

uCosminexus DocumentBroker Version 5 概説

解説書

3021-3-402

対象製品

R-1595F-13 uCosminexus DocumentBroker Platform Version 5 05-00 (適用 OS : Windows Server 2008 R2)
R-1L95F-13 uCosminexus DocumentBroker Platform Version 5 05-00 (適用 OS : Red Hat Enterprise Linux 6 (x86_64))
R-1595F-43 uCosminexus DocumentBroker Developer Version 5 05-00 (適用 OS : Windows Server 2008 R2 , Windows 7 (x86) , Windows 7 (x64))
R-1595F-53 uCosminexus DocumentBroker Runtime Library Version 5 05-00 (適用 OS : Windows Server 2008 R2 , Windows 7 (x86) , Windows 7 (x64))
R-1L95F-53 uCosminexus DocumentBroker Runtime Library Version 5 05-00 (適用 OS : Red Hat Enterprise Linux 6 (x86_64))

印の適用 OS については、サポート時期をご確認ください。

輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

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

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

Internet Explorer は、米国 Microsoft Corp. の米国及びその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Internet Information Services は、米国 Microsoft Corp. の商品名称です。

Microsoft Office Word は、米国 Microsoft Corp. の商品名称です。

Microsoft Word は、米国 Microsoft Corp. の商品名称です。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

Microsoft および 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。

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

製品名称	略称	
Microsoft(R) Active Directory(R)	Active Directory	
Microsoft(R) Internet Explorer(R)	Internet Explorer	
Microsoft(R) Internet Information Services	Internet Information Services	
Microsoft(R) Office Word	Word	
Microsoft(R) Word		
Microsoft(R) Windows(R) 7 Professional 日本語版 (32 ビット版)	Windows 7 (x86)	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)	Windows 7 (x64)	

製品名称	略称
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)	
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	

発行

2012 年 9 月 (第 1 版) 3021-3-402

著作権

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

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

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機能概要について説明したものです。

- R-1595F-13 uCosminexus DocumentBroker Platform Version 5
- R-1L95F-13 uCosminexus DocumentBroker Platform Version 5
- R-1595F-43 uCosminexus DocumentBroker Developer Version 5
- R-1595F-53 uCosminexus DocumentBroker Runtime Library Version 5
- R-1L95F-53 uCosminexus DocumentBroker Runtime Library Version 5

対象読者

このマニュアルは、DocumentBroker の機能を理解し、システムの構成を検討、または運用・管理する方を対象としています。

- UNIX または Windows に関する知識
- Java™ 言語に関する知識
- SQL 言語に関する知識

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」の意味を表します。 (例) A B A または B を指定することを示します。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号 で区切られている場合は、そのうちの一つを選択します。 (例) { A B C } A, B または C のどれかを指定することを示します。
[]	この記号で囲まれている項目は省略してもよいことを意味します。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。
—	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場合にシステムが取る標準値を示します。 (例) [<u>A</u> B] 何も指定しない場合は A が仮定されます。
< >	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。
...	この記号の直前の項目を繰り返し、複数個指定できることを示します。 (例) A ... A を複数個指定できることを示します。

このマニュアルで使用する構文要素記号

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。

構文要素記号	意味
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
16進数字	0 ~ 9 A ~ F a ~ f
記号	! " # \$ % & ' () + , _ . / : ; < = > @ [] ^ - { } 空白 ¥

注 すべての半角文字を使用してください。

目次

1	DocumentBroker の概要	1
1.1	DocumentBroker とは	2
1.1.1	DocumentBroker の特長	2
1.1.2	DocumentBroker クラスライブラリの特長	2
1.2	DocumentBroker で実現する機能	4
1.2.1	文書空間（データベース）に接続する機能	4
1.2.2	文書を作成，管理する機能	4
1.2.3	文書やフォルダを検索する機能	6
1.2.4	独立データを管理する機能	6
1.2.5	そのほかの機能	6
1.3	DocumentBroker を使用する場合のシステム構成	8
1.3.1	DocumentBroker の前提プログラム	8
1.3.2	ユーザアプリケーションプログラムの開発および実行に必要な Java の環境	10
1.3.3	連携できるプログラム	10
1.3.4	使用できる WWW ブラウザ	11
1.4	DocumentBroker を使用する場合の処理の流れ	12
2	文書管理で使用する概念	13
2.1	DocumentBroker の操作で使用する概念	14
2.2	文書管理に使用するオブジェクト	15
2.2.1	DocumentBroker のオブジェクト	15
2.2.2	文書管理オブジェクトと DocumentBroker オブジェクトの関係	15
2.2.3	オブジェクトとデータベースの関係	16
2.2.4	文書管理オブジェクトの種類	16
2.2.5	DocumentBroker のクラス	17
2.2.6	文書管理オブジェクトクラスと DocumentBroker クラスの対応	17
2.3	文書	19
2.3.1	バージョンなし文書	19
2.3.2	バージョン付き文書	19
2.4	バージョン管理	20
2.5	レンディション管理	21
2.6	リファレンスファイル管理	22
2.7	リンク	23
2.7.1	文書間リンク	23
2.8	フォルダ	25
2.8.1	バージョンなしフォルダ	25
2.9	独立データ	26
2.10	アクセス制御	27

2.11 排他制御	29
-----------	----

3

文書管理モデル	31
3.1 バージョン管理モデル	32
3.1.1 バージョン管理の概要	32
3.1.2 バージョン付き文書を構成する DocumentBroker オブジェクト	34
3.1.3 バージョン付き文書による文書管理	36
3.2 レンディション管理モデル	37
3.2.1 レンディション管理の概要	37
3.2.2 レンディションを管理する文書を構成する DocumentBroker オブジェクト	38
3.2.3 レンディションによる文書管理	39
3.2.4 レンディションのプロパティ	42
3.2.5 レンディション状態を表すプロパティ (dbrProp_RenditionStatus プロパティ)	44
3.3 リファレンスファイル文書の管理モデル	53
3.3.1 リファレンスファイル文書の管理の概要	53
3.3.2 リファレンスファイル文書を構成する DocumentBroker オブジェクト	54
3.4 リンクモデル	56
3.4.1 リンクによる文書管理の概要	56
3.4.2 リンクオブジェクトを構成する DocumentBroker オブジェクト	56
3.4.3 リンクによる文書管理	59
3.5 バージョンなしフォルダによる管理モデル	62
3.5.1 バージョンなしフォルダによる文書管理の概要	62
3.5.2 バージョンなしフォルダを構成する DocumentBroker オブジェクト	62
3.5.3 バージョンなしフォルダによる文書管理	63
3.6 独立データ管理モデル	66
3.6.1 独立データを構成する DocumentBroker オブジェクト	66
3.7 属性管理モデル	67
3.7.1 プロパティの管理例	67
3.8 アクセス制御モデル	70
3.8.1 アクセス権の判定に使用される情報	70
3.8.2 アクセス制御情報に設定できるパーミッションの種類	72
3.8.3 文書管理オブジェクトごとのアクセス制御情報の種類	77
3.8.4 アクセス制御情報の管理	82
3.8.5 アクセス制御モデルで使用するプロパティ	84
3.8.6 オブジェクトの操作とアクセス制御情報の関係	93
3.8.7 アクセス制御モデルの運用例	98
3.9 排他制御モデル	104
3.9.1 ロック種別	104
3.9.2 ロックによる排他制御	105
3.9.3 ロックが設定されているオブジェクトに対する接続	105
3.9.4 ロック種別の変更	106

3.9.5	ロックに関する注意事項	106
-------	-------------	-----

4

検索機能	109
4.1 検索の概要	110
4.1.1 DocumentBroker での検索	110
4.1.2 検索条件と検索結果	110
4.2 検索の種類	114
4.2.1 属性検索	114
4.2.2 全文検索	114
4.2.3 全文検索機能付き文字列型プロパティを使用した全文検索	117
4.3 検索の実行方法の種類	119
4.3.1 ?パラメタを使用した検索	119
4.3.2 ロック指定検索	119
4.3.3 アクセス制御機能付き検索	119
4.3.4 名前付き検索結果を取得する検索	120
4.3.5 検索結果キャッシュと検索結果取得情報を指定した検索	120
4.4 検索対象になる文書管理オブジェクト	122
4.4.1 検索対象になるクラス	122
4.4.2 検索結果として取得できるプロパティ	122
4.4.3 検索条件に指定できるプロパティ	122
4.5 全文検索の対象になる文書の作成	123
4.5.1 全文検索対象文書の作成手順	123
4.5.2 全文検索インデクスの作成	123
4.5.3 全文検索インデクスの削除	124

付録

付録 A 文書管理オブジェクトのプロパティ一覧	125
付録 A.1 文書管理オブジェクトクラスのプロパティ一覧	126
付録 A.2 DocumentBroker のクラスのプロパティ一覧	129
付録 A.3 各プロパティの説明	136
付録 B このマニュアルの参考情報	141
付録 B.1 関連マニュアル	141
付録 B.2 このマニュアルでの表記	141
付録 B.3 DocumentBroker のマニュアルで使用する略語	142
付録 B.4 KB (キロバイト) などの単位表記について	143
付録 C 用語解説	144

索引

索引	161
----	-----

1

DocumentBroker の概要

この章では、DocumentBroker の概要、および DocumentBroker で実現できる機能について説明します。

1.1 DocumentBroker とは

1.2 DocumentBroker で実現する機能

1.3 DocumentBroker を使用する場合のシステム構成

1.4 DocumentBroker を使用する場合の処理の流れ

1.1 DocumentBroker とは

DocumentBroker は、膨らみ続ける企業内文書がもたらす情報の分散化や管理の煩雑化を解消するため、業務で扱った文書を統合的に保管・管理する文書管理基盤となる製品です。

きめ細かなアクセス制御や、文書の属性やキーワードを意識した検索機能も備えています。また、文書管理するユーザアプリケーションプログラムを開発する際に使用できる、Java 言語に対応したクラスライブラリを提供しています。

1.1.1 DocumentBroker の特長

DocumentBroker の特長について説明します。

(1) システムの運用性、信頼性を確保

(a) すべての文書をデータベースシステムで保管

すべての文書および文書に付随する情報を、データベースシステムである HiRDB に格納します。小規模な基幹文書管理システムはもちろん、大規模、大容量の基幹文書管理システムを運用する上での運用性と、高い信頼性を確保しています。増え続けていくデータ量や業務の規模の拡大に合わせて段階的にシステムを拡張できます。

(b) きめ細かなアクセス制御を実現

すべての文書に対して、担当業務単位、役割単位、ユーザ単位などに細分化したアクセス権を設定できます。これによって、文書に対する不正なアクセスによる情報の漏えいを防止できます。

(2) 必要な文書をすばやく確実に入手

文書中の単語をキーワードにした高速の全文検索、属性を利用した絞り込み検索、指定した文章と類似する概念を持つ文書の全文検索（概念検索）など、豊富な検索機能を提供しています。したがって、管理している膨大な量の文書の中から目的の文書をすばやく確実に入手できる検索システムを構築できます。

(3) オープンな環境でシステムを構築

アプリケーションの実行環境の構築は、Java アプリケーションサーバの標準仕様である J2EE に準拠しています。アプリケーション開発時には、アプリケーション内で J2EE フォルダ、J2EE サービス、J2EE リソースなどで提供されている API を呼び出すことで、煩雑なコーディングをしないで、複数の業務で共通の処理を実現できます。

1.1.2 DocumentBroker クラスライブラリの特長

DocumentBroker クラスライブラリの特長を次に示します。

ユーザアプリケーションプログラムをコンポーネント化して再利用できます。

Java のコンポーネント化技術である JavaBeans を使用してユーザアプリケーションプログラムを開発できます。ユーザアプリケーションプログラムのコンポーネント化による再利用性の向上と文書管理システムの構築期間の短縮化が図れます。

