf("Version=Fix¥n");
}
else{
    printf("Version=UnFix¥n");
}
}
}
dbrDelete (pVTObjList);
}
    
```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合または前回の読み出しですべて取得した場合は、DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドのコールの継続が必要かどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 17-5 ほかの引数の扱われ方 (SDBR_VTOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppVTObjList		

(凡例)

○ : 有効

× : 無効

注 前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

表 17-6 ほかの引数の扱われ方 (SDBR_VTLINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lClassType	×	
lPropDefListCount	×	
pPropDefList	×	

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
lLinkPropDefListCount	×	
pLinkPropDefList	×	
lObjListCount		
ppVTLinkedObjList		

(凡例)

○ : 有効

× : 無効

注 前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したもとして処理されます。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合、pPropDefList の内容は無視され、プロパティは取得されません。負の値を指定した場合は、エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定します。なお、IPropDefListCount が正の整数のときに pPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

lObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppVTObjList (出力)

構成管理オブジェクトリスト構造体である、SDBR_VTOBJLIST 構造体へのポインタが設定されます。構成管理オブジェクトリスト構造体へのポインタを格納する領域を指定してください。

**ppVTObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで、dbrDelete 関数を使用して解放してください。

SDBR_VTOBJLIST 構造体および SDBR_VTOBJINFO 構造体に設定される内容については、「付録 C.1(26) SDBR_VTOBJLIST 構造体 (構成管理オブジェクトリスト構造体)」および「付録 C.1(25) SDBR_VTOBJINFO 構造体 (構成管理オブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

lClassType(入力)

取得する要素のクラス識別子を指定します。これによって、指定されたクラス識別子の要素だけが取得できます。ここでは、指しているバージョンのオブジェクトに該当するクラス識別子を指定する必要があります。クラス識別子の一覧を次に示します。

表 17-7 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件として指定した取得もできます。例えば、CdbrDocument クラスの要素と CdbrVersionTraceableContainer クラスの要素を取得したい場合は、次のように指定します。

DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONTRACEABLECONTAINER

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

lLinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合、リンクのプロパティは取得されません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。lLinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定すると、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppVTLinkedObjList (出力)

取得した情報を格納する構成管理リンク付きオブジェクトリスト構造体である、SDBR_VTLINKEDOBJLIST 構造体へのポインタが設定されます。**ppVTLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_VTLINKEDOBJLIST 構造体および SDBR_VTLINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(24) SDBR_VTLINKEDOBJLIST 構造体 (構成管理リンク付きオブジェクトリスト構造体)」および「付録 C.1(23) SDBR_VTLINKEDOBJINFO 構造体 (構成管理リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

CdbVersionTraceableContainer::GetVTContaineedListAndLock メソッド

機能

構成管理しているオブジェクトの一覧取得と同時に指定したロックの設定

形式

```
//要素一覧を取得する場合
DmaBoolean CdbVersionTraceableContainer:: GetVTContaineedListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTOBJLIST**    ppVTObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

//種類別の要素一覧とリンク情報を取得する場合
DmaBoolean CdbVersionTraceableContainer:: GetVTContaineedListAndLock (
    pDmaBoolean          pbContinue,
    DmaInteger32         lClassType,
    DmaInteger32         lPropDefListCount,
    SDBR_PROPDEF*       pPropDefList,
    DmaInteger32         lLinkPropDefListCount,
    SDBR_PROPDEF*       pLinkPropDefList,
    DmaInteger32         lObjListCount,
    SDBR_VTLINKEDOBJLIST** ppVTLinkedObjList,
    DmaInteger32         lLockType = DMA_LOCK_WRITE)

// プロパティ定義情報構造体
struct SDBR_PROPDEF {
    DmaId          PropId;          // プロパティ識別子
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;          // プロパティ識別子
    DmaInteger32  lType;           // データ型
    DmaInteger32  lCardinality;    // 基本単位
    DmaInteger32  lCount;         // 要素数
    union {
        pDmaBoolean    pbBoolean; // Boolean型
        pDmaInteger32  plInteger32; // Integer32型
        pDmaId         pId;        // ID型
        pDmapv         ppObject;   // Object型
        ppDmaString_T  ppString;   // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32  lCount;          // プロパティ情報の個数
    SDBR_PROP*   pItem;          // プロパティ情報
};

// 構成管理オブジェクトリスト構造体
struct SDBR_VTOBJLIST{
    DmaInteger32  lCount;          // 構成管理オブジェクト情報の個数
    SDBR_VTOBJINFO* pItem;        // 構成管理オブジェクト情報
};

// 構成管理オブジェクト情報構造体
```

```

struct SDBR_VTOBJINFO{
    pDmaString_T    pOIID;        // オブジェクトのOIID
    pDmaString_T    pVerOIID;     // バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T    pLinkID;      // オブジェクトのリンク識別子
    DmaInteger32    lTraceMode;    // 構成管理モード
    DmaInteger32    lType;        // オブジェクトの種別
    SDBR_PROPLIST   PropList;     // プロパティ情報のリスト
};

// 構成管理リンク付きオブジェクトリスト構造体
struct SDBR_VTLINKEDOBJLIST{
    DmaInteger32    lCount; // リンク付きオブジェクト情報の個数
    SDBR_VTLINKEDOBJINFO* pItem; // リンク付きオブジェクト情報
};

// 構成管理リンク付きオブジェクト情報構造体
struct SDBR_VTLINKEDOBJINFO{
    pDmaString_T    pOIID;        // オブジェクトのOIID
    pDmaString_T    pVerOIID;     // バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T    pLinkID;      // リンク識別子
    DmaInteger32    lTraceMode;    // 構成管理モード
    DmaInteger32    lType;        // オブジェクトの種別
    SDBR_PROPLIST   PropList;     // プロパティリスト
    SDBR_PROPLIST   LinkPropList; // リンクのプロパティリスト
};

```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトが構成管理しているオブジェクトの一覧を取得するメソッドです。pPropDefList でプロパティを指定した場合は、指しているバージョン (構成管理モードが FLOATING モードの場合は最新のバージョン, FIX モードの場合は固定のバージョン) のプロパティも取得します。また、pLinkPropDefList で指定したユーザ定義プロパティを持つリンクについては、そのプロパティの値も取得します。なお、リンクのプロパティとは、DMA オブジェクトの VersionTraceableContainmentRelationship オブジェクトのプロパティです。

1 回のメソッドのコールですべての要素の取得が完了しなかった場合、*pbContinue に DMA_TRUE が設定されるので、再度メソッドをコールして残りを取得してください。

複数回に分けて情報を取得する場合、取得される内容はすべて、1 回目のメソッドのコール時点の状態に依存します。したがって、1 回目のメソッドのコール以降に変更した要素数やプロパティ値は、2 回目以降のコールに反映されません。

また、要素の個数に変更がなく、同じオブジェクトに対するメソッドのコールであっても、取得される情報の順序は常に異なる可能性があります。

次に、構成管理している CdbVersionableDocument オブジェクトの一覧を 10 個ずつ取得して、その OIID とバージョンが固定されているかどうかを出力する例を示します。なお、エラー処理については省略しています。

構成管理している CdbVersionableDocument オブジェクトの一覧を取得する例

```

SDBR_VTOBJLIST* pVTObjList = NULL;
DmaInteger32    lLoop;
DmaBoolean      bContinue = DMA_UNKNOWN;
// すべての要素を取得します。
while (bContinue != DMA_FALSE){
    //情報の取得
    pDbrCfgRfct->GetVTContaineeListAndLock(&bContinue,
        NULL, 0, NULL, 10, &pVTObjList, DMA_LOCK_WRITE);
    //取得数だけループして,

```

```

//OIIDとバージョンが固定されているかどうかを出力します。
for (lLoop = 0; lLoop < pVTObjList->lCount; lLoop++) {
    printf("OIID=%s¥n", pVTObjList->pItem[lLoop].pOIID);
    if (pVTObjList->pItem[lLoop].lTraceMode == DBR_MODE_FIX)
    {
        printf("Version=Fix¥n");
    }else{
        printf("Version=UnFix¥n");
    }
}
dbrDelete (pVTObjList);
}

```

引数

pbContinue (入力 / 出力)

リストの取得方法を示す値を格納する領域を指定します。

DMA_TRUE を指定すると、オブジェクトに対する前回のメソッドのコールで取得した要素の次の要素から取得します。DMA_FALSE または DMA_UNKNOWN を指定すると、前回取得の残りを破棄して、新たに先頭から取得します。なお、オブジェクトの作成後初めてメソッドをコールする場合または前回の読み出しですべて取得した場合は、DMA_FALSE を指定したものととして処理されます。

正常終了すると、メソッドのコールの継続が必要かどうかを示す値が設定されます。リストの呼び出しがすべて完了した場合は DMA_FALSE が設定されます。まだ取得されていない要素がある場合、DMA_TRUE が設定されます。

2 回目以降に取得する場合に、DMA_TRUE を指定した場合と DMA_FALSE を指定した場合の、ほかの引数に指定した値の扱われ方について、次に示します。

表 17-8 ほかの引数の扱われ方 (SDBR_VTOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		
lPropDefListCount	×	
pPropDefList	×	
lObjListCount		
ppVTObjList		
lLockType	×	

(凡例)

○ : 有効

× : 無効

注 前回と異なる値を指定した場合は、DMA_TRUE の指定が無効になり、DMA_FALSE を指定したものととして処理されます。

表 17-9 ほかの引数の扱われ方 (SDBR_VTLINKEDOBJLIST 構造体を取得する場合)

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
pVersionId		

ほかの引数	pbContinue の値	
	DMA_TRUE	DMA_FALSE
IClassType	×	
IPropDefListCount	×	
pPropDefList	×	
ILinkPropDefListCount	×	
pLinkPropDefList	×	
IObjListCount		
ppVTLinkedObjList		
ILockType	×	

(凡例)

：有効

×：無効

注 前回と異なる値を指定した場合は，DMA_TRUE の指定が無効になり，DMA_FALSE を指定したものとして処理されます。

IPropDefListCount (入力)

pPropDefList の要素数を指定します。0 を指定した場合，pPropDefList の内容は無視され，プロパティは取得されません。負の値を指定した場合は，エラーになります。

pPropDefList (入力)

同時に取得するプロパティの定義情報である，SDBR_PROPDEF 構造体の配列を指定します。プロパティを取得しない場合は NULL を指定します。なお，IPropDefListCount が正の整数のときに pPropDefList に NULL を指定すると，エラーになります。

SDBR_PROPDEF 構造体に指定する内容については，「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

IObjListCount (入力)

取得する情報の個数を指定します。0 以下の数を指定するとエラーになります。

ppVTObjList (出力)

構成管理オブジェクトリスト構造体である，SDBR_VTOBJLIST 構造体へのポインタが設定されます。構成管理オブジェクトリスト構造体へのポインタを格納する領域を指定してください。

**ppVTObjList がポイントするメモリは，ライブラリが新たに確保したものです。このメモリは，ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_VTOBJLIST 構造体および SDBR_VTOBJINFO 構造体に設定される内容については，「付録 C.1(26) SDBR_VTOBJLIST 構造体 (構成管理オブジェクトリスト構造体)」および「付録 C.1(25) SDBR_VTOBJINFO 構造体 (構成管理オブジェクト情報構造体)」参照してください。

また，プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については，「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

IClassType(入力)

取得する要素のクラス識別子を指定します。これによって、指定されたクラス識別子の要素だけが取得できます。クラス識別子の一覧を次に示します。

表 17-10 クラス識別子の一覧

クラス識別子	クラスの種別
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_ALL	すべてのクラス

それぞれのクラスの識別子を OR 条件として指定した取得もできます。例えば、CdbrDocument クラスの要素と CdbrVersionTraceableContainer クラスの要素を取得したい場合は、次のように指定します。

```
DBR_CLASS_DOCUMENT | DBR_CLASS_VERSIONTRACEABLECONTAINER
```

すべてのクラスの要素を取得する場合は、DBR_CLASS_ALL を指定してください。

ILinkPropDefListCount (入力)

pLinkPropDefList の要素数を指定します。0 を指定した場合、リンクのプロパティは取得されません。負の値を指定した場合は、エラーになります。

pLinkPropDefList (入力)

同時に取得するリンクのプロパティの定義情報である、SDBR_PROPDEF 構造体の配列を指定します。指定できるリンクのプロパティ識別子は、ユーザが定義したプロパティの識別子だけです。

プロパティを取得しない場合は NULL を指定します。なお、ILinkPropDefListCount に 0 を指定した場合、pLinkPropDefList の指定内容は無視されます。また、ILinkPropDefListCount が正の整数のときに pLinkPropDefList に NULL を指定した場合は、エラーになります。

SDBR_PROPDEF 構造体に指定する内容については、「付録 C.1(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)」を参照してください。

ppVTLinkedObjList (出力)

取得した情報を格納する構成管理リンク付きオブジェクトリスト構造体である、SDBR_VTLINKEDOBJLIST 構造体へのポインタが設定されます。**ppVTLinkedObjList がポイントするメモリは、ライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_VTLINKEDOBJLIST 構造体および SDBR_VTLINKEDOBJINFO 構造体に設定される内容については、「付録 C.1(24) SDBR_VTLINKEDOBJLIST 構造体 (構成管理リンク付きオブジェクトリスト構造体)」および「付録 C.1(23) SDBR_VTLINKEDOBJINFO 構造体 (構成管理リンク付きオブジェクト情報構造体)」を参照してください。

また、プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ILockType (入力)

ロックの種別を指定します。次の値のどれかを設定します。

- DMA_LOCK_READ
read ロックを設定します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_READ | DBR_RLT_FOR_UPDATE
read ロックを設定し、更新・削除用の一覧を取得します。read ロックを設定すると、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。
- DMA_LOCK_WRITE | DBR_RLT_FOR_UPDATE
排他的な write ロックを設定し、更新・削除用の一覧を取得します。排他的な write ロックを設定すると、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
このロック種別は、一覧を分割取得する場合に、取得した一覧のオブジェクトに対して、更新・削除処理を実行するときに指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::LinkVTFix メソッド

機能

構成管理するオブジェクトの FIX モードでの関連づけ

形式

```
DmaBoolean CdbVersionTraceableContainer::LinkVTFix(
    CdbVersionable*    pDbrVersionable,
    pDmaString_T      pCompoVersionId,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T     ppLinkId)

// プロパティ構造体
struct SDBR_PROP {
    DmaId              PropId;           // プロパティ識別子
    DmaInteger32       lType;           // データ型
    DmaInteger32       lCardinality;    // 基本単位
    DmaInteger32       lCount;         // 要素数
    union {            // プロパティの値
        pDmaBoolean    pbBoolean;      // Boolean型
        pDmaInteger32  plInteger32;    // Integer32型
        pDmaId         pId;            // ID型
        pDmapv         ppObject;       // Object型
        ppDmaString_T  ppString;      // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32       lCount;
    SDBR_PROP*        pItem;
};
```

詳細

CdbVersionTraceableContainer オブジェクトと CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを、構成管理型のコンテナメントで関連づけるメソッドです。このとき、常に固定されたバージョンの構成要素を指す FIX モードを構成管理モードとして設定します。FIX モードを設定すると、構成要素がバージョンアップされた場合も CdbVersionTraceableContainer オブジェクトが指すリンク先は変更されません。

構成要素のバージョンアップに伴って CdbVersionTraceableContainer オブジェクトが常にカレントバージョンの構成要素を指すようにするには、CdbVersionTraceableContainer::SetVTFloat メソッドを使用して構成管理モードを FLOATING モードに変更してください。

引数

pDbrVersionable (入力)

構成管理するオブジェクトを指定します。CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを指定できます。

pCompoVersionId (入力)

関連づける構成要素のバージョン識別子を指定します。NULL を指定した場合、カレントバージョンを指定したことになります。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリストを指定します。

edmClass_VersionTraceableContainmentRelationship クラスに定義されているプロパティを指定できます。なお、リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbVersionTraceableContainer::GetLinkPropertyValues メソッド、CdbVersionTraceableContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbVersionTraceableContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。また、関連づけを解除するためのメソッドである CdbVersionTraceableContainer::UnlinkVT メソッドまたは CdbVersionTraceableContainer::UnlinkVTAndLock でも使用できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。

17. CdbvVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切でない値を指定しています。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	未サポートのプロパティ型です。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACT	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::LinkVTFixAndLock メソッド

機能

構成管理するオブジェクトの FIX モードでの関連づけと同時に指定したロックの設定

形式

```
DmaBoolean CdbVersionTraceableContainer::LinkVTFixAndLock (
    CdbVersionable*    pDbrVersionable,
    pDmaString_T      pCompoVersionId,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T     ppLinkId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;           // プロパティ識別子
    DmaInteger32         lType;           // データ型
    DmaInteger32         lCardinality;    // 基本単位
    DmaInteger32         lCount;         // 要素数
    union {              // プロパティの値
        pDmaBoolean      pbBoolean;     // Boolean型
        pDmaInteger32    plInteger32;   // Integer32型
        pDmaId           pId;           // ID型
        pDmapv           ppObject;      // Object型
        ppDmaString_T    ppString;     // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトと CdbVersionableDocument オブジェクトまたは CdbConfiguredReferentialContainer オブジェクトを、構成管理型のコンテインメントで関連づけるメソッドです。このとき、常に固定されたバージョンの構成要素を指す FIX モードを構成管理モードとして設定します。FIX モードを設定すると、構成要素がバージョンアップされた場合も CdbVersionTraceableContainer オブジェクトが指すリンク先は変更されません。

構成要素のバージョンアップに伴って CdbVersionTraceableContainer オブジェクトが常にカレントバージョンの構成要素を指すようにするには、CdbVersionTraceableContainer::SetVTFloat メソッドを使用して構成管理モードを FLOATING モードに変更してください。

引数

pDbrVersionable (入力)

構成管理するオブジェクトを指定します。CdbVersionableDocument オブジェクトまたは CdbConfiguredReferentialContainer オブジェクトを指定できます。

pCompoVersionId (入力)

関連づける構成要素のバージョン識別子を指定します。NULL を指定した場合、カレントバージョンを指定したことになります。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリストを指定します。
edmClass_VersionTraceableContainmentRelationship クラスに定義したプロパティを指定できます。なお、リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、
CdbVersionTraceableContainer::GetLinkPropertyValues メソッド、
CdbVersionTraceableContainer::GetLinkPropertyValuesAndLock メソッドまたは
CdbVersionTraceableContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。また、関連づけを解除するためのメソッドである
CdbVersionTraceableContainer::UnlinkVT メソッドまたは
CdbVersionTraceableContainer::UnlinkVTAndLock でも使用できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切でない値を指定しています。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	未サポートのプロパティ型です。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

CdbVersionTraceableContainer::LinkVTFloat メソッド

機能

構成管理するオブジェクトの FLOATING モードでの関連づけ

形式

```
DmaBoolean CdbVersionTraceableContainer::LinkVTFloat (
    CdbVersionable*    pDbrVersionable,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T      ppLinkId)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;                // プロパティ識別子
    DmaInteger32         lType;                 // データ型
    DmaInteger32         lCardinality;          // 基本単位
    DmaInteger32         lCount;                // 要素数
    union {              // プロパティの値
        pDmaBoolean      pbBoolean;            // Boolean型
        pDmaInteger32    plInteger32;          // Integer32型
        pDmaId           pId;                  // ID型
        pDmapv           ppObject;             // Object型
        ppDmaString_T    ppString;            // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};
```

詳細

CdbVersionTraceableContainer オブジェクトと CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを、構成管理型のコンテインメントで関連づけるメソッドです。このとき、常にカレントバージョンの構成要素を指す FLOATING モードを構成管理モードとして設定します。FLOATING モードを設定すると、構成要素がバージョンアップされた場合、CdbVersionTraceableContainer オブジェクトが指すリンク先は常にカレントバージョンに変更されません。

構成要素のバージョンアップがあっても CdbVersionTraceableContainer オブジェクトが常に特定のバージョンを指すようにするには、CdbVersionTraceableContainer::SetVTFix メソッドを使用して構成管理モードを FIX モードに変更してください。

引数

pDbrVersionable (入力)

構成管理するオブジェクトを指定します。CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを指定できます。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリストを指定します。edmClass_VersionTraceableContainmentRelationship クラスに定義したプロパティを指定できます。なお、リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別

子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbVersionTraceableContainer::GetLinkPropertyValues メソッド、CdbVersionTraceableContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbVersionTraceableContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。また、関連づけを解除するためのメソッドである CdbVersionTraceableContainer::UnlinkVT メソッドまたは CdbVersionTraceableContainer::UnlinkVTAndLock でも使用できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切でない値を指定しています。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	未サポートのプロパティ型です。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::LinkVTFloatAndLock メソッド

機能

構成管理するオブジェクトの FLOATING モードでの関連づけと同時に指定したロックの設定

形式

```
DmaBoolean CdbVersionTraceableContainer::LinkVTFloatAndLock(
    CdbVersionable*    pDbrVersionable,
    SDBR_PROPLIST*    pLinkPropList,
    ppDmaString_T     ppLinkId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)

// プロパティ構造体
struct SDBR_PROP {
    DmaId                PropId;                // プロパティ識別子
    DmaInteger32         lType;                // データ型
    DmaInteger32         lCardinality;         // 基本単位
    DmaInteger32         lCount;              // 要素数
    union {
        pDmaBoolean      pbBoolean;          // Boolean型
        pDmaInteger32    plInteger32;        // Integer32型
        pDmaId           pId;                // ID型
        pDmapv           ppObject;          // Object型
        ppDmaString_T    ppString;          // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32         lCount;
    SDBR_PROP*          pItem;
};
```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトと CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを、構成管理型のコンテインメントで関連づけるメソッドです。このとき、常にカレントバージョンの構成要素を指す FLOATING モードを構成管理モードとして設定します。FLOATING モードを設定すると、構成要素がバージョンアップされた場合、CdbVersionTraceableContainer オブジェクトが指すリンク先は常にカレントバージョンに変更されます。

構成要素のバージョンアップがあっても CdbVersionTraceableContainer オブジェクトが常に特定のバージョンを指すようにするには、CdbVersionTraceableContainer::SetVTFix メソッドを使用して構成管理モードを FIX モードに変更してください。

引数

pDbrVersionable (入力)

構成管理するオブジェクトを指定します。CdbVersionableDocument オブジェクトまたは CdbConfiguratedReferentialContainer オブジェクトを指定できます。

pLinkPropList (入力)

リンクに設定するプロパティのプロパティリストを指定します。

edmClass_VersionTraceableContainmentRelationship クラスに定義したプロパティを指定できます。なお、リンクにプロパティの値を設定できるのは、ユーザが定義したプロパティだけです。プロパティ識別子にはユーザが定義したプロパティ識別子を指定してください。

プロパティを設定しない場合は、NULL を指定してください。

SDBR_PROPLIST 構造体および SDBR_PROP 構造体については、「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

ppLinkId (出力)

リンク識別子へのポインタが設定されます。このリンク識別子は、CdbVersionTraceableContainer::GetLinkPropertyValues メソッド、CdbVersionTraceableContainer::GetLinkPropertyValuesAndLock メソッドまたは CdbVersionTraceableContainer::PutLinkPropertyValues メソッドで、処理対象とするリンクを指定する場合に使用します。また、関連づけを解除するためのメソッドである CdbVersionTraceableContainer::UnlinkVT メソッドまたは CdbVersionTraceableContainer::UnlinkVTAndLock でも使用できます。

*ppLinkId がポイントするメモリはライブラリが新たに確保したものです。このメモリは、ユーザプログラムで dbrDelete 関数を使用して解放してください。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切でない値を指定しています。	正しい型を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_NOT_SUPPORTED	未サポートのプロパティ型です。	この操作は実行できません。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::SetVTFix メソッド

機能

構成要素のバージョンの固定

形式

```
DmaBoolean CdbVersionTraceableContainer::SetVTFix(
    pDmaString_T      pLinkId,
    pDmaString_T      pCompoVersionId)
```

詳細

CdbVersionTraceableContainer オブジェクトの pLinkId で指定した構成要素のバージョンを、pCompoVersionId で指定したバージョンで固定する (構成管理モードに FIX モードを設定する) メソッドです。

このメソッドによってバージョンを固定したあとは、構成要素がバージョンアップしても、CdbVersionTraceableContainer オブジェクトは固定したバージョンの構成要素を参照し続けます。

すでに FIX モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は、FIX モードのまま正常終了します。

引数

pLinkId (入力)

バージョンを固定する構成要素のリンク識別子を指定します。リンク識別子は、CdbVersionTraceableContainer::GetVTContaineerList メソッドまたは CdbVersionTraceableContainer::GetVTContaineerListAndLock メソッドで取得します。

pCompoVersionId (入力)

固定するバージョンのバージョン識別子を指定します。

なお、DBR_VTVERSION を指定した場合には現時点で関連づけられているバージョンで固定されます。NULL を指定した場合にはカレントバージョンで固定されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID で不正な OIID が設定されています。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_TRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::SetVTFixAndLock メソッド

機能

構成要素のバージョンの固定と同時に指定したロックの設定

形式

```
DmaBoolean CdbVersionTraceableContainer::SetVTFixAndLock(
    pDmaString_T      pLinkId,
    pDmaString_T      pCompoVersionId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)
```

詳細

明示的にロックの種別を指定して、CdbVersionTraceableContainer オブジェクトの pLinkId で指定した構成要素のバージョンを、pCompoVersionId で指定したバージョンで固定する (構成管理モードに FIX モードを設定する) メソッドです。

このメソッドによってバージョンを固定したあとは、構成要素がバージョンアップしても、CdbVersionTraceableContainer オブジェクトは固定したバージョンの構成要素を参照し続けます。

すでに FIX モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は、FIX モードのまま正常終了します。

引数

pLinkId (入力)

バージョンを固定する構成要素のリンク識別子を指定します。リンク識別子は、CdbVersionTraceableContainer::GetVTContaineerList メソッドまたは CdbVersionTraceableContainer::GetVTContaineerListAndLock メソッドで取得します。

pCompoVersionId (入力)

固定するバージョンのバージョン識別子を指定します。

なお、DBR_VTVERSION を指定した場合には現時点で関連づけられているバージョンで固定されます。NULL を指定した場合にはカレントバージョンで固定されます。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID で不正な OIID が設定されています。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼びられました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::SetVTFloat メソッド

機能

構成要素のバージョン固定の解除

形式

```
DmaBoolean CdbVersionTraceableContainer::SetVTFloat (
    pDmaString_T    pLinkId)
```

詳細

CdbVersionTraceableContainer オブジェクトが構成管理している pLinkId で指定した構成要素のバージョンの固定を解除する (構成管理モードに FLOATING モードを設定する) メソッドです。

このメソッドをコールすると、CdbVersionTraceableContainer オブジェクトは、常に構成要素のカレントバージョンを参照するようにリンクを切り替えます。

すでに FLOATING モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は、FLOATING モードのまま正常終了します。

引数

pLinkId (入力)

バージョン固定を解除する構成要素のリンク識別子を指定します。リンク識別子は、CdbVersionTraceableContainer::GetVTContaineelist メソッドで取得します。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECT	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::SetVTFloatAndLock メソッド

機能

構成要素のバージョン固定の解除と同時に指定したロックの設定

形式

```
DmaBoolean CdbVersionTraceableContainer::SetVTFloatAndLock(
    pDmaString_T    pLinkId,
    DmaInteger32    lLockType = DMA_LOCK_WRITE)
```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトが構成管理している pLinkId で指定した構成要素のバージョンの固定を解除する (構成管理モードに FLOATING モードを設定する) メソッドです。

このメソッドをコールすると、CdbVersionTraceableContainer オブジェクトは、常に構成要素のカレントバージョンを参照するようにリンクを切り替えます。

すでに FLOATING モードで構成管理しているオブジェクトに対してこのメソッドをコールした場合は、FLOATING モードのまま正常終了します。

引数

pLinkId (入力)

バージョン固定を解除する構成要素のリンク識別子を指定します。リンク識別子は、CdbVersionTraceableContainer::GetVTContaineelist メソッドで取得します。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_CORBA	CORBA の戻り値が設定されます。	CORBA でエラーが発生しました。	ユーザプログラムでは対処できません。エラー情報として文字列の値を取得して、ユーザプログラムを終了してください。
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_FAILED	操作が失敗しました。	時間を置いて再度実行するか、処理を見直してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッド が呼ばれました。	Begin メソッドの呼び出し後 に、この操作を実行してくださ い。

CdbVersionTraceableContainer::UnlinkVT メソッド

機能

構成管理しているオブジェクトとの関連づけの解除

形式

```
DmaBoolean CdbVersionTraceableContainer::UnlinkVT(
    pDmaString_T      pLinkId)
```

詳細

CdbVersionTraceableContainer オブジェクトに関連づけて構成管理しているオブジェクトとの関連づけを解除するメソッドです。

指定したオブジェクトと関連づけされていないリンク識別子を指定した場合は、エラーになります。

引数

pLinkId (入力)

関連づけを解除する構成要素のリンク識別子を指定します。指定するリンク識別子は、CdbVersionTraceableContainer::GetVTContaineelist メソッドまたは CdbVersionTraceableContainer::GetVTContaineelistAndLock メソッドで取得できます。また、CdbVersionTraceableContainer::LinkVTFix メソッド、CdbVersionTraceableContainer::LinkVTFixAndLock メソッド、CdbVersionTraceableContainer::LinkVTFloat メソッドまたは CdbVersionTraceableContainer::LinkVTFloatAndLock メソッドで関連づけした時に出力されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECT	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSAC TION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

CdbVersionTraceableContainer::UnlinkVTAndLock メソッド

機能

構成管理しているオブジェクトとの関連づけの解除と同時に指定したロックの設定

形式

```
DmaBoolean CdbVersionTraceableContainer::UnlinkVTAndLock(
    pDmaString_T      pLinkId,
    DmaInteger32      lLockType = DMA_LOCK_WRITE)
```

詳細

ロックタイプを明示的に指定して、CdbVersionTraceableContainer オブジェクトに関連づけて構成管理しているオブジェクトとの関連づけを解除するメソッドです。

指定したオブジェクトと関連づけされていないリンク識別子を指定した場合は、エラーになります。

引数

pLinkId (入力)

関連づけを解除する構成要素のリンク識別子を指定します。指定するリンク識別子は、CdbVersionTraceableContainer::GetVTContaineelist メソッドまたは CdbVersionTraceableContainer::GetVTContaineelistAndLock メソッドで取得できます。また、CdbVersionTraceableContainer::LinkVTFix メソッド、CdbVersionTraceableContainer::LinkVTFixAndLock メソッド、CdbVersionTraceableContainer::LinkVTFloat メソッドまたは CdbVersionTraceableContainer::LinkVTFloatAndLock メソッドで関連づけした時に出力されます。

lLockType (入力)

ロックの種別を指定します。次のどちらかの値を設定します。

- DMA_LOCK_READ
read ロックを設定します。このロックを設定した場合、ほかのクライアントから write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。
- DMA_LOCK_WRITE
排他的な write ロックを設定します。このロックを設定した場合は、ほかのクライアントから read ロックおよび write ロックを設定できません。ロックがすでに設定されている場合には、ロックが解除されるまで待ちます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。

17. CdbVersionTraceableContainer クラス (バージョンなし構成管理コンテナを扱うクラス)

major_code	minor_code	説明	対処
ERR_DB	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。
ERR_DB	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
ERR_DBR	ERR_ACCESS_NOT_PERMISSIONED	アクセス権がありません。	この操作は実行できません。
ERR_DBR	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
ERR_DBR	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対しては、一度に一つの処理要求しか出さないように見直してください。
ERR_DBR	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
ERR_DBR	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
ERR_DBR	ERR_SESSION_NOT_CONNECTED	セッションが接続されていません。	セッションを接続してください。
ERR_DMA	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
ERR_DMA	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。	正しい OIID を指定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
ERR_DMA	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドの呼び出し後に、この操作を実行してください。

18 CdbrXmlTranslator クラス (XML 変換を扱うクラス)

XML プロパティマッピングに使用する定義ファイル類を解析して、XML 文書管理機能を実行するための環境を管理する機能を提供するクラスです。プロパティリストの取得や全文検索インデクス作成用データの生成など、個々の XML ファイルに対する変換機能を提供します。

CdbrXmlTranslator クラスの詳細

CdbrXmlTranslator::~CdbrXmlTranslator メソッド

CdbrXmlTranslator::CdbrXmlTranslator メソッド

CdbrXmlTranslator::GetDmalInfoList メソッド

CdbrXmlTranslator::GetIndexData メソッド

CdbrXmlTranslator クラスの詳細

CdbrXmlTranslator クラスのインスタンスは、CdbrXmlTranslatorFactory::CreateTranslator メソッドで CdbrXmlTranslator オブジェクトを生成することによって使用できるようになります。なお、CdbrXmlTranslatorFactory オブジェクトが破棄されると、このクラスのインスタンスも CdbrXmlTranslator オブジェクト生成前の状態に戻り、このクラスの主要なメソッドはエラーになります。

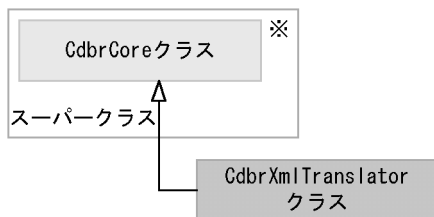
また、このクラスのインスタンスは複数のスレッドで共有できません。一つのスレッド内で占有して使用してください。CdbrXmlTranslator オブジェクトの生成時と異なるスレッドでこのクラスのメソッドを実行した場合の動作は保証しません。

生成処理に時間を要するので、生成した CdbrXmlTranslator オブジェクトはスレッド内で使い回すことを推奨します。

このクラスを使用するには HiRDB Adapter for XML が必要です。

なお、全文検索インデクス作成用データを作成するときは、Preprocessing Library for Text Search があわせて必要です。

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

なし

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrXmlTranslator クラスで導入されるメソッド

メソッド	機能
~CdbrXmlTranslator	デストラクタ
CdbrXmlTranslator	コンストラクタ
GetDmaInfoList	DMA 情報リストの取得
GetIndexData	全文検索インデクス作成用データの取得

以降, CdbrXmlTranslator クラスで導入されるメソッドについて説明します。

CdbrXmlTranslator::~CdbrXmlTranslator メソッド

機能

デストラクタ

形式

`~CdbrXmlTranslator()`

詳細

CdbrXmlTranslator オブジェクトのデストラクタです。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrXmlTranslator::CdbrXmlTranslator メソッド

機能

コンストラクタ

形式

```
CdbrXmlTranslator()
```

詳細

CdbrXmlTranslator オブジェクトのコンストラクタです。

領域を確保した直後の状態では必要な情報が設定されていないため、このクラスのほかのメソッドを実行できません。ほかのメソッドを実行できるようにするには、CdbrXmlTranslatorFactory::CreateTranslator メソッドで CdbrXmlTranslator オブジェクトを生成してください。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbXmlTranslator::GetDmaInfoList メソッド

機能

DMA 情報リストの取得

形式

```
DmaBoolean GetDmaInfoList (
    const pDmaString_T    pXmlFilePath,
    DmaInteger32          lParseLevel,
    const pDmaString_T    pMappingId,
    DmaInteger32          lIndexType,
    const pDmaString_T    pFilterFilePath,
    const pDmaString_T    pIndexFilePath,
    SDBR_DMAINFOLIST**   ppDmaInfoList,
    ppDmaString_T        ppParseMessage)

// DMA情報リスト構造体
struct SDBR_DMAINFOLIST {
    DmaInteger32    lCount;           //DMAオブジェクトの情報の個数
    SDBR_DMAINFO*  pItem;           //DMAオブジェクトの情報
};

// DMAオブジェクト生成用の構造体
struct SDBR_DMAINFO {
    DmaId          ClassId;           //クラス識別子
    SDBR_PROPLIST PropList;          //プロパティ一覧
};

// プロパティ構造体
struct SDBR_PROP {
    DmaId          PropId;           // プロパティ識別子
    DmaInteger32   lType;           // データ型
    DmaInteger32   lCardinality;    // 基本単位
    DmaInteger32   lCount;         // 要素数
    union {
        pDmaBoolean    pbBoolean;   // Boolean型
        pDmaInteger32  plInteger32;  // Integer32型
        pDmaId         pId;         // ID型
        pDmapv         ppObject;    // Object型
        ppDmaString_T  ppString;    // String型
    } uniValue;
};

// プロパティリスト構造体
struct SDBR_PROPLIST {
    DmaInteger32    lCount;
    SDBR_PROP*     pItem;
};
```

詳細

指定された XML ファイルの構文解析を行い、マッピング定義情報に従って XML ファイルからプロパティを抽出して DMA 情報リスト (文書オブジェクトを作成するときのプロパティの初期値のリスト) を作成するメソッドです。また、オプションで、全文検索インデクス作成用のデータをファイルに出力します。ただし、全文検索インデクス作成用データの出力には Preprocessing Library for Text Search が必要です。

CdbXmlTranslatorFactory::CreateTranslator メソッドで CdbXmlTranslator オブジェクトを生成していない場合や、CdbXmlTranslator オブジェクト生成元の CdbXmlTranslatorFactory オブジェクトがすでに破棄されている場合、オブジェクトが未初期化状態であるとして、このメソッドはエラーになります。

引数

pXmlFilePath (入力)

入力する XML ファイルのパス名を, "file:///" で始まる URL 形式で指定します。NULL を指定するとエラーになります。

IParseLevel (入力)

構文解析のレベルを指定します。次の値のどれかを指定してください。

- DBR_XMLPARSE_NO_EXTERNAL_ENTITIES
DTD の解析をしないで, 外部エンティティは無視します。
- DBR_XMLPARSE_WITH_EXTERNAL_ENTITIES
DTD を含めた構文解析をして外部エンティティを読み込みますが, 検証はしません。
- DBR_XMLPARSE_VALIDATE
DTD を使って検証を実行します。

pMappingId (入力)

マッピング定義名を指定します。NULL を指定するとエラーになります。

lIndexType (入力)

作成対象となる全文検索インデクスの種別を指定します。次の値のどれかを指定してください。

- DBR_INDEXTYPE_NOTHING
全文検索インデクスを作成しない場合に指定します。全文検索インデクス作成用データは出力されません。
- DBR_INDEXTYPE_PLAIN
構造指定検索なしの全文検索インデクスを作成する場合に指定します。プレーンテキスト形式のインデクスデータを出力します。
- DBR_INDEXTYPE_STRUCTURED
構造指定検索用の全文検索インデクスを作成する場合に指定します。構造指定検索用のインデクスデータ (ESIS-B 形式のデータ) を出力します。

なお, DBR_INDEXTYPE_NOTHING を指定した場合, 引数 pFilterFilePath および pIndexPath の値は無視されます。

また, Preprocessing Library for Text Search を利用できない場合に, DBR_INDEXTYPE_NOTHING 以外の値を指定するとエラーになります。

pFilterFilePath (入力)

全文検索インデクス作成用データの生成の際に使用するフィルタリング定義ファイルを指定します。引数 lIndexType に DBR_INDEXTYPE_NOTHING を指定した場合, pFilterFileName の値は無視されます。

ファイルは, UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <DocumentBroker Development Kit のインストールディレクトリ>%etc%xml_files を起点とした相対パス名か, "file:///" で始まる絶対パス名で指定します。また, NULL を指定した場合, UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <DocumentBroker Development Kit のインストールディレクトリ>%etc%xml_files にあるデフォルトのファイル "<文書空間識別子><マッピング定義名>.tfd" を使用して処理を実行します。なお, NULL を指定した場合でデフォルトのファイルが存在しない場合, フィルタリング定義ファイルを使用しないで処理を実行します。

pIndexFilePath (入力)

全文検索インデクス作成用データを出力するファイルのパス名を, "file:/" で始まる URL 形式で指定します。引数 lIndexType に DBR_INDEXTYPE_NOTHING を指定した場合, pIndexFilePath の値は無視されます。それ以外の場合, NULL を指定するとエラーになります。

ppDmaInfoList (出力)

抽出したプロパティを格納する DMA 情報リスト構造体である, SDBR_DMAINFOLIST 構造体へのポインタが設定されます。DMA 情報リスト構造体へのポインタを格納する領域を指定してください。NULL を指定するとエラーになります。それ以外の場合 NULL が格納されます。