OS に依存しないユーザアプリケーションプログラムを開発できます。

Java 言語を使用して開発するユーザアプリケーションプログラムは、Java 仮想マシン上で動作します。このため、OS に依存しないユーザアプリケーションプログラムを開発できます。

サンプル Web アプリケーションを提供しています。

DocumentBroker を使用して開発したアプリケーションプログラムを、サンプルとして提供しています。このサンプルを、サンプル Web アプリケーションといいます。サンプル Web アプリケーションは、文書の検索、ダウンロード、登録、削除、フォルダを使用した管理、アクセス制御など、基本的な文書管理機能を備えたアプリケーションプログラムです。

サンプル Web アプリケーションは、Servlet、JSP、JavaBeans、そのほかの Java クラスなどのコンポーネントによって構成されています。サンプル Web アプリケーションのそれぞれのコンポーネントでは、DocumentBroker のクラス、インターフェースおよびメソッドの具体的な使用方法が示されています。

ユーザアプリケーションプログラムを開発する際に、サンプル Web アプリケーションで示されているアーキテクチャを参考に設計・構築すると、効率良くユーザアプリケーションプログラムが開発できます。

サンプル Web アプリケーションの詳細については、マニュアル「DocumentBrokerVersion 5 サンプル Web アプリケーション」を参照してください。

1.2 DocumentBroker で実現する機能

この節では、DocumentBroker によって実現する文書管理機能について説明します。

1.2.1 文書空間（データベース）に接続する機能

DocumentBroker では、文書および文書を管理する仕組みをオブジェクトの概念で表現し、オブジェクトを操作することによって、文書の参照、更新などの文書管理を実現しています。オブジェクトに対する操作を設計したユーザアプリケーションプログラムは、DocumentBroker の文書空間にアクセスして、データベースに格納されているオブジェクトを操作できます。

DocumentBroker では、ユーザアプリケーションプログラムと、DocumentBroker の文書空間を接続する機能を提供しています。文書空間に接続してオブジェクトを作成すると、その時点で、データベースの内容も更新されます。この更新を確認またはキャンセルしたり、文書空間との接続を確認したり、トランザクションを制御したりする機能も提供しています。

なお、DocumentBroker の文書空間では、Shift-JIS または UTF-8 のどちらかの文字コード種別を使用できます。

1.2.2 文書を作成、管理する機能

DocumentBroker で管理する文書には、次の種類があります。

- 一つのファイルから構成される文書
- 一つの文書の中に複数のバージョンに対応するファイルを持つ文書
- 一つの文書の中に複数の形式に対応するファイルを持つ文書
- ファイルの実体が任意のディレクトリに登録されている文書

文書を管理する方法には、次の種類があります。

- 文書と文書を関連付けて管理する
- 文書とフォルダを関連付けて管理する方法
- 文書やフォルダの属性を管理する方法
- 文書やフォルダを一括して管理する方法

(1) 一つのファイルから構成される文書を管理する

一つのファイルから構成される文書は、最も基本的な文書になります。

例えば、Word で作成したファイル「報告書.doc」を、DocumentBroker の文書のコンテンツ（Word やテキストエディタなどのアプリケーションプログラムで作成された文書ファイル）として管理できます。

(2) 一つの文書の中に複数のバージョンに対応するファイルを持つ文書を管理する

文書を更新するときに、履歴を残して管理できます。文書に対して、第1版、第2版などのバージョンを付けておくことで、過去のある時点での文書を取り出したり、その文書を基に新たな文書を作成したりできます。また、その文書が幾つのバージョンを持っているかということも管理できます。

この機能を、バージョン管理機能といいます。

(3) 一つの文書の中に複数の形式に対応するファイルを持つ文書を管理する

複数の形式に対応するファイルを一つの文書のコンテンツとして管理できます。

この機能は、一つのファイルの内容を、対応するアプリケーションプログラムごとの複数の形式に変換した場合などに使用できます。

例えば、文書「報告書」のコンテンツとして、Word で作成したファイル「報告書.doc」のほかに、「報告書.doc」を PDF 形式に変換したファイル「報告書.pdf」も一緒に登録できます。これによって、ユーザの目的やユーザが使用するマシンの環境などの利用形態に応じて、Word 形式のファイルをダウンロードしたり、PDF 形式のファイルをダウンロードしたりできます。

この機能を、マルチレンディション機能といいます。

(4) ファイルの実体が任意のディレクトリに登録されている文書を管理する

文書の実体であるコンテンツを、DocumentBroker が存在するマシンから接続できるファイルシステムの任意のディレクトリに格納し、文書のプロパティとコンテンツの格納先の情報（コンテンツロケーション）をデータベースに登録して管理します。コンテンツロケーションには、コンテンツの格納先の基点となるディレクトリパス（コンテンツ格納先ベースパス）からの相対パスが登録されます。コンテンツの格納先を移行する場合は、データベースに影響を与えることなく、コンテンツ格納先ベースパスを変更するだけで、コンテンツの格納領域を変更できます。また、コンテンツをデータベースに格納しないため、データベースの容量を削減できます。

この機能を、リファレンスファイル管理機能といいます。

(5) 文書と文書を関連付けて管理する

文書をほかの文書と関連付けて管理できます。ある文書の中で、ほかの文書を参照する記述がある場合などに、文書間にリンクを設定しておくことで、参照元の文書から参照先の文書をたどることができます。また、参照先の文書から参照元の文書をたどることもできます。

DocumentBroker では、文書と文書、文書とフォルダなどの関連付けを表すリンクをオブジェクト（リンクオブジェクト）として管理できます。

(6) 文書とフォルダを関連付けて管理する

文書は、フォルダに格納してまとめて管理したり、フォルダに格納した文書を、別の観点から分類して管理したりできます。

DocumentBroker では、複数の文書をまとめて管理するために、フォルダというオブジェクトを使用します。フォルダと文書にリンクを設定することによって、文書をフォルダにまとめて格納して管理したり、文書を複数の観点から分類して管理したりできます。設定するリンクの種類によって、フォルダと文書は $1:n$ （ n は 1 以上の整数）の関係で関連付けたり、 $m:n$ （ m, n は 1 以上の整数）の関係で関連付けたりできます。

例えば、文書をプロジェクト単位でまとめて管理したい場合、プロジェクトごとにフォルダを作成して、 $1:n$ の関係でフォルダと文書のリンクを設定すれば、プロジェクトごとに文書を管理できます。また、フォルダに格納された文書を作成者とテーマで分類して管理したい場合、作成者を表すフォルダとテーマを表すフォルダをそれぞれ作成して、 $m:n$ の関係でフォルダと文書のリンクを設定すれば、文書を作成者やテーマごとに分類して管理できます。

また、フォルダとフォルダを関連付けることによって、フォルダや分類に階層を持たせることもできます。

(7) 文書やフォルダの属性を管理する

DocumentBroker では、文書やフォルダにさまざまな属性を付けて管理できます。この属性をプロパティといいます。文書やフォルダにプロパティを定義すると、プロパティを指定して文書やフォルダを検索できるので便利です。

プロパティには、システムによって定義されるプロパティと、ユーザが任意に追加定義するプロパティがあります。例えば、「報告書.doc」を管理する場合、ユーザが追加したプロパティの値として、「作成者」、「担当部署」、「コメント」などの情報も一緒に管理できます。

バージョン管理できる文書では、バージョンに共通するプロパティと、個々のバージョンのプロパティを個別に管理できます。また、リンクオブジェクトにプロパティを設定して管理することもできます。

(8) 文書やフォルダを一括して管理する

DocumentBroker では、複数の文書やフォルダを一括して管理できます。このため、複数の文書のプロパティを一括して取得したり、複数の文書を一括して削除したりできます。

1.2.3 文書やフォルダを検索する機能

すでに登録されている文書やフォルダは、検索によって取得できます。

DocumentBroker では、文書やフォルダに設定されているプロパティの値を使用した属性検索や、文書に含まれる文字列を指定した全文検索ができます。

全文検索では、文章を指定して、その文章と類似した概念を持つ文書を検索（概念検索）できます。

全文検索を実行するには、前提プログラムとして HiRDB Text Search Plug-in、または HiRDB XML Extension が必要です。

検索を実行する場合の検索条件は、edmSQL によって指定できます。edmSQL は DocumentBroker が提供する機能であり、SQL に基づいた構文で記述できます。このため、SQL の知識を活用して検索を実行できます。

1.2.4 独立データを管理する機能

文書やフォルダの体系に関係なく、独立したデータを作成、管理できます。また、独立データには、プロパティを定義して管理できます。

1.2.5 そのほかの機能

そのほかの機能について説明します。

(1) アクセス制御機能

DocumentBroker で管理している文書やフォルダに対して参照や更新などのアクセスをする場合、そのユーザが持つアクセス権に応じてアクセスを制御できます。例えば、作成した文書をほかのユーザに参照はさせても更新はさせないようにしたい場合などに、この機能を使用します。これを、アクセス制御機能といいます。アクセスを制御するためのアクセス権は、ユーザやグループごとに設定できます。

(2) ユーザ認証機能

DocumentBroker では、次の 2 種類のシステムと連携したユーザ認証機能を提供します。

LDAP ディレクトリサーバ

ユーザが作成したアクセスルーチン

これによって、システムへのユーザの不当なログインを制御できます。

文書空間へのログイン要求があった場合、DocumentBroker は、ユーザから提示されたユーザ ID とパスワードに対して、ユーザ管理システムの機能を利用してユーザを認証します。

(3) 排他制御機能

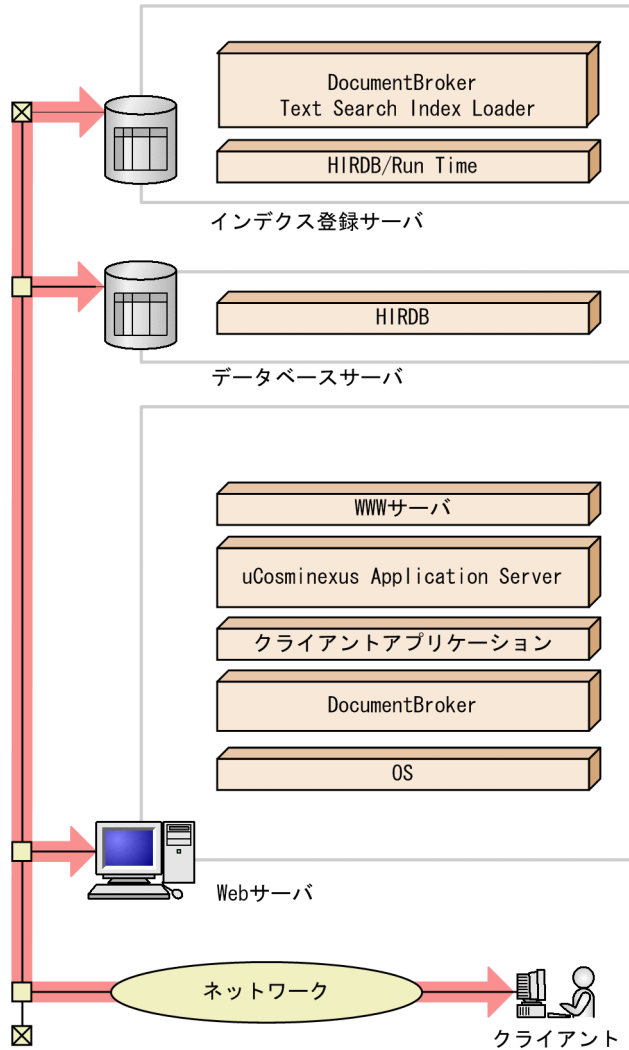
DocumentBroker で管理している文書やフォルダにアクセスする場合、排他制御が実行されます。排他制御とは、あるユーザが文書やフォルダにアクセスしているときに、ほかのユーザによってその文書やフォルダの状態が変更されたり、削除されたりしないようにロックを設定して制御する機能です。