**ppDmaInfoList がポイントするメモリは, ライブラリが新たに確保したものです。このメモリは, ユーザプログラムで dbrDelete 関数を使用して解放してください。

SDBR_DMAINFOLIST 構造体および SDBR_DMAINFO 構造体に設定される内容については, 「付録 C.1(4) SDBR_DMAINFOLIST 構造体 (DMA 情報リスト構造体)」および「付録 C.1(3) SDBR_DMAINFO 構造体 (DMA オブジェクト生成用の構造体)」を参照してください。

また, プロパティ情報が設定される SDBR_PROPLIST 構造体および SDBR_PROP 構造体については, 「付録 C.1(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)」および「付録 C.1(12) SDBR_PROP 構造体 (プロパティ構造体)」を参照してください。

なお, SDBR_DMAINFOLIST 構造体のメンバ pItem に設定される DMA オブジェクト生成用構造体の配列の要素数は, マッピング定義情報で指定したクラスの数と同じです。

また, ユーザがマッピング定義として二つのクラスを指定した場合, 1 番目の要素には dmaClass_ConfigurationHistory クラスまたはそのサブクラスの情報が, 2 番目の要素には dmaClass_DocVersion クラスまたはそのサブクラスの情報が, それぞれ設定されます。

ppParseMessage (出力)

XML ファイルの構文解析などで発生したエラーやワーニング (警告) の詳細情報が設定されます。詳細情報を格納する領域のアドレスを指定してください。NULL を指定するとエラーになります。なお, 返却する詳細情報がない場合は NULL が設定されます。ppParseMessage の領域は, dbrDelete 関数を使ってユーザプログラムで解放してください。

NULL が設定されるかどうかはこのメソッドの戻り値に依存しません。メソッドが正常終了した場合でも NULL 以外の値が設定される場合があります, また, 異常終了した場合でも NULL が設定される場合があります。

取得した文字列は改行コードで区切られ, 複数行で構成される場合があります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_NOT_INITIALIZE	オブジェクトが初期化されていません。	初期化を実行してください。

major_code	minor_code	説明	対処
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_BAD_VALUE	プロパティやリストの値が、許容範囲外の値であるかまたは最大値を超えました。	値の範囲を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません (全文検索インデックス作成用データの出力には Preprocessing Library for Text Search が必要です)。
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指定されたリソースがありません。	指定したリソースパスを見直してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_PARSE_ERROR	XML ファイルの構文解析エラーが発生しました。	XML ファイルの内容を見直してください。

CdbrXmlTranslator::GetIndexData メソッド

機能

全文検索インデクス作成用データの取得

形式

```
DmaBoolean GetIndexData (
    const pDmaString_T    pXmlFilePath,
    DmaInteger32          lParseLevel,
    DmaInteger32          lIndexType,
    const pDmaString_T    pFilterFilePath,
    const pDmaString_T    pIndexPath,
    ppDmaString_T         ppParseMessage)
```

詳細

XML ファイルの構文解析をして全文検索インデクス作成用データを生成し、ファイルに出力するメソッドです。出力されたファイルは、CdbrDocument クラスなどの文書系クラスの CreateIndex メソッドで利用します。

CdbrXmlTranslatorFactory::CreateTranslator メソッドで CdbrXmlTranslator オブジェクトを生成していない場合や、CdbrXmlTranslator オブジェクト生成元の CdbrXmlTranslatorFactory オブジェクトがすでに破棄されている場合、オブジェクトが未初期化状態であるとして、このメソッドはエラーになります。

全文検索インデクス作成用データは、CdbrXmlTranslator::GetDmaInfoList メソッドでも取得できます。このメソッドも CdbrXmlTranslator::GetDmaInfoList メソッドも構文解析を伴うので、プロパティリストも必要な場合は処理性能を考慮して、CdbrXmlTranslator::GetDmaInfoList メソッドで一括して取得することを推奨します。

なお、このメソッドの実行には Preprocessing Library for Text Search が必要です。

引数

pXmlFilePath (入力)

入力する XML ファイルのパス名を、"file://" で始まる URL 形式で指定します。NULL を指定するとエラーになります。

lParseLevel (入力)

構文解析のレベルを指定します。次の値のどれかを指定してください。

- DBR_XMLPARSE_NO_EXTERNAL_ENTITIES
DTD の解析をしないで、外部エンティティは無視します。
- DBR_XMLPARSE_WITH_EXTERNAL_ENTITIES
DTD を含めた構文解析をして外部エンティティを読み込みますが、検証はしません。
- DBR_XMLPARSE_VALIDATE
DTD を使って検証を実行します。

lIndexType (入力)

作成対象となる全文検索インデクスの種別を指定します。次の値のどちらかを指定してください。

- DBR_INDEXTYPE_PLAIN
構造指定検索なしの全文検索インデクスを作成する場合に指定します。プレーンテキスト形式のインデクスデータを出力します。

- DBR_INDEXTYPE_STRUCTURED

構造指定検索用の全文検索インデクスを作成する場合に指定します。構造指定検索用のインデクスデータ (ESIS-B 形式のデータ) を出力します。

pFilterFilePath (入力)

全文検索インデクス作成用データの生成の際に使用するフィルタリング定義ファイルを指定します。

ファイルは、UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <DocumentBroker Development Kit のインストールディレクトリ>%etc%xml_files を起点とした相対パス名か, "file:///" で始まる絶対パス名で指定します。なお、NULL を指定した場合、フィルタリング定義ファイルを使用しないで処理を実行します。

pIndexPath (入力)

全文検索インデクス作成用データを出力するファイルのパス名を, "file:///" で始まる URL 形式で指定します。NULL を指定するとエラーになります。

ppParseMessage (出力)

XML ファイルの構文解析などで発生したエラーやワーニング (警告) の詳細情報が設定されます。詳細情報を格納する領域のアドレスを指定してください。NULL を指定するとエラーになります。なお、返却する詳細情報がない場合は NULL が設定されます。ppParseMessage の領域は、dbrDelete 関数を使ってユーザプログラムで解放してください。

NULL が設定されるかどうかはこのメソッドの戻り値に依存しません。メソッドが正常終了した場合でも NULL 以外の値が設定される場合があります、また、異常終了した場合でも NULL が設定される場合があります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_NOT_INITIALIZED	オブジェクトが初期化されていません。	初期化を実行してください。
ERR_DMA	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_NOT_SUPPORTED	このセッションまたはオブジェクトではサポートされていないメソッドです。	この操作は実行できません (Preprocessing Library for Text Search が必要です)。

18. CdbrXmlTranslator クラス (XML 変換を扱うクラス)

major_code	minor_code	説明	対処
ERR_DMA	DMARC_RESOURCE_NOT_FOUND	指定されたリソースがありません。	指定したリソースパスを見直してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。
ERR_DMA	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
ERR_DMA	EDMRC_PARSE_ERROR	XML ファイルの構文解析エラーが発生しました。	XML ファイルの内容を見直してください。

19 CdbrXmlTranslatorFactory クラス (XML の操作環境を管理するクラス)

XML 文書管理を実行する機能を提供するクラスです。マッピング定義など XML の操作環境を管理し、CdbrXmlTranslator オブジェクトを生成します。CdbrXmlTranslatorFactory オブジェクトは、接続する文書空間単位にプロセスで一つ作成します。

CdbrXmlTranslatorFactory クラスの詳細

CdbrXmlTranslatorFactory::~CdbrXmlTranslatorFactory メソッド

CdbrXmlTranslatorFactory::CdbrXmlTranslatorFactory メソッド

CdbrXmlTranslatorFactory::CreateTranslator メソッド

CdbrXmlTranslatorFactory::Initialize メソッド

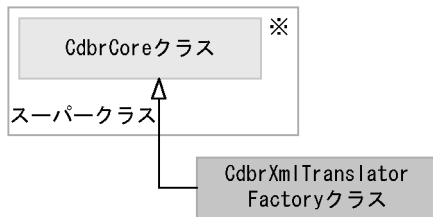
CdbrXmlTranslatorFactory クラスの詳細

CdbrXmlTranslatorFactory クラスのインスタンスは複数のスレッドで共有できます。

ただし、CdbrXmlTranslatorFactory オブジェクトの生成と破棄 (コンストラクタとデストラクタの実行) は、必ずシングルスレッド環境下で実行してください。

このクラスを使用するには HiRDB Adapter for XML が必要です。

スーパークラス



(凡例)

——▷ : クラスの継承関係を表します。

注※ そのクラスの単独のインスタンスを生成しない、抽象クラスです。

プロパティ一覧

なし

メソッド一覧

CdbrCore クラスから継承されるメソッド

メソッド	機能
~CdbrCore	デストラクタ
CdbrCore	コンストラクタ
GetDBError	データベースのエラーの取得
GetLastError	エラー番号の取得

これらのメソッドの詳細については、「6. CdbrCore クラス (エラー情報を取得するクラス)」を参照してください。

CdbrXmlTranslatorFactory クラスで導入されるメソッド

メソッド	機能
~CdbrXmlTranslatorFactory	デストラクタ
CdbrXmlTranslatorFactory	コンストラクタ
CreateTranslator	CdbrXmlTranslator オブジェクトの生成
Initialize	オブジェクトの初期化

以降、CdbrXmlTranslatorFactory クラスで導入されるメソッドについて説明します。

CdbrXmlTranslatorFactory::~CdbrXmlTranslatorFactory メソッド

機能

デストラクタ

形式

`~CdbrXmlTranslatorFactory()`

詳細

CdbrXmlTranslatorFactory オブジェクトのデストラクタです。

CdbrXmlTranslatorFactory オブジェクトの破棄は、必ずシングルスレッド環境下で実行してください。

なお、CdbrXmlTranslatorFactory オブジェクトの破棄によって、CdbrXmlTranslatorFactory::CreateTranslator メソッド実行前の状態に戻り、CdbrXmlTranslator オブジェクトは使用できなくなります。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrXmlTranslatorFactory::CdbrXmlTranslatorFactory メソッド

機能

コンストラクタ

形式

`CdbrXmlTranslatorFactory()`

詳細

`CdbrXmlTranslatorFactory` オブジェクトのコンストラクタです。

`CdbrXmlTranslatorFactory` オブジェクトの生成は、必ずシングルスレッド環境下で実行してください。

引数

なし

戻り値

なし

戻り値の詳細

なし

CdbrXmlTranslatorFactory::CreateTranslator メソッド

機能

CdbrXmlTranslator オブジェクトの生成

形式

```
DmaBoolean CreateTranslator(
    CdbrXmlTranslator* pXmlTrans)
```

詳細

CdbrXmlTranslator オブジェクトを生成し、指定された領域に格納するメソッドです。CdbrXmlTranslatorFactory::Initialize メソッドを実行する前にこのメソッドを実行すると、CdbrXmlTranslatorFactory オブジェクトが未初期化状態のためエラーになります。

生成した CdbrXmlTranslator オブジェクトは、CdbrXmlTranslatorFactory オブジェクトが破棄されるまで有効です。CdbrXmlTranslatorFactory オブジェクトが破棄されると CdbrXmlTranslator オブジェクトは無効状態となり、CdbrXmlTranslator クラスの主要なメソッドは実行時にエラーになります。

また、生成した CdbrXmlTranslator オブジェクトは、生成時と異なるスレッド上では使用できません。

引数

pXmlTrans (出力)

生成された CdbrXmlTranslator オブジェクト (CdbrXmlTranslator 型の変数) が設定されます。生成されたオブジェクトを格納する領域のアドレスを指定してください。領域はユーザ側で確保してください。NULL を指定するとエラーになります。指定した領域に CdbrXmlTranslator オブジェクトがすでに格納されている場合、そのオブジェクトは破棄されます。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_NOT_INITIALIZED	オブジェクトが初期化されていません。	初期化を実行してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

CdbrXmlTranslatorFactory::Initialize メソッド

機能

オブジェクトの初期化

形式

```
DmaBoolean Initialize(
    const DmaId*      pDocSpaceId,
    const pDmaString_T pXmsFileName,
    ppDmaString_T    ppMessage)
```

詳細

指定した文書空間用に、CdbrXmlTranslatorFactory オブジェクトを初期化するメソッドです。

このメソッドでは、指定した xms ファイル (マッピングセット定義ファイル) に基づいて、プロパティマッピング情報の読み込みと解析を実行します。xms ファイル名に NULL を指定した場合、デフォルトの xms ファイル (ファイル名 "<文書空間識別子>.xms") を読み込みます。

デフォルトの xms ファイルおよび xms ファイルから参照される xmp ファイル (マッピング定義ファイル) は、原則として UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <インストールディレクトリ>%client%etc%xml_files に配置します。

マッピング定義情報の解析には秒単位の時間がかかるので、このメソッドの実行回数が少なくなるよう、初期化した CdbrXmlTranslatorFactory オブジェクトはプロセス内で使い回すことを推奨します。

定義情報の解析でエラーやワーニング (警告) が発生した場合、引数 ppMessage にその内容を示す文字列が設定されます。

このメソッドがすでに実行されている場合、初期化済みとしてエラーになります。

引数

pDocSpaceId (入力)

対象となる文書空間識別子を指定します。指定した文書空間が実際に存在するかどうかのチェックはしません。ただし、NULL を指定した場合はエラーになります。

pXmsFileName (入力)

使用する xms ファイルを指定します。

ファイルは、UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <インストールディレクトリ>%client%etc%xml_files を起点とした相対パスか、"file:/" で始まる絶対パスを指定します。また、NULL を指定した場合、UNIX の場合は <インストールディレクトリ>/client/etc/xml_files, Windows の場合は <インストールディレクトリ>%client%etc%xml_files にあるデフォルトの xms ファイル "<文書空間識別子>.xms" を使用して処理を実行します。

ppMessage (出力)

定義情報の解析などで発生したエラーやワーニング (警告) の詳細情報が設定されます。詳細情報を格納する領域のアドレスを指定してください。NULL を指定するとエラーになります。なお、返却する詳細情報がない場合は NULL が設定されます。ppMessage の領域は、dbrDelete 関数を使ってユーザプログラムで解放してください。

NULL が設定されるかどうかはこのメソッドの戻り値に依存しません。メソッドが正常終了した場合でも NULL 以外の値が設定される場合があります、また、異常終了した場合でも NULL が設定される場合があります。

取得した文字列は改行コードで区切られ、複数行で構成される場合があります。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

major_code	minor_code	説明	対処
ERR_DBR	ERR_OBJECT_ALREADY_INITIALIZED	オブジェクトは初期化済みです。	この操作は実行できません。
ERR_DBR	ERR_INVALID_DEFINITION	定義情報が不正です。	定義ファイルの内容を見直してください。
ERR_DMA	DMARC_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。
ERR_DMA	DMARC_NO_MEMORY	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。
ERR_DMA	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

20 関数

ここでは、クラスライブラリで提供する関数について説明します。

関数

dbrDelete 関数

dbrDmaldFromString 関数

関数

クラスライブラリでは、次の関数を提供しています。

関数名	説明
dbrDelete	メモリの解放
dbrDmaIdFromString	文字列から DmaId 型への変換

以降、関数について説明します。

dbrDelete 関数

機能

メモリの解放

形式

```

void dbrDelete(SDBR_PROPLIST* ptr)
void dbrDelete(SDBR_OBJLIST* ptr)
void dbrDelete(SDBR_VTOBJLIST* ptr)
void dbrDelete(SDBR_LINKEDOBJLIST* ptr)
void dbrDelete(SDBR_VTLINKEDOBJLIST* ptr)
void dbrDelete(DmaInteger32 cnt,
               SDBR_QUERYRESULT* ptr)
void dbrDelete(DmaBinaryValue* ptr)
void dbrDelete(pDmaString_T ptr)
void dbrDelete(SDBR_ELEMINFO* ptr)
void dbrDelete(SDBR_NODE* ptr)
void dbrDelete(SDBR_NODEINFO* ptr)
void dbrDelete(SDBR_RESOURCEINFO* ptr)
void dbrDelete(SDBR_VERSIONINFO* ptr)
void dbrDelete(SDBR_VROBJLIST* ptr)
void dbrDelete(SDBR_RENDITIONLIST* ptr)
void dbrDelete(SDBR_RELATIONLIST* ptr)
void dbrDelete(SDBR_DMAINFOLIST* ptr)
void dbrDelete(SDBR_CONTENTLIST* ptr)
void dbrDelete(SDBR_PATHLIST* ptr)

```

詳細

クラスライブラリのメソッドが確保したメモリを解放する関数です。

この関数は、ユーザプログラム側が new や malloc などによって確保したメモリに対しては使用できません。

dbrDelete 関数では、次に示す構造体として確保された領域を解放できます。

- SDBR_PROPLIST 構造体
- SDBR_OBJLIST 構造体
- SDBR_VTOBJLIST 構造体
- SDBR_LINKEDOBJLIST 構造体
- SDBR_VTLINKEDOBJLIST 構造体
- SDBR_QUERYRESULT 構造体
- SDBR_ELEMINFO 構造体
- SDBR_NODE 構造体
- SDBR_NODEINFO 構造体
- SDBR_RESOURCEINFO 構造体
- SDBR_VERSIONINFO 構造体
- SDBR_VROBJLIST 構造体
- SDBR_RENDITIONLIST 構造体
- SDBR_RELATIONLIST 構造体
- SDBR_DMAINFOLIST 構造体
- SDBR_CONTENTLIST 構造体
- SDBR_PATHLIST 構造体

引数

cnt (入力)

CdbrEqlStatement::GetResult メソッドによって確保されたメモリを解放する場合、
CdbrEqlStatement::GetResult メソッドで plSearchCount に設定された値を指定してください。指定し
ない場合、メモリが正常に解放されません。動作は保証されません。

ptr (入力)

解放するメモリのポインタを指定します。クラスライブラリのメソッドの戻り値として返されたポインタ
を指定してください。

戻り値

なし

戻り値の詳細

なし

dbrDmaIdFromString 関数

機能

文字列から DmaId 型への変換

形式

```
DmaBoolean dbrDmaIdFromString (
    pDmaString_T   pCvtString,
    pDmaId         pCvtId )
```

詳細

指定された文書空間識別子などの GUID 文字列を DmaId 型に変換する関数です。

ただし、次の場合は異常終了します。

- 指定された文字列または変換後 DmaId 型の格納領域のアドレスの値が NULL の場合
- 指定された文字列が 36 バイト以外の場合
- 指定された文字列の形式が、XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX ではない場合 (XXX は、16 進数文字列)
- 指定された文字列中の「-」以外の文字が「0 ~ 9」、「a ~ f」または「A ~ F」以外の場合 (ただし、「0 ~ 9」、「a ~ f」または「A ~ F」は、混在していてもかまいません)

なお、異常終了した場合に、変換後 DmaId 型の格納領域に格納される値は、不定です。

引数

pCvtString (入力)

変換対象の「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX」の形式 (XXX は、16 進数文字列) の文字列を指定してください。

pCvtId (出力)

変換後 DmaId 型の格納領域のアドレスを指定してください。

戻り値

DMA_TRUE 正常終了

DMA_FALSE 異常終了

戻り値の詳細

なし

付録

付録 A コーディング例

付録 B DMA オブジェクトのプロパティ

付録 C クラスライブラリのメソッドで使用する構造体

付録 D メソッドの実行に必要なパーミッション（アクセス制御機能を使用する場合）

付録 E メソッド実行時の排他制御（ロック）

付録 F OIID 指定時の URL の文法

付録 G 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メソッドと関数

付録 H 戻り値一覧

付録 I 障害が発生した時に DocumentBroker が出力する情報

付録 J 用語解説

付録 A コーディング例

ここでは、クラスライブラリを使用したコーディング例について説明します。

付録 A.1 作成するコーディング例

作成するコーディング例について説明します。

このコーディング例では、クラスライブラリのメソッドのコールシーケンスを例示します。また、クラスライブラリの各メソッドの使用方法を例示します。

ここで示すある業務の流れに従ったアプリケーションプログラムのコーディング例を参考にして、実際のアプリケーションプログラムを作成してください。

(1) 前提条件

前提条件は次のとおりです。

(a) 記述範囲

このコーディング例では、アプリケーションプログラムのうち、クラスライブラリを使用したインターフェースの個所だけを記述します。GUI や定型業務に依存した個所については説明しません。

また、このコーディング例は、クラスライブラリのすべてのメソッドを使用しているものではありません。

(b) データの定義

このコーディング例はデータを定義する機能は持たないため、あらかじめデータは定義されていることを前提とします。

したがって、クラス識別子やプロパティ識別子はすでにわかっているものとします。

また、文書およびフォルダのクラス定義は実装しません。別のツールを使用して定義してあり、このコーディング例からは変更できないものとします。

(c) プロパティ

このコーディング例で扱うプロパティは次のとおりです。

- 名前：DmaString_T 型
- 日付：DmaInteger32 型
- 作成者：DmaString_T 型

文書、フォルダおよび構成管理フォルダがこれらのプロパティを持ちます。これ以外のプロパティはないものとします。

(2) 作成するプログラム

コーディング例の仕様について説明します。

(a) コンポーネント

このコーディング例は次のコンポーネントを持ちます。

文書

管理の対象になる文書です。文書のデータ形式は規定しません。ファイルシステム上の 1 ファイルが 1 文書に対応します。CdbVersionableDocument クラスを使用して実装します。

フォルダ

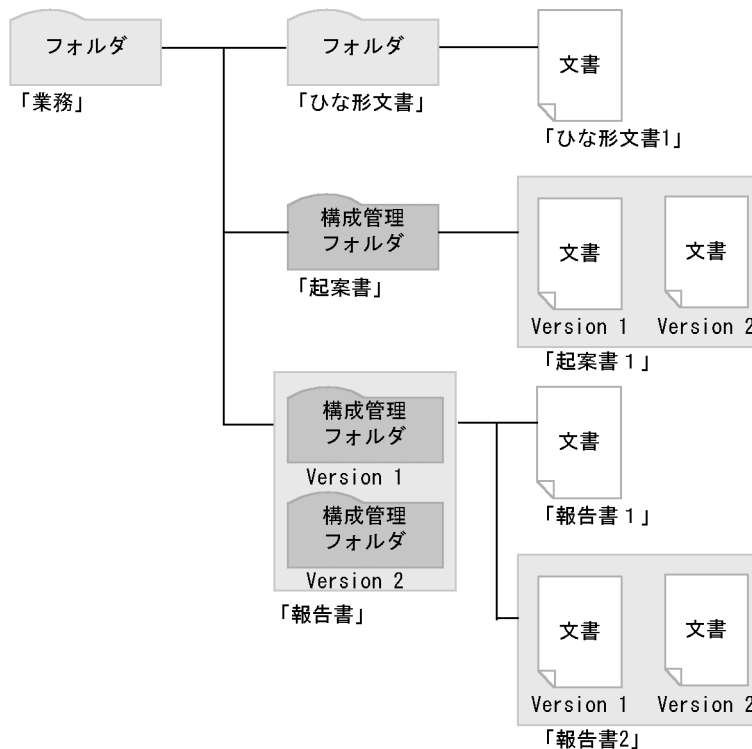
文書、フォルダまたは構成管理フォルダを格納して、一つのグループとして管理します。CdbReferentialContainer クラスを使用して実装します。

構成管理フォルダ

フォルダ自身にバージョンがあり、かつ管理する要素の特定のバージョンを指し示すことができるフォルダです。文書を格納できます。CdbConfiguredReferentialContainer クラスを使用して実装する、バージョン付き構成管理コンテナです。

ここでは、2階層のフォルダを持ち、文書を管理・検索する業務について説明します。コーディング例で扱うコンポーネントの構成について、図 A-1 に示します。

図 A-1 コーディング例で扱うコンポーネントの構成



「業務」というフォルダが最上位のフォルダです。このフォルダ下位にフォルダ「ひな形文書」と構成管理フォルダ「起案書」および「報告書」があります。

(b) 機能

このコーディング例の機能について説明します。

新規作成機能

文書、フォルダおよび構成管理フォルダを新規に作成します。

削除機能

文書、フォルダおよび構成管理フォルダを削除します。

関連づけ機能

文書、フォルダおよび構成管理フォルダを、別のフォルダまたは構成管理フォルダと関連づけます。ここで、文書是最も下位の要素になります。また、文書と文書を関連づけることはできません。

チェックアウト機能

文書をチェックアウトして、参照・編集します。

チェックイン機能

文書をチェックインして、文書のバージョンを一つ上げます。

バージョンアップ機能

構成管理フォルダのバージョンを上げます。

履歴一覧取得機能

文書および構成管理フォルダの履歴一覧を取得します。

検索機能

文書、フォルダおよび構成管理フォルダを、属性値を基に検索します。

付録 A.2 コーディング例

ここでは、これまでに説明したコーディング例の機能を実現する、コーディング例を示します。

なお、(1) でクラスライブラリの処理の開始から終了までを示します。それ以降は、文書空間とクライアント環境のセッションはすでに確立されているものとして、各機能に当たる部分のコーディング例だけを示します。

コーディング例についての注意

ここに示すコーディング例は UNIX の場合の例となっています。したがって、Windows をご使用の場合、コーディング例中の "file:///tmp/sample.txt" などのパス名、ファイル名は、"file:///c:/temp/sample.txt" のように読み替えてください。

(1) 文書の参照と登録

(a) 特定のフォルダへの文書の新規登録

ここでは、文書をフォルダ「業務」に格納するために、文書のパス名を業務のフォルダに関連づけるプログラムのコーディング例を示します。

図 A-2 に、特定のフォルダに文書を新規に登録するシーケンスを示します。

図 A-2 特定のフォルダに文書を新規に登録するシーケンス



コーディング例

```

1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //機能：特定のフォルダに文書を新規に登録する
3 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4 Bool InsertDocument(pDmaId pDocspID,
5     pDmaString_T pUsername, //認証用ユーザID
6     pDmaString_T pPasswd, //認証用パスワード
7     pDmaString_T pFilePath, //登録文書パス

```

```

8         pDmaString_T pFolderName, //登録フォルダ名
9         pDmaId pTargetId,        //名前プロパティID
10        pDmaString_T RootOIID    //ルートのフォルダID
11    )
12
13
14 {
15     CdbrSession* pSession;
16     SDBR_PROP*  pList = 0;
17     pDmaString_T pCFolderName = NULL;
18     pDmaString_T pErrorStrings = NULL;
19
20     DmaBoolean rc;
21     pDmaString_T pTargetOIID = NULL;
22     DmaInteger32 lTotal;
23     SDBR_DMAINFO TmpDma[2];
24     DmaInteger32 lmajor,lminor;
25
26     //////////////////////////////////////
27     //1. 開始処理
28     //////////////////////////////////////
29
30     //1.1 CdbrSessionオブジェクトの生成
31     pSession = new CdbrSession();
32     if(pSession == NULL){
33         return FALSE;
34     }
35     //1.2 文書空間との接続
36     rc = pSession->Connect (pDocspID,pUsername,pPasswd);
37     if(rc != DMA_TRUE){
38         lmajor = pSession->GetLastError(&lminor,&pErrorStrings);
39         printf(" ERROR !! major %x minor %x %s¥n",
40             lmajor,lminor,
41             (pErrorStrings == NULL) ? "NULL":pErrorStrings);
42         return FALSE;
43     }
44     pSession->Begin();
45     //////////////////////////////////////
46     //2. 文書の生成と登録
47     //////////////////////////////////////
48     CdbrVersionableDocument VRDoc;
49     pDmaString_T pOIID;
50     //テンプレートの作成
51     TmpDma[0].ClassId = mdmClass_ConfigHist; //すでに知っている
52     TmpDma[0].PropList.lCount = 0;         //プロパティリストは0
53     TmpDma[0].PropList.pItem = NULL;      //NULLを指定すると
54                                           //プロパティ値なしで生成
55     TmpDma[1].ClassId = mdmClass_DocVer;  //すでに知っている
56     TmpDma[1].PropList.lCount = 0;         //プロパティリストは0
57     TmpDma[1].PropList.pItem = NULL;      //NULLを指定すると
58                                           //プロパティ値なしで生成
59
60     //2.1 CdbrVersionableDocumentオブジェクトの生成
61     // (プロパティ値はNULL)
62     VRDoc.CreateObject (pSession,
63         2,
64         TmpDma,
65         "file:///tmp/sample.txt",
66         "MIME::text/plain",
67         &pOIID
68     );
69
70     //////////////////////////////////////
71     //3. 登録フォルダを検索
72     //////////////////////////////////////
73

```

```

74 //3.1 ルートフォルダのOIIDの設定
75 CdbReferentialContainer RootContainer;
76 rc = RootContainer.SetOIID(pSession,RootOIID);
77 if(rc != DMA_TRUE){
78     lmajor = RootContainer.GetLastError(&lminor,
79                                         &pErrorStrings);
80     printf(" ERROR !! major %x minor %x %s¥n",
81           lmajor,lminor,
82           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
83     return FALSE;
84 }
85
86 //3.2 登録するフォルダの検索
87 //指定フォルダの子フォルダの一覧を取得する
88 //     フォルダの個数をカウント
89 SDBR_PROPDEF PropA;
90 SDBR_PROPLIST* pPropList;
91 DmaBoolean bContinue = DMA_FALSE;
92 DmaInteger32 cnt;
93
94 PropA.PropId = dbrProp_ContaineesCount;
95
96 rc = RootContainer.GetPropertyValues(1,&PropA,&pPropList);
97 if(rc == DMA_TRUE &&
98     *pPropList->pItem[0].uniValue.plInteger32 > 0){
99     lTotal = *pPropList->pItem[0].uniValue.plInteger32;
100 }
101
102 //全子フォルダのプロパティ情報を取得
103 SDBR_OBJLIST* pObjList = NULL;
104 SDBR_PROPDEF PropB;
105 PropB.PropId = usrProp_FolderName; //ユーザ定義のプロパティ
106 rc = RootContainer.GetContainableList(&bContinue,
107                                     DBR_CONTAINMENT_REFERENTIAL,
108                                     1,
109                                     &PropB,
110                                     lTotal,
111                                     &pObjList);
112 if(rc != DMA_TRUE){
113     lmajor = RootContainer.GetLastError(&lminor,
114                                         &pErrorStrings);
115     printf(" ERROR !! major %x minor %x %s¥n",
116           lmajor,lminor,
117           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
118     dbrDelete(pObjList);
119     return FALSE;
120 }
121
122 //指定された名前のフォルダを取得する
123 int j;
124 for(j = 0; j < pObjList->lCount; j++){
125     pList = pObjList->pItem[j].PropList.pItem;
126     pFolderName = GetPropValue(pList,
127                               pObjList->pItem[j].PropList.lCount,pTargetId);
128
129     if(strcmp(pFolderName,pCFolderName) == 0)
130     {
131         break;
132     }
133 }
134 //フォルダのOIIDを取得する
135 pTargetOIID = pObjList->pItem[j].pOIID;
136 //目的のフォルダのOIIDの設定
137 CdbReferentialContainer TargetContainer;
138 rc = TargetContainer.SetOIID(pSession,pTargetOIID);
139 if(rc != DMA_TRUE){

```

```

140     lmajor = TargetContainer.GetLastError(&lminor,
141                                           &pErrorStrings);
142     printf("ERROR !! major %x minor %x %s¥n",
143           lmajor,lminor,
144           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
145     return FALSE;
146 }
147 //3.3フォルダと文書の関連づけ
148 rc = TargetContainer.Link(&VRDoc);
149 if(rc != DMA_TRUE){
150     lmajor = TargetContainer.GetLastError(&lminor,
151                                           &pErrorStrings);
152     printf(" ERROR !! major %x minor %x %s¥n",
153           lmajor,lminor,
154           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
155     return FALSE;
156 }
157 //ファイル実体のアップデート
158 VRDoc.UpdateContent(pFilePath,NULL);
159 //3.4 確定処理
160 pSession->Commit();
161
162 ///////////////////////////////////////////////////////////////////
163 //4. 終了処理
164 ///////////////////////////////////////////////////////////////////
165 //4.1 オブジェクトとの接続の解除
166 dbrDelete (pObjList);
167 VRDoc.ReleaseObject();
168 RootContainer.ReleaseObject();
169 TargetContainer.ReleaseObject();
170 //4.2 セッションの終了
171 pSession->Disconnect();
172 delete pSession;
173 return TRUE;
174 }
175 ///////////////////////////////////////////////////////////////////
176 //SUB: プロパティIDがtargetIdであるもののプロパティ値を取得する
177 ///////////////////////////////////////////////////////////////////
178 pDmaString_T GetPropValue(SDBR_PROP* pPropList,
179                           DmaInteger32 lListCount,
180                           pDmaId       pTargetId)
181 {
182     int i;
183     for (i = 0 ; i < lListCount; i++) {
184         //目的のプロパティIDと一致したらプロパティ値をリターンする
185         if (memcmp(&pPropList[i].PropId,
186                 pTargetId,
187                 sizeof *pTargetId) == 0)
188             {
189                 return *pPropList[i].uniValue.ppString;
190             }
191     }
192     return NULL;
193 }

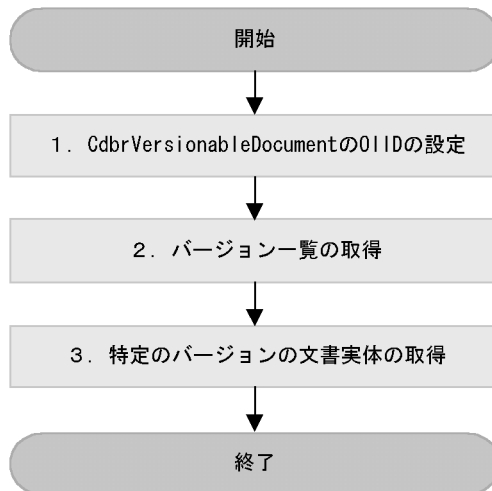
```

(b) 特定の文書を取り出した参照

ここでは、特定の文書を指定して参照するプログラムのコーディング例を示します。文書はバージョン管理されています。ここで参照とは、文書の実体を、指定したファイルに複写することです。このコーディング例では、文書の OIID およびバージョン番号および取り出した文書の実体を複写するファイルを指定して、関数を発行します。

図 A-3 に、特定の文書を取り出して参照するシーケンスを示します。

図 A-3 特定の文書を取り出して参照するシーケンス



コーディング例

```

1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //機能：特定の文書を取り出して参照する
3 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4 Bool GetVersionDoc(CdbrSession* pSession,
5     pDmaString_T pVRDOCId,
6     pDmaString_T pFilePath,
7     DmaInteger32 VersionCount)
8 {
9     DmaBoolean rc;
10    SDBR_DMAINFO TmpDmaConfigHist;
11    SDBR_DMAINFO TmpDmaDocVer;
12    pDmaString_T pErrorStrings = NULL;
13    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
14    //1. CdbrVersionableDocumentのOIIDの設定
15    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
16    CdbrVersionableDocument VRDoc;
17    pSession->Begin();
18    //VRDOCオブジェクトのOIIDの設定
19    rc = VRDoc.SetOIID(pSession,pVRDocId);
20    if(rc != DMA_TRUE)
21    {
22        lmajor = VRDoc.GetLastError(&lminor,&pErrorStrings);
23        printf(" ERROR !! major %x minor %x %s¥n",
24            lmajor,lminor,
25            (pErrorStrings == NULL) ? "NULL":pErrorStrings);
26    }
27    return FALSE;
28 }
29 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
30 //2. バージョン一覧の取得
31 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
32 SDBR_OBJLIST* pOBJList = NULL; //バージョンは10個
33 DmaBoolean bContinue = DMA_FALSE;
34 DmaInteger32 lFetchCount;
35 rc = VRDoc.GetVersionList(&bContinue,
36     0, //プロパティリストは0
37     NULL, //プロパティは取得しない
38     10,
39     &pOBJList);
40 if(rc != DMA_TRUE)
41 {
42     lmajor = VRDoc.GetLastError(&lminor,&pErrorStrings);

```

```

43     printf(" ERROR !! major %x minor %x %s¥n",
44           lmajor, lminor,
45           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
46     return FALSE;
47   }
48   //////////////////////////////////////
49   //3. 特定のバージョンの文書実体の取得
50   //////////////////////////////////////
51   rc = VRDoc.GetContent(pFilePath,
52                       pOBJList->pItem[VersionCount].pOIID);
53   if(rc != DMA_TRUE){
54     lmajor = VRDoc.GetLastError(&lminor, &pErrorStrings);
55     printf(" ERROR !! major %x minor %x %s¥n",
56           lmajor, lminor,
57           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
58     return FALSE;
59   }
60   dbrDelete(pOBJList);
61   VRDoc.ReleaseObject();
62   pSession->Commit();
63   return TRUE;
64 }

```

(c) CdbrEqStatement クラスを使用した文書の検索

ここでは CdbrEqStatement クラスを使用して、edmSQL 文で記述した検索条件を実行し、検索結果を取得するプログラムのコーディング例を示します。

次の図に、CdbrEqStatement クラスを使用して検索するシーケンスを示します。

図 A-4 CdbrEqIStatement クラスを使用して検索するシーケンス



コーディング例

```

1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //機能：CdbrEqIStatementクラスを使用して文書を検索する
3 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4 Bool SearchObject(
5     CdbrSession* pSession
6 )
7 {
8     CdbrEqIStatement* pEqI = NULL;
9     DmaBoolean rc;
10    pDmaString_T pErrorStrings = NULL;
11    pSession->Begin();
12
13 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
14 //1. CdbrEqIStatementオブジェクトの生成
15 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
16    pEqI = new CdbrEqIStatement();
17

```

```

18 ///////////////////////////////////////////////////////////////////
19 //2. オブジェクトの初期化とセッションオブジェクトへの登録
20 ///////////////////////////////////////////////////////////////////
21 rc = pEq1->Initialize(pSession);
22 if ( DMA_TRUE != rc ) {
23     //エラー処理
24     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
25     printf("ERROR !! major %x minor %x %s¥n",
26           lmajor, lminor,
27           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
28 }
29
30 ///////////////////////////////////////////////////////////////////
31 //3. edmsSQL文の設定
32 ///////////////////////////////////////////////////////////////////
33 rc = pEq1->Set( (pDmaString_T)
34               "Select dmaProp_OIID"
35               " From usrClass_Folder"
36               " Where (usrProp_Date = ?) Or (usrProp_No = ?)");
37 if ( DMA_TRUE != rc ) {
38     //エラー処理
39     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
40     printf("ERROR !! major %x minor %x %s¥n",
41           lmajor, lminor,
42           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
43 }
44
45 ///////////////////////////////////////////////////////////////////
46 //4. edmsSQL文の?パラメタの設定
47 //   ?パラメタの値の設定(?(1):19991201 ?(2):100)
48 ///////////////////////////////////////////////////////////////////
49 rc = pEq1->SetParam(1, (pDmaString_T)"19991201");
50 if ( DMA_TRUE != rc ) {
51     //エラー処理
52     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
53     printf("ERROR !! major %x minor %x %s¥n",
54           lmajor, lminor,
55           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
56 }
57
58 rc = pEq1->SetParam(2, 100);
59 if ( DMA_TRUE != rc ) {
60     //エラー処理
61     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
62     printf("ERROR !! major %x minor %x %s¥n",
63           lmajor, lminor,
64           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
65 }
66
67 ///////////////////////////////////////////////////////////////////
68 //5. ロック種別の設定 (Writeロック)
69 ///////////////////////////////////////////////////////////////////
70 DmaInteger32 lLockType = DBR_LOCK_WRITE;
71 rc = pEq1->ChangeLockType(lLockType);
72 if ( DMA_TRUE != rc ) {
73     //エラー処理
74     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
75     printf("ERROR !! major %x minor %x %s¥n",
76           lmajor, lminor,
77           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
78 }
79
80 ///////////////////////////////////////////////////////////////////
81 //6. 検索結果取得件数の設定 (100件)
82 ///////////////////////////////////////////////////////////////////
83 DmaInteger32 lGetObjCount = 100;
84 rc = pEq1->ChangeGetObjCount(lGetObjCount);
85 if ( DMA_TRUE != rc ) {
86     //エラー処理
87     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
88     printf("ERROR !! major %x minor %x %s¥n",
89           lmajor, lminor,
90           (pErrorStrings == NULL) ? "NULL":pErrorStrings);

```



```

91  }
92
93  ///////////////////////////////////////////////////////////////////
94  //7. アクセス制御モードの設定 (DBR_QUERY_WITH_ACL)
95  ///////////////////////////////////////////////////////////////////
96  DmaInteger32 lACLMode = DBR_QUERY_WITH_ACL;
97  rc = pEq1->ChangeACLMode(lACLMode);
98  if ( DMA_TRUE != rc ) {
99      //エラー処理
100     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
101     printf("ERROR !! major %x minor %x %s¥n",
102           lmajor, lminor,
103           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
104 }
105
106 ///////////////////////////////////////////////////////////////////
107 //8. 設定済みedmsSQL文の実行
108 ///////////////////////////////////////////////////////////////////
109 pDmaString_T pErrorMessage = NULL;
110 rc = pEq1->Execute(&pErrorMessage);
111 if ( DMA_TRUE != rc ) {
112     //エラー処理
113     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
114     printf("ERROR !! major %x minor %x %s¥n",
115           lmajor, lminor,
116           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
117     if ((ERR_DBR == lmajor) &&
118         (ERR_EQL_BAD_STATEMENT == lminor)) {
119         printf("StatementError !! %s¥n", pErrorMessage);
120     }
121 }
122 dbrDelete(pErrorMessage);
123 pErrorMessage = NULL;
124
125 ///////////////////////////////////////////////////////////////////
126 //9. 結果を100件ずつ最後まで取得
127 ///////////////////////////////////////////////////////////////////
128 DmaBoolean bContinue = DMA_TRUE;
129 DmaInteger32 lSearchCount = lGetObjCount;
130 SDBR_QUERYRESULT* pQueryResult = NULL;
131
132 while ( lGetObjCount <= lSearchCount )
133 {
134     rc = pEq1->GetResult( bContinue, &lSearchCount,
135                          &pQueryResult );
136     if ( DMA_TRUE != rc ) {
137         //エラー処理
138         lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
139         printf("ERROR !! major %x minor %x %s¥n",
140               lmajor, lminor,
141               (pErrorStrings == NULL) ? "NULL":pErrorStrings);
142     }
143     dbrDelete(lSearchCount, pQueryResult);
144 }
145
146 ///////////////////////////////////////////////////////////////////
147 //10. セッションオブジェクトへの登録抹消
148 ///////////////////////////////////////////////////////////////////
149 rc = pEq1->Terminate();
150 if ( DMA_TRUE != rc ) {
151     //エラー処理
152     lmajor = pEq1->GetLastError(&lminor, &pErrorStrings);
153     printf("ERROR !! major %x minor %x %s¥n",
154           lmajor, lminor,
155           (pErrorStrings == NULL) ? "NULL":pErrorStrings);
156 }
157
158 ///////////////////////////////////////////////////////////////////
159 //11. CdbrEq1Statementオブジェクトを削除
160 ///////////////////////////////////////////////////////////////////
161 delete pEq1;
162 pEq1 = NULL;
163

```

```

164  pSession->Commit();
165  return TRUE;
166  }

```

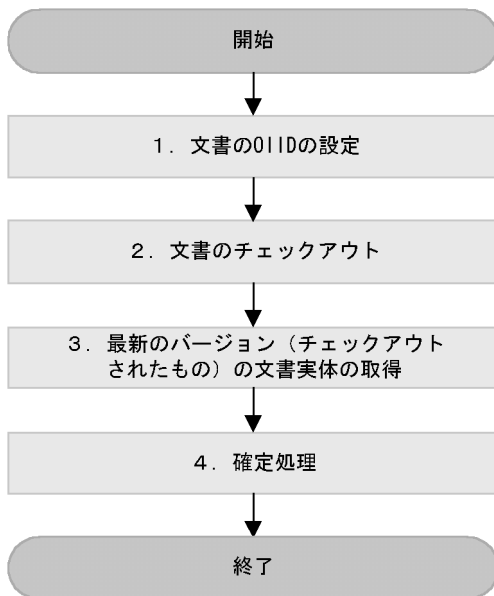
(2) 文書のバージョン管理

(a) 文書のチェックアウト

ここでは、最新バージョンの文書をチェックアウトして、編集できるようにするプログラムのコーディング例を示します。

次の図に、文書をチェックアウトするシーケンスを示します。

図 A-5 文書をチェックアウトするシーケンス



コーディング例

```

1  ////////////////////////////////////////////////////////////////////
2  //機能：文書をチェックアウトする
3  ////////////////////////////////////////////////////////////////////
4  Bool GetVersionDoc(CdbrSession* pSession,
5                    pDmaString_T pVRDOCId,
6                    pDmaString_T pFilePath,
7                    ppDmaString_T ppVersionID)
8  {
9      DmaBoolean rc;
10     CdbrVersionableDocument VRDOC;
11     ppDmaString_T ppVerId=NULL;
12     pDmaString_T pErrorStrings = NULL;
13     pSession->Begin();
14     ////////////////////////////////////////////////////////////////////
15     //1.文書のOIDの設定
16     ////////////////////////////////////////////////////////////////////
17     rc = VRDOC.SetOID(pSession, pVRDOCId);
18     if(rc != DMA_TRUE){
19         lmajor = VRDOC.GetLastError(&lminor,&pErrorStrings);
20         printf(" ERROR !! major %x minor %x %s¥n",
21              lmajor,lminor,
22              (pErrorStrings == NULL) ? "NULL":pErrorStrings);
23     }
24     return FALSE;
25     ////////////////////////////////////////////////////////////////////

```

```

26 //2.文書のチェックアウト
27 //////////////////////////////////////
28 rc = VRDOC.VersionCheckOut(&pVerId);
29 if(rc != DMA_TRUE){
30     lmajor = VRDOC.GetLastError(&lminor,&pErrorStrings);
31     printf(" ERROR !! major %x minor %x %s¥n",
32         lmajor,lminor,
33         (pErrorStrings == NULL) ? "NULL":pErrorStrings);
34     return FALSE;
35 }
36 //////////////////////////////////////
37 //3.最新のバージョン(チェックアウトされたもの)の文書実体の取得
38 //////////////////////////////////////
39 rc = VRDOC.GetContent(pFilePath,pVerId);
40 if(rc != DMA_TRUE){
41     lmajor = VRDOC.GetLastError(&lminor,&pErrorStrings);
42     printf(" ERROR !! major %x minor %x %s¥n",
43         lmajor,lminor,
44         (pErrorStrings == NULL) ? "NULL":pErrorStrings);
45     return FALSE;
46 }
47 //////////////////////////////////////
48 //4. 確定処理
49 //////////////////////////////////////
50 pSession->Commit();
51
52 VRDOC.ReleaseObject();
53 *ppVersionID = (char *)malloc(strlen(pVerId)+1);
54 strcpy(*ppVersionID,pVerId);
55 dbrDelete(pVerId);
56 return TRUE;
57 }

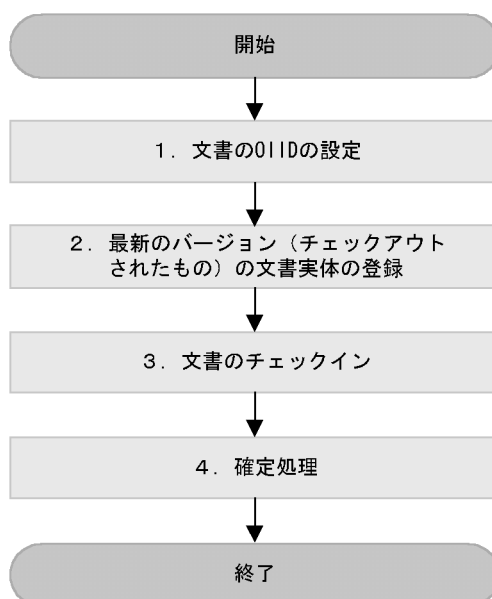
```

(b) チェックアウトした文書をチェックインして登録

ここでは、チェックアウトした文書を、チェックインして登録するプログラムのコーディング例を示します。

次の図に、チェックアウトした文書をチェックインして登録するシーケンスを示します。

図 A-6 チェックアウトした文書をチェックインして登録するシーケンス



コーディング例

```

1 ////////////////////////////////////////////////////////////////////
2 //機能：チェックアウトした文書をチェックインして登録する
3 ////////////////////////////////////////////////////////////////////
4 Bool EntryDoc(CdbrSession* pSession,
5             pDmaString_T pFilePath,
6             pDmaString_T pVRDOCId,
7             pDmaString_T pVerId //CheckOutメソッドで取得したもの
8 )
9 {
10     DmaBoolean rc;
11     CdbrVersionableDocument VRDOC;
12     pDmaString_T pErrorStrings = NULL;
13     pSession->Begin();
14 ////////////////////////////////////////////////////////////////////
15 //1. 文書のOIDの設定(プロパティ値はNULL)
16 ////////////////////////////////////////////////////////////////////
17     rc = VRDOC.SetOID(pSession, pVRDOCId);
18     if(rc != DMA_TRUE){
19         lmajor = VRDOC.GetLastError(&lminor, &pErrorStrings);
20         printf(" ERROR !! major %x minor %x %s¥n",
21             lmajor, lminor,
22             (pErrorStrings == NULL) ? "NULL":pErrorStrings);
23         return FALSE;
24     }
25
26 ////////////////////////////////////////////////////////////////////
27 //2. 最新のバージョン(チェックアウトされたもの)の文書実体の登録
28 ////////////////////////////////////////////////////////////////////
29     rc = VRDOC.UpdateContent(pFilePath, pVerId);
30     if(rc != DMA_TRUE){
31         lmajor = VRDOC.GetLastError(&lminor, &pErrorStrings);
32         printf(" ERROR !! major %x minor %x %s¥n",
33             lmajor, lminor,
34             (pErrorStrings == NULL) ? "NULL":pErrorStrings);
35         return FALSE;
36     }
37 ////////////////////////////////////////////////////////////////////
38 //3. 文書のチェックイン
39 ////////////////////////////////////////////////////////////////////
40     rc = VRDOC.VersionCheckIn();
41     if(rc != DMA_TRUE){
42         lmajor = VRDOC.GetLastError(&lminor, &pErrorStrings);
43         printf(" ERROR !! major %x minor %x %s¥n",
44             lmajor, lminor,
45             (pErrorStrings == NULL) ? "NULL":pErrorStrings);
46         return FALSE;
47     }
48 ////////////////////////////////////////////////////////////////////
49 //4. 確定処理
50 ////////////////////////////////////////////////////////////////////
51     pSession->Commit();
52     VRDOC.ReleaseObject();
53     return TRUE;
54 }

```

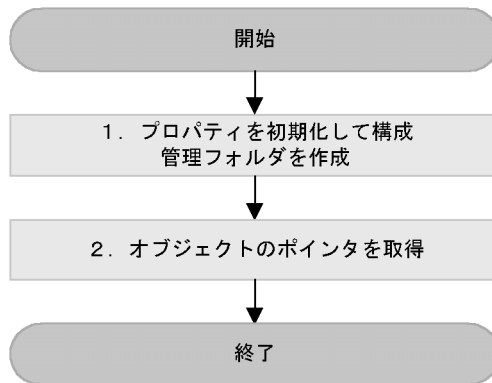
(3) 文書の構成管理

(a) 構成管理フォルダの作成

ここでは、構成管理フォルダを新規に作成するプログラムのコーディング例を示します。

次の図に、構成管理フォルダを新規に作成するシーケンスを示します。

図 A-7 構成管理フォルダを新規に作成するシーケンス



コーディング例

```

1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //機能：構成管理フォルダの新規作成
3 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4 Bool CreateVRFolder(CdbrSession* pSession,
5     ConfiguratedReferentialContainer** ppCFRFCT)
6 {
7     SDBR_DMMAINFO TmpDmaCFRFCT;
8     SDBR_PROPLIST PropList;
9     CdbrConfiguratedReferentialContainer* pCFRFContainer;
10    pCFRFContainer = new CdbrConfiguratedReferentialContainer();
11    DmaBoolean rc;
12    pDmaString_T pErrorStrings = NULL;
13    pSession->Begin();
14    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
15    //1. プロパティを初期化して構成管理フォルダを作成
16    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
17    pDmaString_T pOIID = NULL; //生成するフォルダのOIID
18    TmpDmaCFRFCT.ClassId = ID_ConfigHist; //すでに知っている
19    TmpDmaCFRFCT.PropList.lCount = 0; //プロパティ値なしで生成
20    TmpDmaCFRFCT.PropList.pItem = NULL; //プロパティ値なしで生成
21
22    rc = pCFRFContainer->CreateObject(pSession,
23        1,
24        &TmpDmaCFRFCT,
25        &pOIID
26    );
27    if(rc != DMA_TRUE){
28        lmajor = pCFRFContainer->GetLastError(&lminor,
29            &pErrorStrings);
30        printf(" ERROR !! major %x minor %x %s\n",
31            lmajor, lminor,
32            (pErrorStrings == NULL) ? "NULL":pErrorStrings);
33        return FALSE;
34    }
35    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
36    //2. オブジェクトのポインタを取得
37    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
38    *ppCFRFCT = pCFRFContainer;
39
40    pCFRFContainer->ReleaseObject();
41    pSession->Commit();
42    return TRUE;
43 }
  
```

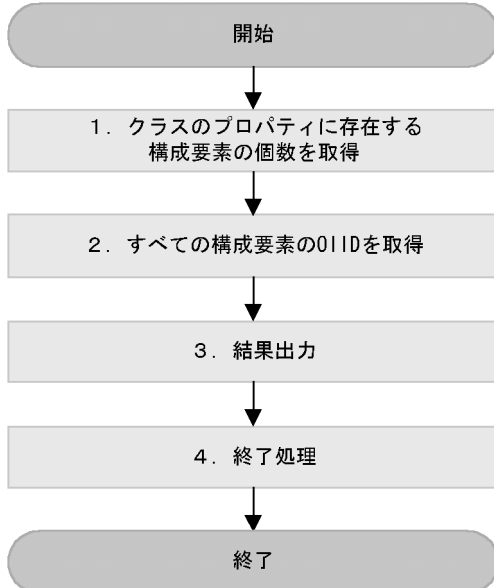
(b) 構成管理フォルダの構成要素の一覧取得

ここでは、構成管理フォルダが管理している特定のバージョンの文書の一覧を取得するプログラムのコー

コーディング例を示します。

次の図に、構成管理フォルダの構成要素の一覧を取得するシーケンスを示します。

図 A-8 構成管理フォルダの構成要素の一覧を取得するシーケンス



コーディング例

```

1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //機能：構成管理フォルダの構成要素一覧の取得
3 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4 Bool GetContaineer(CdbrSession *pSession,
5     CdbrConfiguredReferentialContainer* pCFRFCT)
6 {
7     DmaBoolean bContinue = DMA_TRUE;
8     DmaInteger32 FetchCount;
9     DmaInteger32 ObjListCount;
10    DmaBoolean rc;
11    pDmaString_T pErrorStrings = NULL;
12    pSession->Begin();
13 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
14 //1. クラスのプロパティに存在する構成要素の個数を取得する
15 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
16    SDBR_PROPDEF PropA;
17    PropA.PropId = dbrProp_ContaineesCountVT;
18                                //クラスのプロパティを指定
19    PropA.lType = DMA_DATATYPE_INTEGER32;
20    PropA.lCardinality = DMA_CARDINALITY_SINGLE;
21    SDBR_PROP* pPropT;
22    SDBR_PROPLIST* pPropList;
23    DmaInteger32 cnt;
24    DmaInteger32 iPropCount = 1;
25    rc = pCFRFCT->GetPropertyValues(NULL,
26        iPropCount, &PropA, &pPropList); //一つだけ取得
27    if(rc == DMA_TRUE){
28        lmajor = pCFRFCT->GetLastError(&lminor, &pErrorStrings);
29        printf(" ERROR !! major %x minor %x %s¥n",
30            lmajor, lminor,
31            (pErrorStrings == NULL) ? "NULL":pErrorStrings);
32        return FALSE;
33    }
34    pPropT = &pPropList->pItem[0];

```

```

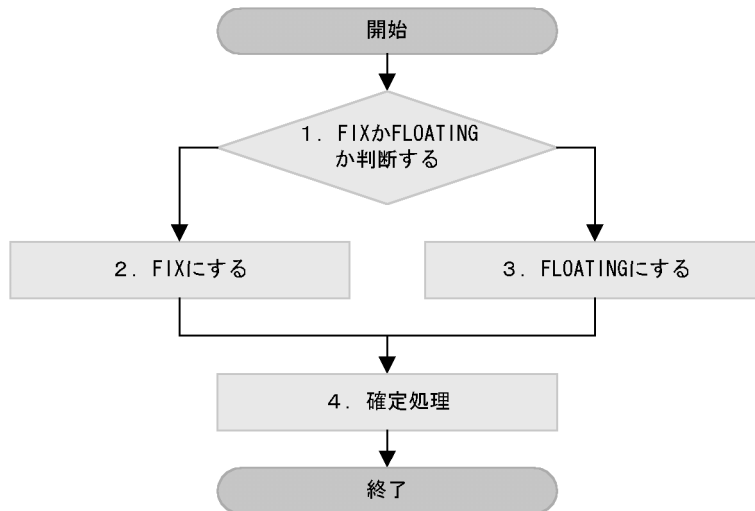
35  if (rc == DMA_TRUE && *pPropT->uniValue.plInteger32 > 0){
36      ObjListCount = *pPropT->uniValue.plInteger32;
37  }
38  ///////////////////////////////////////////////////////////////////
39  //2. すべての構成要素のOIDを取得
40  ///////////////////////////////////////////////////////////////////
41  SDBR_VTOBJLIST* pObjList = NULL;
42  rc = pCFRFCT->GetVTContaineelist( &bContinue,
43      NULL,          //プロパティは取得しない
44      0,            //プロパティは取得しない
45      NULL,
46      ObjListCount,
47      &pObjList
48  );
49  if(rc != DMA_TRUE){
50      lmajor = pCFRFCT->GetLastError(&lminor, &pErrorStrings);
51      printf(" ERROR !! major %x minor %x %s¥n",
52          lmajor, lminor,
53          (pErrorStrings == NULL) ? "NULL":pErrorStrings);
54      return FALSE;
55  }
56  ///////////////////////////////////////////////////////////////////
57  //3. 結果出力
58  ///////////////////////////////////////////////////////////////////
59  int i;
60  for(i = 0; i < pObjList->lCount; i++)
61  {
62      printf("Id = %s", pObjList->pItem[i].pOID);
63  }
64  dbrDelete(pObjList);
65  ///////////////////////////////////////////////////////////////////
66  //4. 終了処理
67  ///////////////////////////////////////////////////////////////////
68  dbrDelete(pPropList);
69  pCFRFCT->ReleaseObject();
70  delete pCFRFCT;
71  pSession->Commit();
72  return TRUE;
73 }

```

(c) 構成管理フォルダが管理する文書の構成管理モードの切り替え

ここでは、構成管理フォルダが管理する文書のバージョンの構成管理モードである、FIX モードと FLOATING モードを切り替えるプログラムのコーディング例を示します。

図 A-9 構成管理フォルダが管理する文書の構成管理モードを切り替えるシーケンス



コーディング例

```

1 ///////////////////////////////////////////////////////////////////
2 //機能：構成管理フォルダが管理する文書の構成管理モードを切り替える
3 ///////////////////////////////////////////////////////////////////
4 Bool ChangeFix(CdbrSession* pSession,
5               CdbrConfiguredReferentialContainer* pCFRFCT,
6               pDmaString_T pLinkId,
7               pDmaString_T pAction,
8               pDmaString_T pOIID)
9 {
10    DmaBoolean rc;
11    pDmaString_T pErrorStrings = NULL;
12    pSession->Begin();
13    pCFRFCT->SetOIID(pSession, pOIID);
14 ///////////////////////////////////////////////////////////////////
15 //1. FIXかFLOATか判断する
16 ///////////////////////////////////////////////////////////////////
17 ///////////////////////////////////////////////////////////////////
18 //2. FIXにする
19 ///////////////////////////////////////////////////////////////////
20    if(!(strcmp("FIX",pAction)))
21    {
22        rc = pCFRFCT->SetVTFix(pLinkId,NULL);
23        if(rc != DMA_TRUE)
24        {
25            lmajor = pCFRFCT->GetLastError(&lminor, &pErrorStrings);
26            printf(" ERROR !! major %x minor %x %s¥n",
27                lmajor,lminor,
28                (pErrorStrings == NULL) ? "NULL":pErrorStrings);
29            return FALSE;
30        }
31    }
32 ///////////////////////////////////////////////////////////////////
33 //3. FLOATにする
34 ///////////////////////////////////////////////////////////////////
35    else if(!(strcmp("FLOAT",pAction)))
36    {
37        rc = pCFRFCT->SetVTFloat(pLinkId);
38        if(rc != DMA_TRUE)
39        {
40            lmajor = pCFRFCT->GetLastError(&lminor, &pErrorStrings);
41            printf(" ERROR !! major %x minor %x %s¥n",
42                lmajor,lminor,

```



```

43         (pErrorStrings == NULL) ? "NULL":pErrorStrings);
44     return FALSE;
45 }
46 }else{
47     lmajor = pCFRFCT->GetLastError(&lminor,&pErrorStrings);
48     printf(" ERROR !! major %x minor %x %s¥n",
49         lmajor,lminor,
50         (pErrorStrings == NULL) ? "NULL":pErrorStrings);
51     return FALSE;
52 }
53 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
54 //4. 確定処理
55 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
56     pSession->Commit();
57     return TRUE;
58 }

```

(4) XML 文書の管理

ここでは、XML 文書を新規登録する方法として、XML ファイルから DMA 情報リストと構造指定検索用のインデクスデータを生成し、その情報を基に文書オブジェクトの作成と構造指定検索インデクスの登録をするコーディング例で説明します。なお、コーディング例中で使用しているエラー判定関数 `checkAndThrow` の定義は次のとおりです。

エラー判定関数 `checkAndThrow` の定義

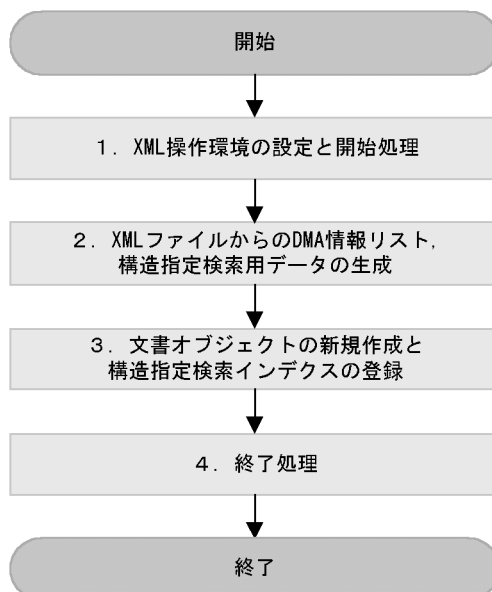
```

// 条件式の判定と例外のスロー
void checkAndThrow(int condition, CdbrCore& rObj)
{
    // 条件式が成り立たない場合、マイナーコードをスローする
    if (!condition) {
        DmaInteger32 lMinor = 0;
        rObj.GetLastError(&lMinor);
        throw DmaInteger32(lMinor);
    }
}

```

次の図に、XML 文書を新規に登録するシーケンスを示します。

図 A-10 XML 文書を新規に登録するシーケンス



コーディング例

```

1 ///////////////////////////////////////////////////////////////////
2 // XML文書の新規登録
3 ///////////////////////////////////////////////////////////////////
4
5 void CreateObject(
6     pDmaId      pDocSpaceId, // 文書空間ID
7     pDmaString_T pUserId,    // ユーザID
8     pDmaString_T pPassword,  // パスワード
9     pDmaString_T pXmlFile,   // XMLファイル
10    pDmaString_T pMapDefId)   // マッピング定義ID
11 {
12     CdbrSession          session;
13     CdbrVersionableDocument verDoc;
14     CdbrXmlTranslatorFactory xmlTransFactory;
15     CdbrXmlTranslator      xmlTrans;
16     SDBR_DMAINFOLIST*     pDmaInfoList = NULL;
17     pDmaString_T          pOIID = NULL;
18     pDmaString_T          pMessage = NULL;
19     DmaBoolean            bRc;
20
21 ///////////////////////////////////////////////////////////////////
22 // 1. XML操作環境の設定と開始処理
23 ///////////////////////////////////////////////////////////////////
24 // XML管理オブジェクトの初期化
25     bRc = xmlTransFactory.Initialize(pDocSpaceId,
26                                     NULL, &pMessage);
27     if (pMessage != NULL) {
28         cout << "message=" << pMessage << endl;
29     }
30     dbrDelete(pMessage);
31     checkAndThrow(bRc, xmlTransFactory);
32
33 // CdbrXmlTranslatorオブジェクトの生成
34     bRc = xmlTransFactory.CreateTranslator(&xmlTrans);
35     checkAndThrow(bRc, xmlTransFactory);
36
37 // セッションの接続
38     bRc = session.Connect(pDocSpaceId, pUserId, pPassword);
39     checkAndThrow(bRc, session);
40
41 // トランザクションの開始
42     bRc = session.Begin();
43     checkAndThrow(bRc, session);
44
45 ///////////////////////////////////////////////////////////////////
46 // 2. XMLファイルからのDMA情報リスト, 構造指定検索用データの生成
47 ///////////////////////////////////////////////////////////////////
48     bRc = xmlTrans.GetDmaInfoList(pXmlFile,
49                                   DBR_XMLPARSE_NO_EXTERNAL_ENTITIES,
50                                   pMapDefId, DBR_INDEXTYPE_STRUCTURED,
51                                   "file:///tmp/NorParamFile.txt",
52                                   "file:///tmp/index.txt",
53                                   &pDmaInfoList, &pMessage);
54     if (pMessage != NULL) {
55         cout << "message=" << pMessage << endl;
56     }
57     dbrDelete(pMessage);
58     checkAndThrow(bRc, xmlTrans);
59
60 ///////////////////////////////////////////////////////////////////
61 // 3. 文書オブジェクトの新規作成と構造指定検索インデクスの登録
62 ///////////////////////////////////////////////////////////////////
63 // DMA情報リストを使用して文書オブジェクトを新規作成する
64     bRc = verDoc.CreateObject(&session,
65                               pDmaInfoList->lCount,
66                               pDmaInfoList->pItem,
67                               pXmlFile, "text:.xml", &pOIID);
68     dbrDelete(pDmaInfoList);
69     checkAndThrow(bRc, verDoc);
70
71 // 構造指定検索インデクスの登録
72     bRc = verDoc.CreateIndex("file:///tmp/index.txt");

```

```

73     checkAndThrow(bRc, verDoc);
74
75     //////////////////////////////////////
76     // 4. 終了処理
77     //////////////////////////////////////
78     // トランザクションのコミット
79     bRc = session.Commit();
80     checkAndThrow(bRc, session);
81
82     // セッションの切断
83     bRc = session.Disconnect();
84     checkAndThrow(bRc, session);
85 }

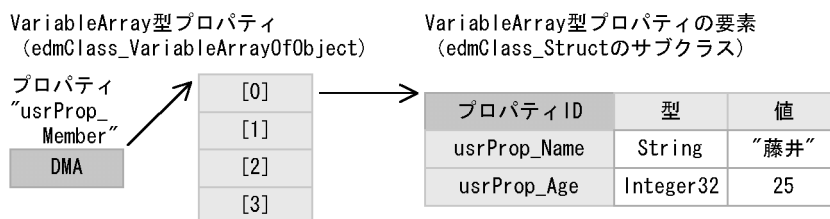
```

(5) 可変長配列の操作

ここでは、CdbrVariableArray クラスと CdbrCompound クラスを使用した VariableArray 型プロパティの操作について、コーディング例を示します。

このコーディング例では、次の図に示す、プロパティ「Name」とプロパティ「Age」を定義した edmClass_Struct クラスのオブジェクトを要素に持つ VariableArray 型のプロパティについて説明します。

図 A-11 サンプルコーディングで使用する VariableArray 型プロパティ



プロパティ「Member」は、基本単位が VariableArray 型で、四つの要素を持つ。プロパティの各要素は「Name」、「Age」という二つのプロパティを持つ。

(a) VariableArray 型プロパティの取得

可変長配列を取得するためには、CdbrDMA::GetPropertyValues メソッドなどの、プロパティを取得するメソッドを使用します。

取得したプロパティ構造体 (SDBR_PROP) が、データ型が Object 型 (IType = DMA_DATATYPE_OBJECT) であり、かつ基本単位が VariableArray (ICardinality = EDM_DMA_CARDINALITY_VARRAY) の場合、メンバ uniValue.ppObject[0] に可変長配列クラス CdbrVariableArray クラスのインスタンスへのポインタが格納されます。

コーディング例を示します。

コーディング例

```

1 // VariableArray型プロパティを取得するサンプルコード
2 SDBR_PROPDEF propdef;
3 SDBR_PROPLIST* pPropList;
4 SDBR_PROP* pProp;
5 DmaInteger32 n;
6 pDmaString_T pName;
7 CdbrVariableArray* pVArray;
8 CdbrCompound CompoValue;
9 propdef.PropId = usrProp_Member;
10 dma.GetPropertyValues(1, &propdef, &pPropList);
11 pProp = &pPropList->pItem[0];

```

```

12 if (pProp->lType == DMA_DATATYPE_OBJECT
13     && pProp->lCardinality == EDM_DMA_CARDINALITY_VARRAY)
14 {
15     pVArray = (CdbVariableArray*) pProp->uniValue.ppObject[0];
16 }

```

(b) 可変長配列の値の取得

可変長配列の要素は、CdbVariableArray::GetValue メソッドにインデクスを指定して取得します。

VariableArray 型プロパティから取得した可変長配列オブジェクトの要素のデータ型は、複合データ型 (DBR_DATATYPE_COMPOUND) です。したがって、CdbCompound クラスの変数のアドレスを指定します。

複合データ型からの値は、CdbCompound::GetValue メソッドにプロパティ ID を指定して、取得します。

コーディング例を次に示します。なお、ここでは、(a) で可変長配列を取得した続きから記述します。

コーディング例 ((a) のコーディング例の続き)

```

17 if (pVArray->GetDataType() == DBR_DATATYPE_COMPOUND) {
18     // 要素の数だけ繰り返す
19     for (int n = 0; n < pVArray->GetCount(); n++) {
20         // 要素 (複合データ型) の取得
21         pVArray->GetValue(n, &CompoValue);
22         // "Name" プロパティの取得と表示
23         CompoValue.GetValue(&usrProp_Name, &pName);
24         printf("Name=%s ", pName);
25         // "Age" プロパティの取得と表示
26         CompoValue.GetValue(&usrProp_Age, &lAge);
27         printf("Age=%d¥n", lAge);
28         dbrDelete(pName);
29     }
30 }

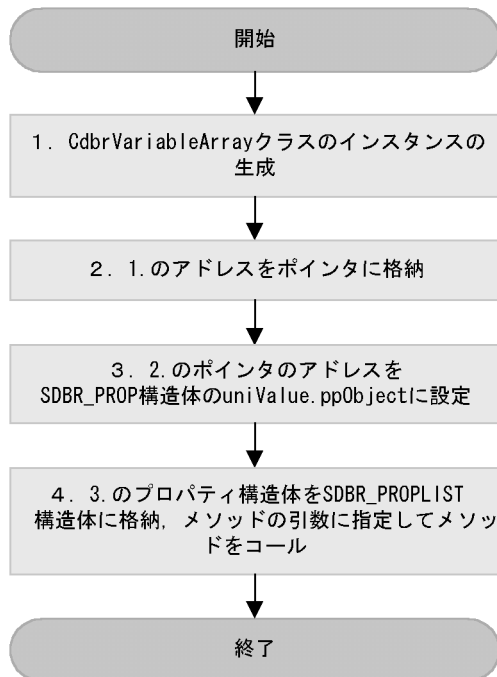
```

(c) VariableArray 型プロパティの設定

VariableArray 型プロパティを設定する場合は、CdbDMA::PutPropertyValues メソッドなどの、プロパティを設定するメソッドを使用します。

次の図に可変長配列を設定するシーケンスを示します。

図 A-12 可変長配列を設定するシーケンス



コーディング例

```

1 // 変数dmaはCdbrDMAクラスのインスタンス
2 SDBR_PROP Prop;
3 SDBR_PROPLIST PropList;
4 Dmapv pObj;
5 CdbCompound CompoValue;
6 CdbVariableArray VArray(DBR_DATATYPE_COMPOUND);
7
8 // 要素となる複合データを作成する
9 CompoValue.SetValue( &usrProp_NAME, "中井");
10 CompoValue.SetValue( &usrProp_AGE, 25);
11 // 可変長配列の要素数を1にして、複合データを可変長配列に格納する
12 VArray.Resize(1);
13 VArray.SetValue(0, CompoValue);
14 // 可変長配列をプロパティ構造体に設定する
15 Prop.PropId = usrProp_Author;
16 Prop.lType = DMA_DATATYPE_OBJECT;
17 Prop.lCardinality = EDM_DMA_CARDINALITY_VARRAY;
18 Prop.lCount = 1;
19 pObj = &VArray;
20 Prop.uniValue.ppObject = &pObj;
21 // プロパティ構造体の値 (=可変長配列) をDMAオブジェクトに設定する
22 PropList.pItem = &Prop;
23 PropList.lCount = 1;
24 dma.PutPropertyValues(&PropList);
  
```

(d) 可変長配列の要素数の追加と削除

可変長配列の要素は、CdbVariableArray::Add メソッドによって追加できます。追加した要素は、可変長配列の末尾に追加されます。

また、可変長配列の要素の削除は、CdbVariableArray::Delete メソッドで、引数に削除する要素のインデックスを指定して実行します。削除された要素以降の要素は、前に詰まります。変更した内容は、PutPropertyValues メソッドなどによってプロパティを更新した時にオブジェクトに反映されます。

コーディング例を次に示します。なお、整数リテラルを指定する場合は、型指定をするか、サフィックスを指定する必要があります。

コーディング例

```

1      CdbrVariableArray  VArray(DBR_DATATYPE_COMPOUND);
2  //要素を作成する
3      CdbrCompound  CompoValue1,CompoValue2;
4      CompoValue1.SetValue( &usrProp_NAME, "島田");
5      CompoValue1.SetValue( &usrProp_AGE, (DmaInteger32)30);
6      CompoValue2.SetValue( &usrProp_NAME, "中井");
7      CompoValue2.SetValue( &usrProp_AGE, 25L);
8  //要素を二つ追加する
9      VArray.Add(CompoValue1);
10     VArray.Add(CompoValue2);
11 //先頭の要素を削除する
12     VArray.Delete(0);

```

(e) すべての要素の削除

可変長配列のすべての要素を一括して削除するには、CdbrVariableArray::Resize メソッドによって要素数に 0 を指定します。要素数 0 の可変長配列を VariableArray 型プロパティに設定すると、VariableArray 型プロパティのすべての要素が削除されます。

次に、コーディング例を示します。なお、ここでは、(a) で VariableArray 型プロパティを取得した続きを記述します。pVArray は、可変長配列のポインタです。

コーディング例 ((a) のサンプルコーディングの続き)

```

17 SDBR_PROP      Prop;
18 SDBR_PROPLIST  PropList;
19 Dmapv          pObj;
20
21 // 要素の数を 0 にする
22 pVArray->Resize(0)
23 // 変更内容をオブジェクトに反映する
24 Prop.PropId = usrProp_Author;
25 Prop.lType = DMA_DATATYPE_OBJECT;
26 Prop.lCardinality = EDM_DMA_CARDINALITY_VARRAY;
27 Prop.lCount = 1;
28 pObj = pVArray;
29 Prop.uniValue.ppObject = &pObj;
30 // プロパティ構造体の値 (= 可変長配列) をDMAオブジェクトに設定する
31 // VariableArray型プロパティの要素はすべて削除される
32 PropList.pItem = &Prop;
33 PropList.lCount = 1;
34 dma.PutPropertyValues(&PropList);

```

(6) エラー処理

ここでは、エラー処理をするプログラムのコーディング例を示します。

(a) データベースエラーが発生した場合のエラー処理

ここでは、データベースエラーが発生した場合のエラー処理をするコーディング例を示します。

コーディング例

```

1  int  MainFunc( ... )
2  {
3      int  retry_count = 3;
4      ...
5      while (retry_count > 0) {
6          rc = UserFunc1(...);

```

```

7         If (rc == RC_RETRY) {
8             retry_count--;
9         }
10        else {
11            retry_count = 0;
12        }
13    }
14    ...
15 }
16
17 Int  UserFunc1( ... )
18 {
19     CdbrReferentialContainer FolderObj;
20     DmaInteger32  lmajor;
21     DmaInteger32  lminor;
22     DmaInteger32  lSQLCode;
23     pDmaString_T  pSQLState;
24
25     pSession->Begin();
26
27     brc = FolderObj.SetOIID( pSession, pOIID );
28     if (brc == DMA_FALSE) goto ERROR_EXIT;
29
30     brc = FolderObj.PutPropertyValues( 引数省略 );
31
32     if (brc == DMA_FALSE) {
33         lmajor = FolderObj.GetLastError(&lminor, &pErrorStrings);
34         if (lmajor == ERR_DB && (lminor == ERR_DB_LOCKED ||
35             lminor == ERR_DB_DEADLOCK_OCCURRED) ) {
36             pSession->Rollback(); // ロールバック
37             return RC_RETRY;
38         }
39         else if (lmajor == ERR_DB && lminor == ERR_DB_FAILED) {
40             pSession->Rollback(); // ロールバック
41             FolderObj.GetDBError(&lSQLCode, &pSQLState);
42             cout << "DB ERROR! SQLCode:" << lSQLCode << endl;
43                 //SQLCodeを出力する
44             cout << "DB ERROR! SQLState:" << pSQLState << endl;
45                 //SQLSTATEを出力する
46             return RC_FAILED;
47         }
48         else {
49             ...
50         }
51     }
52
53     pSession->Commit(); // コミット
54     return RC_OK;
55
56 ERROR_EXIT:
57     ... // 適切なエラー処理
58     return RC_FAILED;
59 }
60

```

(b) それ以外のエラーが発生した場合のエラー処理

ここでは、データベースエラー以外が発生した場合の、単純なエラー処理をするコーディング例を示します。

コーディング例

```

1 int  MainFunc( ... )
2 {
3     rc = UserFunc2( ... );
4     If (rc == RC_FAILED) {

```

```

5         return FALSE;
6     }
7 }
8
9 Int  UserFunc2( ... )
10 {
11     CdbrReferentialContainer FolderObj;
12     DmaInteger32  lmajor;
13     DmaInteger32  lminor;
14
15     pSession->Begin();
16
17     brc = FolderObj.SetOIID( pSession, pOIID );
18     if (brc == DMA_FALSE) goto ERROR_EXIT;
19
20     brc = FolderObj.PutPropertyValues( 引数省略 );
21     if (brc == DMA_FALSE) goto ERROR_EXIT;
22
23     pSession->Commit();           //コミット
24     return RC_OK;
25
26 ERROR_EXIT:
27     pSession->Rollback();        //ロールバック
28     return RC_FAILED;
29 }

```

(c) 詳細メッセージを取得する場合のエラー処理

ここでは、エラー発生時に詳細メッセージを取得するコーディング例を示します。

コーディング例

```

1  int MainFunc(...)
2  {
3      /*セッションオブジェクトのポインタは*/
4      /*pSessionに格納されているものとする*/
5
6      rc = UserFunc3(pSession, ...);
7      if(rc == RC_FAILED) {
8          return FALSE;
9      }
10 }
11
12 int UserFunc3(CdbrSession* pSession,...)
13 {
14     CdbrReferentialContainer FolderObj;
15     DmaInteger32 lmajor;
16     DmaInteger32 lminor;
17     DmaInteger32 lSQLCode;
18     pDmaString_T pSQLState;
19     pDmaString_T pDetailErrorMessage = NULL;//詳細メッセージ用
20
21     pSession->Begin();
22
23     brc = FolderObj.SetOIID(pSession,pOIID);
24     if(brc == DMA_FALSE) goto ERROR_EXIT;
25
26     brc = FolderObj.PutPropertyValues( 引数省略 );
27
28     if (brc == DMA_FALSE) {
29         lmajor = FolderObj.GetLastError(&lminor, &pErrorStrings);
30         if (lmajor == ERR_DB) {
31             FolderObj.GetDBError(&lSQLCode,&pSQLState);
32             cout << "DB ERROR! SQLCode:"<< lSQLCode << endl;
33             //SQLCodeを出力する
34             cout << "DB ERROR! SQLState:"<< pSQLState << endl;

```



```
35                                     //SQLStateを出力する
36
37     }
38
39     //
40     // 詳細メッセージを取得する
41     //
42     pSession->GetLastDetailError(&pDetailErrorMessage);
43     if(pDetailErrorMessage == NULL)
44     {
45         //詳細メッセージは存在しない
46     }
47     else
48     {
49         //詳細メッセージを画面に出力する
50         cout << "DETAIL ERROR :" << pDetailErrorMessage << endl;
51         dbrDelete(pDetailErrorMessage);
52     }
53
54     pSession->Rollback(); //ロールバック
55     return RC_FAILED;
56 }
57
58 pSession->Commit(); //コミット
59 return RC_OK;
60
61 ERROR_EXIT:
62     ... // 適切なエラー処理を行う
63         return RC_FAILED;
64     }
```