DocumentBroker の排他制御には、メソッドによって暗黙的に設定されるロックと、ユーザが明示的に設定するロックがあります。

1.3 DocumentBroker を使用する場合のシステム構成

この節では、DocumentBroker を使用して開発する場合のシステム構成について説明します。システム構成を次の図に示します。

図 1-1 DocumentBroker を使用する場合のシステム構成



また、サンプル Web アプリケーションを使用する場合のシステム構成および前提プログラムについては、マニュアル「DocumentBroker Version 5 サンプル Web アプリケーション」を参照してください。

1.3.1 DocumentBroker の前提プログラム

DocumentBroker には、次に示すプログラムが必要です。なお、WWW サーバ、ユーザアプリケーションプログラムは同じマシン上に構築する必要があります。

(1) OS

DocumentBroker では、次に示すオペレーティングシステムを使用します。

DocumentBroker Platform の場合

- UNIX の場合 : Red Hat Enterprise Linux 6.1 (AMD/Intel 64)
- Windows の場合 : Windows Server 2008 R2

DocumentBroker Developer の場合

- Windows 7 (x86)
- Windows 7 (x64)
- Windows Server 2008 R2

DocumentBroker Runtime Library の場合

- UNIX の場合 : Red Hat Enterprise Linux 6.1 (AMD/Intel 64)
- Windows の場合 :
 - Windows 7 (x86)
 - Windows 7 (x64)
 - Windows Server 2008 R2

印の OS については、サポート時期をご確認ください。

(2) WWW サーバ

DocumentBroker では、次に示す WWW サーバを使用します。

DocumentBroker Platform の場合

- UNIX の場合 : iPlanet Web Server Enterprise Edition , または Hitachi Web Server
- Windows の場合 : Hitachi Web Server , または Internet Information Services

DocumentBroker Developer の場合

- Hitachi Web Server , または Internet Information Services

DocumentBroker Runtime Library の場合

- UNIX の場合 : iPlanet Web Server Enterprise Edition , または Hitachi Web Server
- Windows の場合 : Hitachi Web Server , または Internet Information Services

(3) ユーザアプリケーションプログラム

ユーザが、DocumentBroker を使用して開発するアプリケーションプログラムです。ユーザアプリケーションプログラムの開発および実行に必要な Java の環境については、「1.3.2 ユーザアプリケーションプログラムの開発および実行に必要な Java の環境」を参照してください。

(4) アプリケーションサーバ

DocumentBroker では、次に示すアプリケーションサーバを使用します。

- uCosminexus Application Server Standard 09-00 以降
- uCosminexus Application Server Enterprise 09-00 以降
- uCosminexus Developer Standard 09-00 以降
- uCosminexus Developer Professional 09-00 以降

(5) データベース

DocumentBroker では、次に示すデータベースを使用します。

- HiRDB Server Version 9 (09-00 以降)

(6) 全文検索機能を使用する場合に必要なプログラム

全文検索機能，概念検索機能を使用するためには，次に示すプログラムが必要です。

全文検索機能，または文字列型プロパティに対する全文検索機能の場合

- HiRDB Text Search Plug-in Version 9 (09-00 以降)
- HiRDB XML Extension Version 9 (09-00 以降)

概念検索機能の場合

- HiRDB Text Search Plug-in Conceptual Extension

(7) ディレクトリサーバ

DocumentBroker では，次に示すディレクトリサーバを使用します。

- Active Directory
- Oracle Directory Server Enterprise Edition (11g Release 1 以降)

(8) サンプル Web アプリケーションを使用する場合

サンプル Web アプリケーションを使用する場合は，次の Web ブラウザを使用できます。

- Microsoft Internet Explorer 8
- Microsoft Internet Explorer 9
- Mozilla Firefox

1.3.2 ユーザアプリケーションプログラムの開発および実行に必要な Java の環境

DocumentBroker でユーザアプリケーションプログラムを開発する場合およびユーザアプリケーションプログラムを実行する場合に必要な Java の環境は，Cosminexus を使用して構築してください。

Cosminexus は，Java 2 Platform, Enterprise Edition に準拠した Java の開発環境および実行環境を提供するアプリケーションサーバであり，DocumentBroker で必要な Java の環境を提供しています。このほかに，Enterprise JavaBeans などの各種 API も提供しています。このため，DocumentBroker の API を使用した Java コンポーネントを開発できます。また，Cosminexus が提供する JSP/Servlet エンジン上で DocumentBroker の API を使用した Java コンポーネントを実行できます。

1.3.3 連携できるプログラム

DocumentBroker を使用して連携できるプログラムを次に示します。必要に応じて使用してください。

(1) ユーザ認証やアクセス制御情報の取得に必要なプログラム

ディレクトリサービス

アクセス制御機能を使用する場合，LDAP 対応のディレクトリサービスを使用したユーザ管理システムと連携することをお勧めします。LDAP 対応のディレクトリサービスとして使用できる製品については，マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

ログイン時のユーザ認証に必要な情報，およびアクセス制御機能に必要な情報を取得する場合は，DocumentBroker を介してユーザ管理システムにアクセスします。DocumentBroker は，LDAP 対応のディレクトリサービスやユーザ作成のアクセスルーチンなどを使用したユーザ認証機能を提供しています。

DocumentBroker が LDAP 対応のディレクトリサービスを使用したユーザ管理システムと連携している場合、アクセス制御機能に必要な情報は、LDAP 対応のディレクトリサービスで管理されている情報を基に生成されます。

なお、ユーザアプリケーションプログラムを開発する際に、JNDI を使用して LDAP 対応のディレクトリサービスと連携すれば、LDAP 対応のディレクトリサービスで管理されている情報を直接取得できます。

アクセス制御については、「2.10 アクセス制御」または「3.8 アクセス制御モデル」を参照してください。

(2) レンディション変換機能のプログラム

マルチレンディション文書のコンテンツの登録・更新状態によって、DocumentBroker から出された要求に従い、レンディション変換を実行する機能を持つプログラムです。レンディション変換とは、登録済みのマスタレンディションのコンテンツを基に、ほかの形式のファイルを作成し、サブレンディションのコンテンツとして登録することです。マルチレンディション文書を管理する場合は、このプログラムと連携すると、マスタレンディションとサブレンディションの内容の同期が取りやすくなります。

マルチレンディションおよびレンディション変換機能については、「2.5 レンディション管理」および「3.2 レンディション管理モデル」を参照してください。

なお、レンディション変換機能については、サポート時期をご確認ください。

1.3.4 使用できる WWW ブラウザ

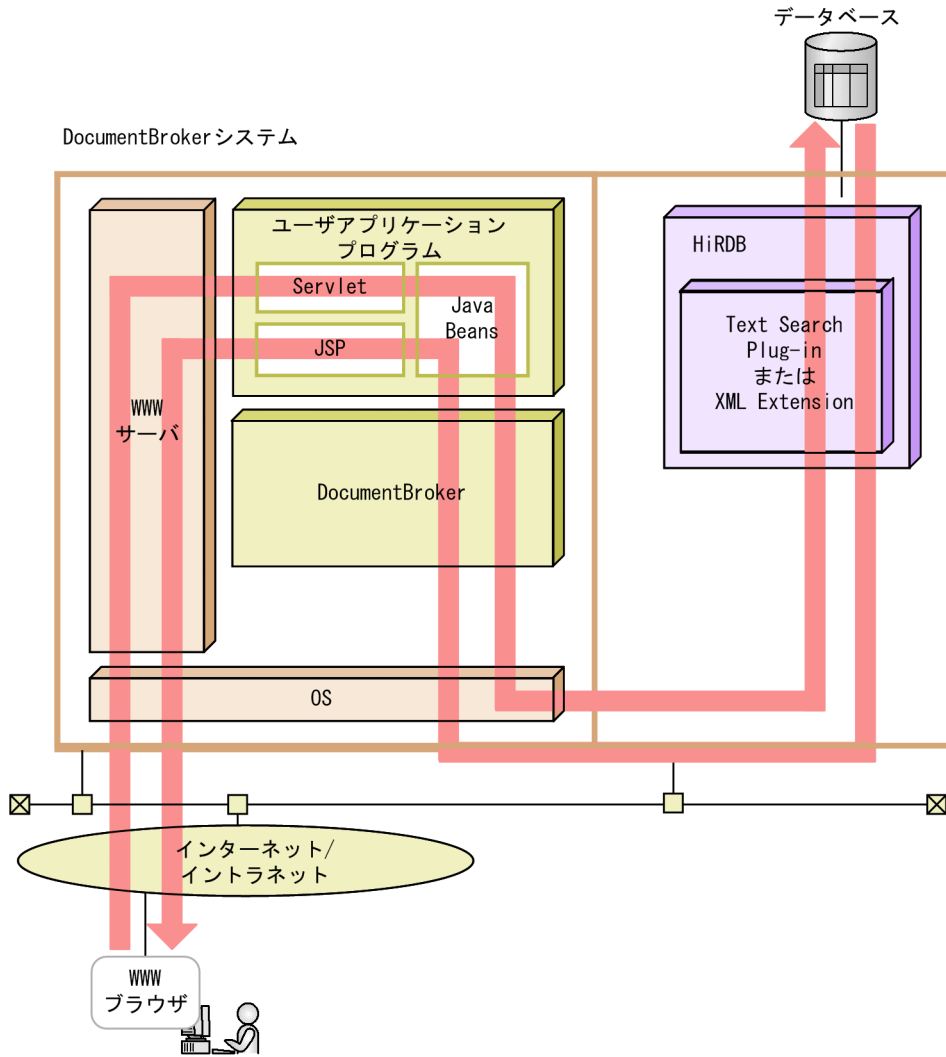
DocumentBroker を使用して開発したユーザアプリケーションプログラムを実行できる WWW ブラウザを次に示します。

Internet Explorer

1.4 DocumentBroker を使用する場合の処理の流れ

この節では、DocumentBroker を使用して開発する場合の処理の流れについて説明します。処理の流れの例を次の図に示します。

図 1-2 DocumentBroker を使用する場合の処理の流れの例



この例では、JavaBeans、Servlet、および JSP を使用してユーザアプリケーションプログラムを作成しています。このユーザアプリケーションプログラムでは、Servlet と JSP の処理を、JavaBeans を経由して実行しています。

例えば、検索を実行する画面が WWW ブラウザ上に表示されているとします。ユーザは、WWW ブラウザ上に表示された画面に検索条件を入力し、「検索実行」ボタンを押して、検索を実行します。ユーザからの処理要求は Servlet が受け付け、Servlet はその処理を JavaBeans に渡します。JavaBeans はデータベースにアクセスして検索処理を実行し、その処理結果を JSP に渡します。JSP は JavaBeans から受け取った処理結果を基に WWW ページを生成して、WWW ブラウザ上に検索結果の画面を表示します。

2

文書管理で使用する概念

この章では、文書管理で使用する概念と各概念の詳細について説明します。

2.1 DocumentBroker の操作で使用する概念

2.2 文書管理に使用するオブジェクト

2.3 文書

2.4 バージョン管理

2.5 レンディション管理

2.6 リファレンスファイル管理

2.7 リンク

2.8 フォルダ

2.9 独立データ

2.10 アクセス制御

2.11 排他制御

2.1 DocumentBroker の操作で使用する概念

この節では、DocumentBroker の操作で使用する概念について説明します。

DocumentBroker では、文書および文書を管理する仕組みをオブジェクトの概念で表現し、オブジェクトを操作することによって、文書の参照、更新などの文書管理を実現しています。オブジェクトに対する操作を設計したユーザアプリケーションプログラムは、DocumentBroker の文書空間にアクセスして、データベースに格納されているオブジェクトを操作できます。

ユーザアプリケーションプログラムと、DocumentBroker の文書空間を接続する機能を提供しています。文書空間に接続してオブジェクトを作成すると、その時点で、データベースの内容も更新されます。この更新を確定またはキャンセルしたり、文書空間との接続を確認したり、トランザクションを制御したりする機能も提供しています。