付録 B DMA オブジェクトのプロパティ

ここでは、クラスライブラリのオブジェクトで参照、設定できるプロパティの定義元になる DMA のクラスのプロパティについて説明します。

付録 B.1 DMA のクラスのプロパティ一覧

ここでは、クラスライブラリのオブジェクトで参照、設定できるプロパティの定義元になる、主な DMA のクラスのプロパティ一覧を示します。

(1) 説明する項目

各クラスの詳細は、次に示す項目に分けて説明しています。なお、DMA のクラス名について説明中では「dmaClass_DMA クラス」のように表記しています。「クラス」を除いた「dmaClass_DMA」の部分がクラス識別子です。

対応するクラスライブラリのクラス

説明する DMA のクラスと対応するクラスライブラリのクラスについて説明しています。

クラスの説明

クラスの特徴と機能、注意事項などについて説明しています。

スーパークラス

クラスの直上のスーパークラスを記述しています。

プロパティ一覧

クラスが保持するプロパティの特性を次のような一覧表で示しています。

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_X		-	Object 型	Scalar 型	dmaClass_X

プロパティの一覧表の各項目について説明します。

名前

プロパティのシステム ID を記述しています。表記の都合上、複数行にわたっている場合がありますが、実際は 1 行で記述します。

システム生成

DocumentBroker が値を設定するプロパティかどうかを記述しています。

： DocumentBroker が値を設定するプロパティであることを示します。

-：クライアントが値を設定するプロパティであることを示します。

読み取り専用

読み取り専用のプロパティかまたは編集できるプロパティかを記述しています。

：読み取り専用のプロパティであることを示します。

-：編集できるプロパティであることを示します。

データ型

プロパティ値のデータ型について記述しています。

基本単位

プロパティの基本単位 (Scalar 型, List 型または Enumeration 型) について記述しています。

要求クラス

プロパティ値のデータ型が Object 型の場合に、オブジェクトの基になっているクラスまたはサブクラスを記述しています。表記の都合上、複数行にわたっている場合がありますが、実際は 1 行で記述します。なお、要求クラスが複数ある場合は、ベースとなる共通のクラスを記述しています。

(2) dmaClass_ConfigurationHistory クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrConfiguratedReferentialContainer クラス
- CdbrVersionableDocument クラス

クラスの説明

dmaClass_ConfigurationHistory クラスは、バージョン管理する文書のバージョン構成を保持するためのクラスです。文書をバージョン管理する時の最上位に位置します。

スーパークラス

dmaClass_Containable クラス

dmaClass_ConfigurationHistory クラスのプロパティ一覧を、次の表に示します。

表 B-1 dmaClass_ConfigurationHistory クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_Class Description			Object 型	Scalar 型	dmaClass_Class Description
dmaProp_This			Object 型	Scalar 型	dmaClass_Configuration History
dmaProp_Parent			Object 型	Scalar 型	dmaClass_Direct Containment Relationship
dmaProp_Parent Container			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_Referential Containment Relationship
dmaProp_Versioned ObjectClass	-	-	Object 型	Scalar 型	dmaClass_Class Description
dmaProp_Primary VersionSeries			Object 型	Scalar 型	dmaClass_Version Series

(3) dmaClass_Container クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrReferentialContainer クラス

クラスの説明

dmaClass_Container クラスは、ほかのオブジェクトを DirectContainmentRelationship オブジェクトまたは ReferentialContainmentRelationship オブジェクトを介して包含するためのクラスです。Container オブジェクトは直接コンテンツを持ちません。

スーパークラス

dmaClass_Containable クラス

dmaClass_Container クラスのプロパティ一覧を、次の表に示します。

表 B-2 dmaClass_Container クラスのプロパティ一覧

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_Container
dmaProp_Parent			Object 型	Scalar 型	dmaClass_DirectContainmentRelationship
dmaProp_ParentContainer			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
dmaProp_Children			Object 型	Enumeration 型	dmaClass_DirectContainmentRelationship
dmaProp_Containees			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship

(4) dmaClass_DirectContainmentRelationship クラスのプロパティ

対応するクラスライブラリのクラスなし。

クラスの説明

dmaClass_DirectContainmentRelationship クラスは、一つの文書空間内でのオブジェクト間の直接型のコンテインメントを規定するクラスで、dmaClass_ContainmentRelationship クラスのサブクラスです。

直接型のコンテインメントでは、Parent 対 Child は 1:n の関係にあるため、包含される Child に相当するオブジェクトは、dmaClass_DirectContainmentRelationship クラスを介さずに、dmaProp_ParentContainer プロパティから、自身を包含する Container オブジェクトを探索できます。

DocumentBroker では、このクラスは検索できません。

スーパークラス

dmaClass_ContainmentRelationship クラス

dmaClass_DirectContainmentRelationship クラスのプロパティ一覧を、次の表に示します。

表 B-3 dmaClass_DirectContainmentRelationship クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_Class Description			Object 型	Scalar 型	dmaClass_Class Description
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Head	-	-	Object 型	Scalar 型	dmaClass_Containable
dmaProp_Tail	-	-	Object 型	Scalar 型	dmaClass_Container

(5) dmaClass_DocVersion クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrDocument クラス

クラスの説明

dmaClass_DocVersion クラスは、文書を表現するオブジェクトのルートとなるクラスです。

dmaClass_Versionable クラスおよび dmaClass_Containable クラスのプロパティを継承します。したがって、文書空間では、dmaClass_DocVersion クラスまたはそのサブクラスのインスタンスをバージョン管理したり、ほかの文書と関連づけて管理したりできます。

スーパークラス

dmaClass_Versionable クラス

dmaClass_DocVersion クラスのプロパティ一覧を、次の表に示します。

表 B-4 dmaClass_DocVersion クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_Class Description			Object 型	Scalar 型	dmaClass_Class Description
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Parent			Object 型	Scalar 型	dmaClass_Direct Containment Relationship
dmaProp_Parent Container			Object 型	Scalar 型	dmaClass_Container

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
dmaProp_VersionDescriptions			Object 型	Enumeration 型	dmaClass_VersionDescription
dmaProp_CurrentOfSeriesCount			Integer32 型	Scalar 型	-
dmaProp_RenditionsPresent			String 型	List 型	-
dmaProp_Renditions	-	-	Object 型	List 型	dmaClass_Rendition
edmProp_RenditionsCount			Integer32 型	Scalar 型	-

(6) dmaClass_ReferentialContainmentRelationship クラスのプロパティ

対応するクラスライブラリのクラスなし。

クラスの説明

dmaClass_ReferentialContainmentRelationship クラスは、文書空間内でのオブジェクト間の参照型のコンテインメントを規定するクラスで、dmaClass_ContainmentRelationship クラスのサブクラスです。

DocumentBroker では、このクラスは検索できません。

スーパークラス

dmaClass_ContainmentRelationship クラス

dmaClass_ReferentialContainmentRelationship クラスのプロパティ一覧を、次の表に示します。

表 B-5 dmaClass_ReferentialContainmentRelationship クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Head	-	-	Object 型	Scalar 型	dmaClass_Containable
dmaProp_Tail	-	-	Object 型	Scalar 型	dmaClass_Container

(7) dmaClass_Rendition クラスのプロパティ

対応するクラスライブラリのクラス
なし。

クラスの説明

dmaClass_Rendition クラスは、特定の形式で DocVersion オブジェクトのコンテンツを表現するためのクラスです。特定の表現形式のコンテンツ (ContentTransfer オブジェクトまたは ContentTransfers オブジェクト) へのリファレンスを保持します。

Rendition オブジェクトは、DocVersion オブジェクトに従属するオブジェクトです。DocumentBroker では、Rendition オブジェクトが属している DocVersion オブジェクトの dmaProp_RenditionsPresent プロパティに、Rendition オブジェクトが保持する表現形式を示す文字列 (dmaProp_RenditionType プロパティの値) を設定します。

スーパークラス

dmaClass_DMA クラス

dmaClass_Rendition クラスのプロパティ一覧を、次の表に示します。

表 B-6 dmaClass_Rendition クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_Class Description			Object 型	Scalar 型	dmaClass_Class Description
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Rendition Type	-	-	String 型	Scalar 型	-
dmaProp_ContentElementsPresent			String 型	List 型	-
dmaProp_ContentElements	-	-	Object 型	List 型	dmaClass_ContentElement

(8) edmClass_ComponentDocVersion クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrDocument クラス

クラスの説明

edmClass_ComponentDocVersion クラスは、dmaClass_DocVersion クラスのサブクラスです。

スーパークラス

dmaClass_DocVersion クラス

edmClass_ComponentDocVersion クラスのプロパティ一覧を、次の表に示します。

表 B-7 edmClass_ComponentDocVersion クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Parent			Object 型	Scalar 型	dmaClass_DirectContainmentRelationship
dmaProp_ParentContainer			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
dmaProp_VersionDescriptions			Object 型	Enumeration 型	dmaClass_VersionDescription
dmaProp_CurrentOfSeriesCount			Integer32 型	Scalar 型	-
dmaProp_RenditionsPresent			String 型	List 型	-
dmaProp_Renditions	-	-	Object 型	List 型	dmaClass_Rendition
edmProp_CDContainers			Object 型	Enumeration 型	-
edmProp_RenditionsCount			Integer32 型	Scalar 型	-

(9) edmClass_ContainerVersion クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrVersionTraceableContainer クラス

クラスの説明

edmClass_ContainerVersion クラスは、構成管理コンテナの機能を持ち、格納しているオブジェクトの集合全体としてのバージョンを管理するためのクラスです。また、edmProp_VTContainers プロパティを保持し、ほかの構成管理コンテナ機能を持つオブジェクトの構成要素にもなれます。

スーパークラス

dmaClass_Versionable クラス

edmClass_ContainerVersion クラスのプロパティ一覧を、次の表に示します。

表 B-8 edmClass_ContainerVersion クラス

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	edmClass_ContainerVersion
dmaProp_Parent			Object 型	Scalar 型	dmaClass_DirectContainmentRelationship
dmaProp_ParentContainer			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
edmProp_VTContainers			Object 型	Enumeration 型	edmClass_VersionTraceableContainmentRelationship
dmaProp_Children			Object 型	Enumeration 型	dmaClass_DirectContainmentRelationship
dmaProp_Containees			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
edmProp_VTContainees			Object 型	Enumeration 型	edmClass_VersionTraceableContainmentRelationship
dmaProp_VersionDescriptions			Object 型	Enumeration 型	dmaClass_VersionDescription
dmaProp_CurrentOfSeriesCount			Integer32 型	Scalar 型	-

(10) edmClass_ContentSearch クラスのプロパティ

対応するクラスライブラリのクラス
なし。

クラスの説明

edmClass_ContentSearch クラスは、dmaClass_DocVersion クラスのサブクラスに仮想的に継承させて全文検索の対象にする文書を登録するためのクラスです。edmClass_ContentSearch クラスのプロパティを仮想的に継承した dmaClass_DocVersion クラスのサブクラスでは、edmClass_ContentSearch

クラスの選択可能なプロパティを検索に使用するプロパティとして指定できます。また、検索可能なプロパティである 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 プロパティを指定して全文検索ができます。

スーパークラス

dmaClass_DMA クラス

edmClass_ContentSearch クラスのプロパティ一覧を、次の表に示します。

表 B-9 edmClass_ContentSearch クラスのプロパティの一覧

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	edmClass_IndependentPersistence
edmProp_TextIndex			String 型	Scalar 型	-
edmProp_StIndex			String 型	Scalar 型	-
edmProp_ConceptTextIndex			String 型	Scalar 型	-
edmProp_ConceptStIndex			String 型	Scalar 型	-
edmProp_Content			String 型	Scalar 型	-
edmProp_DocVersionOIID			Id 型	Scalar 型	-
edmProp_Score			Integer32 型	Scalar 型	-
edmProp_RawScore			Integer32 型	Scalar 型	-
edmProp_DocLength			Integer32 型	Scalar 型	-
edmProp_ContentIndexStatus			Integer32 型	Scalar 型	-

(11) edmClass_IndependentPersistence クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrIndependentPersistence クラス

クラスの説明

edmClass_IndependentPersistence クラスは、ほかのオブジェクトに従属しないで、独立して存在す

るオブジェクトを新たに作成するためのクラスです。DocumentBroker では、このクラスを継承しているサブクラスを基にして作成したオブジェクトを DMA のオブジェクトとして扱います。

DocumentBroker で、このクラスを基にサブクラスを定義する場合は、次の規則に従ってください。

クラス、プロパティの定義に関する規則

クラスまたはプロパティの GUID は、DocumentBroker が予約している値以外の値で定義する複数のクラスに同じ GUID を定義しない。

複数のプロパティに同じ GUID を定義する場合は、プロパティ名およびプロパティの型を同じにする。

edmClass_IndependentPersistence クラスで定義しているプロパティは、すべて継承し、削除できない。

プロパティのデータ型に関する規則

定義できるプロパティのデータ型は次のとおりです。

Boolean 型

Integer32 型

String 型

Object 型 (ただし VariableArray 型だけ)

プロパティ値の設定および取得に関する規則

DocumentBroker またはクライアントアプリケーションが定義する GUID カインデックスを使用して、プロパティ値の設定または取得ができます。

スーパークラス

dmaClass_DMA クラス

edmClass_IndependentPersistence クラスのプロパティ一覧を、次の表に示します。

表 B-10 edmClass_IndependentPersistence クラスのプロパティの一覧

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_Class Description			Object 型	Scalar 型	dmaClass_Class Description
dmaProp_This			Object 型	Scalar 型	edmClass_Independent Persistence

(12) edmClass_PublicACL クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrPublicACL クラス

クラスの説明

edmClass_PublicACL クラスは、パブリック ACL を表すオブジェクト、CdbrPublicACL オブジェクトの構成要素になるクラスです。

スーパークラス

dmaClass_DMA クラス

edmClass_PublicACL クラスのプロパティ一覧を、次の表に示します。

表 B-11 edmClass_PublicACL クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_This			Object 型	Scalar 型	edmClass_PublicACL
edmProp_OwnerId			String 型	Scalar 型	-

(13) edmClass_Relationship クラスのプロパティ

対応するクラスライブラリのクラス
なし。

クラスの説明

edmClass_Relationship クラスは、リレーションによってオブジェクト間の関連づけを設定するためのクラスで、dmaClass_Relationship クラスのサブクラスです。

スーパークラス

dmaClass_Relationship クラス

edmClass_Relationship クラスのプロパティ一覧を、次の表に示します。

表 B-12 edmClass_Relationship クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Head	-	-	Object 型	Scalar 型	dmaClass_DMA
dmaProp_Tail	-	-	Object 型	Scalar 型	dmaClass_DMA
edmProp_RelationType	-	-	Integer32 型	Scalar 型	-

(14) edmClass_VersionTraceableContainmentRelationship クラス

対応するクラスライブラリのクラス
なし。

クラスの説明

edmClass_VersionTraceableContainmentRelationship クラスは、文書空間内でのオブジェクト間の構成管理型のコンテインメントを規定するクラスで、dmaClass_ContainmentRelationship クラスのサブクラスです。コンテナの構成要素の最新バージョンを追跡してコンテインメントを規定したり、任意のバージョンを固定してコンテインメントを規定できるプロパティを保持します。

DocumentBroker では、このクラスは検索できません。

スーパークラス

dmaClass_ContainmentRelationship クラス

edmClass_VersionTraceableContainmentRelationship クラスのプロパティ一覧を、次の表に示します。

表 B-13 edmClass_VersionTraceableContainmentRelationship クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Head	-	-	Object 型	Scalar 型	dmaClass_Containable
dmaProp_Tail	-	-	Object 型	Scalar 型	dmaClass_Container
edmProp_VTMode			Integer32 型	Scalar 型	-
edmProp_VTVersionSeries			Object 型	Scalar 型	dmaClass_VersionSeries
edmProp_VTConfigurationHistory			Object 型	Scalar 型	dmaClass_ConfigurationHistory

(15)edmClass_VersionTracedComponentDocVersion クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrDocument クラス

クラスの説明

edmClass_VersionTracedComponentDocVersion クラスは、構成管理型で包含される文書を表現するためのクラスで、dmaClass_DocVersion クラスのサブクラスです。構成管理型で包含されるための edmProp_VTContainers プロパティを保持します。

スーパークラス

dmaClass_DocVersion クラス

edmClass_VersionTracedComponentDocVersion クラスのプロパティ一覧を、次の表に示します。

表 B-14 edmClass_VersionTracedComponentDocVersion クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_Parent			Object 型	Scalar 型	dmaClass_Direct Containment Relationship
dmaProp_Parent Container			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_Referential Containment Relationship
dmaProp_Version Descriptions			Object 型	Enumeration 型	dmaClass_Version Description
dmaProp_CurrentOf SeriesCount			Integer32 型	Scalar 型	-
dmaProp_Renditions Present			String 型	List 型	-
dmaProp_Renditions	-	-	Object 型	List 型	dmaClass_Rendition
edmProp_CDContainers			Object 型	Enumeration 型	-
edmProp_Renditions Count			Integer32 型	Scalar 型	-
edmProp_VTContainers			Object 型	Enumeration 型	edmClass_VersionTraceableContainment Relationship

(16) edmClass_VersionTracedDocVersion クラスのプロパティ

対応するクラスライブラリのクラス

- CdbrDocument クラス

クラスの説明

edmClass_VersionTracedDocVersion クラスは、構成管理型で包含される文書を表現するためのクラスで、dmaClass_DocVersion クラスのサブクラスです。構成管理型で包含されるための edmProp_VTContainers プロパティを保持します。

スーパークラス

dmaClass_DocVersion クラス

edmClass_VersionTracedDocVersion クラスのプロパティ一覧を、次の表に示します。

表 B-15 edmClass_VersionTracedDocVersion クラスのプロパティ

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_OIID			String 型	Scalar 型	-

名前	システム生成	読み取り専用	データ型	基本単位	要求クラス
dmaProp_ClassDescription			Object 型	Scalar 型	dmaClass_ClassDescription
dmaProp_This			Object 型	Scalar 型	dmaClass_DocVersion
dmaProp_Parent			Object 型	Scalar 型	dmaClass_DirectContainmentRelationship
dmaProp_ParentContainer			Object 型	Scalar 型	dmaClass_Container
dmaProp_Containers			Object 型	Enumeration 型	dmaClass_ReferentialContainmentRelationship
dmaProp_VersionDescriptions			Object 型	Enumeration 型	dmaClass_VersionDescription
dmaProp_CurrentOfSeriesCount			Integer32 型	Scalar 型	-
dmaProp_RenditionsPresent			String 型	List 型	-
dmaProp_Renditions	-	-	Object 型	List 型	dmaClass_Rendition
edmProp_RenditionsCount			Integer32 型	Scalar 型	-
edmProp_VTContainers			Object 型	Enumeration 型	edmClass_VersionTraceableContainmentRelationship

付録 B.2 プロパティの説明

ここでは、それぞれのプロパティについて説明します。

(1) dmaProp_Children プロパティ

Container オブジェクトまたは ContainerVersion オブジェクトが直接含む Containable オブジェクトとの包含関係を表す DirectContainmentRelationship オブジェクトのリストです。

DirectContainmentRelationship オブジェクトは、Container オブジェクトとその Child に相当するオブジェクトとの関連を表します。DirectContainmentRelationship オブジェクトの dmaProp_Tail プロパティの Reflective プロパティです。

(2) dmaProp_ClassDescription プロパティ

オブジェクトの基になっているクラスについて定義する ClassDescription オブジェクトです。

オブジェクトに対して、オブジェクトの基になるクラスについての定義情報へのアクセスを提供します。

(3) dmaProp_Containees プロパティ

Container オブジェクトまたは ContainerVersion オブジェクトが参照的に含む Containable オブジェクトとの包含関係を表す ReferentialContainmentRelationship オブジェクトのリストです。
ReferentialContainmentRelationship オブジェクトは、Container オブジェクトと Containee に相当するオブジェクトとの関連を表します。

(4) dmaProp_Containers プロパティ

指定したオブジェクトを参照型で包含している Container オブジェクトを識別する ReferentialContainmentRelationship オブジェクトのリストです。

(5) dmaProp_ContentElements プロパティ

Rendition オブジェクトに含まれる ContentTransfer オブジェクト、ContentTransfers オブジェクトまたは ContentReference オブジェクトのリストです。

一つの Rendition オブジェクトに対してコンテンツは、一つです。複数のコンテンツをリストの要素としようとする、インデクスが 0 だけがコンテンツになります。

(6) dmaProp_ContentElementsPresent プロパティ

コンテンツが属している DocVersion オブジェクトを最後に永続化した時点で ContentTransfer オブジェクト、ContentTransfers オブジェクトまたは ContentReference オブジェクトの dmaProp_ComponentType プロパティが保持していた文字列のリストです。

DocumentBroker が値を設定します。クライアントがこのプロパティの値を操作すると、実際の永続オブジェクトの状態と不一致となるので、操作しないでください。

dmaProp_ContentElementsPresent プロパティは、dmaProp_ContentElements プロパティのリストの状態を動的に追跡しません。したがって、dmaProp_ContentElementsPresent プロパティと dmaProp_ContentElements プロパティの値は等しくない場合もあります。

Rendition オブジェクトが属している DocVersion オブジェクトに再接続した時点または DocVersion オブジェクトの scratchpad オブジェクトを永続オブジェクトの状態に更新した時点で、dmaProp_ContentElementsPresent プロパティと dmaProp_ContentElements プロパティは、同期します。

(7) dmaProp_CurrentOfSeriesCount プロパティ

VersionSeries オブジェクトに最新バージョンとしてチェックインした文書オブジェクト (DocVersion オブジェクト) に対して DocumentBroker が付与する番号です。

クラスライブラリでは、そのオブジェクトのバージョンの状態を明確にするために、次の値を設定します。

表 B-16 クラスライブラリで設定する dmaProp_CurrentOfSeriesCount プロパティの値

DocVersion オブジェクトの状態	dmaProp_CurrentOfSeriesCount プロパティに設定される値
CdbrVersionableDocument オブジェクトのカレントバージョンに対応する DocVersion オブジェクト	1
チェックアウト中の DocVersion オブジェクト	-2

DocVersion オブジェクトの状態		dmaProp_CurrentOfSeriesCount プロパティに設定される値
チェックイン後の DocVersion オブジェクト	チェックインによって追加された DocVersion オブジェクト	1
	チェックインによって旧バージョンになった DocVersion オブジェクト	0
バージョン管理の対象でない CdbDocument オブジェクトに含まれる DocVersion オブジェクト		-1

クライアントは、このプロパティの値の整合性を維持するために、すでに存在する文書オブジェクト (DocVersion オブジェクト) をチェックインする場合、その文書オブジェクトに編集可能なロック (write ロック) を設定して矛盾が起きないように注意してください。

(8) dmaProp_Head プロパティ

関連づけられた二つのオブジェクトのうち、Head (関連の終点) 側のオブジェクトです。例えば、コンテインメントでは、包含される側のオブジェクトです。

このプロパティは、コンテインメントやリレーションのプロパティです。クラスライブラリでは、コンテインメントやリレーションが設定されているオブジェクトの dmaProp_This プロパティとこのプロパティを組み合わせ、検索実行時に検索対象クラスを結合する場合に利用できます。

(9) dmaProp_OIID プロパティ

オブジェクトを識別する識別子です。

(10) dmaProp_Parent プロパティ

このオブジェクトを直接型で包含している Parent に相当するオブジェクトを識別する DirectContainmentRelationship オブジェクトです。

(11) dmaProp_ParentContainer プロパティ

DirectContainmentRelationship オブジェクトを介してこのオブジェクトを包含している Container オブジェクトです。

クライアントは、この dmaProp_ParentContainer プロパティによって、介在する DirectContainmentRelationship オブジェクトを参照することなく、直接 Parent に相当する Container オブジェクトを探索できます。

DocumentBroker では、このプロパティによる探索を許可します。

(12) dmaProp_PrimaryVersionSeries プロパティ

バージョン管理する文書に対する最初の VersionSeries オブジェクトです。

ConfigurationHistory オブジェクトに、最初に VersionSeries オブジェクトがバインドされる時に値が設定されます。

(13) dmaProp_Renditions プロパティ

DocVersion オブジェクトが保持する Rendition オブジェクトのリストです。

(14) dmaProp_RenditionsPresent プロパティ

DocVersion オブジェクトを最後に永続化した時点で DocVersion オブジェクトの dmaProp_RenditionType プロパティが保持していた値のリストです。

DocVersion オブジェクトに接続すると、DocumentBroker が値を設定します。

dmaProp_RenditionsPresent プロパティは、dmaProp_Renditions プロパティのリストの状態を動的に追跡しません。したがって、dmaProp_RenditionsPresent プロパティと dmaProp_RenditionType プロパティの値は等しくない場合があります。

DocVersion オブジェクトに再接続した時点または DocVersion オブジェクトの scratchpad オブジェクトを永続オブジェクトの状態に更新した時点で、dmaProp_RenditionsPresent プロパティと dmaProp_Renditions プロパティの値は同期します。

(15) dmaProp_RenditionType プロパティ

コンテンツの表現形式を表す文字列です。

文字列の形式は次のようになります。

```
<rendition_type_space>::<typename>
```

DocumentBroker では、rendition_type_space に IANA で定義している MIME を使用します。表現形式がプレーンテキストの場合、次のように指定します。

```
MIME::text/plain
```

クラスライブラリでは、次のオブジェクトの dbrProp_RenditionType プロパティとして操作します。

- CdbrDocument クラス
- CdbrVersionableDocument クラス

(16) dmaProp_Tail プロパティ

関連づけられた二つのオブジェクトのうち、Tail (関連の始点) 側のオブジェクトです。例えば、コンテンツメントでは、包含する側のオブジェクトです。

このプロパティは、コンテンツメントやリレーションのプロパティです。クラスライブラリでは、コンテンツメントやリレーションが設定されているオブジェクトの dmaProp_This プロパティとこのプロパティを組み合わせて、検索実行時に検索対象クラスを結合する場合に利用できます。

(17) dmaProp_This プロパティ

オブジェクト自身です。

問い合わせの中で、オブジェクト間の関連を表現して、問い合わせの結果の中から候補のオブジェクトを選択するために使用できます。

クラスライブラリでは、コンテンツメントやリレーションの dmaProp_Head プロパティや dmaProp_Tail プロパティとこのプロパティを組み合わせて、検索実行時に検索対象クラスを結合する場合に利用できません。

(18)dmaProp_VersionDescriptions プロパティ

バージョン管理する文書と特定の VersionSeries オブジェクトとを結び付ける VersionDescription オブジェクトのリストです。

文書をバージョン管理する場合にだけ必要となります。

(19)dmaProp_VersionedObjectClass プロパティ

バージョン管理する文書のベースとなるクラスについて定義する ClassDescription オブジェクトです。

バージョン管理する文書は、すべてこのプロパティの ClassDescription オブジェクトが定義するクラスを基に作成される必要があります。

DocumentBroker では、このプロパティに設定された ClassDescription オブジェクトで、バージョン管理する文書のオブジェクトであるかどうかを確認しません。このプロパティの値は、あらかじめ DocumentBroker サーバのメタ情報ファイル (edms.ini) に設定してください。DocumentBroker サーバのメタ情報ファイルについては、マニュアル「DocumentBroker Version 3 システム導入・運用ガイド」を参照してください。

(20)edmProp_ConceptStIndex プロパティ

全文検索用に登録した文書の全文データです。

このプロパティは、全文検索エンジンに対して、全文検索用の登録情報を参照することを示すためにあります。したがって、SELECT 句に検索結果として取得するプロパティには指定できません。

このプロパティでは、概念検索および構造指定検索を含む全文検索が実行できます。

(21)edmProp_ConceptTextIndex プロパティ

全文検索用に登録した文書の全文データです。

このプロパティは、全文検索エンジンに対して、全文検索用の登録情報を参照することを示すためにあります。したがって、SELECT 句に検索結果として取得するプロパティには指定できません。

このプロパティでは、概念検索を含む全文検索が実行できます。

(22)edmProp_Content プロパティ (Version 1 との互換用)

全文検索用に登録した文書の全文データです。

このプロパティは、全文検索エンジンに対して、全文検索用の登録情報を参照することを示すためにあります。したがって、SELECT 句に検索結果として取得するプロパティには指定できません。

このプロパティでは、構造指定検索を含む全文検索が実行できます。

(23)edmProp_ContentIndexStatus プロパティ

全文検索インデクスの登録状態を示します。設定される値は次のとおりです。

表 B-17 edmProp_ContentIndexStatus プロパティに設定される値

値	意味
0	文書は未登録です。
1	文書は登録されていますが、全文検索インデクスは未登録です。
2	文書および全文検索インデクスが登録されています。

値	意味
3	文書が更新されていますが、全文検索インデックスが更新されていないため、文書と全文検索インデックスの内容が一致しません。
100 以降	DocumentBroker Text Search Index Loader Version 2 が使用する値です。

(24) edmProp_DocLength プロパティ

検索した文書の長さです。

バイト単位で示します。

(25) edmProp_DocVersionOIID プロパティ

全文検索用に登録した DocVersion オブジェクトの OIID です。

(26) edmProp_OwnerId プロパティ

オブジェクトの所有者を表す文字列です。

クラスライブラリでは、dbrProp_OwnerId プロパティとして扱います。ただし、検索条件に指定する場合は、「edmProp_OwnerId」と指定します。

(27) edmProp_RawScore プロパティ

ランキング検索した場合のスコアです。

検索する時に算出されて値が設定されます。ランキング検索時のスコアを取得するための形式的なプロパティです。スコアは、検索結果集合内で正規化されずに返されます。

(28) edmProp_RelationType プロパティ

Relationship オブジェクトを使用した関連の種類を定義します。ユーザプログラムで任意に設定できますが、0 以下の値は設定できません。

(29) edmProp_RenditionsCount プロパティ

一つの文書中に登録されているレンディションの数を表すカウンタです。マスタレンディションだけが登録されている場合、値は常に 1 になります。サブレンディションが登録されると、マスタレンディションと登録されたサブレンディションの数を合わせた値になります。

(30) edmProp_Score プロパティ

ランキング検索した場合のスコアです。

検索する時に算出されて値が設定されます。ランキング検索時のスコアを取得するための形式的なプロパティです。スコアは、検索結果集合内で正規化されて返されます。

(31) edmProp_StIndex プロパティ

全文検索用に登録した文書の全文データです。

このプロパティは、全文検索エンジンに対して、全文検索用の登録情報を参照することを示すためにあります。したがって、SELECT 句に検索結果として取得するプロパティには指定できません。

このプロパティでは、構造指定検索を含む全文検索が実行できます。

(32)edmProp_TextIndex プロパティ

全文検索用に登録した文書の全文データです。

このプロパティは、全文検索エンジンに対して、全文検索用の登録情報を参照することを示すためにあります。したがって、SELECT 句に検索結果として取得するプロパティには指定できません。

このプロパティでは、プレーンテキストに対する全文検索が実行できます。

(33)edmProp_VTContainees プロパティ

ContainerVersion オブジェクトが構成管理の対象として含む Containable オブジェクトとの包含関係を表す VersionTraceableContainmentRelationship オブジェクトのリストです。

(34)edmProp_VTContainers プロパティ

ContainerVersion オブジェクトが構成管理の対象として含まれる上位の Container との包含関係を表す VersionTraceableContainmentRelationship オブジェクトのリストです。

(35)edmProp_VTConfigurationHistory プロパティ

VersionTraceableContainmentRelationship オブジェクトが指している包含される側のオブジェクトのバージョンを統合している ConfigurationHistory オブジェクトです。

(36)edmProp_VTMode プロパティ

包含するオブジェクトの最新バージョンを追跡するか (FLOATING モード), バージョンを固定するか (FIX モード) を設定します。

(37)edmProp_VTVersionSeries プロパティ

VersionTraceableContainmentRelationship オブジェクトが指している包含される側のオブジェクトのバージョンを管理している VersionSeries オブジェクトです。

付録 C クラスライブラリのメソッドで使用する構造体

クラスライブラリのメソッドでは、オブジェクトを作成したりプロパティを設定したりするときに、設定する情報を構造体として指定します。

また、既存のオブジェクトの情報やプロパティの値を取得する場合、情報を構造体として取得します。

ここでは、クラスライブラリのメソッドで指定する構造体について説明します。

付録 C.1 オブジェクトの情報を設定，参照する構造体

ここでは、オブジェクトを作成，参照または更新するときに、オブジェクトの情報およびプロパティの情報を扱う構造体についてアルファベット順で説明します。

(1) SDBR_CONTENTINFO 構造体 (コンテンツ情報構造体)

説明

マルチファイル管理機能を使用する場合に、一つのコンテンツの情報を指定したり、取得したりする場合に使用する構造体です。

形式

```
struct SDBR_CONTENTINFO {
    pDmaString_T    pComponentType; //ファイルを識別する文字列
    pDmaString_T    pRetrievalName; //ファイル名
    DmaUInteger32  lContentSize;   //コンテンツのサイズ
};
```

メンバの説明

出力されるメンバの説明

- pComponentType
格納しているコンテンツの ComponentType が設定されます。
- pRetrievalName
格納しているコンテンツの RetrievalName が設定されます。
- lContentSize
格納しているコンテンツのサイズが設定されます。

入力するメンバの説明

CdbrDocument::GetContentInfo メソッドまたは CdbrVersionableDocument::GetContentInfo メソッドで取得した構造体を変更しないで、入力する構造体としてください。

この構造体を出力または入力するメソッド

この構造体を出力するメソッド

- CdbrDocument::GetContentInfo メソッド
- CdbrVersionableDocument::GetContentInfo メソッド

この構造体を入力するメソッド

- CdbrDocument::GetContentAndLock メソッド
- CdbrVersionableDocument::GetContentAndLock メソッド

(2) SDBR_CONTENTLIST 構造体 (コンテンツ情報リスト構造体)

説明

マルチファイル管理機能を使用する場合に、マルチファイル文書の情報を指定したり、取得したりする場合に使用する構造体です。

形式

```

struct SDBR_CONTENTLIST {
    DmaInteger32      lCount;           //登録コンテンツ数
    SDBR_CONTENTINFO* pItem;           //コンテンツ情報構造体の
                                        //ポインタ
    DmaInteger32      lValidMode;       //有効モード
    DmaBoolean        bIsMultiContents; //マルチファイル識別
};

```

メンバの説明

出力されるメンバの説明

- lCount
登録したコンテンツ数が設定されます。
- pItem
すべてのコンテンツ情報を格納した SDBR_CONTENTINFO 構造体のポインタが設定されま
す。
- lValidMode
DocumentBroker がシステムとして定義する値が設定されます。
- bIsMultiContents
マルチファイル文書の場合、DMA_TRUE が設定されます。マルチファイル文書ではない場
合、DMA_FALSE が設定されます。

入力するメンバの説明

CdbrDocument::GetContentInfo メソッドまたは CdbrVersionableDocument::GetContentInfo
メソッドで取得した構造体を変更しないで、入力する構造体としてください。

この構造体を出力または入力するメソッド

この構造体を出力するメソッド

- CdbrDocument::GetContentInfo メソッド
- CdbrVersionableDocument::GetContentInfo メソッド

この構造体を入力するメソッド

- CdbrDocument::GetContentAndLock メソッド
- CdbrVersionableDocument::GetContentAndLock メソッド

(3) SDBR_DMAININFO 構造体 (DMA オブジェクト生成用の構造体)

説明

クラスライブラリのオブジェクトを作成するときに、構成要素にする DMA オブジェクトの基になる
DMA のクラスの情報を指定します。
なお、それぞれの DMA オブジェクトにプロパティを設定する場合は、プロパティリスト構造体
(SDBR_PROPLIST 構造体) を指定します。
また、マッピング定義情報に従って XML ファイルからプロパティを抽出する場合、DMA オブジェク
ト生成用の情報が設定される構造体です。

形式

```

// DMAオブジェクト生成用の構造体
struct SDBR_DMAININFO {
    DmaId      ClassId; //DMAオブジェクトのクラス識別子
    SDBR_PROPLIST PropList; //プロパティ情報のリスト
};

```

メンバの説明

出力されるメンバの説明

- ClassId
DMA オブジェクトのクラス識別子が設定されます。
- PropList
マッピング定義情報に従って XML ファイルから抽出したプロパティのリストを格納した SDBR_PROPLIST 構造体のアドレスが設定されます。

入力するメンバの説明

- ClassId
新規に作成する DMA オブジェクトのクラス識別子を指定します。
作成する DMA オブジェクトと指定するクラス識別子の関係を次に示します。

表 C-1 DMA オブジェクトを作成する場合に指定するクラス識別子

作成するオブジェクト	指定するクラス識別子
CdbrConfiguratedReferentialContainer オブジェクト	1 番目の要素として dmaClass_ConfigurationHistory クラスまたはそのサブクラスのクラス識別子, 2 番目の要素として edmClass_ContainerVersion クラスまたはそのサブクラスのクラス識別子を指定します。 2 番目の要素を省略した場合 (メソッドの引数 IDMAInfoListCount が 1 の場合) は, edmClass_ContainerVersion クラスを使用して DMA オブジェクトが作成されます。1 番目の要素を省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_ContainerVersion クラスと dmaClass_ConfigurationHistory クラスを使用して DMA オブジェクトが作成されます。
CdbrDocument オブジェクト	dmaClass_DocVersion クラスまたはそのサブクラスのクラス識別子を指定します。 省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_VersionTracedComponentDocVersion クラスを使用して DMA オブジェクトが作成されます。
CdbrIndependentPersistence オブジェクト	edmClass_IndependentPersistence クラスまたはそのサブクラスのクラス識別子を指定します。 省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_IndependentPersistence クラスを使用して DMA オブジェクトが作成されます。
CdbrPublicACL オブジェクト	edmClass_PublicACL クラスのクラス識別子を指定します。 省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_PublicACL クラスを使用して DMA オブジェクトが作成されます。
CdbrReferentialContainer オブジェクト	dmaClass_Container クラスまたはそのサブクラスのクラス識別子を指定します。 省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), dmaClass_Container クラスを使用して DMA オブジェクトが作成されます。

作成するオブジェクト	指定するクラス識別子
CdbrVersionableDocument オブジェクト	1 番目の要素として dmaClass_ConfigurationHistory クラスまたはそのサブクラスのクラス識別子, 2 番目の要素として dmaClass_DocVersion クラスまたはそのサブクラスのクラス識別子を指定します。 2 番目の要素を省略した場合 (メソッドの引数 IDMAInfoListCount が 1 の場合), edmClass_VersionTracedComponentDocVersion クラスを使用して DMA オブジェクトが作成されます。1 番目の要素を省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_VersionTracedComponentDocVersion クラスを使用して DocVersion オブジェクトが作成され, dmaClass_ConfigurationHistory クラスを使用して ConfigurationHistory オブジェクトが作成されます。
CdbrVersionTraceableContainer オブジェクト	edmClass_ContainerVersion クラスまたはそのサブクラスのクラス識別子を指定します。 省略した場合 (メソッドの引数 IDMAInfoListCount が 0 で pDMAInfoList が NULL の場合), edmClass_ContainerVersion クラスを使用して DMA オブジェクトが作成されます。

- PropList

オブジェクトの初期値として設定するプロパティを設定した SDBR_PROPLIST 構造体のアドレスを指定します。

この構造体を出力または入力するメソッド

この構造体を出力するメソッド

- CdbrXmlTranslator::GetDmaInfoList メソッド

この構造体を入力するメソッド

- CdbrConfiguredReferentialContainer::CreateObject メソッド
- CdbrDocument::CreateObject メソッド
- CdbrIndependentPersistence::CreateObject メソッド
- CdbrPublicACL::CreateObject メソッド
- CdbrReferentialContainer::CreateObject メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionTraceableContainer::CreateObject メソッド

(4) SDBR_DMAINFOLIST 構造体 (DMA 情報リスト構造体)

説明

マッピング定義情報に従って XML ファイルからプロパティを抽出して DMA 情報リスト (文書オブジェクトを作成するときのプロパティの初期値のリスト) を作成する場合に, DMA 情報 (SDBR_DMAININFO 構造体) の個数とポインタが設定される構造体です。

形式

```
struct SDBR_DMAINFOLIST {
    DmaInteger32    lCount;    //DMAオブジェクトの情報の個数
    SDBR_DMAININFO* pItem;    //DMAオブジェクトの情報
};
```