2.2 文書管理に使用するオブジェクト

DocumentBroker のオブジェクトの概要、オブジェクトの構成、オブジェクトとデータベースの関係について説明します。

2.2.1 DocumentBroker のオブジェクト

DocumentBroker では、次の 2 種類のオブジェクトを区別して理解しておく必要があります。

- 文書管理オブジェクト
複数の DocumentBroker オブジェクトを包含した形で構成されるオブジェクトです。文書管理オブジェクトを操作することによって、複数の DocumentBroker オブジェクトをまとめて操作します。
- DocumentBroker オブジェクト
文書管理オブジェクトの構成要素となります。DocumentBroker オブジェクトには、文書を表すオブジェクト、文書の表現形式を管理するオブジェクト、文書の実体であるコンテンツ（文書ファイル）に相当するオブジェクトなどがあります。
DocumentBroker では、DocumentBroker オブジェクトを組み合わせて使用することによって文書を管理しています。一つの DocumentBroker オブジェクトは、一つの DocumentBroker クラスを基に作成されます。

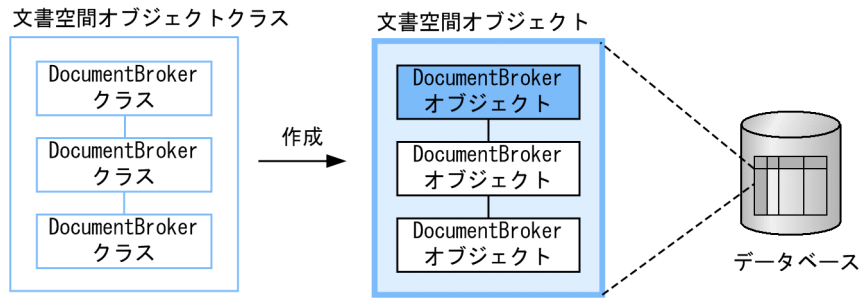
なお、このマニュアルでは、文書管理オブジェクトを構成する DocumentBroker オブジェクトのうち、代表的な DocumentBroker オブジェクトをトップオブジェクトといいます。また、トップオブジェクトのクラスをトップオブジェクトクラスといいます。

2.2.2 文書管理オブジェクトと DocumentBroker オブジェクトの関係

DocumentBroker では、文書管理に使用する複数の DocumentBroker オブジェクトを、文書やフォルダなどのまとまった一つの文書管理オブジェクトとして扱います。文書管理オブジェクトに対応するクラスが、文書管理オブジェクトクラスです。文書管理オブジェクトクラスは、文書管理を実現する機能ごとに、一つまたは複数の DocumentBroker クラスから構成されています。文書管理オブジェクトクラスを基に文書管理オブジェクトを作成することによって、その機能を実現するために必要な DocumentBroker オブジェクトが作成されます。また、文書管理に必要な操作をクラスごとまたはインターフェースごとにメソッドとして提供しています。DocumentBroker メソッドを実行することによって、その操作に必要なプロパティが DocumentBroker オブジェクトに設定されます。したがって、文書を表す文書管理オブジェクトを操作することで、文書を表すオブジェクト、文書の表現形式を管理するオブジェクト、文書の実体であるコンテンツに相当するオブジェクトなどの複数の DocumentBroker オブジェクトを包含した、一つの文書が操作できます。

文書管理オブジェクトと DocumentBroker オブジェクトの関係を次の図に示します。

図 2-1 文書管理オブジェクトと DocumentBroker オブジェクトの関係



2.2.3 オブジェクトとデータベースの関係

データベースは DocumentBroker のオブジェクトモデルと対応して構成されています。DocumentBroker クラスは、データベースの表に対応します。文書管理オブジェクトに対してメソッドを実行すると、複数の DocumentBroker オブジェクトへのメソッドとして配置され、該当するデータベースの表が操作されます。

オブジェクトを生成する DocumentBroker クラスは、通常、それぞれが HiRDB の一つの表と対応します。DocumentBroker オブジェクトは各表の行（レコード）に相当し、各 DocumentBroker クラスが持つプロパティは各表の列（カラム）に相当します。また、DocumentBroker オブジェクト同士の関連は、各オブジェクトが属性として保持する参照先の OIID によって表されます。

2.2.4 文書管理オブジェクトの種類

文書管理オブジェクトの種類を次に示します。

バージョンなし文書

一つのバージョンだけを持つ文書です。

バージョン付き文書

複数のバージョンを持つ文書です。

バージョンなしフォルダ

一つのバージョンだけを持ち、複数の文書をまとめたり、分類したりして構成管理するフォルダです。

独立データ

独立したデータを扱う文書管理オブジェクトです。

パブリック ACL

複数の文書管理オブジェクトで共有するアクセス制御情報を保持する文書管理オブジェクトです。

リンク

複数の文書管理オブジェクト間をリンク付ける文書管理オブジェクトです。なお、リンクオブジェクトは、文書またはフォルダに従属するオブジェクトであり、単独では作成できません。

なお、このマニュアルでは、それぞれの文書管理オブジェクトをまとめて、次のような総称で表記することがあります。

文書

「バージョンなし文書」および「バージョン付き文書」の総称として使用します。また、「文書オブジェクト」と表記することもあります。

フォルダ

「バージョンなしフォルダ」として使用します。また、「フォルダオブジェクト」と表記することもあります。

バージョンなしオブジェクト

「バージョンなし文書」および「バージョンなしフォルダ」の総称として使用します。

バージョン付きオブジェクト

「バージョン付き文書」として使用します。なお、バージョン付きオブジェクトの一連のバージョンを統括するオブジェクトをバージョンングオブジェクト、個々のバージョンを表すオブジェクトをバージョンオブジェクトといいます。

2.2.5 DocumentBroker のクラス

DocumentBroker では、次の 2 種類のクラスを区別して理解しておく必要があります。

- 文書管理オブジェクトクラス
一つまたは複数の DocumentBroker クラスから構成される概念的なクラスです。文書管理オブジェクトクラスは、DocumentBroker オブジェクトをまとめて操作するための機能を持ったクラスです。文書管理を実現する機能ごとに、必要な DocumentBroker オブジェクトを操作する機能を持ったクラスが定義されています。文書管理オブジェクトクラスには、バージョンなし文書クラス、バージョン付き文書クラス、バージョンなしフォルダクラスなどがあります。文書やフォルダを作成して操作するためのクラスは、文書やフォルダに必要な複数の DocumentBroker クラスを構成要素として持っています。
- DocumentBroker クラス
文書管理オブジェクトクラスの構成要素です。DocumentBroker オブジェクトに対応するクラスです。DocumentBroker クラスは、データベースの表に対応します。DocumentBroker クラスには、文書を表すオブジェクトのクラス、フォルダを表すオブジェクトのクラスなどがあります。

2.2.6 文書管理オブジェクトクラスと DocumentBroker クラスの対応

文書管理オブジェクトクラスと、その構成要素である DocumentBroker クラスのトップオブジェクトクラスの対応を次の表に示します。トップオブジェクトクラスとして、次の表に示す DocumentBroker クラスまたはそのサブクラスを指定してください。

表 2-1 文書管理オブジェクトクラスとトップオブジェクトクラスの対応

文書管理オブジェクトクラス (文書管理オブジェクト)	トップオブジェクトクラス (トップオブジェクト)
バージョンなし文書クラス (バージョンなし文書)	dmaClass_DocVersion クラス (DocVersion オブジェクト)
バージョン付き文書クラス (バージョン付き文書)	dmaClass_ConfigurationHistory クラス (ConfigurationHistory オブジェクト)
バージョン付き文書クラス (バージョン付き文書のバージョン ¹⁾)	edmClass_VersionTracedDocVersion クラス ² (VersionTracedDocVersion オブジェクト)
バージョンなしフォルダクラス (バージョンなしフォルダ)	edmClass_ContainerVersion クラス (ContainerVersion オブジェクト)
独立データクラス (独立データ)	edmClass_IndependentPersistence クラス (IndependentPersistence オブジェクト)
パブリック ACL クラス (パブリック ACL)	edmClass_PublicACL クラス ³ (PublicACL オブジェクト)

2. 文書管理で使用する概念

文書管理オブジェクトクラス (文書管理オブジェクト)	トップオブジェクトクラス (トップオブジェクト)
リンクオブジェクトクラス (リンクオブジェクト)	<ul style="list-style-type: none">• dmaClass_DirectContainmentRelationship クラス (DirectContainmentRelationship オブジェクト)• dmaClass_ReferentialContainmentRelationship クラス (ReferentialContainmentRelationship オブジェクト)• edmClass_Relationship クラス³ (Relationship オブジェクト)

注 1 バージョン付き文書の 1 バージョンとして管理するバージョンなし文書です。

注 2 トップオブジェクトクラスとして、dmaClass_DocVersion クラス、または edmClass_VersionTracedDocVersion クラス以外の dmaClass_DocVersion クラスのサブクラスを指定した場合、そのバージョン付き文書は構成管理の対象になりません。

注 3 edmClass_PublicACL クラスと edmClass_Relationship クラスは、サブクラスを作成できません。

なお、このマニュアルでは、それぞれの文書管理オブジェクトクラスをまとめて、次のような総称で表記することがあります。

バージョンなしオブジェクトクラス

「バージョンなし文書クラス」および「バージョンなしフォルダクラス」の総称として使用します。

バージョン付きオブジェクトクラス

「バージョン付き文書クラス」として使用します。

文書オブジェクトクラス

「バージョンなし文書クラス」および「バージョン付き文書クラス」の総称として使用します。

フォルダオブジェクトクラス

「バージョンなしフォルダクラス」として使用します。

2.3 文書

この節では、DocumentBroker での文書の概念について説明します。

DocumentBroker では、文書を表すオブジェクト、文書の表現形式を管理するオブジェクト、文書の実体であるコンテンツ（Word やテキストエディタなどのアプリケーションプログラムで作成された文書ファイル）に相当するオブジェクトなどの複数の DocumentBroker オブジェクトを、まとまった一つの文書として扱います。

DocumentBroker では、次の 2 種類の文書を表す文書管理オブジェクトを扱います。

- バージョンなし文書
- バージョン付き文書

2.3.1 バージョンなし文書

バージョンなし文書は、一つのバージョンだけを持つ文書です。版管理をして更新履歴などを残す必要がない（バージョンを管理する必要がない）文書を管理する場合に使用します。

2.3.2 バージョン付き文書

バージョン付き文書は、複数のバージョンを持つ文書です。バージョンを付けて管理することで、過去のある時点での文書を参照したり、その文書を流用して新たに文書を作成するのに使用したりできます。

DocumentBroker では、複数のバージョンを持つ文書を一つの文書（バージョン付き文書）として管理できます。

2.4 バージョン管理

この節では、DocumentBroker でのバージョン管理の概念について説明します。

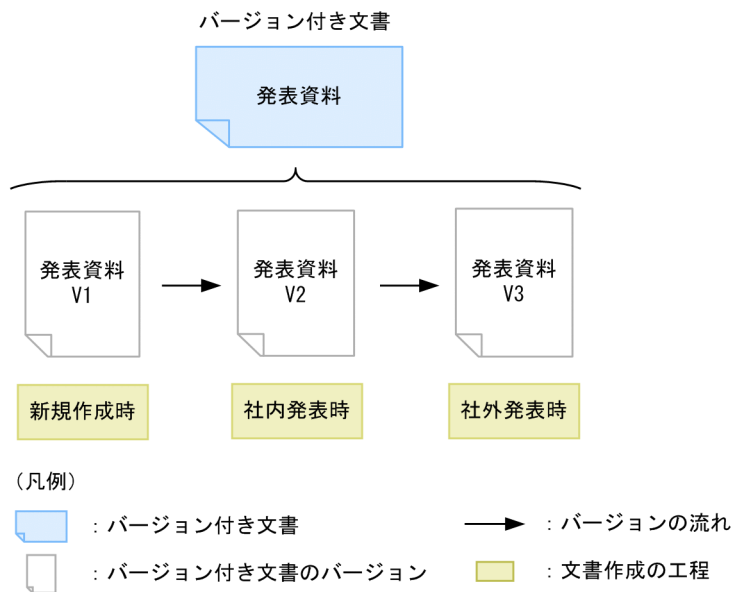
DocumentBroker では、次の文書管理オブジェクトでバージョンを管理できます。

- バージョン付き文書

バージョン付きオブジェクトでは、オブジェクトにバージョンを付けて管理します。新しく文書を作成した時点からバージョンを付け始め、更新のたびにバージョンを追加して、一連のバージョンを管理します。なお、各バージョンから枝分かれしたバージョンを管理することはできません。

バージョン付きオブジェクトの管理例を次の図に示します。

図 2-2 バージョン付きオブジェクトの管理例



この例では、バージョン付き文書「発表資料」を作成して、社内発表時および社外発表時にバージョンを追加して、管理しています。

バージョン付きオブジェクトを更新する場合は、古いバージョンを破棄して新しいバージョンと置き換えるのではなく、古いバージョンを残して新しいバージョンを追加（バージョンアップ）して登録します。バージョンを追加する場合は、チェックアウトとチェックインという操作が必要になります。

チェックアウトは、新しいバージョンの追加を予約するために、最新バージョンの次に仮のバージョンを追加する操作です。また、チェックインは、チェックアウトで追加した仮のバージョンを最新バージョンとして確定する操作です。

なお、バージョン付きオブジェクトの一連のバージョンを統括する文書管理オブジェクトをバージョンニングオブジェクト、個々のバージョンを表す文書管理オブジェクトをバージョンオブジェクトといいます。DocumentBroker では、バージョン付きオブジェクトを操作する場合は、バージョン全体を対象とするときはバージョンニングオブジェクト、個々のバージョンを対象とするときはバージョンオブジェクトを操作します。

2.5 レンディション管理

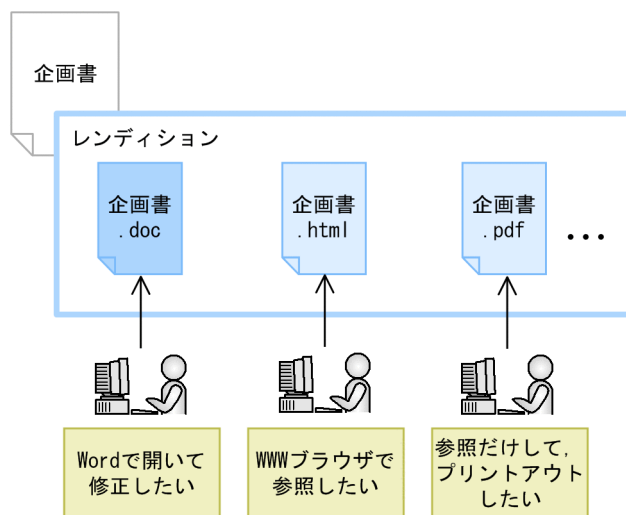
この節では、DocumentBroker でのレンディション管理の概念について説明します。

文書の実体であるコンテンツ（Word やテキストエディタなどのアプリケーションプログラムで作成された文書ファイル）と、その表現形式を表すレンディションタイプ（MIME 形式）などの情報をあわせてレンディションといいます。

DocumentBroker では、バージョンなし文書またはバージョン付き文書に 1 個、または複数のレンディションを登録して管理します。複数のレンディションを登録した文書を特にマルチレンディション文書といいます。マルチレンディションとは、一つの文書に対して同じ内容を表す複数の異なる形式のレンディションを登録する機能です。なお、レンディションのうち、主要なレンディションをマスタレンディション、マスタレンディション以外のレンディションをサブレンディションと区別します。

例えば、同じ内容を表す Word 形式のファイル、HTML 形式のファイルおよび PDF 形式のファイルをマルチレンディション文書のコンテンツとして登録して管理すると、ユーザの目的やユーザが使用するマシンの環境などの利用形態に応じて、Word 形式のファイルをダウンロードしたり、HTML 形式のファイルをダウンロードしたり、PDF 形式のファイルをダウンロードしたりできます。マルチレンディションによる文書の管理例を次の図に示します。

図 2-3 マルチレンディションによる文書の管理例



(凡例)

-  : バージョンなし文書
-  : マスタレンディション
-  : ユーザの作業目的
-  : サブレンディション

DocumentBroker では、バージョンなし文書またはバージョン付き文書をマルチレンディション文書として管理して、一つの文書に対して、1 個のマスタレンディションおよび最大 9 個のサブレンディションを登録できます。

2.6 リファレンスファイル管理

この節では、DocumentBroker でのリファレンスファイル管理の概念について説明します。

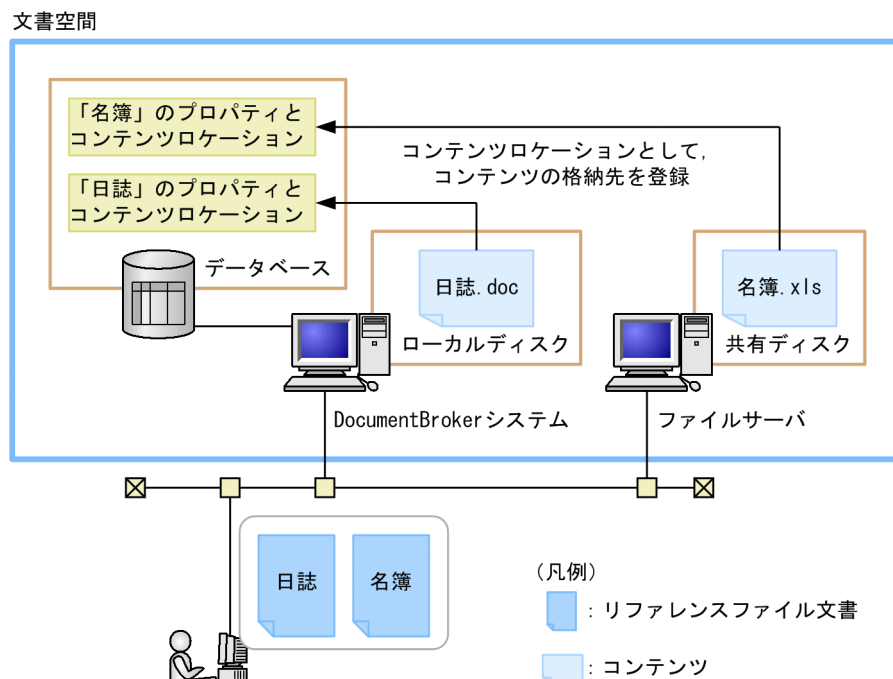
リファレンスファイル管理機能は、コンテンツを任意のディレクトリに格納し、文書のプロパティとコンテンツロケーションだけをデータベースで管理する機能です。データベースとコンテンツを切り離すことによって、ユーザによるコンテンツ格納先の選択やディスク移行時のコンテンツ格納ディレクトリの変更など、柔軟にコンテンツ格納先の管理ができます。また、UNC 形式のパスでコンテンツを操作することもできます。

リファレンスファイル管理機能を使用して管理している文書を、リファレンスファイル文書といいます。

DocumentBroker では、バージョンなし文書およびバージョン付き文書をリファレンスファイル文書として管理できます。

リファレンスファイル文書の管理例を次の図に示します。

図 2-4 リファレンスファイル文書の管理例



2.7 リンク

この節では、DocumentBroker でのリンクの概念について説明します。

文書またはフォルダは、DocumentBroker に登録されているほかの複数の文書またはフォルダと関連づけて管理できます。DocumentBroker では、このオブジェクト間のリンクをオブジェクト（リンクオブジェクト）として扱います。

DocumentBroker では、このリンクオブジェクトを使用して、複数の文書管理オブジェクト間をリンク付けたり、文書管理オブジェクトを検索したりできます。

リンクには次の 3 種類があります。

- 直接型リンク
- 参照型リンク
- 文書間リンク

この節では、文書間リンクについて説明します。直接型リンクおよび参照型リンクについては、「2.8 フォルダ」を参照してください。

2.7.1 文書間リンク

文書間リンクとは、文書と文書を関連づけて管理する機能です。

文書間リンクは、次のような場合に使用できます。

参考文献のある論文などの文書を、参考文献とともに管理したい場合

別文書として登録しているテキストと図データを関連がわかるように管理したい場合

DocumentBroker では、二つの文書を文書間リンクという概念で関連づけます。このとき、文書間リンクを設定するリンク元の文書をリンク元文書、文書間リンクが設定されるリンク先の文書をリンク先文書といます。

DocumentBroker では、次のどちらかの DocumentBroker オブジェクトの間に文書間リンクを設定できます。

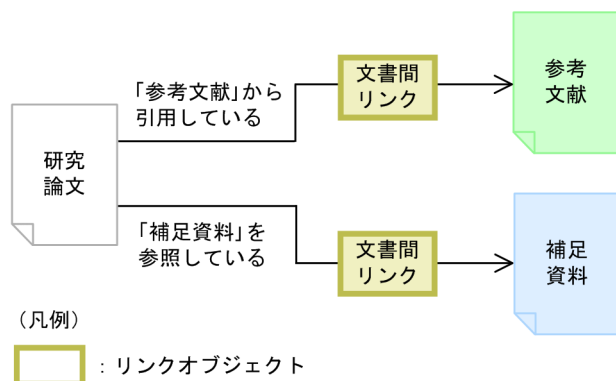
- バージョンなし文書
- バージョン付き文書

一つのリンク元文書は、複数のリンク先文書と関連づけることができます。また、一つのリンク先文書も、複数のリンク元文書から関連づけられることができます。したがって、 $m:n$ (m, n は 1 以上の整数) のリンクを設定することで、 m 個と n 個の文書が文書間リンクによって関連づけられます。

文書間リンクによる文書の管理例を次の図に示します。

2. 文書管理で使用する概念

図 2-5 文書間リンクによる文書の管理例



この例では、文書「研究論文」は、文書「参考文献」から記述を引用し、文書「補足資料」を本文中で参照しています。このため、「研究論文」と「参考文献」、「研究論文」と「補足資料」を、それぞれ文書間リンクで関連づけています。このように文書間を関連づけておくことによって、必要な資料をまとめて管理できます。また、関連づけている「参考文献」が更新された場合や、「補足資料」が削除された場合には、「研究論文」から確認できます。

2.8 フォルダ

この節では、DocumentBroker でのフォルダの概念について説明します。

DocumentBroker では、複数の文書をまとめたり、関連づけて分類したりするためにフォルダという概念を使用します。フォルダを使用すると、複数の文書を目的に応じて一つにまとめて管理したり、一つの文書を複数の分類に関連づけたりできます。また、フォルダを別のフォルダに関連づけることもできます。

DocumentBroker では、次のフォルダを表す文書管理オブジェクトを扱います。

- バージョンなしフォルダ

2.8.1 バージョンなしフォルダ

バージョンなしフォルダは、一つのバージョンだけを持つフォルダです。

バージョンなしフォルダには次の二つの機能があります。

直接型リンク

一つのフォルダに複数の文書管理オブジェクト（文書またはフォルダ）をリンク付けて、ディレクトリのイメージで文書をまとめて管理する機能です。

参照型リンク

一つの文書管理オブジェクト（文書またはフォルダ）を複数のフォルダにリンク付けて、文書を分類管理する機能です。

2.9 独立データ

この節では、DocumentBroker での独立データの概念について説明します。

DocumentBroker では、文書やフォルダの体系に関係なく、独立したデータを作成、管理できます。また、独立データには、プロパティを定義して管理できます。

独立データは、独立したデータを扱う文書管理オブジェクトです。このため、文書のようにまとめたり、フォルダに関連づけたり、バージョンを管理したり、フォルダのように上位・下位の階層構造で管理したりすることはできません。