メンバの説明

- lCount

DMA オブジェクトの情報の個数が設定されます。

- pItem
DMA オブジェクトの情報が SDBR_DMMAINFO 構造体として設定されます。なお、設定される構造体の配列の要素数は、マッピング定義情報で指定したクラスの数と同じです。

この構造体を出力するメソッド

- CdbrXmlTranslator::GetDmaInfoList メソッド

(5) SDBR_FILELINK_PATHINFO 構造体 (File Link パス情報構造体)

説明

File Link 連携機能を使用する場合に、HiRDB File Link で必要な情報を格納する構造体です。

File Link 文書を登録または更新するときには、登録するファイルまたはディレクトリや登録先など、HiRDB File Link が必要とする情報を指定します。File Link 文書を取得するときには、取得先のクライアントのファイルパスまたはディレクトリパスを指定します。

形式

```
// File Linkパス情報構造体
struct SDBR_FILELINK_PATHINFO {
    pDmaString_T    pFilePath;           // 登録ファイルもしくは
                                           // ディレクトリのURL
                                           // または取得先のファイルもしくは
                                           // ディレクトリのURL

    pDmaString_T    pFamName;           // FAM名
    pDmaString_T    pFamFilePath;       // コンテンツ保管先パス
    DmaInteger32    lContentDeleteMode; // コンテンツ削除モード
    DmaInteger32    lFileNamingMode;    // コンテンツ名自動生成モード
};
```

メンバの説明

なお、文書の取得時には、メンバ pFilePath の指定だけが有効になります。

- pFilePath
文書の登録・更新時に、クライアントにあるファイルを登録または更新する場合は、登録するコンテンツとして、クライアントにあるファイルまたはディレクトリのパス名を、URL 形式で指定します。ファイルパスを指定した場合は、指定したファイルが登録されます。ディレクトリパスを指定した場合は、指定した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が登録されます。
文書の登録・更新時に、すでに HiRDB File Link で管理されているデータをコンテンツとして登録する場合は、NULL を指定します。
文書の取得時には、コンテンツの取得先として、クライアントのファイルパスまたはディレクトリパスを指定します。
- pFamName
HiRDB File Link の flsetup コマンドの -fam オプションで指定した FAM 名を指定します。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。
- pFamFilePath
コンテンツ保管先のパスを指定します。保管先のパスの設定については、マニュアル「HiRDB File Link」を参照してください。
- lContentDeleteMode
文書のコンテンツを RemoveObject メソッドまたは DeleteRendition メソッドによって削除するときに、保管先のファイルの実体を削除することを指定します。必ず、DBR_CONTENTDELETE_SYNC を指定してください。
- lFileNamingMode

文書のコンテンツを登録するときに、保管先でのコンテンツ名を自動生成するかどうかを指定します。

自動生成機能を使用しない場合は DBR_FILENAMING_SPECIFY を指定してください。使用する場合は DBR_FILENAMING_AUTO を指定してください。

この構造体を出力するメソッド

- CdbrDocument::AddRendition メソッド
- CdbrDocument::CreateObject メソッド
- CdbrDocument::GetContentAndLock メソッド
- CdbrDocument::UpdateContent メソッド
- CdbrDocument::UpdateContentAndRenditionType メソッド
- CdbrVersionableDocument::AddRendition メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::GetContentAndLock メソッド
- CdbrVersionableDocument::UpdateContent メソッド
- CdbrVersionableDocument::UpdateContentAndRenditionType メソッド

(6) SDBR_LINKEDOBJINFO 構造体 (リンク付きオブジェクト情報構造体)

説明

コンテナによってオブジェクト間の関連づけを設定している場合に、コンテナが包含しているオブジェクトの情報やリンク識別子などが設定される構造体です。また、包含しているオブジェクトのプロパティやリンクのプロパティ情報を取得する場合、プロパティリスト構造体 (SDBR_PROPLIST 構造体) が設定されます。

形式

```
struct SDBR_LINKEDOBJINFO{
    pDmaString_T    pOIID;           //オブジェクトのOIID
    DmaInteger32    lType;           //オブジェクトの種別
    pDmaString_T    pLinkId;        //リンク識別子
    SDBR_PROPLIST   PropList;       //オブジェクトのプロパティリスト
    SDBR_PROPLIST   LinkPropList;   //リンクのプロパティリスト
};
```

メンバの説明

- pOIID
コンテナが指しているオブジェクトの OIID が設定されます。または、オブジェクトを指しているコンテナの OIID が設定されます。
- lType
オブジェクトの種別を表す値として、対応するクラスライブラリのクラスを示す定数が設定されます。定数とクラスライブラリのクラスの関係は、「付録 C.3 構造体で指定する定数」を参照してください。
- pLinkId
オブジェクトのリンク識別子が設定されます。
- PropList
オブジェクトがメソッドの引数で指定したプロパティを持つ場合、プロパティ情報が SDBR_PROPLIST 構造体として設定されます。
- LinkPropList
オブジェクトのリンクがメソッドの引数で指定したプロパティを持つ場合、プロパティの識別子、型および値の一覧が SDBR_PROPLIST 構造体として設定されます。

この構造体を出力するメソッド

- CdbrConfiguredReferentialContainer::GetContainableList メソッド
- CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbrContainable::GetContainerList メソッド
- CdbrContainable::GetContainerListAndLock メソッド
- CdbrReferentialContainer::GetContainableList メソッド
- CdbrReferentialContainer::GetContainableListAndLock メソッド

(7) SDBR_LINKEDOBJLIST 構造体 (リンク付きオブジェクトリスト構造体)

説明

リンク付きオブジェクト情報構造体 (SDBR_LINKEDOBJINFO 構造体) の個数とポインタが設定される構造体です。

形式

```
struct SDBR_LINKEDOBJLIST{
    DmaInteger32      lCount; //リンク付きオブジェクト情報の個数
    SDBR_LINKEDOBJINFO* pItem; //リンク付きオブジェクト情報
};
```

メンバの説明

- lCount
リンク付きオブジェクト情報を表す SDBR_LINKEDOBJINFO 構造体の個数が設定されます。
- pItem
SDBR_LINKEDOBJINFO 構造体の先頭のアドレスへのポインタが設定されます。このとき、SDBR_LINKEDOBJINFO 構造体の個数分の領域が確保されます。

この構造体を出力するメソッド

- CdbrConfiguredReferentialContainer::GetContainableList メソッド
- CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbrContainable::GetContainerList メソッド
- CdbrContainable::GetContainerListAndLock メソッド
- CdbrReferentialContainer::GetContainableList メソッド
- CdbrReferentialContainer::GetContainableListAndLock メソッド

(8) SDBR_OBJINFO 構造体 (オブジェクト情報構造体)

説明

オブジェクト間の情報が設定される構造体です。例えば、コンテナによってオブジェクト間の関連づけを設定している場合、指定したコンテナが包含しているオブジェクトの情報、または指定したオブジェクトが包含されているコンテナの情報が設定されます。

また、オブジェクトのプロパティ情報を取得する場合には、プロパティリスト構造体 (SDBR_PROPLIST 構造体) が設定されます。

形式

```
struct SDBR_OBJINFO {
    pDmaString_T      pOIID; // オブジェクトのOIID
    DmaInteger32      lType; // オブジェクトの種別
    SDBR_PROPLIST     PropList; // プロパティリスト構造体
};
```

メンバの説明

- pOIID
メソッドの実行結果となるオブジェクトの OIID が設定されます。例えば、バージョン付きコンテ

ナの場合は、指定したバージョンに対応するオブジェクトの OIID が設定されます。

なお、次に示すメソッドではバージョン識別子が設定されます。

CdbrVersionable::GetVersionList メソッド

CdbrVersionable::GetVersionListAndLock メソッド

- IType

オブジェクトの種別を表す値として、対応するクラスライブラリのクラスを示す定数が設定されま
す。定数とクラスライブラリのクラスの関係は、「付録 C.3 構造体で指定する定数」を参照してく
ださい。

なお、次に示すメソッドでは IType に相当する情報が設定されても、その内容は保証できません。

CdbrVersionable::GetVersionList メソッド

CdbrVersionable::GetVersionListAndLock メソッド

- PropList

対象オブジェクトが、メソッドの引数で指定したプロパティを持つ場合、プロパティ情報が
SDBR_PROPLIST 構造体として設定されます。

この構造体を出力するメソッド

- CdbrConfiguredReferentialContainer::GetContainableList メソッド
- CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbrContainable::GetContainerList メソッド
- CdbrContainable::GetContainerListAndLock メソッド
- CdbrContainable::GetPublicACLList メソッド
- CdbrContainable::GetPublicACLListAndLock メソッド
- CdbrDocument::GetVersionableList メソッド
- CdbrDocument::GetVersionableListAndLock メソッド
- CdbrIndependentPersistence::GetPublicACLList メソッド
- CdbrIndependentPersistence::GetPublicACLListAndLock メソッド
- CdbrPublicACL::GetBindObjectList メソッド
- CdbrPublicACL::GetBindObjectListAndLock メソッド
- CdbrReferentialContainer::GetContainableList メソッド
- CdbrReferentialContainer::GetContainableListAndLock メソッド
- CdbrVersionable::GetPublicACLList メソッド
- CdbrVersionable::GetPublicACLListAndLock メソッド
- CdbrVersionable::GetVersionList メソッド
- CdbrVersionable::GetVersionListAndLock メソッド
- CdbrVersionTraceableContainer::GetVersionableList メソッド
- CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド

(9) SDBR_OBJLIST 構造体 (オブジェクトリスト構造体)

説明

オブジェクト情報構造体 (SDBR_OBJINFO 構造体) の個数とポインタが設定される構造体です。

形式

```
struct SDBR_OBJLIST {
    DmaInteger32    lCount;    //オブジェクト情報の個数
    SDBR_OBJINFO*  pItem;    //オブジェクト情報
};
```

メンバの説明

- lCount

取得した OBJINFO 構造体の個数が設定されます。

- pItem
SDBR_OBJINFO 構造体の先頭のアドレスへのポインタが設定されます。このとき、SDBR_OBJINFO 構造体の個数分の領域が確保されます。

この構造体を実出力するメソッド

- CdbrConfiguredReferentialContainer::GetContainableList メソッド
- CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbrContainable::GetContainerList メソッド
- CdbrContainable::GetContainerListAndLock メソッド
- CdbrContainable::GetPublicACLList メソッド
- CdbrContainable::GetPublicACLListAndLock メソッド
- CdbrDocument::GetVersionableList メソッド
- CdbrDocument::GetVersionableListAndLock メソッド
- CdbrIndependentPersistence::GetPublicACLList メソッド
- CdbrIndependentPersistence::GetPublicACLListAndLock メソッド
- CdbrPublicACL::GetBindObjectList メソッド
- CdbrPublicACL::GetBindObjectListAndLock メソッド
- CdbrReferentialContainer::GetContainableList メソッド
- CdbrReferentialContainer::GetContainableListAndLock メソッド
- CdbrVersionable::GetPublicACLList メソッド
- CdbrVersionable::GetPublicACLListAndLock メソッド
- CdbrVersionable::GetVersionList メソッド
- CdbrVersionable::GetVersionListAndLock メソッド
- CdbrVersionTraceableContainer::GetVersionableList メソッド
- CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド

(10) SDBR_PATHINFO 構造体 (ファイルパス情報構造体)

説明

マルチファイル管理機能を使用する場合に、一つのファイルの情報を指定する構造体です。

形式

```
struct SDBR_PATHINFO {
    pDmaString_T    pFilePath;          //ファイルのURL
    pDmaString_T    pComponentType;    //ファイルを識別する文字列
};
```

メンバの説明

- pFilePath
ファイルの URL を指定します。"file://" で始まる文字列で指定してください。
URL の値からファイル名が抽出され、そのファイル名がデータベースに登録されます。
- pComponentType
複数のファイルを識別するための任意の文字列を指定します。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 コードで記述してください。
NULL を指定した場合、データベースには空文字列が格納されます。
文字列長は、0 ~ 255 (バイト)

この構造体を入力するメソッド

- CdbrDocument::CreateObject メソッド

- CdbrDocument::UpdateContent メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::UpdateContent メソッド

(11) SDBR_PATHLIST 構造体 (ファイルパス情報リスト構造体)

説明

マルチファイル管理機能を使用する場合に、マルチファイル文書の複数のファイルのファイルパス情報を指定する構造体です。

形式

```
struct SDBR_PATHLIST {
    DmaInteger32      lCount;      //登録ファイル数
    SDBR_PATHINFO*   pItem;      //ファイルパス情報構造体のポインタ
};
```

メンバの説明

- lCount
登録するファイルの数を指定します。指定できるのは、「1 ~ メタ情報の初期設定コマンド (EDMInitMeta) で定義する最大要素数」の範囲の値です。なお、この範囲内の値であっても、BLOB に定義した最大サイズを超えた場合はエラーになります。
- pItem
ファイルパス情報構造体のメンバを配列として確保した領域の先頭アドレスを指定します。

この構造体を入力するメソッド

- CdbrDocument::CreateObject メソッド
- CdbrDocument::UpdateContent メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::UpdateContent メソッド

(12) SDBR_PROP 構造体 (プロパティ構造体)

説明

参照系メソッドによってオブジェクトのプロパティを参照するときに、取得したプロパティの情報が設定される構造体です。

また、更新系メソッドによってオブジェクトのプロパティを設定するときに、設定するプロパティの情報を指定する構造体です。

なお、プロパティ構造体 (SDBR_PROP 構造体) を入力に使用する場合、同じプロパティ識別子を指定したプロパティ構造体 (SDBR_PROP 構造体) をプロパティリスト構造体 (SDBR_PROPLIST 構造体) に複数指定した場合は、エラーになります。

形式

```
struct SDBR_PROP {
    DmaId      PropId;      //プロパティ識別子
    DmaInteger32 lType;      //データ型
    DmaInteger32 lCardinality; //基本単位
    DmaInteger32 lCount;      //値の個数
    union {          //プロパティの値
        pDmaBoolean      pbBoolean;      //Boolean型
        pDmaInteger32    plInteger32;     //Integer32型
        pDmaId           pId;            //ID型
        pDmapv           ppObject;       //Object型
        ppDmaString_T    ppString;       //String型
    } uniValue;
};
```

メンバの説明

出力されるメンバの説明

- PropId
プロパティ識別子が設定されます。
- lType
プロパティのデータ型が設定されます。設定される値を次に示します。
DMA_DATATYPE_BOOLEAN : Boolean 型
DMA_DATATYPE_ID : ID 型
DMA_DATATYPE_INTEGER32 : Integer32 型
DMA_DATATYPE_OBJECT : Object 型
DMA_DATATYPE_STRING : String 型
- lCardinality
基本単位が設定されます。設定される値を次に示します。
DMA_CARDINALITY_SINGLE : Scalar (スカラー) 型
DMA_CARDINALITY_ENUM : Enumeration (列挙) 型
DMA_CARDINALITY_LIST : List (リスト) 型
EDM_DMA_CARDINALITY_VARRAY : VariableArray (可変長配列) 型
- lCount
プロパティの値の要素数が設定されます。要素数は、lCardinality の値に応じて設定されます。lCardinality の値と設定される値を次に示します。
DMA_CARDINALITY_SINGLE の場合 : 1
DMA_CARDINALITY_ENUM または DMA_CARDINALITY_LIST の場合 : 0 以上の値
EDM_DMA_CARDINALITY_VARRAY の場合 : 1
- 共用体 uniValue のメンバ
プロパティの値は、共用体 uniValue のメンバ pbBoolean, plInteger32, pId, ppObject, ppString のうち、lType に設定された型と名前が一致するものが示す領域に設定されます。

注意事項

lType が DMA_DATATYPE_OBJECT で、lCardinality が EDM_DMA_CARDINALITY_VARRAY の場合、CdbVariableArray オブジェクトへのポインタを取得します。このポインタは、構造体を dbrDelete 関数で削除すると、同時に解放または削除されますので、個別に解放する必要はありません。ただし、dbrDelete 関数をコールしたあとで、このポインタは使用できません。

入力するメンバの説明

- PropId
プロパティ識別子を指定します。
同じプロパティ識別子を指定した SDBR_PROP 構造体を SDBR_PROPLIST 構造体に複数指定した場合は、エラーになります。
- lType
プロパティの型を示す値として、次の値を指定します。
DMA_DATATYPE_BOOLEAN : Boolean 型
DMA_DATATYPE_ID : ID 型
DMA_DATATYPE_INTEGER32 : Integer32 型
DMA_DATATYPE_OBJECT : Object 型
DMA_DATATYPE_STRING : String 型
- lCardinality
基本単位として、次の値を指定します。

DMA_CARDINALITY_SINGLE : Scalar (スカラー) 型

DMA_CARDINALITY_LIST : List (リスト) 型

EDM_DMA_CARDINALITY_VARRAY : VariableArray (可変長配列) 型

- lCount

プロパティの値の要素数を指定します。要素数は、lCardinality の値に応じて指定します。

lCardinality の値と指定する値を次に示します。

DMA_CARDINALITY_SINGLE の場合：この値は無視されます。

DMA_CARDINALITY_LIST の場合：0 以上の値を指定します。

EDM_DMA_CARDINALITY_VARRAY の場合：1 を指定します。

- 共用体 uniValue のメンバ

プロパティの値は、共用体 uniValue のメンバ pbBoolean, plInteger32, pId, ppObject, ppString のうち、lType に指定した型と名前が一致するものが示す領域に指定します。

この構造体を出力または入力するメソッド

この構造体を出力するメソッド

- CdbrConfiguredReferentialContainer::GetContainableList メソッド
- CdbrConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbrConfiguredReferentialContainer::GetLinkPropertyValues メソッド
- CdbrConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbrConfiguredReferentialContainer::GetVTContaineelist メソッド
- CdbrConfiguredReferentialContainer::GetVTContaineelistAndLock メソッド
- CdbrContainable::GetContainerList メソッド
- CdbrContainable::GetContainerListAndLock メソッド
- CdbrContainable::GetPublicACLList メソッド
- CdbrContainable::GetPublicACLListAndLock メソッド
- CdbrDMA::GetPropertyValues メソッド
- CdbrDMA::GetPropertyValuesAndLock メソッド
- CdbrDocument::GetRelationList メソッド
- CdbrDocument::GetRelationListAndLock メソッド
- CdbrDocument::GetRenditionList メソッド
- CdbrDocument::GetRenditionListAndLock メソッド
- CdbrDocument::GetVersionableList メソッド
- CdbrDocument::GetVersionableListAndLock メソッド
- CdbrIndependentPersistence::GetPublicACLList メソッド
- CdbrIndependentPersistence::GetPublicACLListAndLock メソッド
- CdbrPublicACL::GetBindObjectList メソッド
- CdbrPublicACL::GetBindObjectListAndLock メソッド
- CdbrReferentialContainer::GetContainableList メソッド
- CdbrReferentialContainer::GetContainableListAndLock メソッド
- CdbrReferentialContainer::GetLinkPropertyValues メソッド
- CdbrReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbrSession::GetUserInfo メソッド
- CdbrVersionable::GetPropertyValues メソッド
- CdbrVersionable::GetPropertyValuesAndLock メソッド
- CdbrVersionable::GetPublicACLList メソッド
- CdbrVersionable::GetPublicACLListAndLock メソッド

- CdbrVersionable::GetVersionList メソッド
- CdbrVersionable::GetVersionListAndLock メソッド
- CdbrVersionableDocument::GetRelationList メソッド
- CdbrVersionableDocument::GetRelationListAndLock メソッド
- CdbrVersionableDocument::GetRenditionList メソッド
- CdbrVersionableDocument::GetRenditionListAndLock メソッド
- CdbrVersionTraceableContainer::GetVersionableList メソッド
- CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド
- CdbrVersionTraceableContainer::GetVTContaineeList メソッド
- CdbrVersionTraceableContainer::GetVTContaineeListAndLock メソッド
- CdbrXmlTranslator::GetDmaInfoList メソッド

この構造体を入力するメソッド

- CdbrConfiguratedReferentialContainer::CreateObject メソッド
- CdbrConfiguratedReferentialContainer::Link メソッド
- CdbrConfiguratedReferentialContainer::LinkAndLock メソッド
- CdbrConfiguratedReferentialContainer::LinkVTFix メソッド
- CdbrConfiguratedReferentialContainer::LinkVTFixAndLock メソッド
- CdbrConfiguratedReferentialContainer::LinkVTFloat メソッド
- CdbrConfiguratedReferentialContainer::LinkVTFloatAndLock メソッド
- CdbrConfiguratedReferentialContainer::PutLinkPropertyValues メソッド
- CdbrDMA::PutPropertyValues メソッド
- CdbrDocument::AddRendition メソッド
- CdbrDocument::CreateObject メソッド
- CdbrDocument::CreateRelation メソッド
- CdbrDocument::PutRelationPropertyValues メソッド
- CdbrDocument::PutRenditionPropertyValues メソッド
- CdbrIndependentPersistence::CreateObject メソッド
- CdbrPublicACL::CreateObject メソッド
- CdbrReferentialContainer::CreateObject メソッド
- CdbrReferentialContainer::Link メソッド
- CdbrReferentialContainer::LinkAndLock メソッド
- CdbrReferentialContainer::PutLinkPropertyValues メソッド
- CdbrVersionable::PutPropertyValues メソッド
- CdbrVersionableDocument::AddRendition メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::CreateRelation メソッド
- CdbrVersionableDocument::PutRelationPropertyValues メソッド
- CdbrVersionableDocument::PutRenditionPropertyValues メソッド
- CdbrVersionTraceableContainer::CreateObject メソッド
- CdbrVersionTraceableContainer::LinkVTFix メソッド
- CdbrVersionTraceableContainer::LinkVTFixAndLock メソッド
- CdbrVersionTraceableContainer::LinkVTFloat メソッド
- CdbrVersionTraceableContainer::LinkVTFloatAndLock メソッド

(13) SDBR_PROPDEF 構造体 (プロパティ定義情報構造体)

説明

オブジェクトのプロパティを参照するときに、参照するプロパティのプロパティ識別子を指定する構造体です。

形式

```
struct SDBR_PROPDEF {
    DmaId      PropId;          // プロパティ識別子
};
```

メンバの説明

- PropId
プロパティ識別子を指定します。

この構造体を入力するメソッド

- CdbConfiguredReferentialContainer::GetContainableList メソッド
- CdbConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbConfiguredReferentialContainer::GetLinkPropertyValues メソッド
- CdbConfiguredReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbConfiguredReferentialContainer::GetVTContaineelist メソッド
- CdbConfiguredReferentialContainer::GetVTContaineelistAndLock メソッド
- CdbContainable::GetContainerList メソッド
- CdbContainable::GetContainerListAndLock メソッド
- CdbContainable::GetPublicACLList メソッド
- CdbContainable::GetPublicACLListAndLock メソッド
- CdbDMA::GetPropertyValues メソッド
- CdbDMA::GetPropertyValuesAndLock メソッド
- CdbDocument::GetRelationList メソッド
- CdbDocument::GetRelationListAndLock メソッド
- CdbDocument::GetRenditionList メソッド
- CdbDocument::GetRenditionListAndLock メソッド
- CdbDocument::GetVersionableList メソッド
- CdbDocument::GetVersionableListAndLock メソッド
- CdbIndependentPersistence::GetPublicACLList メソッド
- CdbIndependentPersistence::GetPublicACLListAndLock メソッド
- CdbPublicACL::GetBindObjectList メソッド
- CdbPublicACL::GetBindObjectListAndLock メソッド
- CdbReferentialContainer::GetContainableList メソッド
- CdbReferentialContainer::GetContainableListAndLock メソッド
- CdbReferentialContainer::GetLinkPropertyValues メソッド
- CdbReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbSession::GetUserInfo メソッド
- CdbVersionable::GetPropertyValues メソッド
- CdbVersionable::GetPropertyValuesAndLock メソッド
- CdbVersionable::GetPublicACLList メソッド
- CdbVersionable::GetPublicACLListAndLock メソッド
- CdbVersionable::GetVersionList メソッド
- CdbVersionable::GetVersionListAndLock メソッド

- CdbVersionableDocument::GetRelationList メソッド
- CdbVersionableDocument::GetRelationListAndLock メソッド
- CdbVersionableDocument::GetRenditionList メソッド
- CdbVersionableDocument::GetRenditionListAndLock メソッド
- CdbVersionTraceableContainer::GetVersionableList メソッド
- CdbVersionTraceableContainer::GetVersionableListAndLock メソッド
- CdbVersionTraceableContainer::GetVTContaineelist メソッド
- CdbVersionTraceableContainer::GetVTContaineelistAndLock メソッド

(14) SDBR_PROPLIST 構造体 (プロパティリスト構造体)

説明

参照系メソッドによってオブジェクトのプロパティを参照するときに、プロパティ構造体 (SDBR_PROP 構造体) の個数とポインタが設定される構造体です。

また、更新系メソッドによってオブジェクトのプロパティを設定するときに、プロパティ構造体 (SDBR_PROP 構造体) とその個数を指定する構造体です。なお、プロパティを設定する場合、同じプロパティ識別子を指定したプロパティ構造体 (SDBR_PROP 構造体) をプロパティリスト構造体 (SDBR_PROPLIST 構造体) に複数指定した場合は、エラーになります。

形式

```
struct SDBR_PROPLIST{
    DmaInteger32    lCount;    //プロパティ情報の個数
    SDBR_PROP*     pItem;     //プロパティ情報
};
```

メンバの説明

出力されるメンバの説明

- lCount
取得した SDBR_PROP 構造体の個数が設定されます。取得する必要がある値がデータベースに設定されていないなどで、プロパティをまったく取得しなかった場合、0 が設定されます。
- pItem
プロパティの識別子と値が格納された SDBR_PROP 構造体の先頭のアドレスへのポインタが設定されます。プロパティをまったく取得しなかった場合、NULL が設定されます。このとき、SDBR_PROP 構造体の個数分の領域が確保されます。

入力するメンバの説明

- lCount
pItem に指定するプロパティ情報の個数を指定します。CreateObject メソッドをコールするときに 0 を指定した場合は、プロパティの値は設定されません。負の数を指定した場合は、エラーになります。
- pItem
初期値として設定するプロパティの識別子と値を格納した SDBR_PROP 構造体の先頭のアドレスを指定します。NULL を指定した場合は、プロパティの値は設定されません。なお、lCount が正の整数のときに pItem に NULL を指定すると、エラーになります。

この構造体を出力または入力するメソッド

この構造体を出力するメソッド

- CdbConfiguredReferentialContainer::GetContainableList メソッド
- CdbConfiguredReferentialContainer::GetContainableListAndLock メソッド
- CdbConfiguredReferentialContainer::GetLinkPropertyValues メソッド

- CdbConfiguratedReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbConfiguratedReferentialContainer::GetVTContaineelist メソッド
- CdbConfiguratedReferentialContainer::GetVTContaineelistAndLock メソッド
- CdbContainable::GetContainerList メソッド
- CdbContainable::GetContainerListAndLock メソッド
- CdbContainable::GetPublicACLList メソッド
- CdbContainable::GetPublicACLListAndLock メソッド
- CdbDMA::GetPropertyValues メソッド
- CdbDMA::GetPropertyValuesAndLock メソッド
- CdbDocument::GetRelationList メソッド
- CdbDocument::GetRelationListAndLock メソッド
- CdbDocument::GetRenditionList メソッド
- CdbDocument::GetRenditionListAndLock メソッド
- CdbDocument::GetVersionableList メソッド
- CdbDocument::GetVersionableListAndLock メソッド
- CdbIndependentPersistence::GetPublicACLList メソッド
- CdbIndependentPersistence::GetPublicACLListAndLock メソッド
- CdbPublicACL::GetBindObjectList メソッド
- CdbPublicACL::GetBindObjectListAndLock メソッド
- CdbReferentialContainer::GetContainableList メソッド
- CdbReferentialContainer::GetContainableListAndLock メソッド
- CdbReferentialContainer::GetLinkPropertyValues メソッド
- CdbReferentialContainer::GetLinkPropertyValuesAndLock メソッド
- CdbSession::GetUserInfo メソッド
- CdbVersionable::GetPropertyValues メソッド
- CdbVersionable::GetPropertyValuesAndLock メソッド
- CdbVersionable::GetPublicACLList メソッド
- CdbVersionable::GetPublicACLListAndLock メソッド
- CdbVersionable::GetVersionList メソッド
- CdbVersionable::GetVersionListAndLock メソッド
- CdbVersionableDocument::GetRelationList メソッド
- CdbVersionableDocument::GetRelationListAndLock メソッド
- CdbVersionableDocument::GetRenditionList メソッド
- CdbVersionableDocument::GetRenditionListAndLock メソッド
- CdbVersionTraceableContainer::GetVersionableList メソッド
- CdbVersionTraceableContainer::GetVersionableListAndLock メソッド
- CdbVersionTraceableContainer::GetVTContaineelist メソッド
- CdbVersionTraceableContainer::GetVTContaineelistAndLock メソッド
- CdbXmlTranslator::GetDmaInfoList メソッド

この構造体を入力するメソッド

- CdbConfiguratedReferentialContainer::CreateObject メソッド
- CdbConfiguratedReferentialContainer::Link メソッド
- CdbConfiguratedReferentialContainer::LinkAndLock メソッド
- CdbConfiguratedReferentialContainer::LinkVTFix メソッド
- CdbConfiguratedReferentialContainer::LinkVTFixAndLock メソッド

- CdbrConfiguredReferentialContainer::LinkVTFloat メソッド
- CdbrConfiguredReferentialContainer::LinkVTFloatAndLock メソッド
- CdbrConfiguredReferentialContainer::PutLinkPropertyValues メソッド
- CdbrDMA::PutPropertyValues メソッド
- CdbrDocument::AddRendition メソッド
- CdbrDocument::CreateObject メソッド
- CdbrDocument::CreateRelation メソッド
- CdbrDocument::PutRelationPropertyValues メソッド
- CdbrDocument::PutRenditionPropertyValues メソッド
- CdbrIndependentPersistence::CreateObject メソッド
- CdbrPublicACL::CreateObject メソッド
- CdbrReferentialContainer::CreateObject メソッド
- CdbrReferentialContainer::Link メソッド
- CdbrReferentialContainer::LinkAndLock メソッド
- CdbrReferentialContainer::PutLinkPropertyValues メソッド
- CdbrVersionable::PutPropertyValues メソッド
- CdbrVersionableDocument::AddRendition メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::CreateRelation メソッド
- CdbrVersionableDocument::PutRelationPropertyValues メソッド
- CdbrVersionableDocument::PutRenditionPropertyValues メソッド
- CdbrVersionTraceableContainer::CreateObject メソッド
- CdbrVersionTraceableContainer::LinkVTFix メソッド
- CdbrVersionTraceableContainer::LinkVTFixAndLock メソッド
- CdbrVersionTraceableContainer::LinkVTFloat メソッド
- CdbrVersionTraceableContainer::LinkVTFloatAndLock メソッド

(15) SDBR_REFERENCE_PATHINFO 構造体 (リファレンスパス情報構造体)

説明

リファレンスファイル管理機能を使用する場合に、リファレンスファイル管理機能に必要な情報が設定される構造体です。

形式

```
// リファレンスパス情報構造体
struct SDBR_REFERENCE_PATHINFO {
    DmaInteger32  lContentOperateMode; // 操作モード
    pDmaString_T  pEntry;             // ファイルのパスおよびロケーション
    pDmaString_T  pTargetPath;       // コンテント格納先パス
    pDmaString_T  pDeleteRootPath;   // 削除ディレクトリのルートパス
};
```

メンバの説明

- lContentOperateMode
リファレンスパス情報構造体で指定するパス操作モードを指定します。リファレンスパス情報構造体で指定する内容は、この指定に基づき、チェックされます。コンテンツロケーションをコンテンツの相対パスで管理する場合は、DBR_OPERATEMODE_USER_RELATIVE_CONTENT を指定します。コンテンツがないオブジェクトを作成する場合は、DBR_OPERATEMODE_NONE を指定します。
- pEntry

文書の作成時に登録するコンテンツのパス、またはコンテンツ取得時のコンテンツのダウンロード先のパスを、URL 形式で指定します。なお、CdbSession::SetReferencePath で指定したコンテンツ格納先ベースパス、pTargetPath、DocumentBroker がコンテンツを管理するためのディレクトリ（35 バイト）、および pEntry で指定したファイル名を合わせた文字列長（パスの区切り文字も含む）が、Windows の場合 1 ~ 259（バイト）、AIX の場合 1 ~ 1022（バイト）となるように指定してください。また、ファイル URL から切り出したファイル名長は、1 ~ 255（バイト）としてください。

- pTargetPath

コンテンツの登録先となるサーバマシン上のディレクトリのパスを指定します。pTargetPath を指定するときの注意事項を次に示します。

DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Origin を指定した場合コンテンツロケーション（ContentLocation プロパティ）には、このメンバで指定された文字列に、DocumentBroker がコンテンツを管理するためのディレクトリおよび pEntry で指定されたファイル名が結合されて格納されます。このとき、"/"（半角）を区切り文字として設定します。

CdbSession::SetReferencePath で指定したコンテンツ格納先ベースパス、pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ（35 バイト）、および pEntry で指定したファイル名を合わせた文字列長（パスの区切り文字も含む）が、Windows のときは 1 ~ 259（バイト）、AIX のときは 1 ~ 1022（バイト）となるように指定してください。

DocumentSpace 構成定義ファイルの ReferenceStorageMode エントリに Divide を指定した場合コンテンツロケーション（ContentLocation プロパティ）には、このメンバで指定された文字列に、DocumentBroker がコンテンツを管理するためのディレクトリおよび DocumentBroker が指定するファイル名が結合されて格納されます。このとき、"/"（半角）を区切り文字として設定します。CdbSession::SetReferencePath で指定したコンテンツ格納先ベースパス、pTargetPath で指定したコンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ（6 バイト）、および DocumentBroker が指定するファイル名（最大 59 バイト）を合わせた文字列長（パスの区切り文字も含む）が、Windows のときは 1 ~ 259（バイト）、AIX のときは 1 ~ 1022（バイト）となるように指定してください。

また、ファイル URL から切り出したファイル名長は、1 ~ 255（バイト）としてください。なお、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 コードで記述できる値を指定してください。

- pDeleteRootPath

文書を削除する場合、登録したコンテンツの削除と同時にコンテンツを格納したディレクトリを削除するときは、削除するディレクトリのルートパスを指定します。ただし、この指定は、コンテンツを登録したオブジェクトの場合だけ有効になります。ディレクトリの削除は、コンテンツロケーションで示されるパス上の格納ディレクトリから、この引数で指定されたディレクトリの下位のディレクトリまでの間のディレクトリが、削除の対象となります。最下位層のディレクトリから順に削除し、エラーとなった時点で処理を終了します。この指定が、コンテンツロケーションのパスと一致していない場合、エラーになります。NULL を指定した場合、登録されているコンテンツおよび DocumentBroker がコンテンツを管理するためのディレクトリが削除されます。

パス長は、Windows の場合 1 ~ 259（バイト）、AIX の場合 1 ~ 1022（バイト）で指定してください。指定できるパスは、CdbSession::SetReferencePath メソッドで指定したコンテンツ格納先ベースパスから、コンテンツ格納先ベースパスと pTargetPath で指定したコンテンツ格納先パス（DocumentBroker がコンテンツを管理するためのディレクトリを含まない）の間のパスです。コンテンツ格納先ベースパスより上位のパスおよび DocumentBroker がコンテンツを管理するためのディレクトリは指定できません。なお、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 コードで記述できる値を指定してください。

この構造体を出力するメソッド

- CdbrDMA::RemoveObject メソッド
- CdbrDocument::AddRendition メソッド
- CdbrDocument:: ConvertContentType メソッド
- CdbrDocument::CreateObject メソッド
- CdbrDocument::GetContentAndLock メソッド
- CdbrDocument::UpdateContentAndRenditionType メソッド
- CdbrVersionableDocument::AddRendition メソッド
- CdbrVersionableDocument::ConvertContentType メソッド
- CdbrVersionableDocument::CreateObject メソッド
- CdbrVersionableDocument::GetContentAndLock メソッド
- CdbrVersionableDocument::UpdateContentAndRenditionType メソッド

(16) SDBR_RELATIONINFO 構造体 (リレーション情報構造体)

説明

文書間リレーション情報の一覧を取得した場合に、個々のリレーションに関する情報が設定される構造体です。

形式

```
// リレーション情報構造体
struct SDBR_RELATIONINFO {
    pDmaString_T    pOIID;           //文書オブジェクトのOIID
    DmaInteger32    lType;           //オブジェクトの種別
    pDmaString_T    pRelId;         //リレーション識別子
    SDBR_PROPLIST   PropList;       //オブジェクトのプロパティリスト
    SDBR_PROPLIST   RelPropList;    //リレーションのプロパティリスト
}
```

メンバの説明

- pOIID
文書を表すオブジェクトの OIID が設定されます。
- lType
オブジェクトの種別を表す値として、対応するクラスライブラリのクラスを示す定数が設定されます。定数とクラスライブラリのクラスの関係は、「付録 C.3 構造体で指定する定数」を参照してください。
- pRelId
リレーション識別子が設定されます。
- PropList
オブジェクトがメソッドの引数 pPropDefList で指定されたプロパティを持つ場合、プロパティの識別子、型、および値の一覧が、SDBR_PROPLIST 構造体として設定されます。
- RelPropList
リレーションがメソッドの引数 pRelLinkPropDefList で指定されたプロパティを持つ場合、プロパティの識別子、型、および値の一覧が、SDBR_PROPLIST 構造体として設定されます。

なお、リレーション先のオブジェクトが削除されていてリレーション情報だけが残っている場合、pOIID = NULL、lType = DBR_CLASS_UNKNOWN、PropList.lCount = 0、PropList.pItem = NULL になり、pRelId と RelPropList だけに正当な値が設定されます。

この構造体を出力するメソッド

- CdbrDocument::GetRelationList メソッド
- CdbrDocument::GetRelationListAndLock メソッド

- CdbrVersionableDocument::GetRelationList メソッド
- CdbrVersionableDocument::GetRelationListAndLock メソッド

(17) SDBR_RELATIONLIST 構造体 (リレーションリスト構造体)

説明

文書間リレーション情報の一覧を取得した場合に、リレーション情報構造体 (SDBR_RELATIONINFO 構造体) の個数とそのポインタが設定される構造体です。

形式

```
// リレーションリスト構造体
struct SDBR_RELATIONLIST {
    DmaInteger32      lCount; //リレーションしている数
    SDBR_RELATIONINFO* pItem; //リレーション情報構造体へのポインタ
}
```

メンバの説明

- lCount
文書間リレーション情報である、SDBR_RELATIONINFO 構造体の個数が設定されます。
- pItem
SDBR_RELATIONINFO 構造体へのポインタが設定されます。この領域は、SDBR_RELATIONINFO 構造体の個数分確保されます。

この構造体を出力するメソッド

- CdbrDocument::GetRelationList メソッド
- CdbrDocument::GetRelationListAndLock メソッド
- CdbrVersionableDocument::GetRelationList メソッド
- CdbrVersionableDocument::GetRelationListAndLock メソッド

(18) SDBR_RENDITION_COMMENTINFO 構造体 (レンディションコメント情報構造体)

説明

コンテンツ変換格納機能を使用する場合に、追加するサブレンディションに付与するコメントの情報をコメント情報構造体で指定します。

形式

```
// レンディションコメント情報構造体
struct SDBR_RENDITION_COMMENTINFO {
    DmaBoolean      bIsInvestSourceComment; // 変換元コメント付与フラグ
    DmaInteger32    lInvestMode;           // コメント付与方式
    pDmaString_T    pRenditionComment;    // レンディションコメント
};
```

メンバの説明

- bIsInvestSourceComment
pRenditionComment で指定するレンディションコメントを変換前のレンディションに付与するか、変換後のレンディションに付与するかを指定します。
 - DMA_TRUE
変換前のレンディションにコメントを付与します。
 - DMA_FALSE
変換後のレンディションにコメントを付与します。
 ユティリティ製品を使用する場合などにマスタレンディションのレンディションタイプを変更したくない場合などでは、bIsChangeMaster と bIsInvestSourceComment の指定によりサブレンディションにコメントを付与します。