2.10 アクセス制御

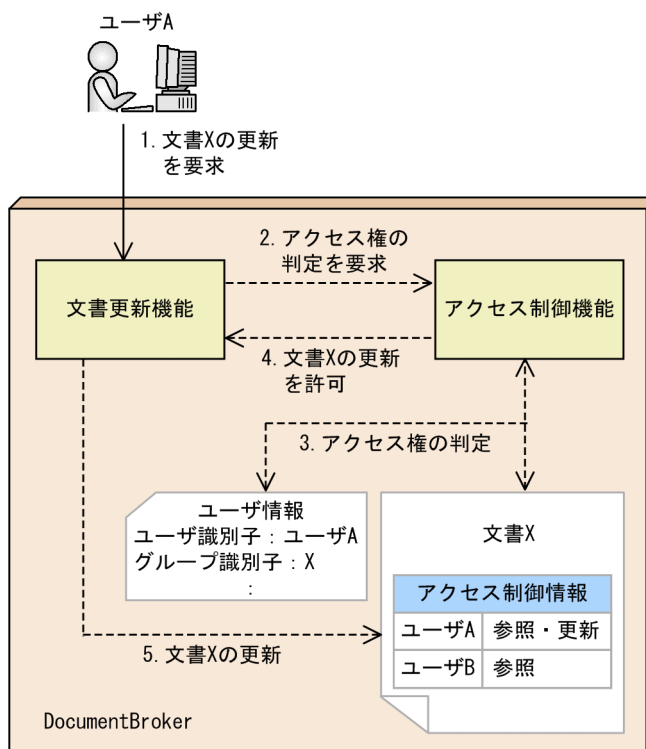
この節では、DocumentBroker でのアクセス制御の概念について説明します。

DocumentBroker では、ユーザごとに文書やフォルダに対するアクセスの範囲を限定することをアクセス制御といいます。DocumentBroker で作成、管理する文書管理オブジェクトは、アクセス制御をしない状態では、DocumentBroker にログインしたユーザであれば、だれでも参照したり更新したりできます。例えば、あるユーザが作成した文書を登録した場合、ほかのユーザがその文書を自由に参照したり、更新したり、削除したりできます。これでは、文書管理オブジェクトに対して、作成者以外のユーザが悪意を持って変更したり、誤って削除してしまったりすることが考えられます。

こうしたことを防ぐために、文書管理オブジェクトに対して、「どのユーザはどの操作が可能」というアクセスの範囲を限定する情報を設定しておく必要があります。DocumentBroker では、文書管理オブジェクトごとに、「だれに対して」「どのような操作を許可するか」というアクセス制御情報を設定できます。

また、ユーザが DocumentBroker にログインした場合、「ログインしたユーザがだれなのか」というユーザ情報が生成されます。ユーザが文書やフォルダに対する操作を要求すると、DocumentBroker は、操作を要求したユーザのユーザ情報、要求された操作の種類および操作対象のオブジェクトに設定されているアクセス制御情報を比較します。比較した結果、ユーザが操作対象のオブジェクトに、要求した操作を実行する権利（アクセス権）があると判定できた場合だけ、要求を処理します。操作を要求したユーザに要求した範囲のアクセス権がない場合は、エラーを返却して要求を処理しません。アクセス制御の概要を次の図に示します。

図 2-6 アクセス制御の概要



(凡例)

——> : ユーザの操作

-----> : DocumentBroker内部の処理

2. 文書管理で使用する概念

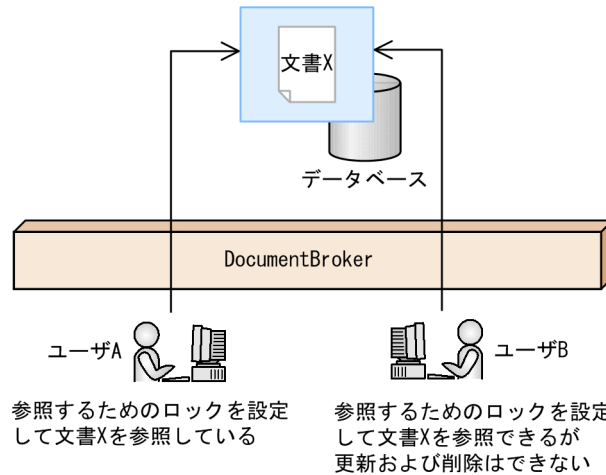
この例では、文書 X に対して、「ユーザ A からは参照と更新ができて、ユーザ B からは参照だけができるようにする」というアクセス制御情報が設定されています。ユーザ A が文書 X を更新しようとする時、ユーザ A のユーザ情報と文書 X のアクセス制御情報が比較され、文書 X の更新についてユーザ A のアクセス権が判定されます。文書 X のアクセス制御情報には、ユーザ A の「参照・更新」のアクセス権が設定されているため、アクセス権の判定の結果、ユーザ A による文書 X の更新が許可されて、文書 X が更新されます。

2.11 排他制御

この節では、DocumentBroker での排他制御の概念について説明します。

DocumentBroker では、文書管理オブジェクトを操作する場合には、ほかのユーザから同時にオブジェクトに対する操作が実行されないようにするために、排他制御をする必要があります。文書管理オブジェクトに対してロックを設定することによって、その文書管理オブジェクトの操作についてユーザ間の排他制御が実現します。ロックの設定例を次の図に示します。

図 2-7 ロックの設定例



この例では、ユーザ A は、参照するためのロックを設定して文書 X を参照しています。このため、ユーザ B は、同じロックを設定して文書 X を参照することはできませんが、更新および削除はできません。

なお、DocumentBroker の排他制御には、メソッドによって暗黙的に設定されるロックと、ユーザが明示的に設定するロックがあります。

3

文書管理モデル

この章では、DocumentBroker での文書管理体系で使用する各文書管理モデルについて説明します。モデルは複数組み合わせで導入できます。

-
- 3.1 バージョン管理モデル
 - 3.2 レンディション管理モデル
 - 3.3 リファレンスファイル文書の管理モデル
 - 3.4 リンクモデル
 - 3.5 バージョンなしフォルダによる管理モデル
 - 3.6 独立データ管理モデル
 - 3.7 属性管理モデル
 - 3.8 アクセス制御モデル
 - 3.9 排他制御モデル
-

3.1 バージョン管理モデル

この節では、文書管理オブジェクトのバージョン管理モデルについて説明します。

バージョンを管理できる文書管理オブジェクトは、バージョン付き文書です。

バージョン管理モデルは、ほかの文書管理モデルと組み合わせて使用できます。

ほかの文書管理モデルとの組み合わせ

- バージョン付き文書は、バージョン管理モデルとレンディション管理モデルを組み合わせで管理できます。レンディション管理モデルについては、「3.2 レンディション管理モデル」を参照してください。

ここでは、バージョン付きオブジェクトによるバージョン管理の概要と、バージョン付き文書によるバージョン管理について説明します。

3.1.1 バージョン管理の概要

ここでは、バージョン付きオブジェクトによるバージョン管理の概要について説明します。

(1) バージョンの追加

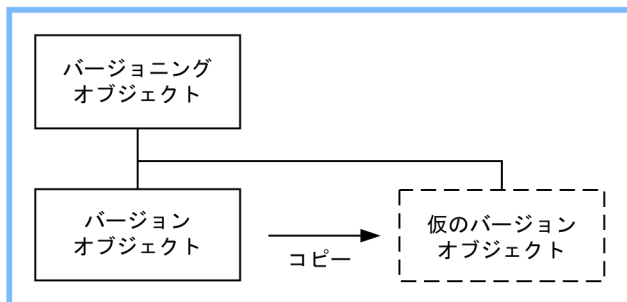
バージョンを追加する場合には、チェックアウトとチェックインという操作が必要になります。

チェックアウト

チェックアウトは、新しいバージョンの追加を予約するために、最新バージョンの次に仮のバージョンを追加する操作です。チェックアウトによって、仮のバージョンのバージョンオブジェクトとして、最新バージョンのバージョンオブジェクトがコピーされて追加されます。バージョン付きオブジェクトのチェックアウトを次の図に示します。

図 3-1 バージョン付きオブジェクトのチェックアウト

バージョン付きオブジェクト



(凡例)

: DocumentBrokerオブジェクト

チェックアウトによって仮のバージョンが追加されたバージョン付きオブジェクトは、ほかのユーザによってチェックアウトできなくなります。チェックアウトは、チェックアウトされたバージョン付きオブジェクトがチェックインされるかまたはチェックアウトが取り消されるまで有効です。チェックアウトを取り消すと仮のバージョンは削除されます。

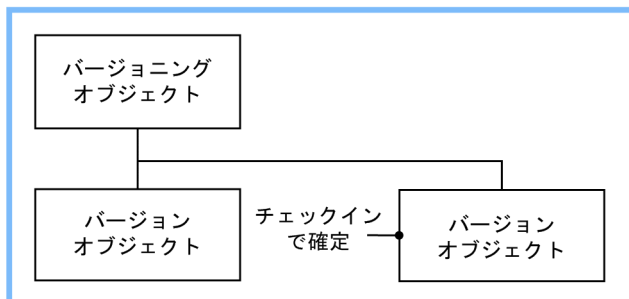
なお、バージョン付き文書をチェックアウトすると、追加された仮のバージョンには最新バージョンのマスタレンディションのコンテンツがコピーされて登録されますが、サブレンディションのコンテンツは、仮のバージョンには登録されません。

チェックイン

チェックインは、チェックアウトで追加した仮のバージョンを最新バージョンとして確定する操作です。チェックインすることでチェックアウトが解除され、バージョンが追加されます。バージョン付きオブジェクトのチェックインを次の図に示します。

図 3-2 バージョン付きオブジェクトのチェックイン

バージョン付きオブジェクト



(凡例)

: DocumentBrokerオブジェクト

(2) バージョンの順序性とカレントバージョン

バージョンには順序性があります。バージョンの順序とは、それぞれのバージョンをチェックインした順序です。なお、最新のバージョンのことをカレントバージョンといいます。

(3) バージョンの指定方法

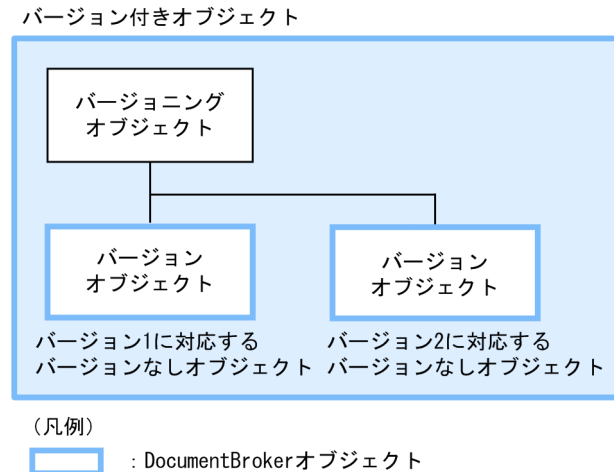
バージョン付きオブジェクトの個々のバージョンは、バージョン識別子によって指定できます。チェックアウトしている文書のバージョンを指定する場合は、チェックアウト時に取得した仮のバージョン識別子を指定します。また、バージョンのチェックアウト状態を確認して、すでにチェックアウトされている文書管理オブジェクトの仮のバージョン識別子を取得することもできます。

仮のバージョン識別子は、チェックイン後のバージョンの識別子とは異なりますので、注意してください。バージョンをチェックアウトしたあとで、チェックアウトを取り消した場合は、仮のバージョンオブジェクトおよび仮のバージョン識別子は削除されます。

(4) バージョン付きオブジェクトとバージョンなしオブジェクトの関係

バージョン付きオブジェクトでは、バージョン付きオブジェクトの1バージョンとしてバージョンなしオブジェクトを管理できます。バージョンなし文書はバージョン付き文書の1バージョンに対応します。つまり、バージョン付きオブジェクトのバージョンオブジェクトは、バージョンなしオブジェクトに対応します。バージョン付きオブジェクトとバージョンなしオブジェクトの関係を次の図に示します。

図 3-3 バージョン付きオブジェクトとバージョンなしオブジェクトの関係



このため、バージョン付きオブジェクトのバージョンオブジェクトをバージョンなしオブジェクトとみなして、バージョンなしオブジェクトクラスの機能を使用して操作できます。また、バージョンなしオブジェクトから、自身をバージョンオブジェクトとして管理しているバージョンニングオブジェクトをたどることもできます。

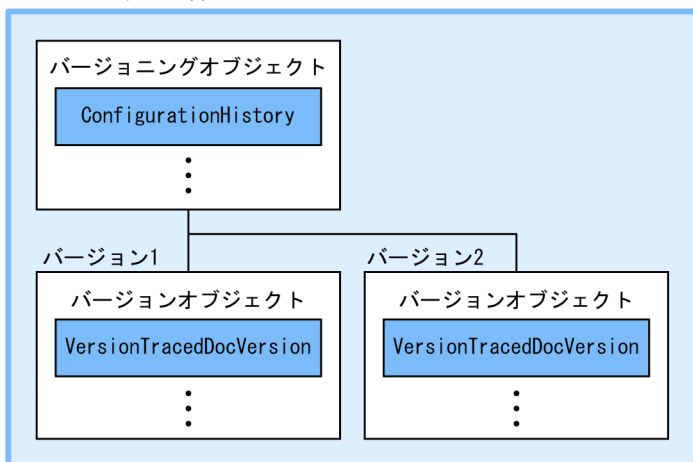
例えば、バージョン付き文書を全文検索する場合、実際に全文検索の対象となるコンテンツは個々のバージョンごとに存在します。このため、検索はバージョンオブジェクトであるバージョンなし文書が対象になり、検索結果としてバージョンオブジェクトのプロパティが取得できます。このとき、例えば、「全文検索で取得したバージョンなし文書を更新して、新しいバージョンとして追加したい」という場合などには、バージョンオブジェクトからバージョンニングオブジェクトをたどることもできます。

3.1.2 バージョン付き文書を構成する DocumentBroker オブジェクト

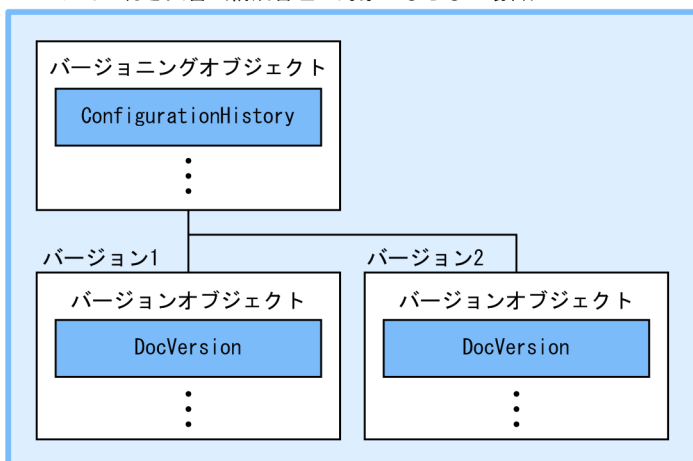
ここでは、バージョン付き文書を構成する DocumentBroker オブジェクトについて説明します。バージョン付き文書を構成する DocumentBroker オブジェクトを次の図に示します。

図 3-4 バージョン付き文書を構成する DocumentBroker オブジェクト

バージョン付き文書




バージョン付き文書（構成管理の対象にならない場合）



(凡例)

 : DocumentBroker オブジェクト

 : トップオブジェクト

バージョン付き文書のバージョンニングオブジェクトのトップオブジェクトクラスは、`dmaClass_ConfigurationHistory` クラスまたはそのサブクラスです。

バージョン付き文書のバージョンオブジェクトのトップオブジェクトクラスは、`dmaClass_DocVersion` クラス、`edmClass_VersionTracedDocVersion` クラス、またはそのサブクラスです。

バージョン付き文書を構成する DocumentBroker オブジェクトについて説明します。

ConfigurationHistory オブジェクト

バージョン付き文書のバージョンニングオブジェクトのトップオブジェクトです。この DocumentBroker オブジェクトのプロパティがバージョン付き文書クラスのデフォルトのプロパティとなり、バージョンニングオブジェクトのプロパティとして使用できます。また、この DocumentBroker オブジェクトの OIID がバージョン付き文書の OIID となります。

VersionTracedDocVersion オブジェクト

バージョンオブジェクトのトップオブジェクトです（トップオブジェクトクラスは、`edmClass_VersionTracedDocVersion` クラスまたはそのサブクラス）。

3. 文書管理モデル

この DocumentBroker オブジェクトのプロパティは、バージョンオブジェクトのプロパティとして使用できます。バージョン付き文書の個々のバージョンの識別には、この DocumentBroker オブジェクトの OIID ではなく、バージョン識別子を使用します。

DocVersion オブジェクト

バージョンオブジェクトのトップオブジェクトです (トップオブジェクトクラスは、dmaClass_DocVersion クラス、または edmClass_VersionTracedDocVersion クラス以外の dmaClass_DocVersion クラスのサブクラス)。

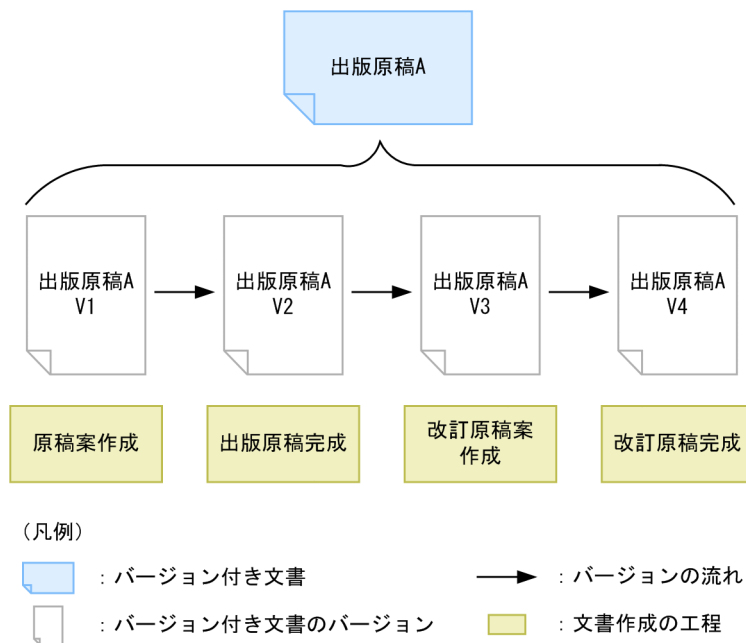
この DocumentBroker オブジェクトのプロパティは、バージョンオブジェクトのプロパティとして使用できます。バージョン付き文書の個々のバージョンの識別には、この DocumentBroker オブジェクトの OIID ではなく、バージョン識別子を使用します。

3.1.3 バージョン付き文書による文書管理

ここでは、バージョン付き文書による文書管理について説明します。

バージョン付き文書の管理例を次の図に示します。

図 3-5 バージョン付き文書の管理例



この例では、文書作成の工程に従って「出版原稿 A」という文書のバージョンを管理しています。この例では、原稿案を作成した時点で「出版原稿 A」を登録しています。そして、出版原稿が完成した時点、改訂版の原稿案を作成した時点および改訂版の出版原稿が完成した時点で「出版原稿 A」を更新してバージョンを追加しています。このように、新しく文書を登録した時点でバージョンを付け始め、履歴を残して一連の複数のバージョンを管理できます。DocumentBroker では、このように連続する複数のバージョンを持つ文書を一つの文書 (バージョン付き文書) として管理できます。

また、バージョン付き文書と個々のバージョンには、プロパティを設定して管理できます。文書のプロパティとして「作成者」などを設定しておくと、文書の検索で使用できます。プロパティの管理については、「3.7 属性管理モデル」を参照してください。

3.2 レンディション管理モデル

この節では、レンディション管理モデルについて説明します。

3.2.1 レンディション管理の概要

ここでは、レンディション管理の概要について説明します。

(1) レンディションの種類

DocumentBroker では、バージョンなし文書またはバージョン付き文書に 1 個、または複数のレンディションを登録して管理します。レンディション管理モデルでは、次の二つのレンディションを区別して扱います。

マスタレンディション

主要なレンディションです。レンディションを 1 個だけ登録した場合は、そのレンディションがマスタレンディションになります。

サブレンディション

マスタレンディション以外のレンディションです。文書をマルチレンディション文書として管理する場合、一つの文書に対して 1 個のマスタレンディション、および最大 9 個のサブレンディションを登録できます。DocumentBroker では文書の作成時、コンテンツの更新時、またはレンディションの追加時に、レンディションを登録できます。また、文書の作成時またはレンディションの追加時には、複数のレンディションを一括して登録できます。サブレンディションは削除できます。ただし、マスタレンディションは削除できません。また、サブレンディションをマスタレンディションに変更することもできます。この場合、それまでマスタレンディションだったレンディションはサブレンディションに変更されます。

(2) レンディションのコンテンツの管理

文書のコンテンツ (Word やテキストエディタなどのアプリケーションプログラムで作成された文書ファイル) は、プロパティおよびレンディションタイプ (MIME 形式) の情報を合わせて、レンディションとして文書に登録して管理します。

文書のコンテンツに対しては、次の操作ができます。

コンテンツのダウンロード

文書のコンテンツをダウンロードして参照します。

コンテンツのアップロード

文書のコンテンツを登録します。また、マスタレンディションのコンテンツの登録時には、登録するコンテンツの全文検索インデックスを作成することもできます。

(3) マルチレンディション文書の用途

バージョンなし文書またはバージョン付き文書に複数のレンディションを登録して、マルチレンディション文書として管理すると、次のような場合に使用できます。

あるアプリケーションプログラムで作成したファイルを、そのアプリケーションプログラムをインストールしていないマシンでも参照できるように、同じ内容の HTML 形式および PDF 形式のファイルを作成して一緒に登録したいという場合

アクセス制御モデルとの連携で、アクセス権によって参照できるファイルの形式を限定するために、更新できる形式のファイルと更新できない形式のファイル (例えば、同じ内容の Word 形式のファイルと

PDF 形式のファイル)を両方登録したい場合

アクセス制御モデルについては、「3.8 アクセス制御モデル」を参照してください。

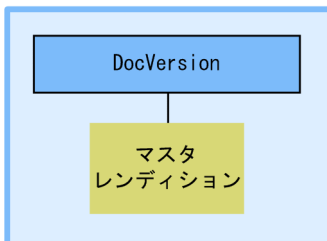
3.2.2 レンディションを管理する文書を構成する DocumentBroker オブジェクト

ここでは、レンディションを管理するバージョンなし文書およびバージョン付き文書を構成する DocumentBroker オブジェクトについて説明します。

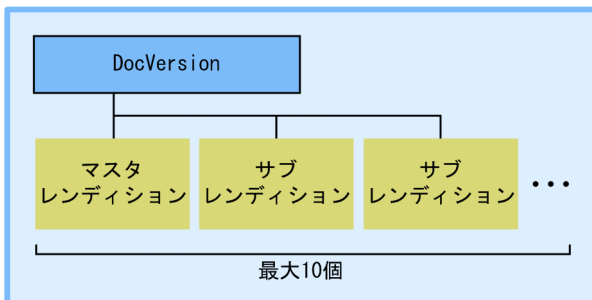
レンディションを登録したバージョンなし文書を構成する DocumentBroker オブジェクトを次の図に示します。

図 3-6 レンディションを登録したバージョンなし文書を構成する DocumentBroker オブジェクト



バージョンなし文書



バージョンなし文書 (マルチレンディション文書の場合)