- `lInvestMode`
コメントの付与方式を指定します。
 - `DBR_INVESTMODE_APPEND`
レンディションタイプの末尾にコメントを追加します。
 - `DBR_INVESTMODE_REPLACE`
レンディションタイプの末尾にコメントがある場合、コメント部分を指定されたコメントで置き換えます。複数のコメントが存在する場合は、末尾のコメントを置き換えます。コメントがない場合は、レンディションタイプの末尾にコメントを追加します。
- `pRenditionComment`
レンディションに付与するレンディションコメントを指定します。コメントの指定には、コンテンツタイプを表す文字列を指定することを推奨します。レンディションタイプとコメントの間には1バイトのスペースが設定されます。NULLを指定した場合は、DocumentBrokerが自動的にコメント(" edms-cnv ")を付与します。空文字列("")は指定できません。
指定するコメントは、変換するレンディションのレンディションタイプの文字列長とレンディションコメントの文字列長を合わせて255バイト以内にしてください。255バイトを超えた場合は、DMARC_BAD_PARAMETERのエラーとなります。なお、レンディションコメントには、印刷可能なASCIIコードで記述できる値を指定してください。

この構造体を入力するメソッド

- `CdbrDocument::ConvertContentType` メソッド
- `CdbrVersionableDocument::ConvertContentType` メソッド

(19) SDBR_RENDITIONINFO 構造体 (レンディション情報構造体)

説明

マルチレンディション文書のレンディション一覧を取得した場合に、個々のレンディションに関する情報が設定される構造体です。この構造体の配列要素の1番目に格納されるのは、マスタレンディションに関する情報です。サブレンディションの情報は二番目以降に格納されます。

形式

```
// レンディション情報構造体
struct SDBR_RENDITIONINFO {
    pDmaString_T pRenditionType; //RenditionTypeを表す文字列
    SDBR_PROPLIST PropList;      //プロパティリスト構造体
};
```

メンバの説明

- `pRenditionType`
RenditionTypeを表す文字列が設定されます。サブレンディションのRenditionTypeは必ず設定されます。
- `PropList`
オブジェクトがメソッドの引数 `pPropDefList` で指定されたプロパティを持つ場合、そのプロパティの個数、プロパティ識別子、型、および値の一覧が、SDBR_PROPLIST 構造体として設定されます。

この構造体を出力するメソッド

- `CdbrDocument::GetRenditionList` メソッド
- `CdbrDocument::GetRenditionListAndLock` メソッド
- `CdbrVersionableDocument::GetRenditionList` メソッド
- `CdbrVersionableDocument::GetRenditionListAndLock` メソッド

(20) SDBR_RENDITIONLIST 構造体 (レンディションリスト構造体)

説明

マルチレンディション文書のレンディション一覧を取得した場合に、レンディション情報構造体 (SDBR_RENDITIONINFO 構造体) の個数とそのポインタが設定される構造体です。

形式

```
// レンディションリスト構造体
struct SDBR_RENDITIONLIST {
    DmaInteger32      lCount; //レンディション情報の数
    SDBR_RENDITIONINFO* pItem; //レンディション情報構造体
};
```

メンバの説明

- lCount
取得したレンディション情報構造体である、SDBR_RENDITIONINFO 構造体の個数が設定されます。
- pItem
レンディション情報構造体 (SDBR_RENDITIONINFO 構造体) の配列の先頭アドレスが設定されます。この領域は、SDBR_RENDITIONINFO 構造体の個数分確保されます。

この構造体を出力するメソッド

- CdbrDocument::GetRenditionList メソッド
- CdbrDocument::GetRenditionListAndLock メソッド
- CdbrVersionableDocument::GetRenditionList メソッド
- CdbrVersionableDocument::GetRenditionListAndLock メソッド

(21) SDBR_VROBJINFO 構造体 (バージョン識別子付きオブジェクト情報構造体)

説明

指定したオブジェクトを 1 バージョンとして包含しているバージョン管理オブジェクト (CdbrVersionable クラスのサブクラスのオブジェクト) の一覧を取得する時に、そのバージョン管理オブジェクトの情報が設定される構造体です。同時に、メソッドをコールしたオブジェクトがバージョン管理オブジェクトのどのバージョンに当たるかを示すバージョン識別子も取得できます。

形式

```
// バージョン識別子付きオブジェクト情報構造体
struct SDBR_VROBJINFO {
    pDmaString_T      pOIID; // オブジェクトのOIID
    DmaInteger32      lType; // オブジェクトの種別
    pDmaString_T      pVersionId; // バージョン識別子
    SDBR_PROPLIST     PropList; // プロパティリスト構造体
};
```

メンバの説明

- pOIID
オブジェクトの OIID が設定されます。
- lType
オブジェクトの種別を表す値として、対応するクラスライブラリのクラスを示す定数が設定されます。定数とクラスライブラリのクラスの関係は、「付録 C.3 構造体で指定する定数」を参照してください。
- pVersionId
バージョン識別子が設定されます。
- PropList

オブジェクトがメソッドの引数で指定されたプロパティを持つ場合、プロパティリスト (SDBR_PROPLIST 構造体) が設定されます。

この構造体を出力するメソッド

- CdbrDocument::GetVersionableList メソッド
- CdbrDocument::GetVersionableListAndLock メソッド
- CdbrVersionTraceableContainer::GetVersionableList メソッド
- CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド

(22) SDBR_VROBJLIST 構造体 (バージョン識別子付きオブジェクトリスト構造体)

説明

バージョン識別子付きオブジェクト情報構造体 (SDBR_VROBJINFO 構造体) の個数とポインタが設定される構造体です。

形式

```
// バージョン識別子付きオブジェクトリスト構造体
struct SDBR_VROBJLIST {
    DmaInteger32    lCount;
    SDBR_VROBJINFO* pItem;
};
```

メンバの説明

- lCount
バージョン識別子付きオブジェクト構造体である、SDBR_VROBJINFO 構造体の個数が設定されます。
- pItem
SDBR_VROBJINFO 構造体へのポインタが設定されます。この領域は SDBR_VROBJINFO 構造体の個数分確保されます。

この構造体を出力するメソッド

- CdbrDocument::GetVersionableList メソッド
- CdbrDocument::GetVersionableListAndLock メソッド
- CdbrVersionTraceableContainer::GetVersionableList メソッド
- CdbrVersionTraceableContainer::GetVersionableListAndLock メソッド

(23) SDBR_VTLINKEDOBJINFO 構造体 (構成管理リンク付きオブジェクト情報構造体)

説明

バージョン付きコンテナが構成管理しているオブジェクトと、そのオブジェクトを関連づけているリンクの情報が設定される構造体です。

形式

```
struct SDBR_VTLINKEDOBJINFO{
    pDmaString_T  pOIID;           //オブジェクトのOIID
    pDmaString_T  pVerOIID;       //バージョントレースの対象になる
                                   // オブジェクトのOIID
    pDmaString_T  pLinkID;        //リンク識別子
    DmaInteger32  lTraceMode;     //構成管理モード
    DmaInteger32  lType;         //オブジェクトの種別
    SDBR_PROPLIST PropList;      //プロパティリスト
    SDBR_PROPLIST LinkPropList;  //リンクのプロパティリスト
};
```

メンバの説明

- pOIID
指している構成要素のバージョン（構成管理モードが FLOATING モードの場合は最新のバージョン、FIX モードの場合は固定のバージョン）に対応するオブジェクトの OIID が設定されます。DMA オブジェクトでは、DocVersion オブジェクトの OIID または ContainerVersion オブジェクトの OIID に該当します。
- pVerOIID
構成要素の OIID が設定されます。DMA オブジェクトでは、ConfigurationHistory オブジェクトの OIID に該当します。
- pLinkID
対応する構成管理型の関連づけのリンク識別子が設定されます。
- lTraceMode
オブジェクトのバージョンが固定されているかどうかを設定されます。バージョンが固定されている（FIX モード）場合は DBR_MODE_FIX が、バージョンが固定されていない（FLOATING モード）場合は DBR_MODE_UNFIX が設定されます。
- lType
オブジェクトの種別を表す値として、対応するクラスライブラリのクラスを示す定数が設定されます。定数とクラスライブラリのクラスの関係は、「付録 C.3 構造体で指定する定数」を参照してください。
- PropList
オブジェクトがメソッドの引数で指定したプロパティを持つ場合、プロパティ情報のリスト（SDBR_PROPLIST 構造体）が設定されます。
- LinkPropList
リンクがメソッドの引数で指定したプロパティを持つ場合、プロパティ情報のリスト（SDBR_PROPLIST 構造体）が設定されます。

この構造体を出力するメソッド

- CdbConfiguredReferentialContainer::GetVTContaineelist メソッド
- CdbConfiguredReferentialContainer::GetVTContaineelistAndLock メソッド
- CdbVersionTraceableContainer::GetVTContaineelist メソッド
- CdbVersionTraceableContainer::GetVTContaineelistAndLock メソッド

(24) SDBR_VTLINKEDOBJLIST 構造体（構成管理リンク付きオブジェクトリスト構造体）

説明

バージョン付きコンテナが構成管理をしているオブジェクトの情報と関連づけているリンクの情報を参照するときに、構成管理リンク付きオブジェクト情報構造体（SDBR_VTLINKEDOBJINFO 構造体）の個数とポインタが設定される構造体です。

形式

```
struct SDBR_VTLINKEDOBJLIST{
    DmaInteger32          lCount; //リンク付きオブジェクト情報の個数
    SDBR_VTLINKEDOBJINFO* pItem; //リンク付きオブジェクト情報
};
```

メンバの説明

- lCount
取得した SDBR_VTLINKEDOBJLIST 構造体の個数が格納されます。
- pItem
SDBR_VTLINKEDOBJINFO 構造体の先頭のアドレスへのポインタが格納されます。このとき、lCount に設定された SDBR_VTLINKEDOBJINFO 構造体の個数分の領域が確保されます。

この構造体を実出力するメソッド

- CdbConfiguredReferentialContainer::GetVTContaineerList メソッド
- CdbConfiguredReferentialContainer::GetVTContaineerListAndLock メソッド
- CdbVersionTraceableContainer::GetVTContaineerList メソッド
- CdbVersionTraceableContainer::GetVTContaineerListAndLock メソッド

(25) SDBR_VTOBJINFO 構造体 (構成管理オブジェクト情報構造体)

説明

バージョン付きコンテナが構成管理しているオブジェクトの情報が設定される構造体です。

形式

```
struct SDBR_VTOBJINFO{
    pDmaString_T  pOIID;           //オブジェクトのOIID
    pDmaString_T  pVerOIID;       //バージョントレースの対象になる
                                // オブジェクトのOIID
    pDmaString_T  pLinkID;        //オブジェクトのリンク識別子
    DmaInteger32  lTraceMode;     //構成管理モード
    DmaInteger32  lType;          //オブジェクトの種別
    SDBR_PROPLIST PropList;       //プロパティ情報のリスト
};
```

メンバの説明

- pOIID
指している構成要素のバージョン (構成管理モードが FLOATING モードの場合は最新のバージョン, FIX モードの場合は固定のバージョン) に対応するオブジェクトの OIID が設定されます。DMA オブジェクトでは, DocVersion オブジェクトの OIID または ContainerVersion オブジェクトの OIID に該当します。
- pVerOIID
構成要素の OIID が設定されます。DMA オブジェクトでは, ConfigurationHistory オブジェクトの OIID に該当します。
- pLinkID
対応する構成管理型の関連づけのリンク識別子が設定されます。
- lTraceMode
オブジェクトのバージョンが固定されているかどうかを設定されます。バージョンが固定されている (FIX モード) 場合は DBR_MODE_FIX が, バージョンが固定されていない (FLOATING モード) 場合は DBR_MODE_UNFIX が設定されます。
- lType
オブジェクトの種別を表す値として, 対応するクラスライブラリのクラスを示す定数が設定されます。定数とクラスライブラリのクラスの関係は, 「付録 C.3 構造体で指定する定数」を参照してください。
- PropList
オブジェクトがメソッドの引数で指定したプロパティを持つ場合, プロパティリスト構造体 (SDBR_PROPLIST 構造体) が設定されます。

この構造体を実出力するメソッド

- CdbConfiguredReferentialContainer::GetVTContaineerList メソッド
- CdbConfiguredReferentialContainer::GetVTContaineerListAndLock メソッド
- CdbVersionTraceableContainer::GetVTContaineerList メソッド
- CdbVersionTraceableContainer::GetVTContaineerListAndLock メソッド

(26) SDBR_VTOBJLIST 構造体 (構成管理オブジェクトリスト構造体)

説明

バージョン付きコンテナが構成管理しているオブジェクトの情報を参照するときに、オブジェクト情報構造体 (SDBR_VTOBJINFO 構造体) の個数とポインタが設定される構造体です。

形式

```
struct SDBR_VTOBJLIST{
    DmaInteger32    lCount; //構成管理オブジェクト情報の個数
    SDBR_VTOBJINFO* pItem; //構成管理オブジェクト情報
};
```

メンバの説明

- lCount
取得した SDBR_VTOBJINFO 構造体の個数が格納されます。
- pItem
SDBR_VTOBJINFO 構造体の先頭のアドレスへのポインタが設定されます。このとき、lCount に設定された SDBR_VTOBJINFO 構造体の個数分の領域が確保されます。

この構造体を出力するメソッド

- CdbrConfiguredReferentialContainer::GetVTContaineelist メソッド
- CdbrConfiguredReferentialContainer::GetVTContaineelistAndLock メソッド
- CdbrVersionTraceableContainer::GetVTContaineelist メソッド
- CdbrVersionTraceableContainer::GetVTContaineelistAndLock メソッド

付録 C.2 検索に使用する構造体

ここでは、検索に使用する構造体について説明します。

(1) 検索結果として取得する構造体

(a) SDBR_QUERYRESULT 構造体 (検索結果構造体)

説明

検索結果が格納される構造体です。検索を実行したとき、検索結果として取得できるプロパティの個数とプロパティ情報が設定されます。検索結果が 0 件の場合には、検索結果構造体の内容は NULL です。

形式

```
struct SDBR_QUERYRESULT {
    DmaInteger32    lPropCount; //プロパティ個数
    SDBR_PROPERTIES* pProperties; //プロパティ情報
};
```

メンバの説明

- lPropCount
検索結果として取得したプロパティの個数が設定されます。
- pProperties
検索結果として取得したプロパティのプロパティ情報を表す SDBR_PROPERTIES 構造体の先頭のアドレスへのポインタが設定されます。

この構造体を出力するメソッド

- CdbrEqlStatement::GetResult メソッド

(b) SDBR_PROPERTIES 構造体

説明

検索結果として取得したプロパティの情報が設定される構造体です。

形式

```
struct SDBR_PROPERTIES {
    DmaInteger32    lType;           //データ型
    DmaInteger32    lCardinality;   //基本単位
    DmaInteger32    lCount;        //プロパティ値の個数
    union {         //プロパティ値を
                    // 格納する共用体
        pDmaBoolean    pbBoolean;  //Boolean
        pDmaId         pId;        //ID
        pDmaInteger32  plInteger;   //Integer
        pDmapv         pplObject;   //Object
        ppDmaString_T ppString;    //String
    } uValue;
};
```

メンバの説明

• lType

プロパティの型を示す値として、設定される値を次に示します。

DMA_DATATYPE_BOOLEAN : Boolean 型

DMA_DATATYPE_ID : ID 型

DMA_DATATYPE_INTEGER32 : Integer32 型

DMA_DATATYPE_OBJECT : Object 型

DMA_DATATYPE_STRING : String 型

• lCardinality

基本単位が設定されます。設定される値を次に示します。

DMA_CARDINALITY_SINGLE : Scalar (スカラー) 型

DMA_CARDINALITY_ENUM : Enumeration (列挙) 型

DMA_CARDINALITY_LIST : List (リスト) 型

EDM_DMA_CARDINALITY_VARRAY : VariableArray (可変長配列) 型

• lCount

プロパティの値の要素数が設定されます。要素数は、lCardinality の値に応じて設定されます。

lCardinality の値と設定される値を次に示します。なお、検索結果である、

SDBR_QUERYRESULT 構造体の内容が NULL の場合、lCount には 0 が設定されます。

DMA_CARDINALITY_SINGLE の場合 : 1

DMA_CARDINALITY_ENUM または DMA_CARDINALITY_LIST の場合 : 0 以上の値

EDM_DMA_CARDINALITY_VARRAY の場合 : 1

• 共用体 uValue のメンバ

プロパティの値は、共用体 uValue のメンバ pbBoolean, plInteger, pId, pplObject, ppString のうち、lType に設定された型と名前が一致するものが示す領域に設定されます。

この構造体を実出力するメソッド

- CdbxEqlStatement::GetResult メソッド

付録 C.3 構造体で指定する定数

ここでは、複数の構造体で共通して使用する定数について説明します。

(1) クラスライブラリのクラスを表す定数

オブジェクト情報を取得する時に、そのオブジェクトがクラスライブラリで提供するどのクラスのオブジェクトであるかは、構造体に設定される定数から判断できます。この定数は、次に示す構造体の IType にオブジェクトの種別を表す値（クラス識別子）として設定されます。

- SDBR_LINKEDOBJINFO 構造体
- SDBR_OBJINFO 構造体
- SDBR_RELATIONINFO 構造体
- SDBR_VROBJINFO 構造体
- SDBR_VTLINKEDOBJINFO 構造体
- SDBR_VTOBJINFO 構造体

オブジェクトの種別を表す値（クラス識別子）とクラスライブラリのクラスとの対応を、次の表に示します。

表 C-2 オブジェクトの種別を表す値（クラス識別子）とクラスライブラリのクラスとの対応

オブジェクトの種別を表す値（クラス識別子）	クラスライブラリのクラス
DBR_CLASS_ALL	すべてのクラス
DBR_CLASS_CONFIGURATEDREFERENTIALCONTAINER	CdbrConfiguratedReferentialContainer クラス
DBR_CLASS_DOCUMENT	CdbrDocument クラス
DBR_CLASS_INDEPENDENTPERSISTENCE	CdbrIndependentPersistence クラス
DBR_CLASS_PUBLICACL	CdbrPublicACL クラス
DBR_CLASS_REFERENTIALCONTAINER	CdbrReferentialContainer クラス
DBR_CLASS_RFCT_OR_VTCT	CdbrReferentialContainer クラスと CdbrVersionTraceableContainer クラスの OR 条件
DBR_CLASS_VERSIONABLEDOCUMENT	CdbrVersionableDocument クラス
DBR_CLASS_VERSIONTRACEABLECONTAINER	CdbrVersionTraceableContainer クラス
DBR_CLASS_UNKNOWN	クラスの種別が不明

どのクラス識別子が指定できるかについては、それぞれのメソッドを参照してください。

付録 D メソッドの実行に必要なパーミッション (アクセス制御機能を使用する場合)

ここでは、アクセス制御機能を使用している場合に、それぞれのメソッドをコールする時に必要なパーミッションについて説明します。

付録 D.1 パーミッション一覧で使用する記号

ここでは、「付録 D.2 メソッドの実行に必要なパーミッション一覧」で使用する記号について説明します。

パーミッションが必要かどうかを表す記号

- ：メソッドの実行に必要なパーミッションです。
 - ：メソッドの実行に不要なパーミッションです。
- なお、すべてのパーミッションに「-」が入っている場合は、そのメソッドの実行に必要なパーミッションはありません。

複数のオブジェクトに関するメソッドの場合の記号

複数のオブジェクトに関するメソッドでは、どのオブジェクトに対して必要なパーミッションか、「」とともに次の記号で示しています。

表 D-1 パーミッションが必要なオブジェクトを表す記号

記号	意味
b	バインドしているオブジェクトに必要なパーミッション
d	リンクによって関連づけるオブジェクトに必要なパーミッション
dp	リンクによって関連づけるバージョン付きオブジェクトに必要なパーミッション
dv	リンクによって関連づけるバージョンなしオブジェクトに必要なパーミッション
o	メソッドをコールしたオブジェクトに必要なパーミッション
p	バージョン付きオブジェクトに必要なパーミッション
q	検索でヒットするオブジェクトに必要なパーミッション
t	ClassType を取得する対象となるオブジェクトに必要なパーミッション
v	1 バージョンに該当するオブジェクト (バージョンなしオブジェクト) に必要なパーミッション
vc	カレントバージョンに対応する DMA オブジェクトの ContainerVersion オブジェクトに必要なパーミッション
vd	カレントバージョンに対応する DMA オブジェクトの DocVersion オブジェクトに必要なパーミッション

付録 D.2 メソッドの実行に必要なパーミッション一覧

ここでは、メソッドの実行に必要なパーミッションの一覧を、クラスごとに示します。なお、クラスおよびメソッドはアルファベット順に並んでいます。

(1) CdbrCompound クラスのメソッドのコールに必要なパーミッション

CdbrCompound クラスのメソッドのコールに必要なパーミッションはありません。

(2) CdbrConfiguredReferentialContainer クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrConfiguredReferentialContainer クラスのメソッドのコールに必要なパーミッションを

示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-2 CdbConfiguredReferentialContainer クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbConfiguredReferentialContainer	-	-	-	-	-	-	-	-
CdbConfiguredReferentialContainer	-	-	-	-	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-
GetContainableList	o,v,d	-	-	-	-	-	-	-
GetContainableList (リンクのプロパティを指定する場合)	o,v,d	-	-	-	-	-	-	-
GetContainableListAndLock	o,v,d	-	-	-	-	-	-	-
GetContainableListAndLock (リンクのプロパティを指定する場合)	o,v,d	-	-	-	-	-	-	-
GetLinkPropertyValues	o,v	-	-	-	-	-	-	-
GetLinkPropertyValuesAndLock	o,v	-	-	-	-	-	-	-
GetVTContaineeList	o,v,dv	-	-	-	-	-	-	-
GetVTContaineeList (リンクのプロパティを指定する場合)	o,v,dv	-	-	-	-	-	-	-
GetVTContaineeListAndLock	o,v,dv	-	-	-	-	-	-	-
GetVTContaineeListAndLock (リンクのプロパティを指定する場合)	o,v,dv	-	-	-	-	-	-	-
Link	o,d	-	-	-	v	-	-	-
Link (リンクのプロパティを指定する場合)	o,d	-	-	-	v	-	-	-
LinkAndLock	o,d	-	-	-	v	-	-	-
LinkAndLock (リンクのプロパティを指定する場合)	o,d	-	-	-	v	-	-	-
LinkVTFix	o,dp,dv	-	-	-	v	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
LinkVTFixAndLock	o,dp,dv	-	-	-	v	-	-	-
LinkVTFloat	o,dp,dv	-	-	-	v	-	-	-
LinkVTFloatAndLock	o,dp,dv	-	-	-	v	-	-	-
PutLinkPropertyValues	o	-	-	-	v	-	-	-
SetVTFix	o	-	-	-	v	-	-	-
SetVTFloat	o	-	-	-	v	-	-	-
Unlink	o	-	-	-	v	-	-	-
Unlink (リンク識別子を指定する場合)	o	-	-	-	v	-	-	-
UnlinkAndLock	o	-	-	-	v	-	-	-
UnlinkAndLock (リンク識別子を指定する場合)	o	-	-	-	v	-	-	-
UnlinkVT	o	-	-	-	v	-	-	-
UnlinkVTAndLock	o	-	-	-	v	-	-	-

(3) CdbrContainable クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrContainable クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-3 CdbrContainable クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbrContainable	-	-	-	-	-	-	-	-
CdbrContainable	-	-	-	-	-	-	-	-
GetContainerList	o,d	-	-	-	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
GetContainerList (リンクのプロパティを指定する場合)	o,d	-	-	-	-	-	-	-
GetContainerListAndLock	o,d	-	-	-	-	-	-	-
GetContainerListAndLock (リンクのプロパティを指定する場合)	o,d	-	-	-	-	-	-	-

次の表に、CdbrContainable クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-4 CdbrContainable クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッション

メソッド	パーミッション		
	PRIM_READ_PROPS	CREATE	CHANGE_PERM
BindPublicACL	-	-	-
GetPublicACLList	-	-	-
GetPublicACLListAndLock	-	-	-
UnbindPublicACL	-	-	-

(4) CdbrCore クラスのメソッドのコールに必要なパーミッション

CdbrCore クラスのメソッドをコールする場合に必要なパーミッションはありません。

(5) CdbrDMA クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrDMA クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-5 CdbrDMA クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbrDMA	-	-	-	-	-	-	-	-
ConnectObject	-	-	-	-	-	-	-	-
GetOID	-	-	-	-	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
GetPropertyValues	1	-	-	-	-	-	-	-
GetPropertyValuesAndLock	1	-	-	-	-	-	-	-
GetType	-	-	-	-	-	-	-	-
PutPropertyValues	-	-	2	-	-	-	-	-
ReleaseObject	-	-	-	-	-	-	-	-
RemoveObject	-	-	-	-	-	-	3	-
SetOID	-	-	-	-	-	-	-	-

注 1

CdbrPublicACL オブジェクトに対してコールする場合は不要です。

注 2

アクセス制御情報を表すプロパティを更新する場合は、別のアクセス制御情報変更権 (DBR_PERM_CHANGE_PERM) が必要です。ただし、所有者 (dbrProp_OwnerId プロパティ) およびセキュリティ ACL (dbrProp_SACL プロパティ) を更新できるのは、所有者またはセキュリティ管理者だけです。

注 3

バージョン付きオブジェクトとバージョン関係にあるバージョンなしオブジェクトの場合は、バージョン付きオブジェクトに基本バージョン管理権が必要です。

(6) CdbrDocument クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrDocument クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-6 CdbrDocument クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbrDocument	-	-	-	-	-	-	-	-
AddRendition	-	-	-	-	-	-	-	-
CdbrDocument	-	-	-	-	-	-	-	-
ChangeMasterRendition	-	-	-	-	-	-	-	-
CreateIndex	-	-	-	-	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
CreateRelation	d	-	-	-	o	-	-	-
DeleteRendition	-	-	-	-	-	-	-	-
GetContent	-	-	-	-	-	-	-	-
GetContentAndLock	-	-	-	-	-	-	-	-
GetRelationList	o,d	-	-	-	-	-	-	-
GetRelationListAndLock	o,d	-	-	-	-	-	-	-
GetRenditionList	-	-	-	-	-	-	-	-
GetRenditionListAndLock	-	-	-	-	-	-	-	-
GetVersionableList	o,p	-	-	-	-	-	-	-
GetVersionableListAndLock	o,p	-	-	-	-	-	-	-
PutRelationPropertyValues	-	-	-	-	o	-	-	-
PutRenditionPropertyValues	-	-	-	-	-	-	-	-
RemoveIndex	-	-	-	-	-	-	-	-
RemoveRelation	-	-	-	-	o	-	-	-
UpdateContent	-	-	-	-	-	-	-	-
UpdateContentAndRenditionType	-	-	-	-	-	-	-	-

注

リンク先オブジェクトのプロパティを取得しない場合、リンク先オブジェクトの基本プロパティ参照権 (DBR_PERM_PRIM_READ_PROPS) は不要です。

(7) CdbrEqStatement クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrEqStatement クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-7 CdbrEqStatement クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbrEqStatement	-	-	-	-	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
CdbrEqlStatement	-	-	-	-	-	-	-	-
ChangeACLMode	-	-	-	-	-	-	-	-
ChangeGetObjCount	-	-	-	-	-	-	-	-
ChangeLockType	-	-	-	-	-	-	-	-
DeleteResult	-	-	-	-	-	-	-	-
Execute	-	-	-	-	-	-	-	-
GetACLMode	-	-	-	-	-	-	-	-
GetResult	q	q	-	-	-	-	-	-
Initialize	-	-	-	-	-	-	-	-
Set	-	-	-	-	-	-	-	-
SetParam	-	-	-	-	-	-	-	-
SetObjParam	-	-	-	-	-	-	-	-
SetOIDParam	-	-	-	-	-	-	-	-
Terminate	-	-	-	-	-	-	-	-

注

全文検索を実行する場合、基本コンテンツ参照権（PRIM_READ_CONTENTS）がないオブジェクトは取得できません。

（8）CdbrIndependentPersistence クラスのメソッドのコールに必要なパーミッション

次の表に、CdbrIndependentPersistence クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-8 CdbrIndependentPersistence クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbrIndependentPersistence	-	-	-	-	-	-	-	-
CdbrIndependentPersistence	-	-	-	-	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-

次の表に、CdbrIndependentPersistence クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧

で使用する記号」を参照してください。

表 D-9 CdbIndependentPersistence クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッション

メソッド	パーミッション		
	PRIM_READ_PROPS	CREATE	CHANGE_PERM
BindPublicACL	-	-	
GetPublicACLList		-	-
GetPublicACLListAndLock		-	-
UnbindPublicACL	-	-	

(9) CdbPublicACL クラスのメソッドのコールに必要なパーミッション

次の表に、CdbPublicACL クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-10 CdbPublicACL クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッション

メソッド	パーミッション		
	PRIM_READ_PROPS	CREATE	CHANGE_PERM
~CdbPublicACL	-	-	-
CdbPublicACL	-	-	-
CreateObject	-		-
GetBindObjectList	b	-	-
GetBindObjectListAndLock	b	-	-

(10) CdbReferentialContainer クラスのメソッドのコールに必要なパーミッション

次の表に、CdbReferentialContainer クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-11 CdbReferentialContainer クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbReferentialContainer	-	-	-	-	-	-	-	-
CdbReferentialContainer	-	-	-	-	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-
GetContainableList	o,d	-	-	-	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
GetContainableList (リンクの識別子を指定する場合)	o,d	-	-	-	-	-	-	-
GetContainableListAndLock	o,d	-	-	-	-	-	-	-
GetContainableListAndLock (リンクの識別子を指定する場合)	o,d	-	-	-	-	-	-	-
GetLinkPropertyValues		-	-	-	-	-	-	-
GetLinkPropertyValuesAndLock		-	-	-	-	-	-	-
Link	d	-	-	-	o	-	-	-
Link (リンクの識別子を指定する場合)	d	-	-	-	o	-	-	-
LinkAndLock	d	-	-	-	o	-	-	-
LinkAndLock (リンクの識別子を指定する場合)	d	-	-	-	o	-	-	-
PutLinkPropertyValues	-	-	-	-		-	-	-
Unlink	-	-	-	-		-	-	-
Unlink (リンクの識別子を指定する場合)	-	-	-	-		-	-	-
UnlinkAndLock	-	-	-	-		-	-	-
UnlinkAndLock (リンクの識別子を指定する場合)	-	-	-	-		-	-	-

(11) CdbSession クラスのメソッドのコールに必要なパーミッション

CdbSession クラスのメソッドのコールに必要なパーミッションはありません。

(12) CdbVariableArray クラスのメソッドのコールに必要なパーミッション

CdbVariableArray クラスのメソッドのコールに必要なパーミッションはありません。

(13) CdbVersionable クラスのメソッドのコールに必要なパーミッション

次の表に、CdbVersionable クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-12 CdbVersionable クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbVersionable	-	-	-	-	-	-	-	-
CdbVersionable	-	-	-	-	-	-	-	-
DeleteVersion	-	-	-	-	-	o	v	-
GetPropertyValues	o,v	-	-	-	-	-	-	-
GetPropertyValuesAndLock	o,v	-	-	-	-	-	-	-
GetReservationStatus	o	-	-	-	-	-	-	-
GetReservationStatusAndLock	o	-	-	-	-	-	-	-
GetVersionList	o,v	-	-	-	-	-	-	-
GetVersionListAndLock	o,v	-	-	-	-	-	-	-
PutPropertyValues	o	-	v	-	-	-	-	-
VersionCheckIn	-	-	-	-	-	o	-	-
VersionCheckOut	vc	vd	-	-	-	o	-	-
VersionRevoke	-	-	-	-	-	o	-	-

次の表に、CdbVersionable クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-13 CdbVersionable クラスのアクセス制御情報を操作するメソッドのコールに必要なパーミッション

メソッド	パーミッション		
	PRIM_READ_PROPS	CREATE	CHANGE_PERM
BindPublicACL	o	-	v
GetPublicACLList	o,v	-	-
GetPublicACLListAndLock	o,v	-	-
UnbindPublicACL	o	-	v

(14) CdbVersionableDocument クラスのメソッドのコールに必要なパーミッション

次の表に、CdbVersionableDocument クラスのメソッドのコールに必要なパーミッションを示します。
 なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-14 CdbVersionableDocument クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbVersionableDocument	-	-	-	-	-	-	-	-
AddRendition	o	-	-	v	-	-	-	-
CdbVersionableDocument	-	-	-	-	-	-	-	-
ChangeMasterRendition	o	-	-	v	-	-	-	-
CreateIndex	o	-	-	v	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-
CreateRelation	d	-	-	-	o	-	-	-
DeleteRendition	o	-	-	v	-	-	-	-
GetContent	o	v	-	-	-	-	-	-
GetContentAndLock	o	v	-	-	-	-	-	-
GetRelationList	o,d	-	-	-	-	-	-	-
GetRelationListAndLock	o,d	-	-	-	-	-	-	-
GetRenditionList	o,v	-	-	-	-	-	-	-
GetRenditionListAndLock	o,v	-	-	-	-	-	-	-
PutRelationPropertyValues	-	-	-	-	o	-	-	-
PutRenditionPropertyValues	o	-	v	-	-	-	-	-
RemoveIndex	o	-	-	v	-	-	-	-
RemoveRelation	-	-	-	-	o	-	-	-
UpdateContent	o	-	-	v	-	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
UpdateContentAndRenditionType	o	-	-	v	-	-	-	-

注

リンク先オブジェクトのプロパティを取得しない場合、リンク先オブジェクトの基本プロパティ参照権 (DBR_PERM_PRIM_READ_PROPS) は不要です。

(15) CdbVersionTraceableContainer クラスのメソッドのコールに必要なパーミッション

次の表に、CdbVersionTraceableContainer クラスのメソッドのコールに必要なパーミッションを示します。なお、この表で使用する記号については、「付録 D.1 パーミッション一覧で使用する記号」を参照してください。

表 D-15 CdbVersionTraceableContainer クラスのメソッドのコールに必要なパーミッション

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
~CdbVersionTraceableContainer	-	-	-	-	-	-	-	-
CdbVersionTraceableContainer	-	-	-	-	-	-	-	-
CreateObject	-	-	-	-	-	-	-	-
GetVersionableList	o,p	-	-	-	-	-	-	-
GetVersionableListAndLock	o,p	-	-	-	-	-	-	-
GetVTContaineelist	o,dv	-	-	-	-	-	-	-
GetVTContaineelist (リンクのプロパティ指定あり)	o,dv	-	-	-	-	-	-	-
GetVTContaineelistAndLock	o,dv	-	-	-	-	-	-	-
GetVTContaineelistAndLock (リンクのプロパティ指定あり)	o,dv	-	-	-	-	-	-	-
LinkVTFix	dp,dv	-	-	-	o	-	-	-
LinkVTFixAndLock	dp,dv	-	-	-	o	-	-	-
LinkVTFloat	dp,dv	-	-	-	o	-	-	-
LinkVTFloatAndLock	dp,dv	-	-	-	o	-	-	-

メソッド名	パーミッション							
	PRIM_READ_PROPS	PRIM_READ_CONTENTS	PRIM_WRITE_PROPS	PRIM_WRITE_CONTENTS	PRIM_LINK	PRIM_VERSION	PRIM_DELETE	CREATE
SetVTFix	-	-	-	-		-	-	-
SetVTFixAndLock	-	-	-	-		-	-	-
SetVTFloat	-	-	-	-		-	-	-
SetVTFloatAndLock	-	-	-	-		-	-	-
UnlinkVT	-	-	-	-		-	-	-
UnlinkVTAndLock	-	-	-	-		-	-	-

(16) CdbrXmlTranslator クラスのメソッドのコールに必要なパーミッション

CdbrXmlTranslator クラスのメソッドのコールに必要なパーミッションはありません。

(17) CdbrXmlTranslatorFactory クラスのメソッドのコールに必要なパーミッション

CdbrXmlTranslatorFactory クラスのメソッドのコールに必要なパーミッションはありません。

(18) 関数のコールに必要なパーミッション

クラスライブラリで提供している関数, dbrDelete 関数を実行する場合に必要なパーミッションはありません。

付録 E メソッド実行時の排他制御 (ロック)

クラスライブラリの各メソッドには、ユーザが明示的にロックの種類を設定して実行するものと、DocumentBroker によってあらかじめ決められた種類のロックで実行するものがあります。ここでは、これらのロックについて説明します。

なお、クラスライブラリのオブジェクトのうち、どの DMA オブジェクトに対してロックが設定されるかについては、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。

付録 E.1 メソッド実行時の明示的なロックと暗黙のロック

オブジェクトを操作する時には、複数のユーザから同時にオブジェクトに対して操作が実行されないように、排他制御をする必要があります。排他制御はオブジェクトに対してロックを設定することで実現できます。

ロックはメソッドで設定します。クラスライブラリのメソッドによるロックの設定方法には、次に示す方法があります。

処理と同時にロックを設定するメソッド (~ AndLock メソッド) による明示的なロックの設定

処理を実行するときに明示的にロックを設定するメソッドを使用する方法です。

これらのメソッドでは、事前に SetOIID メソッドによって接続したオブジェクトに対して、処理と同時に指定したロックが設定できます。

SetOIID メソッドと ~ AndLock メソッドを組み合わせた操作を実行することで、ConnectObject メソッドによってロックを設定して操作するよりも、データベースへのアクセス回数が減り、高速な処理ができます。

ConnectObject メソッドによる明示的なロックの設定

オブジェクトに接続するときに明示的にロックを設定する、ConnectObject メソッドを使用する方法です。このメソッドでは、クラスライブラリを構成する DMA オブジェクトのうち、トップオブジェクトに当たるオブジェクトにロックを設定します。

検索の実行結果に対する明示的なロックの設定

検索の実行結果に対して明示的にロックを設定する方法です。

CdbrEqLStatement::ChangeLockType メソッドの引数でロックの種類を明示的に指定します。このメソッドで指定したロックが、そのあとで実行する CdbrEqLStatement::Execute メソッドの実行結果 (検索結果集合) に対して設定されます。

なお、この方法で明示的なロックを設定しない場合は、検索の実行結果に対してロックは設定されません。暗黙に設定されるロックはありません。

それ以外のメソッドによる暗黙のロックの設定

ユーザが明示的にロックを設定するメソッドをコールしなかった場合、DocumentBroker によってデータベースのアクセスに必要なロックを暗黙に設定させる方法です。設定されるロックの種類は、メソッドごとに異なります。

付録 E.2 メソッド実行時のロック一覧

ここでは、クラスごとに、明示的にロックを設定できるメソッドの一覧と、あらかじめ DocumentBroker によってメソッドに設定されている暗黙のロックの一覧を示します。

（１）CdbrCompound クラス

（a）明示的にロックを設定できるメソッド

CdbrCompound クラスには、明示的にロックを設定できるメソッドはありません。

（b）暗黙のロックが設定されているメソッドとロックの種類

CdbrCompound クラスには、暗黙のロックが設定されているメソッドはありません。

（２）CdbrConfiguredReferentialContainer クラス

（a）明示的にロックを設定できるメソッド

CdbrConfiguredReferentialContainer クラスの明示的にロックを設定できるメソッドを次に示します。

- GetContainableListAndLock メソッド
- GetLinkPropertyValuesAndLock メソッド
- GetVTContaineedListAndLock メソッド
- LinkAndLock メソッド
- LinkVTFixAndLock メソッド
- LinkVTFloatAndLock メソッド
- UnlinkAndLock メソッド
- UnlinkVTAndLock メソッド

（b）暗黙のロックが設定されているメソッドとロックの種類

CdbrConfiguredReferentialContainer クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-1 CdbrConfiguredReferentialContainer クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrConfiguredReferentialContainer	-
CdbrConfiguredReferentialContainer	-
CreateObject	write ロック
GetContainableList	read ロック
GetLinkPropertyValues	read ロック
GetVTContaineedList	read ロック
Link ¹	read ロック
LinkVTFix ²	read ロック
LinkVTFloat ²	read ロック
PutLinkPropertyValues	write ロック
SetVTFix	read ロック
SetVTFloat	read ロック
Unlink ¹	read ロック
UnlinkVT ²	read ロック

（凡例） - : 該当しません。

注 1

関連づける要素または関連づけを削除する要素にもロックが設定されます。
 DirectContainmentRelationship による関連づけの場合は write ロックが、
 ReferentialContainmentRelationship による関連づけの場合は read ロックが設定されます。

注 2

関連づける要素または関連づけを削除する要素に read ロックが設定されます。

(3) CdbrContainable クラス

(a) 明示的にロックを設定できるメソッド

CdbrContainable クラスの明示的にロックを設定できるメソッドを次に示します。

- GetContainerListAndLock メソッド
- GetPublicACLListAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrContainable クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-2 CdbrContainable クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrContainable	-
BindPublicACL	write ロック
CdbrContainable	-
GetContainerList	read ロック
GetPublicACLList	read ロック
UnbindPublicACL	write ロック

(凡例) - : 該当しません。

(4) CdbrCore クラス

(a) 明示的にロックを設定できるメソッド

CdbrCore クラスには、明示的にロックを設定できるメソッドはありません。

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrCore クラスには、暗黙のロックが設定されているメソッドはありません。

(5) CdbrDMA クラス

(a) 明示的にロックを設定できるメソッド

CdbrDMA クラスの明示的にロックを設定できるメソッドを次に示します。

- GetPropertyValuesAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrDMA クラスの暗黙のロックが設定されているメソッドとロックの種類を次の表に示します。

表 E-3 CdbrDMA クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrDMA	-
CdbrDMA	-
ConnectObject	明示的に指定したロック
GetOIID	-
GetPropertyValues	read ロック
GetType	-
PutPropertyValues	write ロック
ReleaseObject	-
RemoveObject	write ロック
SetOIID	-

（凡例） - : 該当しません。

（6）CdbrDocument クラス

（a）明示的にロックを設定できるメソッド

CdbrDocument クラスの明示的にロックを設定できるメソッドを次に示します。

- GetContentAndLock メソッド
- GetContentInfo メソッド
- GetRelationListAndLock メソッド
- GetRenditionListAndLock メソッド
- GetVersionableListAndLock メソッド

（b）暗黙のロックが設定されているメソッドとロックの種類

CdbrDocument クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-4 CdbrDocument クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrDocument	-
AddRendition	write ロック
CdbrDocument	-
ChangeMasterRendition	write ロック
CreateIndex	write ロック
CreateObject	write ロック
CreateRelation	write ロック
DeleteRendition	write ロック
GetContent	read ロック
GetRelationList	read ロック
GetRenditionList	read ロック
GetVersionableList	read ロック

メソッド	設定されているロックの種類
PutRelationPropertyValues	write ロック
PutRenditionPropertyValues	write ロック
RemoveIndex	write ロック
RemoveRelation	write ロック
UpdateContent	write ロック
UpdateContentAndRenditionType	write ロック

（凡例） - : 該当しません。

（7）CdbrEqStatement クラス

（a）明示的にロックを設定できるメソッド

CdbrEqStatement クラスの明示的にロックを設定できるメソッドを次に示します。

- ChangeLockType メソッド

ChangeLockType メソッドで指定したロックが、そのあとで実行する CdbrEqStatement::Execute メソッドの実行結果（検索結果集合）に対して設定されます。

（b）暗黙のロックが設定されているメソッドとロックの種類

CdbrEqStatement クラスには、暗黙のロックが設定されているメソッドはありません。

（8）CdbrIndependentPersistence クラス

（a）明示的にロックを設定できるメソッド

CdbrIndependentPersistence クラスの明示的にロックを設定できるメソッドを次に示します。

- GetPublicACLListAndLock メソッド

（b）暗黙のロックが設定されているメソッドとロックの種類

CdbrIndependentPersistence クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-5 CdbrIndependentPersistence クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrIndependentPersistence	-
BindPublicACL	write ロック
CdbrIndependentPersistence	-
CreateObject	write ロック
GetPublicACLList	read ロック
UnbindPublicACL	write ロック

（凡例） - : 該当しません。

(9) CdbrPublicACL クラス

(a) 明示的にロックを設定できるメソッド

CdbrPublicACL クラスの明示的にロックを設定できるメソッドを次に示します。

- GetBindObjectListAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrPublicACL クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-6 CdbrPublicACL クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrPublicACL	-
CdbrPublicACL	-
CreateObject	write ロック
GetBindObjectList	read ロック

(凡例) - : 該当しません。

(10) CdbrReferentialContainer クラス

(a) 明示的にロックを設定できるメソッド

CdbrReferentialContainer クラスの明示的にロックを設定できるメソッドを次に示します。

- GetContainableListAndLock メソッド
- GetLinkPropertyValuesAndLock メソッド
- LinkAndLock メソッド
- UnlinkAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrReferentialContainer クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-7 CdbrReferentialContainer クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrReferentialContainer	-
CdbrReferentialContainer	-
CreateObject	write ロック
GetContainableList	read ロック
GetLinkPropertyValues	read ロック
Link	read ロック
PutLinkPropertyValues	write ロック
Unlink	read ロック

(凡例) - : 該当しません。

注

関連づける要素または関連づけを削除する要素にもロックが設定されます。
 DirectContainmentRelationship による関連づけの場合は write ロックが、
 ReferentialContainmentRelationship による関連づけの場合は read ロックが設定されます。

(11) CdbrSession クラス

(a) 明示的にロックを設定できるメソッド

CdbrSession クラスには、明示的にロックを設定できるメソッドはありません。

(b) 暗黙のロックが設定されているメソッド

CdbrSession クラスには、暗黙のロックが設定されているメソッドはありません。

(12) CdbrVariableArray クラス

(a) 明示的にロックを設定できるメソッド

CdbrVariableArray クラスには、明示的にロックを設定できるメソッドはありません。

(b) 暗黙のロックが設定されているメソッド

CdbrVariableArray クラスには、暗黙のロックが設定されているメソッドはありません。

(13) CdbrVersionable クラス

(a) 明示的にロックを設定できるメソッド

CdbrVersionable クラスの明示的にロックを設定できるメソッドを次に示します。

- GetPropertyValuesAndLock メソッド
- GetPublicACLListAndLock メソッド
- GetReservationStatusAndLock メソッド
- GetVersionListAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrVersionable クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-8 CdbrVersionable クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrVersionable	-
BindPublicACL	write ロック
CdbrVersionable	-
DeleteVersion	write ロック
GetPropertyValues	read ロック
GetPublicACLList	read ロック
GetReservationStatus	read ロック
GetVersionList	read ロック
PutPropertyValues	write ロック
UnbindPublicACL	write ロック
VersionCheckIn	write ロック

メソッド	設定されているロックの種類
VersionCheckOut	write ロック
VersionRevoke	write ロック

（凡例） - : 該当しません。

(14) CdbrVersionableDocument クラス

(a) 明示的にロックを設定できるメソッド

CdbrVersionableDocument クラスの明示的にロックを設定できるメソッドを次に示します。

- GetContentAndLock メソッド
- GetContentInfo メソッド
- GetRelationListAndLock メソッド
- GetRenditionListAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbrVersionableDocument クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-9 CdbrVersionableDocument クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbrVersionableDocument	-
AddRendition	write ロック
CdbrVersionableDocument	-
ChangeMasterRendition	write ロック
CreateIndex	write ロック
CreateObject	write ロック
CreateRelation	write ロック
DeleteRendition	write ロック
GetContent	read ロック
GetRelationList	read ロック
GetRenditionList	read ロック
PutRelationPropertyValues	write ロック
PutRenditionPropertyValues	write ロック
RemoveIndex	write ロック
RemoveRelation	write ロック
UpdateContent	write ロック
UpdateContentAndRenditionType	write ロック

（凡例） - : 該当しません。

(15) CdbVersionTraceableContainer クラス

(a) 明示的にロックを設定できるメソッド

CdbVersionTraceableContainer クラスの明示的にロックを設定できるメソッドを次に示します。

- GetVersionableListAndLock メソッド
- GetVTContaineelistAndLock メソッド
- LinkVTFixAndLock メソッド
- LinkVTFloatAndLock メソッド
- SetVTFixAndLock メソッド
- SetVTFloatAndLock メソッド
- UnlinkVTAndLock メソッド

(b) 暗黙のロックが設定されているメソッドとロックの種類

CdbVersionTraceableContainer クラスの暗黙のロックが設定されているメソッドとロックの種類を、次の表に示します。

表 E-10 CdbVersionTraceableContainer クラスの暗黙のロックが設定されているメソッドとロックの種類

メソッド	設定されているロックの種類
~CdbVersionTraceableContainer	-
CdbVersionTraceableContainer	-
CreateObject	write ロック
GetVersionableList	read ロック
GetVTContaineelist	read ロック
LinkVTFix	write ロック
LinkVTFloat	write ロック
SetVTFix	write ロック
SetVTFloat	write ロック
UnlinkVT	write ロック

(凡例) - : 該当しません。

(16) CdbXmlTranslator クラス

(a) 明示的にロックを設定できるメソッド

CdbXmlTranslator クラスには、明示的にロックを設定できるメソッドはありません。

(b) 暗黙のロックが設定されているメソッド

CdbXmlTranslator クラスには、暗黙のロックが設定されているメソッドはありません。

(17) CdbXmlTranslatorFactory クラス

(a) 明示的にロックを設定できるメソッド

CdbXmlTranslatorFactory クラスには、明示的にロックを設定できるメソッドはありません。

(b) 暗黙のロックが設定されているメソッド

CdbrXmlTranslatorFactory クラスには、暗黙のロックが設定されているメソッドはありません。

付録 F 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。

「付録 F(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。

「付録 F(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。

「付録 F(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
```

```

dmasystem = 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 G 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メソッドと関数

ここでは、文書空間で使用する文字コード種別が UTF-8 の場合に使用できるクラス、メソッドおよび関数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、すべてのクラス、メソッドおよび関数が使用できます。

付録 G.1 文字コード種別が UTF-8 の場合に使用できるメソッド

表 G-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メソッド

クラス	メソッド名	使用可否
CdbCompound	~CdbCompound	
	CdbCompound	
	GetCount	
	GetDataType	
	GetId	
	GetValue	
	SetValue	
CdbConfiguratedReferentialContainer	~CdbConfiguratedReferentialContainer	
	CdbConfiguratedReferentialContainer	
	CreateObject	
	GetContainableList	
	GetContainableListAndLock	
	GetLinkPropertyValues	
	GetLinkPropertyValuesAndLock	
	GetVTContaineeList	
	GetVTContaineeListAndLock	
	Link	
	LinkAndLock	
	LinkVTFix	
	LinkVTFixAndLock	
	LinkVTFloat	
	LinkVTFloatAndLock	
	PutLinkPropertyValues	
	SetVTFix	
	SetVTFloat	
	Unlink	
	UnlinkAndLock	
UnlinkVT		
UnlinkVTAndLock		

クラス	メソッド名	使用可否
CdbrContainable	~CdbrContainable	
	BindPublicACL	
	CdbrContainable	
	GetContainerList	
	GetContainerListAndLock	
	GetPublicACLList	
	GetPublicACLListAndLock	
	UnbindPublicACL	
CdbrCore	~CdbrCore	
	CdbrCore	
	GetDBError	
	GetLastError	
CdbrDMA	~CdbrDMA	
	CdbrDMA	
	ConnectObject	
	GetOIID	
	GetPropertyValues	
	GetPropertyValuesAndLock	
	GetType	
	PutPropertyValues	
	ReleaseObject	
	RemoveObject	
	SetOIID	
	CdbrDocument	~CdbrDocument
AddRendition		
CdbrDocument		
ChangeMasterRendition		
ConvertContentType		
CreateIndex		
CreateObject		
CreateRelation		
DeleteRendition		
GetContent		
GetContentAndLock		
GetContentInfo		
GetRelationList		
GetRelationListAndLock		
GetRenditionList		
GetRenditionListAndLock		
GetVersionableList		

クラス	メソッド名	使用可否
	GetVersionableListAndLock	
	PutRelationPropertyValues	
	PutRenditionPropertyValues	
	RemoveIndex	
	RemoveRelation	
	UpdateContent	
	UpdateContentAndRenditionType	
CdbrEqlStatement	~CdbrEqlStatement	
	CdbrEqlStatement	
	ChangeACLMode	
	ChangeGetObjCount	
	ChangeLockType	
	DeleteResult	
	Execute	
	GetACLMode	
	GetResult	
	Initialize	
	Set	
	SetObjParam	
	SetOIIDParam	
	SetParam	
	Terminate	
CdbrIndependentPersistence	~CdbrIndependentPersistence	
	BindPublicACL	
	CdbrIndependentPersistence	
	CreateObject	
	GetPublicACLList	
	GetPublicACLListAndLock	
	UnbindPublicACL	
CdbrPublicACL	~CdbrPublicACL	
	CdbrPublicACL	
	CreateObject	
	GetBindObjectList	
	GetBindObjectListAndLock	

クラス	メソッド名	使用可否
CdbrReferentialContainer	~CdbrReferentialContainer	
	CdbrReferentialContainer	
	CreateObject	
	GetContainableList	
	GetContainableListAndLock	
	GetLinkPropertyValues	
	GetLinkPropertyValuesAndLock	
	Link	
	LinkAndLock	
	PutLinkPropertyValues	
	Unlink	
	UnlinkAndLock	
CdbrSession	~CdbrSession	
	Begin	
	CdbrSession	
	Commit	
	Connect	
	Disconnect	
	GetClassType	
	GetDocSpaceCharacterSet	
	GetLastDetailError	
	GetReferencePath	
	GetUserInfo	
	OrbComponentShutdown	
	OrbComponentStart	
	Rollback	
	SetReferencePath	
CdbrVariableArray	~CdbrVariableArray	
	Add	
	CdbrVariableArray	
	Delete	
	GetCount	
	GetDataType	
	GetValue	
	Resize	
	SetValue	

クラス	メソッド名	使用可否
CdbrVersionable	~CdbrVersionable	
	BindPublicACL	
	CdbrVersionable	
	DeleteVersion	
	GetPropertyValues	
	GetPropertyValuesAndLock	
	GetPublicACLList	
	GetPublicACLListAndLock	
	GetReservationStatus	
	GetReservationStatusAndLock	
	GetVersionList	
	GetVersionListAndLock	
	PutPropertyValues	
	UnbindPublicACL	
	VersionCheckIn	
	VersionCheckOut	
VersionRevoke		
CdbrVersionableDocument	~CdbrVersionableDocument	
	AddRendition	
	CdbrVersionableDocument	
	ChangeMasterRendition	
	ConvertContentType	
	CreateIndex	
	CreateObject	
	CreateRelation	
	DeleteRendition	
	GetContent	
	GetContentAndLock	
	GetContentInfo	
	GetRelationList	
	GetRelationListAndLock	
	GetRenditionList	
	GetRenditionListAndLock	
	PutRelationPropertyValues	
	PutRenditionPropertyValues	
	RemoveIndex	
	RemoveRelation	
	UpdateContent	
UpdateContentAndRenditionType		

クラス	メソッド名	使用可否
CdbVersionTraceableContainer	~CdbVersionTraceableContainer	
	CdbVersionTraceableContainer	
	CreateObject	
	GetVersionableList	
	GetVersionableListAndLock	
	GetVTContaineerList	
	GetVTContaineerListAndLock	
	LinkVTFix	
	LinkVTFixAndLock	
	LinkVTFloat	
	LinkVTFloatAndLock	
	SetVTFix	
	SetVTFixAndLock	
	SetVTFloat	
	SetVTFloatAndLock	
	UnlinkVT	
UnlinkVTAndLock		
CdbXmlTranslator	~CdbXmlTranslator	-
	CdbXmlTranslator	-
	GetDmaInfoList	-
	GetIndexData	-
CdbXmlTranslatorFactory	~CdbXmlTranslatorFactory	-
	CdbXmlTranslatorFactory	-
	CreateTranslator	-
	Initialize	-

(凡例)

- : 使用できます。
- : 使用できません。

付録 G.2 文字コード種別が UTF-8 の場合に使用できる関数

表 G-2 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関数

関数名	使用可否
dbrDelete	
dbrDmaIdFromString	

(凡例)

- : 使用できます。

付録 H 戻り値一覧

ここでは、クラスライブラリのメソッドを実行したときに返される戻り値について説明します。また、クラスライブラリの戻り値一覧を掲載します。

付録 H.1 クラスライブラリの戻り値

クラスライブラリの戻り値は、次の 4 種類です。

- DMA の戻り値
- データベースエラーが発生した場合の戻り値
- クラスライブラリ独自の戻り値
- CORBA の戻り値

クラスライブラリの戻り値には、`major_code` と `minor_code` があります。`major_code` の種類によって、エラーの種類を特定できます。`major_code` および `minor_code` は、`CdbrCore` で定義されている `GetLastError` メソッドによって返却されます。`major_code` はメソッドの戻り値として、`minor_code` はメソッドの引数として返却されます。

次の表に、戻り値の種類と `major_code`、`minor_code` の関係を示します。

表 H-1 戻り値の種類と `major_code`、`minor_code` の関係

戻り値の種類	<code>major_code</code>	<code>major_code</code> の戻り値
エラーがない場合	<code>ERR_NOERROR</code>	0
DMA の戻り値	<code>ERR_DMA</code>	1
CORBA の戻り値	<code>ERR_CORBA</code>	2
クラスライブラリ独自の戻り値	<code>ERR_DBR</code>	3
データベースエラーが発生した場合の戻り値	<code>ERR_DB</code>	4

`major_code` が、`ERR_DMA`、`ERR_DBR` または `ERR_DB` の場合、`minor_code` が取得できます。

なお、`major_code` が `ERR_CORBA` の場合の戻り値については、マニュアル「`VisiBroker for C++` プログラマーズガイド」を参照してください。

付録 H.2 DMA の戻り値

`major_code` が `ERR_DMA` の場合の `minor_code` の一覧を次の表に示します。

表 H-2 `major_code` が `ERR_DMA` の場合の DMA の戻り値一覧

<code>minor_code</code> の戻り値	<code>minor_code</code>	説明	対処
-2147024809	<code>DMARC_BAD_PARAMETER</code>	不正な引数が入力されました。	正しい引数を指定してください。
-2147024882	<code>DMARC_NO_MEMORY</code>	メモリ不足で実行できません。	メモリを増設してください。またはメモリを大量に消費しているアプリケーションを停止させて、処理を再実行してください。

minor_code の 戻り値	minor_code	説明	対処
-2147024891	DMARC_ACCESS_DENIED	要求された操作に対してアクセス権がありません。	正しいアクセス権を設定してください。
-2147217384	EDMRC_NOT_OUTTRANSACTION	トランザクション内でメソッドが呼ばれました。	CdbrSession::Begin() メソッドを呼び出す前に、この操作を実行してください。
-2147217386	EDMRC_FTPSERVICE_NOT_AVAILABLE	ファイル転送サービスが利用できません。	サーバのバージョンに対応したクライアントを利用してください。
-2147217387	EDMRC_FTPSERVICE_MAX_SESSION	ファイル転送サービスに同時に接続できるセッション数の上限に達しています。	時間を置いて、Connect ユーザ数が減った後、再度実行してください。または、ファイル転送サービス環境定義ファイルの FtpSessionMax エントリを見直してください。
-2147217388	EDMRC_NO_FTPSERVICE	使用できるファイル転送サービスがありません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 _HIEDMS_FTPMODE=STATIC を指定してファイル転送サービスを起動しているかどうかを、確認してください。 時間を置いて、再度実行してください。 ファイル転送サービスのセットアップコマンドを実行しているかどうかを、確認してください。
-2147217389	EDMRC_FTPSERVICE_LOST_CONNECTION	ファイル転送サービスへのセッションが切断されました。	ネットワーク環境を確認してください。または、時間を置いて、再度実行してください。
-2147217390	EDMRC_QPARAM_INDEX_OUT_OF_RANGE	? パラメタの設定で指定した値の個数を超える出現順序が指定されています。	? パラメタの設定を見直して、正しい出現順序を指定してください。
-2147217391	EDMRC_QPARAM_BAD_VALUE	? パラメタの設定で指定した値が、不正である可能性があります。	設定した ? パラメタの値とデータ型を見直してください。
-2147217402	EDMRC_IN_ASYNC_EXECUTE	SQL 文の非同期実行中です。	この操作は実行できません。
-2147217404	EDMRC_NOT_INTRANSACTION	トランザクション外でメソッドが呼ばれました。	Begin メソッドを呼び出した後に、この操作を実行してください。
-2147217407	EDMRC_PARSE_ERROR	XML 文書の構文解析エラーが発生しました。	XML 文書の内容を見直してください。
-2147220868	DMARC_CONFLICTING_OPERATION	メソッドの呼び出し順序が矛盾しています。	正しい呼び出し順序でメソッドをコールしてください。
-2147220869	DMARC_MULTIPLE_CHECKIN_NOT_SUPPORTED	同一オブジェクト上の複数の異なるチェックインをサポートしません。	この操作は実行できません。
-2147220870	DMARC_BAD_DOCUMENTSPACE	指定された識別子の文書空間は存在しません。	正しい文書空間を指定してください。
-2147220873	DMARC_AT_ENUM_END	EnumerationOfObject オブジェクトの現在位置は、要素であるオブジェクト群の末尾です。	DocumentBroker サーバのメタ情報ファイル (slocalreg.ini) を見直してください。

minor_code の戻り値	minor_code	説明	対処
-2147220875	DMARC_RESULTS_TRUNCATED	結果集合は IdmaResultSet:TerminateResults の呼び出しにより、切断されます。	この操作は実行できません。
-2147220876	DMARC_NETWORK_ERROR	ネットワーク環境でエラーが発生しました。	ネットワーク環境を確認してください。
-2147220885	DMARC_REQUIRED_VALUE_ABSENT	必要なプロパティ値が設定されていません。	値が必須のプロパティに値が設定されていません。
-2147220887	DMARC_CONSTRAINT_VIOLATED	制約違反の操作です。	この操作は実行できません。
-2147220889	DMARC_DATATYPE_MISMATCH	プロパティのデータ型に適切ではないメソッドです。	正しい型を指定してください。
-2147220900	DMARC_ILLEGAL_OPERATION	DMA 規約で禁止されている操作です。	この操作は実行できません。
-2147220903	DMARC_NOT_AUTHENTICATED	ユーザは認証されていません。	正しいユーザ ID およびパスワードを指定してください。
-2147220906	DMARC_NO_RESERVATION	バージョン識別子が予約されていません。または、チェックアウトされていません。	チェックイン後に割り当てられる正式なバージョン識別子を、入力引数に設定してください。または、チェックアウト状態を確認してください。
-2147220908	DMARC_BAD_PROTECTION_LEVEL	同時更新に対する保護レベルが不正です。	正しい保護レベルを設定してください。
-2147220912	DMARC_TIMEOUT	制限時間を超過しました。	タイムアウトしました。
-2147220921	DMARC_BAD_OPERAND	オペランドのデータ型、形式またはクラスが不正です。	オペランドに正しいデータ型や形式、クラスを指定してください。
-2147220925	DMARC_MAX_ROWS	検索結果の最大項目数を超えたため、検索結果の一部が切り捨てられました。	最大項目数を変更して、再度実行してください。
-2147220932	DMARC_RESERVATION_EXISTS	Version Series は Reservation をすでに保持しています。	この操作は実行できません。
-2147220933	DMARC_RESERVATION_NOT_ALLOWED	Reservation が現在存在しないのに、Version Series の条件により Reservation が許可されませんでした。	この操作は実行できません。
-2147220934	DMARC_RESERVATION_PENDING	コネクションはすでに Reservation 要求を保持しています。一度に一つ以上は存在できません。	この操作は実行できません。
-2147220936	DMARC_NO_CURRENT_VERSION	VersionSeries が最新バージョンを保持していません。	この操作は実行できません。

minor_code の戻り値	minor_code	説明	対処
-2147220938	DMARC_NETWORK_UNAVAILABLE	この操作を実行するために必要なネットワークが利用できません。	<ul style="list-style-type: none"> ネットワーク環境を確認してください。 クライアント側で接続する Document Broker サーバを選択している場合は、接続先の Document Broker サーバの docspace.ini の [Entry0001] セクションに「SelectServerInMultiServer=Yes」の定義があるか確認してください。クライアント側で接続する DocumentBroker サーバを選択していない場合は、「SelectServerInMultiServer=No」と定義しているか確認してください。 DocumentBroker サーバが起動しているか確認してください。 EDMRefresher コマンドが実行中の場合は、コマンド終了後、再度実行してください。
-2147220943	DMARC_DEVICE_ERROR	データベースのアクセスエラーが発生しました。	エラーメッセージを参照してエラーの要因を取り除いてください。
-2147220945	DMARC_BAD_OIID	SetOIID メソッドで設定された OIID が不正です。 または、引数で指定された OIID が不正です。	正しい OIID を指定してください。
-2147220947	DMARC_READ_ONLY	読み取り専用のオブジェクトまたはプロパティのため、異常終了しました。	このプロパティの値は更新できません。
-2147220951	DMARC_BAD_VALUE	次の要因が考えられます。 1. プロパティやリストの値が、許容範囲外の値であるか、または最大値を超えました。 2. VariableArray 型プロパティに設定する可変長配列の要素数が最大値を超えました。 3. docspace.ini の [Entry0001] セクションに「VArrayElementCheck=Yes」の定義がある場合に、VariableArray 型プロパティの要素の値が、許容範囲外の値であるか、または最大値を超えました。	<p>要因に応じて、次の対策を実施してください。</p> <ol style="list-style-type: none"> プロパティやリストの値の範囲を見直してください。 VariableArray 型プロパティに設定する可変長配列の要素数を見直してください。 VariableArray 型プロパティの要素の値の範囲を見直してください。
-2147220953	DMARC_BAD_URL	リソースの URL が不正です。	URL の内容を見直してください。
-2147220955	DMARC_URL_PROTOCOL_NOT_SUPPORTED	サポートしていない URL プロトコルです。	file:/// で始まっているか確認してください。
-2147220957	DMARC_RESOURCE_NOT_FOUND	指示されたリソースが見つかりません。	正しいリソースを指定してください。
-2147220958	DMARC_VALUE_NOT_SET	要求されたプロパティは値を保持していません。	この操作は実行できません。
-2147220962	DMARC_OBJECT_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。

minor_code の戻り値	minor_code	説明	対処
-2147220964	DMARC_NOT_SUPPORTED	次の要因が考えられます。 1. このセッションまたはオブジェクトではサポートされていないメソッドです。または、サポートされていないデータ型です。 2. 未サポートのモードを指定しました。 3. シングルスレッド対応のクライアントアプリケーションから発行されました。	要因に応じて、次の対策を実施してください。 1. この操作は実行できません。 2. このモードは指定できません。 3. マルチスレッド対応のクライアントアプリケーションから発行してください。
-2147220966	DMARC_NOT_FOUND	要求された要素が見つかりません。	正しいオブジェクトを設定してください。
-2147220968	DMARC_DISCONNECTED	文書空間との接続が切断されました。	セッションを接続してください。
-2147220971	DMARC_LOST_CONNECTION	現在のオブジェクトからデータベースへの論理的な接続が失われました。操作は完了できません。	文書空間に再接続してください。
-2147220979	DMARC_FOREIGN_OBJECT	そのオブジェクトは現在の文書空間ではありません。	正しいオブジェクトを指定してください。
-2147220981	DMARC_BAD_PROPID	プロパティの識別子が不正です。	正しいプロパティ識別子を指定してください。
-2147220984	DMARC_BAD_LOCK_TYPE	不正な種類のロックが指定されています。	正しいロックタイプを指定してください。
-2147220986	DMARC_BAD_INDEX	指定されたインデックスは、正当なプロパティまたはリスト要素を参照していません。	正しいインデックスを指定してください。
-2147220988	DMARC_BAD_CLASSID	指定された識別子は、利用できるオブジェクトのクラスではありません。	正しいクラス識別子を指定してください。
-2147418113	DMARC_UNEXPECTED	予期しないエラーが発生しました。	処理を終了してください。

minor_code の戻り値	minor_code	説明	対処
-2147467259	DMARC_FAILED	次の要因が考えられます。 1. 実行した操作の処理に失敗しました。 2. DocumentBroker サーバのメタ情報ファイルが不正です。 3. メモリが不足しています。 4. DocumentSpace 構成定義ファイルの ProcessOrbBoaOption エントリに、-OathreadMax オプションを 129 以上の値で指定している環境で、同時に 129 以上のクライアントから CdbrSession::Connect メソッドが発行されました。 5. メソッド発行順序が不正です。	要因に応じて、次の対策を実施してください。 1. 時間を置いてから再度実行するか、処理を見直してください。 2. DocumentBroker の実行環境ディレクトリ /etc/meta_files (UNIX の場合) または DocumentBroker のインストールディレクトリ %Server%\etc\meta_files (Windows の場合) の下位にあるファイルを削除してから、-F オプションを指定して、EDMPrintMeta コマンドを実行してください。 3. メモリを増設するか、メモリを大量に消費しているアプリケーションを終了して、処理を再度実行してください。 4. 時間を置いて再度実行するか、-OathreadMax オプションの値を 128 以下にしてください。 5. メソッド発行順序を見直してください。
-2147467262	DMARC_BAD_INTERFACE	要求したインターフェースは、このオブジェクトではサポートしていません。	正しいインターフェースを指定してください。

付録 H.3 クラスライブラリ独自の戻り値 (ERR_DBR の場合)

major_code が ERR_DBR の場合の minor_code の一覧を次の表に示します。

表 H-3 major_code が ERR_DBR の場合のクラスライブラリ独自の戻り値一覧

minor_code の戻り値	minor_code	説明	対処
4	ERR_SESSION_ALREADY_CONNECTED	セッションがすでに接続されています。	処理を継続してください。または、いったん切断してから、再度このメソッドをコールしてください。
5	ERR_SESSION_NOT_CONNECTED	次の要因が考えられます。 • セッションが接続されていません。 • EDMRefresher コマンドが実行中です。 • メモリサイズ監視によるサービスプロセスのリフレッシュが実行中です。	要因に応じて、次の対策のどちらかを実施してください。 • セッションを接続してください。 • EDMRefresher コマンドが終了したあと、再度実行してください。 • メモリサイズ監視によるサービスプロセスのリフレッシュが終了したあと、再度実行してください。
7	ERR_BAD_INTERFACE	指定されたインターフェースが不正です。	正しいインターフェースを指定してください。
8	ERR_BAD_PARAMETER	不正な引数が入力されました。	正しい引数を指定してください。

minor_code の 戻り値	minor_code	説明	対処
10	ERR_OBJECT_ALREADY_CONNECTED	オブジェクトはすでに接続されています。	この操作は実行できません。
11	ERR_OBJECT_NOT_CONNECTED	オブジェクトは接続されていません。	オブジェクトを接続してください。
25	ERR_NONEXISTS_VERSION	指定されたバージョンは存在しません。	正しいバージョンを指定してください。
29	ERR_ACLIB_OBJECT_INVALID	クラスライブラリのオブジェクトは無効です。	有効なクラスライブラリのオブジェクトを指定してください。
30	ERR_OIID_NOT_SET	OIID が設定されていません。	SetOIID メソッドをコールしてから、このメソッドをコールしてください。
31	ERR_OBJECT_REMOVED	オブジェクトはすでに削除されています。	正しいオブジェクトを指定してください。
32	ERR_LAST_VERSION	指定されたバージョンは唯一のバージョンであるため、削除できません。	この操作は実行できません。
33	ERR_SESSION_OVERFLOW	制限を超えて Connect しようとしてしました。	時間を置いて、Connect ユーザ数が減った後、再度実行してください。
34	ERR_INVALID_STATUS	セッションの状態が不正です。一つのセッションに同時に複数のメソッド処理要求が出されています。	一つのセッションに対して一度に一つの処理要求しか出さないように見直してください。
35	ERR_NO_SERVICE	使用できるサービスプロセスがありません。	時間を置いて、再度実行してください。 EDMRefresher コマンドが実行中の場合は、コマンド終了後、再度実行してください。
36	ERR_NOTSUPPORT_CLIENT_VERSION	操作を実行するためのネットワークが使用できません。	サーバのバージョンに対応したクライアントライブラリを利用してください。
37	ERR_ACCESS_NOT_PERMITTED	アクセス権がありません。	必要なアクセス権を設定して、再度実行してください。
40	ERR_ACCESSCONTROL_NOT_SUPPORTED	アクセス制御機能が未サポートです。	この操作は実行できません。
41	ERR_BAD_PERMISSION	不正なパーミッションが指定されました。	正しいパーミッションを指定してください。
42	ERR_BAD_SUBJECT_TYPE	不正なサブジェクト種別が指定されました。	正しいサブジェクト種別を指定してください。

minor_code の 戻り値	minor_code	説明	対処
43	ERR_INVALID_SUBJECT_LENGTH	サブジェクトの長さが不正です。	次に示す範囲内でサブジェクトを指定してください。 <ul style="list-style-type: none"> サブジェクトの種別がユーザ識別子の場合、1文字以上、ユーザ識別子の最大長以下の文字列を指定してください。ユーザ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がグループ識別子の場合、1文字以上、グループ識別子の最大長以下の文字列を指定してください。グループ識別子の最大長は、通常 254 文字ですが、DocumentBroker の環境設定時に 512 文字まで拡張できます。 サブジェクトの種別がシステムサブジェクトの場合、1文字以上、254 文字以下の文字列を指定してください。 ユーザ識別子とグループ識別子の最大長を拡張する方法の詳細は、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。
44	ERR_INVALID_ACE_COUNT	設定する ACE の個数が制限値を超えています。	ACE は 64 個以内で指定してください。
45	ERR_INVALID_PUBLICACL_COUNT	バインドするパブリック ACL の個数が制限値を超えています。	パブリック ACL は 10 個以内で指定してください。
47	ERR_INVALID_ACE	不正な ACE が指定されました。	ACE として正しい CdbCompound を指定してください。
48	ERR_PUBLICACL_ALREADY_BOUND	指定されたパブリック ACL はすでにバインドされています。	すでにバインドされているパブリック ACL を指定しないで処理を再実行してください。
49	ERR_PUBLICACL_NOT_BOUND	指定されたパブリック ACL はバインドされていません。	バインドされているパブリック ACL を指定して処理を再実行してください。
50	ERR_NONEXISTS_PUBLICACL	指定されたパブリック ACL は存在しません。	存在しないパブリック ACL を指定しないで処理を再実行してください。
51	ERR_CONTENT_IS_EMPTY	コンテンツが変換処理されてなくて空です。	この操作は実行できません。
52	ERR_IS_MASTERRENDITION	指定された RenditionType はマスタレンディションです。	RenditionType を見直してください。
53	ERR_LIMITNUM_OF_RENDITIONS	保持できるレンディション数の上限に達しています。	この操作は実行できません。

minor_code の戻り値	minor_code	説明	対処
54	ERR_MASTERRENDTYPE_NOT_SET	マスタレンディションの RenditionType に値が設定されていません。	この操作は実行できません。
55	ERR_NONEXISTS_RENDITION	指定された RenditionType は存在しません。	RenditionType を見直してください。
56	ERR_RENDITIONTYPE_DUPLICATED	RenditionType が重複しています。	重複しない RenditionType を指定してください。
57	ERR_RENDITION_NOT_CONVERTED	コンテンツが更新処理されていません。	この操作は実行できません。
58	ERR_CONTENTTYPE_MISMATCH	指定したレンディションタイプのコンテンツに対して実行できません。	正しい文書オブジェクトを指定してメソッドを実行してください。
59	ERR_EQL_ALREADY_INITIALIZED	すでに初期化されています。	この操作は実行できません。
60	ERR_EQL_NOT_INITIALIZED	初期化が実行されていません。	初期化を実行してください。
61	ERR_EQL_STATEMENT_NOT_SET	実行する edmSQL 文が登録されていません。	CdbrEqStatement::Set メソッドで実行する edmSQL 文を登録してください。
62	ERR_EQL_STATEMENT_NOT_EXEC	edmSQL 文が実行されていません。	CdbrEqStatement::Execute メソッドを実行し、edmSQL 文を実行してください。
63	ERR_EQL_BAD_STATEMENT	不正な edmSQL 文を実行しようとしてしました。	edmSQL 文を見直してください。必要に応じて構文解析結果情報を取得してください。
65	ERR_NOT_VERSION_TRACEABLE	バージョントレース可能なオブジェクトではありません。	この操作はできません。オブジェクトのクラスを見直してください。
66	ERR_OBJECT_NOT_INITIALIZE	オブジェクトが初期化されていません。	初期化を実行してください。
67	ERR_OBJECT_ALREADY_INITIALIZED	オブジェクトは初期化済みです。	この操作は実行できません。
68	ERR_INVALID_DEFINITION	定義情報が不正です。	定義ファイルの内容を見直してください。
71	ERR_SESSION_ALREADY_AUTHENTICATED	文書空間にすでにログインしています。	処理を継続するか、文書空間からログアウトしてから、再度このメソッドをコールしてください。
72	ERR_SESSION_NOT_AUTHENTICATE	文書空間にログインしていません。	文書空間にログインしてください。
73	ERR_INVALIDENV	実行環境が不正です。	環境変数「DOCBROKERDIR」に、DocumentBroker の実行環境ディレクトリを正しく指定するか、syslog ファイルに出力されているメッセージを基に対処してください。

minor_code の戻り値	minor_code	説明	対処
74	ERR_OPERATION_FAILED	初期化処理または終了処理に失敗しました。次の要因が考えられます。 <ul style="list-style-type: none"> • docspace.ini メモリ常駐失敗 • OIID 通番メモリ常駐失敗 • メタ情報常駐失敗 • 文書空間接続準備失敗 • 実行環境情報のメモリ常駐失敗 • プロセス間ロック制御失敗 	syslog ファイルに出力されているメッセージを基に対処してください。
75	ERR_NOT_MULTI_CONTENTS	マルチファイル文書ではありません。	マルチファイル管理機能を使用しない場合の形式で実行してください。
76	ERR_BAD_CONTENTSCOUNT	登録しようとするファイルの数が多過ぎます。	登録するファイルの数を見直してください。
77	ERR_RETRIEVALNAME_DUPLICATED	RetrievalName が重複して指定されています。	RetrievalName が重複しないように指定してください。
78	ERR_COMPONENTTYPE_DUPLICATED	ComponentType が重複して指定されています。	ComponentType が重複しないように指定してください。
80	ERR_SYSTEMDBSERVER_NOT_NOTIFY	DB コネクション管理オブジェクトのリファレンスが未通知です。	処理を終了してから、再度実行してください。
81	ERR_FILELINK_FAILED	HiRDB File Link でエラーが発生しました。	詳細メッセージを基に対処してください。
82	ERR_FAL_NOT_LOADED	HiRDB File Link のライブラリがロードできません。	詳細メッセージを基に対処してください。
83	ERR_CONTENT_NOT_REGISTERED	コンテンツが登録されていません。	コンテンツを登録してください。
85	ERR_FILE_REFERENCE_OPERATION_FAILED	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生しました。	詳細メッセージを基に対処してください。
86	ERR_FILE_REFERENCE_MISMATCH_STATUS	リファレンスファイル管理機能で扱うサーバマシンでのコンテンツの操作でエラーが発生し、オブジェクトとコンテンツが不整合な状態となりました。	エラーが発生した操作に応じて、次の対策を実施してください。 <ul style="list-style-type: none"> • 文書の更新のとき 文書を削除したあと、更新後のコンテンツを使用して文書を作成してください。 • 文書の削除のとき 再度実行して文書を削除してください。 • バージョンの削除のとき 再度実行してバージョンを削除してください。 • チェックアウトの取り消しのとき 再度実行してチェックアウトを取り消してください。

minor_code の戻り値	minor_code	説明	対処
87	ERR_TARGETCONTENTPATH_NOT_SET	リファレンスファイル管理機能でコンテンツ格納先ベースパスが設定されていません。または、UNC 形式のパスの場合 (Windows だけ)、コンテンツ格納先ベースパスにコンピュータ名または IP アドレスだけが使用されています。	コンテンツ格納先ベースパスを設定してください。
88	ERR_REFERENCETYPE_MISMATCH	オブジェクト種別が一致していません。	オブジェクト種別または指定した操作モード (IContentOperateMode) を見直してください。
91	ERR_FILE_REFERENCE_CURRENT_CONTENT_NOT_FOUND	リファレンスファイル管理機能でカレントバージョンのコンテンツが存在しません。	<p>エラーが発生したメソッドに応じて、次の対策を実施してください。</p> <ul style="list-style-type: none"> • VersionCheckOut 該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新から順に確認し、コンテンツが存在しないバージョンを削除してください。 • VersionCheckIn CdbrVersionable::VersionRevoke メソッドを実行し、VersionCheckOut を実行してください。必要に応じてコンテンツを更新してください。 • DeleteVersion 該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新 (カレントバージョンを削除した場合は一つ前のバージョン) から順に確認し、コンテンツが存在しないバージョンを削除してください。 • RemoveObject 該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新 (カレントバージョンを削除した場合は一つ前のバージョン) から順に確認し、コンテンツが存在しないバージョンを削除してください。

minor_code の戻り値	minor_code	説明	対処
			<ul style="list-style-type: none"> VersionRevoke 該当するレンディションを削除し、必要に応じて、削除したレンディションを追加してください。または、バージョンを最新から順に確認し、コンテンツが存在しないバージョンを削除してください。
92	ERR_CONVERT_CONTENT_TARGET_NOTFOUND	変換対象のコンテンツは存在しません。	オブジェクトのコンテンツ種別または指定した変換対象とするコンテンツ種別 (ISourceContentType) を見直してください。
93	ERR_CONVERT_CONTENT_NOT_RETRIEVALNAME	変換対象コンテンツの dmaProp_RetrievalName プロパティの値が設定されていない (値が NULL または空文字 ("")) ため、変換処理を実行できません。	dmaProp_RetrievalName プロパティの値を設定してメソッドを実行してください。

付録 H.4 データベースエラーが発生した場合の戻り値 (ERR_DB の場合)

major_code が ERR_DB の場合の minor_code の一覧を、次の表に示します。

表 H-4 major_code が ERR_DB の場合のクラスライブラリ独自の戻り値一覧

minor_code の戻り値	minor_code	説明	対処
26	ERR_DB_LOCKED	オブジェクトがロックされています。	時間を置いて、再度実行してください。
27	ERR_DB_DEADLOCK_OCCURRED	データベースでデッドロックが発生しました。	時間を置いて、再度実行してください。
28	ERR_DB_FAILED	データベースで障害が発生しました。	syslog を参照して、障害の原因を取り除いてください。

付録 H.5 詳細メッセージの出力について

アプリケーション開発でのデバッグ時および実行時のエラーに具体的に対処できるように、DocumentBroker では、より詳細なエラー情報 (詳細メッセージ) を提供しています。

(1) 詳細メッセージの出力先

詳細メッセージの出力先は、次のとおりです。なお、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。

詳細エラーログファイル

クライアント環境変数 DBR_DETAIL_ERRORLOG が ON のとき、DBR_DETAIL_ERRORLOG_DIR で指定されたディレクトリの下に「EDMErrTraceCLXXXXX_1.log」として出力されます。XXXX はプロセス ID 番号を示します。詳細エラーログファイルには、エラーが発生しないときでもログが出力されますので、必要に応じて詳細エラーログファイルを削除するようにしてください。

クライアント環境変数については、マニュアル「DocumentBroker Version 3 クラスライブラリ C++ 解説」の「5. 環境設定」を参照してください。

CdbrSession::GetLastDetailError メソッドの引数

詳細メッセージは、それを取得するためのメソッドである CdbrSession::GetLastDetailError メソッドをコールした場合に、引数の値として出力されます。

このメソッドは、クラスライブラリのメソッドのコール時にエラーが発生した場合に、CdbrCore::GetLastError メソッドで major_code, minor_code を取得した後で必要に応じてコールします。メソッドの詳細については、「13. CdbrSession クラス (セッションを確立するクラス)」の CdbrSession::GetLastDetailError メソッドを参照してください。

(2) 詳細メッセージ取得の対象メソッド, 取得できる条件および注意事項

(a) 詳細メッセージを取得できる対象メソッド

次に示すクラスのメソッドがエラーリターンした場合に、CdbrSession::GetLastDetailError メソッドで詳細メッセージを取得できます。

クラス名	メソッド名
CdbrConfiguratedReferentialContainer	<ul style="list-style-type: none"> • CreateObject • GetLinkPropertyValues • GetLinkPropertyValuesAndLock • PutLinkPropertyValues
CdbrDocument	CreateObject
CdbrEqlStatement	<ul style="list-style-type: none"> • ChangeACLMode • ChangeGetObjCount • ChangeLockType • DeleteResult • Execute • GetACLMode • GetResult • Initialize • Set • SetObjParam • SetOIIDParam • SetParam • Terminate
CdbrIndependentPersistence	CreateObject
CdbrPublicACL	CreateObject
CdbrReferentialContainer	<ul style="list-style-type: none"> • CreateObject • GetLinkPropertyValues • GetLinkPropertyValuesAndLock • PutLinkPropertyValues
CdbrVersionableDocument	CreateObject
CdbrVersionTraceableContainer	CreateObject

(b) 詳細メッセージを取得できる条件

CdbrSession::GetLastDetailError メソッドは、取得するメッセージがない場合、引数 ppDetailErrorMessage の指すポインタの値が NULL になります。引数 ppDetailErrorMessage の指すポインタの値が NULL 以外の場合に詳細メッセージを取得できます。

(c) 注意事項

詳細メッセージの取得についての注意事項を次に示します。

すべてのエラーに対して、詳細メッセージとして KMBR20000-E が出力されます。

クライアントで検知した引数不正に対しても詳細メッセージとして KMBR26000-E が出力されます。

クライアントで検知する引数不正には、次のような場合があります。

- NULL 値を指定できないポインタ引数に対して NULL 値が設定された場合

なお、この引数不正が発生した場合でも、次に示すメソッドに関しては詳細メッセージを取得できません。

ConnectObject メソッド

CreateObject メソッド

SetOID メソッド

Initialize メソッド

アクセス権を操作するメソッドのエラー時およびアクセス権エラー時に詳細メッセージは取得できません。

付録 I 障害が発生した時に DocumentBroker が出力する情報

ここでは、障害が発生した時に、DocumentBroker が出力する情報について説明します。

(1) メッセージ

運用中のメッセージは、syslog に出力されます。

(2) トレースファイル

発生した障害の切り分けや、原因究明のためにトレースファイルを出力します。関数トレースやメッセージも出力されます。

(a) 出力先ディレクトリ

トレースファイルは、環境変数「_HIEDMS_TRACE_DIR」に指定したディレクトリに出力されます。

デフォルトの出力先は、UNIX の場合は<インストールディレクトリ> /spool/client カレントディレクトリ、Windows の場合は<インストールディレクトリ> %spool%\client カレントディレクトリです。

(b) 出力ファイル名

トレースファイルの出力ファイル名は、環境変数に指定できません。次のファイル名で出力されます。

```
EDMRasTraceCL"PID"_"NO".log
```

注意：「PID」はプロセス ID です。「NO」は項番です。

(c) 切り替えファイル数

トレース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指定できます。環境変数「_HIEDMS_TRACE_NUM」に、切り替えることができるファイルの数を 0、または 2 ~ 16 の値で指定します。デフォルトは 2 です。切り替えるファイルの数の上限を超えると、最初のファイルに戻って出力します。このとき、ファイルは上書きされます。0 を指定した場合、ファイルは作成されません。この場合、障害の切り分けや障害の原因究明などに必要な情報は出力されませんの注意してください。

(d) ファイルサイズ

トレースを出力するファイルのサイズを、環境変数「_HIEDMS_TRACE_SIZE」に 4096 ~ 2147483647 の値で指定します。

デフォルトは 1048576 (1 メガバイト) です。

(e) トレースの出力レベル

トレースの出力レベルを変更できます。環境変数「_HIEDMS_TRACE_LEVEL」にトレースレベルを指定します。デフォルトは 10 です。トレースレベルと出力情報について、次の表に示します。

表 I-1 トレースレベルと出力情報

トレースレベル	出力情報
-1	<ul style="list-style-type: none"> トレースヘッダ
0	<ul style="list-style-type: none"> エラーなどの必須情報 サーバの開始 / 終了

トレースレベル	出力情報
10 (デフォルト)	<ul style="list-style-type: none"> • トレースレベルが 0 の場合に出力される情報 • ユーザインターフェースの情報 • ほかのプログラムとのインターフェースの情報 • データベースへの接続 / 切断

トレースレベルが -1 の場合、環境変数「_HIEDMS_TRACE_NUM」にデフォルトの値または 2 以上の値が指定されていても、トレースを出力するファイルの数は一つです。

(f) トレースファイルの保存日数

トレースファイルは、環境変数「_HIEDMS_TRACE_KEEP_DAYS_CLT」に指定した日数だけ保存されます。1 ~ 365 の日数を指定します。デフォルトは 0 (日) です。0 を指定した場合はトレースファイルは削除しません。

クライアントアプリケーションの停止時、トレースファイルの出力先ディレクトリに、環境変数「_HIEDMS_TRACE_KEEP_DAYS_CLT」で指定した保存日数を超えるトレースファイルが存在する場合、これらのファイルは削除されます。

削除対象は、「(a) 出力先ディレクトリ」の指定内容に対応した出力先のファイルです。

一つのシステムでは環境変数「_HIEDMS_TRACE_KEEP_DAYS」と「_HIEDMS_TRACE_KEEP_DAYS_CLT」は同じ値を設定することを推奨します。

なお、環境変数「_HIEDMS_TRACE_KEEP_DAYS」の詳細については、マニュアル「uCosminexus DocumentBroker Version 3 システム導入・運用ガイド」の、障害が発生した時に DocumentBroker が出力する情報に関する説明を参照してください。

(3) サービスのワークディレクトリの削除

ファイル転送サービス起動時に、環境変数「_HIEDMS_TRACE_KEEP_DAYS_CLT」で指定した保存日数を超えるサービスのワークディレクトリを削除します。ファイル転送サービスが作成していないディレクトリの場合でも、名称が 14 桁の数字で構成されているディレクトリは削除の対象となります。

削除対象となるディレクトリの例を次に示します。

- <インストールディレクトリ>¥DevKit¥spool¥YYYYMMDDhhmmss
- <インストールディレクトリ>¥Runtime¥spool¥YYYYMMDDhhmmss

注

サービスを起動した時刻を表す文字列です。

- YYYY : 西暦年号 (4 桁)
- MM : 月 (2 桁)
- DD : 日 (2 桁)
- hh : 時 (2 桁)
- mm : 分 (2 桁)
- ss : 秒 (2 桁)

付録 J 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。

CdbXmlTranslatorFactory オブジェクト

XML プロパティマッピング機能や XML インデクスデータ作成機能を実行するための操作環境を管理するオブジェクトです。

また、CdbXmlTranslator オブジェクトを作成します。

CdbXmlTranslator オブジェクト

XML プロパティマッピング機能および XML インデクスデータ作成機能を実行するオブジェクトです。

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。

DCD (Document Content Description)

W3C で定義している, タグセットを記述するためのタグセットです。マッピング元 XML タグ定義はこの DCD の形式で記述します。DTD と同じように XML 文書の構造を記述しますが, DTD と異なり, DCD 自体が XML のタグセットとして定義されています。また, タグおよびその属性について, データの型を指定できるといった特長を持ちます。

DeclSet オブジェクト

DTD を管理する DMA オブジェクトです。宣言の集合は, 内部 DTD と外部 DTD を合併して, パラメタエンティティを展開した形で格納されます。

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。CdbrDocument クラスまたは CdbrVersionableDocument 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』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

直接型のコンテナメントの場合、オブジェクト (Child) を包含するオブジェクトを指します。

RD エリア

データベースの表、インデクスおよびデータディクショナリを格納するデータ領域のことです。データの格納単位の一つで、1 ~ 16 個の HiRDB ファイルから構成されます。

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。

VersionTracedComponentDocVersion オブジェクト

直接型、参照型および構成管理型のコンテナ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_VersionTracedComponentDocVersion クラスを基に作成します。

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ナメントで、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ナメントの場合、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ナメントの場合、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。

XML インデクスデータ作成機能

XML 文書の文書ファイルからプレーンテキスト形式または構造指定検索用の全文検索インデクスデータを作成する機能です。XML 形式のファイルの構文解析もあわせて実行します。

XML プロパティマッピング機能

XML 形式のファイルを構文解析して、DocumentBroker サーバのプロパティマッピング定義に従って XML 形式のファイル内に記述されているタグ間の文字列または属性値を、XML 文書のプロパティに割り当てて設定する機能です。

(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 (OIID) を利用して、そのオブジェクトを探索することです。

異表記展開検索

全文検索条件として指定する検索タームまたは検索タームの異表記を含む文書を検索する方法です。例えば、検索タームとして「バイオリン」を指定した場合に、「ヴァイオリン」という検索タームの異表記を含む文書も検索できます。

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います。

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。

(カ行)

概念検索

ユーザが任意に指定した文章や文字列を手がかりにして、その条件と似た概念を持つ文書を検索する方法です。全文検索の一種です。概念検索で指定する条件のことを種文章といいます。

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

コンテナメントの型 (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、コンテインメントの設定、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 コマンドでも作成できます。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。

検索結果構造体

検索結果が格納される構造体です。

構成管理型のコンテインメント (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イン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イン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。

コンテインメント (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。コンテインメントには直接型 (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

コンテインメントで、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

全文検索の対象と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認することです。

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。

同義語展開検索

全文検索条件として指定する検索タームまたは検索タームの同義語を含む文書を検索する方法です。例えば、検索タームとして「パソコン」を指定した場合に、「電子計算機」、「パーソナルコンピュータ」、「PC」など、検索タームと同じ意味を持つ単語を含む文書も検索できます。

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。CdbrVersionTraceableContainer 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。

プロパティマッピング定義ファイル（DPM）

XML 文書中のタグ名とその内容をマッピングするクラス名，およびプロパティ名の対応関係の定義（プロパティマッピング定義）を記述したファイルです。この定義ファイルを基に，XML 定義ファイルの追加 / 更新 / 削除コマンド（EDMXmlMap）によってマッピング定義ファイルを生成します。

プロパティマッピング定義は，登録時にマッピング定義名を付けて登録します。プロパティマッピングの実行時には，このマッピング定義名を指定して使用するプロパティマッピング定義を選択します。

文書

CdbrVersionableDocument オブジェクトおよび CdbrDocument オブジェクトに相当します。

CdbrVersionableDocument オブジェクトを，「バージョン付き文書」と呼びます。CdbrDocument 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。

マッピングセット定義ファイル (XMS)

HiRDB Adapter for XML が使用する、登録されたマッピング定義の一覧を管理するファイルです。

マッピング定義ファイル (XMP)

HiRDB Adapter for XML が使用できる文法に変換したマッピング定義が記述されているファイルです。ユーザ作成のプロパティマッピング定義ファイル (DPM) とマッピング元 XML タグ定義ファイル (DCD) から生成します。

マッピング元 XML タグ定義ファイル (DCD)

登録する XML 文書の構造を DCD の形式で記述したファイルです。

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 サーバが存在するマシン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 サーバが存在するマシン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、データベースでは、文書のプロパティおよび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。

レンディションのコンテンツ種別変換機能

レンディションのコンテンツの格納先を、最適な格納先へ変換することができる機能です。

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

記号

~CdbCompound メソッド 23
~CdbConfiguratedReferentialContainer メソッド 39
~CdbContainable メソッド 120
~CdbCore メソッド 147
~CdbDMA メソッド 154
~CdbDocument メソッド 187
~CdbEqlStatement メソッド 290
~CdbIndependentPersistence メソッド 324
~CdbPublicACL メソッド 345
~CdbReferentialContainer メソッド 366
~CdbSession メソッド 411
~CdbVariableArray メソッド 437
~CdbVersionableDocument メソッド 509
~CdbVersionable メソッド 452
~CdbVersionTraceableContainer メソッド 610
~CdbXmlTranslatorFactory メソッド 681
~CdbXmlTranslator メソッド 670

A

ACE 817
ACFlag 817
ACL 817
AddRendition メソッド 188, 510
Add メソッド 438
AND-NOT 検索 817
AND 検索 817
API 817

B

Begin メソッド 412
BindPublicACL メソッド 121, 325, 453

C

CdbCompound オブジェクト 817
CdbCompound クラス 21
CdbCompound クラスの詳細 22
CdbCompound メソッド 24
CdbConfiguratedReferentialContainer オブジェクト 817
CdbConfiguratedReferentialContainer クラス 31
CdbConfiguratedReferentialContainer クラスの詳細 33
CdbConfiguratedReferentialContainer メソッド 40

CdbContainable クラス 115
CdbContainable クラスの詳細 116
CdbContainable メソッド 123
CdbCore クラス 145
CdbCore クラスの詳細 146
CdbCore メソッド 148
CdbDMA クラス 151
CdbDMA クラスの詳細 152
CdbDMA メソッド 155
CdbDocument オブジェクト 817
CdbDocument クラス 179
CdbDocument クラスの詳細 181
CdbDocument メソッド 195
CdbEqlStatement オブジェクト 817
CdbEqlStatement クラス 287
CdbEqlStatement クラスの詳細 288
CdbEqlStatement メソッド 291
CdbIndependentPersistence オブジェクト 817
CdbIndependentPersistence クラス 319
CdbIndependentPersistence クラスの詳細 320
CdbIndependentPersistence メソッド 327
CdbPublicACL オブジェクト 817
CdbPublicACL クラス 341
CdbPublicACL クラスの詳細 342
CdbPublicACL メソッド 346
CdbReferentialContainer オブジェクト 817
CdbReferentialContainer クラス 361
CdbReferentialContainer クラスの詳細 362
CdbReferentialContainer メソッド 367
CdbSession オブジェクト 818
CdbSession クラス 407
CdbSession クラスの詳細 409
CdbSession メソッド 413
CdbVariableArray オブジェクト 818
CdbVariableArray クラス 435
CdbVariableArray クラスの詳細 436
CdbVariableArray メソッド 439
CdbVersionableDocument オブジェクト 818
CdbVersionableDocument クラス 501
CdbVersionableDocument クラスの詳細 503
CdbVersionableDocument メソッド 517
CdbVersionable クラス 447
CdbVersionable クラスの詳細 448
CdbVersionable メソッド 455
CdbVersionTraceableContainer オブジェクト 818
CdbVersionTraceableContainer クラス 603
CdbVersionTraceableContainer クラスの詳細 605

CdbrVersionTraceableContainer メソッド 611
 CdbrXmlTranslatorFactory オブジェクト 818
 CdbrXmlTranslatorFactory クラス 679
 CdbrXmlTranslatorFactory クラスの詳細 680
 CdbrXmlTranslatorFactory メソッド 682
 CdbrXmlTranslator オブジェクト 818
 CdbrXmlTranslator オブジェクトの生成 683
 CdbrXmlTranslator クラス 667
 CdbrXmlTranslator クラスの詳細 668
 CdbrXmlTranslator メソッド 671
 ChangeACLMode メソッド 292
 ChangeGetObjCount メソッド 294
 ChangeLockType メソッド 295
 ChangeMasterRendition メソッド 196, 518
 Child 818
 Commit メソッド 414
 ConfigurationHistory オブジェクト 818
 ConnectObject メソッド 156
 Connect メソッド 415
 Containable オブジェクト 818
 Containable オブジェクトとの関連づけ 79
 Containable オブジェクトとの関連づけと同時に指定したロックの設定 82
 Containable オブジェクトとの関連づけの解除 105
 Containable オブジェクトとの関連づけの解除と同時に指定したロックの設定 107
 Containee 818
 Container 818
 ContainerVersion オブジェクト 818
 Container オブジェクト 818
 ContentElement 819
 ContentFileLink オブジェクト 819
 ContentReference オブジェクト 819
 ContentTransfers オブジェクト 819
 ContentTransfer オブジェクト 819
 ConvertContentType メソッド 198, 521
 CORBA 819
 CreateIndex メソッド 203, 527
 CreateObjectメソッド 41, 206, 328, 347, 368, 530, 612
 CreateRelation メソッド 216, 540
 CreateTranslator メソッド 683

D

dbrDelete 関数 689
 dbrDmaIdFromString 関数 691
 DCD 819
 DeclSet オブジェクト 819
 DeleteRendition メソッド 219, 543

DeleteResult メソッド 297
 DeleteVersion メソッド 456
 Delete メソッド 440
 Disconnect メソッド 418
 DMA 819
 dmaClass_ConfigurationHistory クラスのプロパティ 723
 dmaClass_Container クラスのプロパティ 723
 dmaClass_DirectContainmentRelationship クラスのプロパティ 724
 dmaClass_DocVersion クラスのプロパティ 725
 dmaClass_ReferentialContainmentRelationship クラスのプロパティ 726
 dmaClass_Rendition クラスのプロパティ 727
 dmaProp_Children プロパティ 735
 dmaProp_ClassDescription プロパティ 735
 dmaProp_Containees プロパティ 736
 dmaProp_Containers プロパティ 736
 dmaProp_ContentElementsPresent プロパティ 736
 dmaProp_ContentElements プロパティ 736
 dmaProp_CurrentOfSeriesCount プロパティ 736
 dmaProp_Head プロパティ 737
 dmaProp_OIID プロパティ 737
 dmaProp_ParentContainer プロパティ 737
 dmaProp_Parent プロパティ 737
 dmaProp_PrimaryVersionSeries プロパティ 737
 dmaProp_RenditionsPresent プロパティ 738
 dmaProp_Renditions プロパティ 737
 dmaProp_RenditionType プロパティ 738
 dmaProp_Tail プロパティ 738
 dmaProp_This プロパティ 738
 dmaProp_VersionDescriptions プロパティ 739
 dmaProp_VersionedObjectClass プロパティ 739
 DMA URL 819
 DMA オブジェクト 819
 DMA オブジェクト生成用の構造体 743
 DMA 情報リスト構造体 745
 DMA 情報リストの取得 672
 DMA の戻り値 801
 DocSpace との接続 415
 DocumentSpace 構成定義ファイル 819
 DocVersion オブジェクト 820

E

edmClass_ComponentDocVersion クラスのプロパティ 727
 edmClass_ContainerVersion クラスのプロパティ 728
 edmClass_ContentSearch クラスのプロパティ 729

edmClass_IndependentPersistence クラスのプロパティ 730

edmClass_PublicACL クラスのプロパティ 731

edmClass_Relationship クラス 820

edmClass_Relationship クラスのプロパティ 732

edmClass_VersionTraceableContainmentRelationship クラス 732

edmClass_VersionTracedComponentDocVersion クラスのプロパティ 733

edmClass_VersionTracedDocVersion クラスのプロパティ 734

edmProp_ConceptStIndex プロパティ 739

edmProp_ConceptTextIndex プロパティ 739

edmProp_ContentIndexStatus プロパティ 739

edmProp_Content プロパティ 739

edmProp_DocLength プロパティ 740

edmProp_DocVersionOIID プロパティ 740

edmProp_OwnerId プロパティ 740

edmProp_RawScore プロパティ 740

edmProp_RelationType プロパティ 740

edmProp_RenditionsCount プロパティ 740

edmProp_Score プロパティ 740

edmProp_StIndex プロパティ 740

edmProp_TextIndex プロパティ 741

edmProp_VTConfigurationHistory プロパティ 741

edmProp_VTContainees プロパティ 741

edmProp_VTContainers プロパティ 741

edmProp_VTMode プロパティ 741

edmProp_VTVersionSeries プロパティ 741

edmSQL 820

edmSQL 検索 820

edmSQL 文の?パラメタ (DMA オブジェクト) の設定 310

edmSQL 文の?パラメタ (OIID) の設定 312

edmSQL 文の?パラメタの設定 314

edmSQL 文の設定 308

edmSQL 文を操作, 実行するクラス 287

ERR_DBR の場合 806

ERR_DBR の場合のクラスライブラリ独自の戻り値一覧 806

ERR_DB の場合 812

ERR_DB の場合のクラスライブラリ独自の戻り値一覧 812

ERR_DMA の場合の DMA の戻り値一覧 801

Execute メソッド 299

F

FAM 820

File Link パス情報構造体 746

File Link 文書 820

File Link 連携機能 820

G

GetACLMode メソッド 302

GetBindObjectListAndLock メソッド 356

GetBindObjectList メソッド 351

GetClassType メソッド 419

GetContainableListAndLock メソッド 52, 379

GetContainableList メソッド 45, 373

GetContainerListAndLock メソッド 130

GetContainerList メソッド 124

GetContentAndLock メソッド 224, 548

GetContentInfo メソッド 231, 555

GetContent メソッド 221, 545

GetCount メソッド 25, 441

GetDataType メソッド 26, 442

GetDBError メソッド 149

GetDmaInfoList メソッド 672

GetDocSpaceCharacterSet メソッド 421

GetId メソッド 27

GetIndexData メソッド 676

GetLastDetailError メソッド 422

GetLastError メソッド 150

GetLinkPropertyValuesAndLock メソッド 62, 389

GetLinkPropertyValues メソッド 59, 386

GetOIID メソッド 158

GetPropertyValuesAndLock メソッド 162, 462

GetPropertyValues メソッド 159, 459

GetPublicACLListAndLock メソッド 140, 335, 469

GetPublicACLList メソッド 137, 332, 466

GetReferencePath メソッド 423

GetRelationListAndLock メソッド 239, 563

GetRelationList メソッド 234, 558

GetRenditionListAndLock メソッド 248, 573

GetRenditionList メソッド 245, 569

GetReservationStatusAndLock メソッド 474

GetReservationStatus メソッド 472

GetResult メソッド 304

GetType メソッド 166

GetUserInfo メソッド 424

GetValue メソッド 28, 443

GetVersionableListAndLock メソッド 257, 621

GetVersionableList メソッド 252, 616

GetVersionListAndLock メソッド 481

GetVersionList メソッド 477

GetVTContaineelistAndLock メソッド 72, 633

GetVTContaineelist メソッド 65, 626

GUID 820

I

ID ファイル 820
Initialize メソッド 307, 684

L

LDAP 820
LinkAndLock メソッド 82, 395
LinkVTFixAndLock メソッド 89, 643
LinkVTFix メソッド 86, 640
LinkVTFloatAndLock メソッド 96, 650
LinkVTFloat メソッド 93, 647
Link メソッド 79, 392

N

NOT 条件 820

O

OIID 820
OIID (DMA URL) の BNF 記法 793
OIID 指定時の URL の文法 793
OIID の形式 793
OIID の取得 158
OIID の設定 177
OIID の設定やプロパティを扱うクラス 151
OrbComponentShutdown メソッド 427
OrbComponentStart メソッド 429
OR 検索 820

P

Parent 821
PutLinkPropertyValues メソッド 99, 399
PutPropertyValues メソッド 168, 486
PutRelationPropertyValues メソッド 262, 577
PutRenditionPropertyValues メソッド 265, 580

R

RD エリア 821
ReleaseObject メソッド 172
RemoveIndex メソッド 268, 584
RemoveObject メソッド 174
RemoveRelation メソッド 270, 586
Rendition オブジェクト 821
Reservation オブジェクト 821
Resize メソッド 444
Rollback メソッド 431

S

Scalar 型 821
SDBR_CONTENTINFO 構造体 742
SDBR_CONTENTLIST 構造体 742
SDBR_DMAINFOLIST 構造体 745
SDBR_DMAININFO 構造体 743
SDBR_FILELINK_PATHINFO 構造体 746
SDBR_LINKEDOBJINFO 構造体 747
SDBR_LINKEDOBJLIST 構造体 748
SDBR_OBJINFO 構造体 748
SDBR_OBJLIST 構造体 749
SDBR_PATHINFO 構造体 750
SDBR_PATHLIST 構造体 751
SDBR_PROPDEF 構造体 755
SDBR_PROPERTIES 構造体 768
SDBR_PROPLIST 構造体 756
SDBR_PROP 構造体 751
SDBR_QUERYRESULT 構造体 767
SDBR_REFERENCE_PATHINFO 構造体 758
SDBR_RELATIONINFO 構造体 760
SDBR_RELATIONLIST 構造体 761
SDBR_RENDITION_COMMENTINFO 構造体 761
SDBR_RENDITIONINFO 構造体 762
SDBR_RENDITIONLIST 構造体 763
SDBR_VROBJINFO 構造体 763
SDBR_VROBJLIST 構造体 764
SDBR_VTLINKEDOBJINFO 構造体 764
SDBR_VTLINKEDOBJLIST 構造体 765
SDBR_VTOBJINFO 構造体 766
SDBR_VTOBJLIST 構造体 767
SetObjParam メソッド 310
SetOIIDParam メソッド 312
SetOIID メソッド 177
SetParam メソッド 314
SetReferencePath メソッド 432
SetValue メソッド 29, 445
SetVTFixAndLock メソッド 655
SetVTFix メソッド 101, 653
SetVTFloatAndLock メソッド 660
SetVTFloat メソッド 103, 658
Set メソッド 308
SystemManager オブジェクト 821
System オブジェクト 821

T

Terminate メソッド 317
TPBroker の ORB コンポーネントの開始 429
TPBroker の ORB コンポーネントの終了 427

U

UnbindPublicACL メソッド 143, 338, 490
 UnlinkAndLock メソッド 107, 404
 UnlinkVTAndLock メソッド 112, 665
 UnlinkVT メソッド 110, 663
 Unlink メソッド 105, 402
 UOC 821
 UpdateContentAndRenditionType メソッド 278, 594
 UpdateContent メソッド 272, 588

V

VariableArray 型 821
 Versionable オブジェクト 821
 VersionCheckIn メソッド 493
 VersionCheckOut メソッド 495
 VersionDescription オブジェクト 821
 VersionRevoke メソッド 498
 VersionSeries オブジェクト 821
 VersionTracedComponentDocVersion オブジェクト 821
 VersionTracedDocVersion オブジェクト 821
 VTContainee 822
 VTContainer 822

W

W3C 822

X

XML インデクスデータ作成機能 822
 XML の操作環境を管理するクラス 679
 XML プロパティマッピング機能 822
 XML 変換を扱うクラス 667

あ

アクセス権 822
 アクセス制御エレメント 822
 アクセス制御機能 822
 アクセス制御機能付き検索 822
 アクセス制御情報 822
 アクセス制御情報変更権 822
 アクセス制御フラグ 822
 アクセス制御モードの取得 302
 アクセス制御モードの変更 292
 アクセス制御モデル 822
 アクセス制御リスト 822
 値の取得 28, 443

値の設定 29, 445
 アドレス呼び出し 823

い

異表記展開検索 823
 インデクス情報ファイル 823

え

永続オブジェクト 823
 永続オブジェクトの確定 414
 永続プロパティ 823
 エラー情報を取得するクラス 145
 エラー番号の取得 150

お

オブジェクト作成権 823
 オブジェクト作成権限 823
 オブジェクト情報構造体 748
 オブジェクト操作権限 823
 オブジェクト操作ツール 823
 オブジェクトとの接続の解除 172
 オブジェクトの削除 174
 オブジェクトの種別の取得 166
 オブジェクトの初期化 684
 オブジェクトの初期化とセッションオブジェクトへの登録 307
 オブジェクトの新規作成 41, 206, 328, 368, 530, 612
 オブジェクトリスト構造体 749
 オブジェクトリファレンス 823
 オブジェクトを指定したモードでロックして接続 156

か

概念検索 823
 可変長配列を扱うクラス 435
 仮のバージョン識別子 823
 関数 688
 関連オブジェクト 823
 関連づけの解除 402
 関連づけの解除と同時に指定したロックの設定 404
 関連づけの設定 392
 関連づけの設定と同時に指定したロックの設定 395

き

基本コンテンツ更新権 824
 基本コンテンツ参照権 824

基本削除権 824
 基本単位 824
 基本バージョン管理権 824
 基本パーミッション 824
 基本プロパティ更新権 824
 基本プロパティ参照権 824
 基本リンク権 824
 近傍条件検索 824

 く

組み合わせパーミッション 825
 クラス定義情報ファイル 825
 クラスライブラリで提供するクラス一覧 2
 クラスライブラリ独自の戻り値 806
 クラスライブラリの戻り値 801

 け

継承 825
 検索結果構造体 767, 825
 検索結果の削除 297
 検索結果の取得 304
 検索結果の取得件数の変更 294

 こ

構成管理オブジェクト情報構造体 766
 構成管理オブジェクトリスト構造体 767
 構成管理型のコンテナメント 825
 構成管理コンテナ 825
 構成管理しているオブジェクトとの関連づけの解除
 110, 663
 構成管理しているオブジェクトとの関連づけの解除と
 同時に指定したロックの設定 112, 665
 構成管理しているオブジェクトの一覧取得 65, 626
 構成管理しているオブジェクトの一覧取得と同時に指
 定したロックの設定 72, 633
 構成管理するオブジェクトの FIX モードでの関連づ
 け 86, 640
 構成管理するオブジェクトの FIX モードでの関連づ
 けと同時に指定したロックの設定 89, 643
 構成管理するオブジェクトの FLOATING モードでの
 関連づけ 93, 647
 構成管理するオブジェクトの FLOATING モードでの
 関連づけと同時に指定したロックの設定 650
 構成管理するオブジェクトの FLOATING モードでの
 関連づけと同時に指定したロックの設定 96
 構成管理リンク付きオブジェクト情報構造体 764
 構成管理リンク付きオブジェクトリスト構造体 765
 構成要素のバージョン固定の解除 103, 658

構成要素のバージョン固定の解除と同時に指定した
 ロックの設定 660
 構成要素のバージョンの固定 101, 653
 構成要素のバージョンの固定と同時に指定したロック
 の設定 655
 構造指定検索 825
 コストラクタ 24, 40, 123, 148, 155, 195,
 291, 327, 346, 367, 413, 439, 455, 517,
 611, 671, 682
 コンテナメント 825
 コンテナ 825
 コンテナの一覧取得 124
 コンテナの一覧取得と同時に指定したロックの設定
 130
 コンテナメント 825
 コンテナメント格納先パス 825
 コンテナメント格納先ベースパス 826
 コンテナメント格納先ベースパスの取得 423
 コンテナメント格納先ベースパスの設定 432
 コンテナメント種別 826
 コンテナメント情報構造体 742
 コンテナメント情報リスト構造体 742
 コンテナメントロケーション 826

 さ

最後に CdbSession::Commit メソッドがコールされ
 た状態まで戻す 431
 削除権 826
 サブクラス 826
 サブジェクト 826
 サブジェクト種別 826
 サブレンディション 826
 参照型のコンテナメント 826
 参照権 826
 参照更新権 826

 し

システム管理者 826
 実行環境制御ファイル 826
 指定した OIID のクラスの判定 419
 詳細エラー情報の取得 422
 詳細メッセージの出力について 812
 状態フラグ 826
 所有者 827
 シングルファイル文書 827

 す

スーパークラス 827

せ

セキュリティ ACL 827
 セキュリティ運用者 827
 セキュリティ管理者 827
 セキュリティ定義ファイル 827
 セッションオブジェクトへの登録抹消 317
 セッションの切断 418
 セッションを確立するクラス 407
 設定済み edmSQL 文の実行 299
 全文検索 827
 全文検索インデクス 827
 全文検索インデクス作成用データの取得 676
 全文検索インデクスの削除 268, 584
 全文検索インデクスの作成 203, 527
 全文検索機能付き文書クラス 827

ち

チェックアウト 827
 チェックアウト状態の取得 472
 チェックアウト状態の取得と同時に指定したロックの
 設定 474
 チェックアウトの取り消し 498
 チェックイン 827
 直接型のコンテインメント 828

て

定義情報ファイル 828
 データベースエラーが発生した場合の戻り値 812
 データベースのエラーの取得 149
 デストラクタ 23, 39, 120, 147, 154, 187,
 290, 324, 345, 366, 411, 437, 452, 509,
 610, 670, 681

と

同義語展開検索 828
 動作環境メタ情報ファイル 828
 独立データ 828
 独立データを扱うクラス 319
 特権 828
 トップオブジェクト 828
 トランザクションの開始 412

な

ナビゲーション 828

は

バージョン管理機能を提供するクラス 447

バージョン管理権 828
 バージョン管理情報一覧の取得 252, 616
 バージョン管理情報一覧の取得と同時に指定したロッ
 クの設定 257, 621
 バージョン識別子付きオブジェクト情報構造体 763
 バージョン識別子付きオブジェクトリスト構造体
 764
 バージョン情報一覧の取得 477
 バージョン情報一覧の取得と同時に指定したロックの
 設定 481
 バージョン付き構成管理コンテナ 828
 バージョン付き構成管理コンテナを扱うクラス 31
 バージョン付き文書を扱うクラス 501
 バージョンなし構成管理コンテナ 828
 バージョンなし構成管理コンテナを扱うクラス 603
 バージョンなしコンテナを扱うクラス 361
 バージョンなし文書を扱うクラス 179
 バージョンの削除 456
 バージョンのチェックアウト 495
 バージョンのチェックイン 493
 パーミッション 828
 バインドしているオブジェクトの一覧情報の取得
 351
 バインドしているオブジェクトの一覧情報の取得と同
 時に指定したロックの設定 356
 バインドしているパブリック ACL 一覧情報の取得
 137, 332
 バインドしているパブリック ACL 一覧情報の取得
 (バージョン指定あり) 466
 バインドしているパブリック ACL 一覧情報の取得と
 同時に指定したロックの設定 140, 335
 バインドしているパブリック ACL 一覧情報の取得と
 同時に指定したロックの設定 (バージョン指定あ
 り) 469
 パブリック ACL 829
 パブリック ACL のアンバインド 143, 338
 パブリック ACL のアンバインド (バージョン指定あ
 り) 490
 パブリック ACL の新規作成 347
 パブリック ACL のバインド 121, 325
 パブリック ACL のバインド (バージョン指定あり)
 453
 パブリック ACL を扱うクラス 341

ふ

ファイルサーバ 829
 ファイルの更新 272, 588
 ファイルの更新および RenditionType の変更 278,
 594

ファイルの取得 221, 545
 ファイルの取得と同時に指定したロックの設定 224, 548
 ファイルパス情報構造体 750
 ファイルパス情報リスト構造体 751
 フィルタリング定義ファイル (TFD) 829
 複合データを扱うクラス 21
 プライマリグループ 829
 フルコントロール 829
 プロパティ更新権 829
 プロパティ構造体 751
 プロパティ参照権 829
 プロパティ定義情報構造体 755
 プロパティの取得 159
 プロパティの取得 (バージョン指定あり) 459
 プロパティの取得と同時に指定したロックの設定 162
 プロパティの取得と同時に指定したロックの設定 (バージョン指定あり) 462
 プロパティの設定 168
 プロパティの設定 (バージョン指定あり) 486
 プロパティマッピング定義ファイル (DPM) 829
 プロパティリスト構造体 756
 文書 829
 文書間リレーションの解除 270, 586
 文書間リレーションの設定 216, 540
 文書間リレーションのプロパティの設定 262, 577
 文書空間 829
 文書のリレーション情報の一覧取得 234, 558
 文書のリレーション情報の一覧取得と同時に指定したロックの設定 239, 563

へ

変換フラグ 829

ほ

包含している Containable オブジェクトの一覧取得 45
 包含している Containable オブジェクトの一覧取得と同時に指定したロックの設定 52
 包含要素の一覧取得 373
 包含要素の一覧取得と同時に指定したロックの設定 379
 包含要素を扱うクラス 115

ま

マスタレンディション 830
 マスタレンディションへの変更 196, 518

マッピングセット定義ファイル (XMS) 830
 マッピング定義ファイル (XMP) 830
 マッピング元 XML タグ定義ファイル (DCD) 830
 マルチファイル管理機能 830
 マルチファイル文書 830
 マルチファイル文書の情報の取得 231, 555
 マルチレンディション機能 830
 マルチレンディション文書 830

め

メソッド実行時の排他制御 (ロック) 783
 メソッド実行時の明示的なロックと暗黙のロック 783
 メソッド実行時のロック一覧 783
 メソッドの実行に必要なパーミッション 770
 メソッドの実行に必要なパーミッション一覧 770
 メタ情報ファイル 830
 メタデータ 830
 メタデータ空間 830
 メモリの解放 689

も

文字列から DmaId 型への変換 691
 戻り値一覧 801
 戻り値の種類と major_code, minor_code の関係 801

ゆ

ユーザ権限 830
 ユーザ権限定義ファイル 830
 ユーザ情報 831
 ユーザ情報の取得 424

よ

要素 ID の取得 27
 要素数の取得 25, 441
 要素数の変更 444
 要素の型の取得 26, 442
 要素の削除 440
 要素の追加 438

ら

ランキング検索 831

り

リファレンスパス情報構造体 758
 リファレンスファイル管理機能 831

リファレンスファイル文書 831
リレーション 831
リレーション先文書 831
リレーション識別子 831
リレーション情報構造体 760
リレーション元文書 831
リレーションリスト構造体 761
リンク 831
リンク権 831
リンク識別子 831
リンク付きオブジェクト情報構造体 747
リンク付きオブジェクトリスト構造体 748
リンクのプロパティの取得 59, 386
リンクのプロパティの取得と同時に指定したロックの
設定 62, 389
リンクのプロパティの設定 99, 399

れ

レンディション 831
レンディション一覧の取得 245, 569
レンディション一覧の取得と同時に指定したロックの
設定 248, 573
レンディションコメント情報構造体 761
レンディション情報構造体 762
レンディションタイプ 831
レンディションのコンテンツ種別変換機能 832
レンディションの削除 219, 543
レンディションの追加 188, 510
レンディションのプロパティの設定 265, 580
レンディション変換 832
レンディションリスト構造体 763

ろ

ローカル ACL 832
ロケール 832
ロック種別の変更 295