(凡例)

-  : DocumentBrokerオブジェクト
-  : トップオブジェクト

バージョンなし文書のトップオブジェクトクラスは、`dmaClass_DocVersion` クラスまたはそのサブクラスです。

バージョンなし文書を構成する DocumentBroker オブジェクトについて説明します。

DocVersion オブジェクト

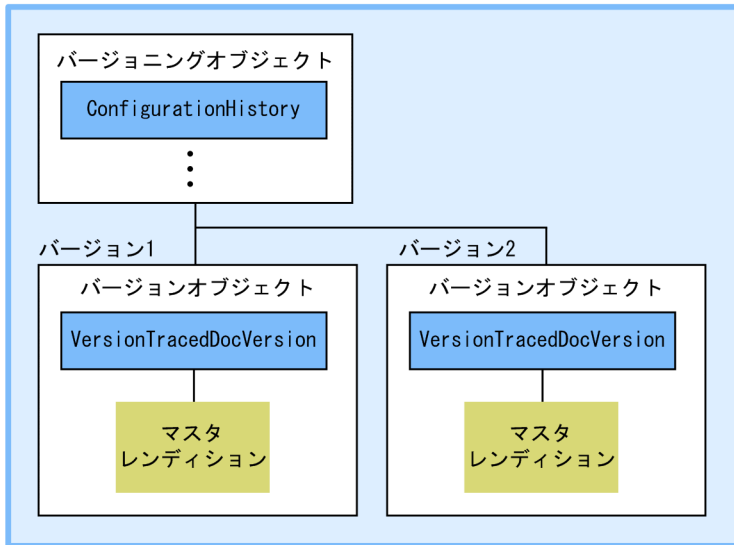
バージョンなし文書のトップオブジェクトです。

バージョンなし文書をマルチレンディション文書として管理する場合は、一つの DocVersion オブジェクトに複数のレンディションを登録できます。

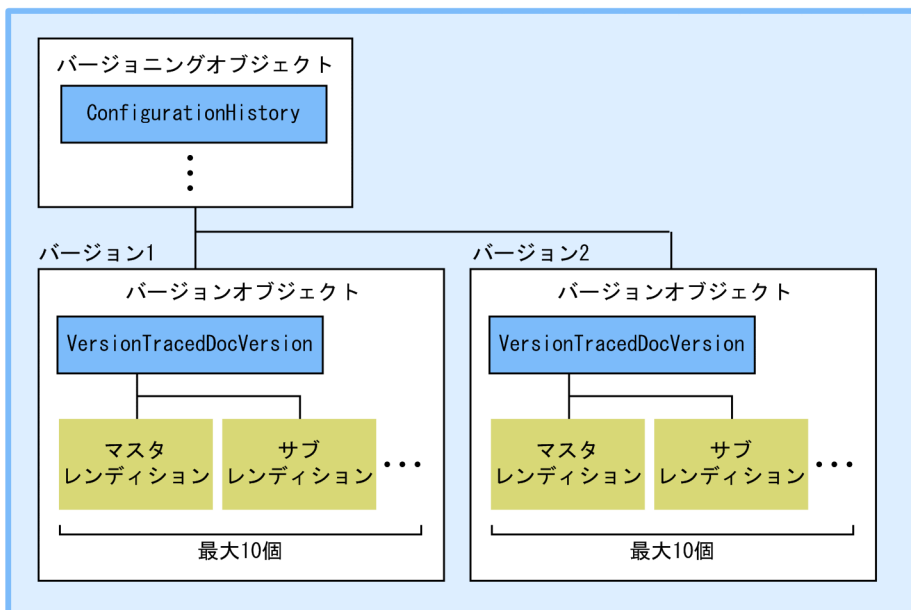
次に、バージョン付き文書について説明します。レンディションを登録したバージョン付き文書を構成する DocumentBroker オブジェクトを次の図に示します。

図 3-7 レンディションを登録したバージョン付き文書を構成する DocumentBroker オブジェクト


バージョン付き文書




バージョン付き文書（マルチレンディション文書の場合）



(凡例)

 : DocumentBroker オブジェクト

 : トップオブジェクト

バージョン付き文書を構成する DocumentBroker オブジェクトの詳細については、「3.1.2 バージョン付き文書を構成する DocumentBroker オブジェクト」を参照してください。

バージョン付き文書をマルチレンディション文書として管理する場合は、一つの VersionTracedDocVersion オブジェクト（または DocVersion オブジェクト）に複数のレンディションを登録します。

3.2.3 レンディションによる文書管理

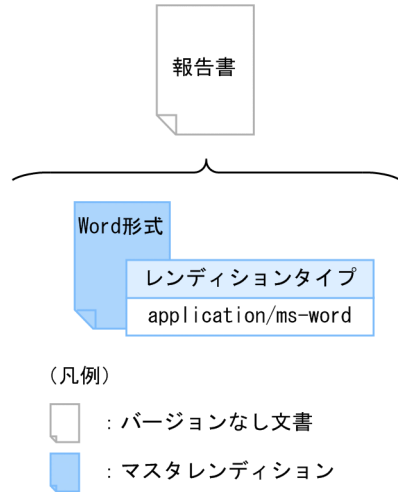
ここでは、レンディションによる文書管理について説明します。

3. 文書管理モデル

(1) レンディションの登録

バージョンなし文書に1個のレンディションを登録して管理している例を次の図に示します。

図 3-8 バージョンなし文書のレンディションの管理例

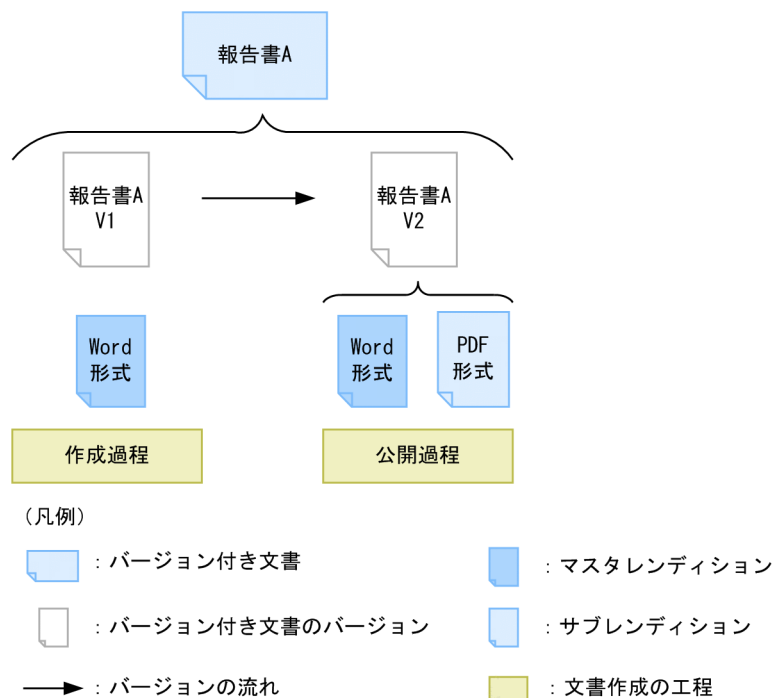


この例では、バージョンなし文書「報告書」に、1個のレンディションを登録して管理しています。バージョンなし文書「報告書」には、コンテンツとして Word 形式のファイルと、レンディションタイプとしてコンテンツを作成したアプリケーションプログラムの名前を登録しています。

(2) レンディションの追加

バージョン付き文書にレンディションを追加して、マルチレンディション文書として管理する例を次の図に示します。

図 3-9 マルチレンディション文書のレンディションの追加例



この例では、文書の作成過程で、文書の作成者が更新用にコンテンツをダウンロードできるように、コンテンツを作成しているアプリケーションプログラム（この例の場合 Word）の形式のレンディションを登録しています。この Word 形式のレンディションがマスタレンディションとなります。そして、文書が完成して公開過程になったタイミングでバージョンを追加して、追加したバージョンをほかのユーザが参照しやすいように、PDF 形式のレンディションを追加登録しています。この追加した PDF 形式のレンディションがサブレンディションとなります。

なお、マルチレンディション文書をチェックアウトすると、最新バージョンのマスタレンディションが仮のバージョンにコピーされます。サブレンディションはコピーされません。例えば、図 3-9 の場合、「報告書 A」にさらにバージョンを追加するときは、チェックアウトすると、Word 形式のレンディションはコピーされますが、PDF 形式のレンディションはコピーされません。

(3) レンディションの参照

マルチレンディション文書は、レンディションタイプを指定して参照できます。例えば、図 3-9 の場合、「報告書 A」のバージョン 2 を参照するときは、レンディション一覧を取得して、Word 形式か PDF 形式かを指定して参照します。文書の作成者が文書をバージョンアップしたいときには、Word でファイルを編集するために Word 形式を指定してダウンロードしたり、Word をインストールしていない環境のユーザは PDF 形式を指定してダウンロードしたりするなど、目的や、コンテンツを参照するマシン環境に応じた参照ができます。

また、レンディション管理モデルとアクセス制御モデルを組み合わせると、マルチレンディション文書を管理することもできます。例えば、Word 形式のコンテンツと PDF 形式のコンテンツを登録した文書がある場合に、「基本コンテンツ更新権のあるユーザには Word 形式のコンテンツを参照させる」、「基本コンテンツ参照権しかないユーザには PDF 形式のコンテンツを参照させる」というように、ユーザの保持するアクセス権によって、ユーザが参照できるコンテンツを限定するという運用が考えられます。このとき、DocumentBroker では文書ごとのアクセス制御しかできないため、さらにレンディションごとのアクセス制御をユーザアプリケーションプログラムで処理する必要があります。アクセス制御モデルについては、「3.8 アクセス制御モデル」を参照してください。

(4) 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の更新

マルチレンディション文書のコンテンツを更新する場合、レンディションタイプを指定することによって、レンディションを指定して更新できます。例えば、600dpi の GIF ファイルをマスタレンディション、300dpi の GIF ファイルをサブレンディションとして管理している場合に、300dpi の GIF ファイルを更新するときは、該当するレンディションのレンディションタイプを指定してください。

マスタレンディションのコンテンツを更新した場合には、サブレンディションのコンテンツも更新してください。

コンテンツの更新時には、レンディションタイプを変更できます。また、マスタレンディションのコンテンツの更新時には、全文検索インデックスを作成できます。

(5) レンディションの削除

マルチレンディション文書のレンディションを削除する場合、削除対象のレンディションのレンディションタイプを指定することによって、複数のレンディションを一括して削除できます。

マスタレンディションは削除できません。なお、マルチレンディション文書自体を削除した場合は、すべてのレンディションが削除されます。

(6) レンディションの検索

レンディションタイプを表す `dbrProp_RenditionType` プロパティは、検索条件には指定できません。例えば、「マスタレンディションのレンディションタイプが "application/ms-word" である文書を検索する」のような指定はできません。また、検索結果として、レンディションタイプを取得することもできません。レンディションタイプを参照したい場合は、ほかの検索条件によって参照したい文書を特定してから取得してください。

(7) レンディションのコンテンツ種別変換

レンディションのコンテンツの格納先（コンテンツ種別）を変換する機能です。文書の実体であるコンテンツの格納先を変化するデータの利用価値に応じて、最適な格納先へ変換したい場合などに使用できます。コンテンツ種別の変換は、変換後のコンテンツをサブレンディションとして追加します。

3.2.4 レンディションのプロパティ

ここでは、レンディションのプロパティについて説明します。

レンディションには、プロパティを設定して管理できます。ただし、レンディションのプロパティとしてユーザ定義プロパティは設定できません。レンディションに設定されているプロパティは、次の4種類です。

`dbrProp_RenditionType` プロパティ
レンディションタイプを表すプロパティです。

`dbrProp_RetrievalName` プロパティ
レンディションに登録されたコンテンツのファイル名を表すプロパティです。

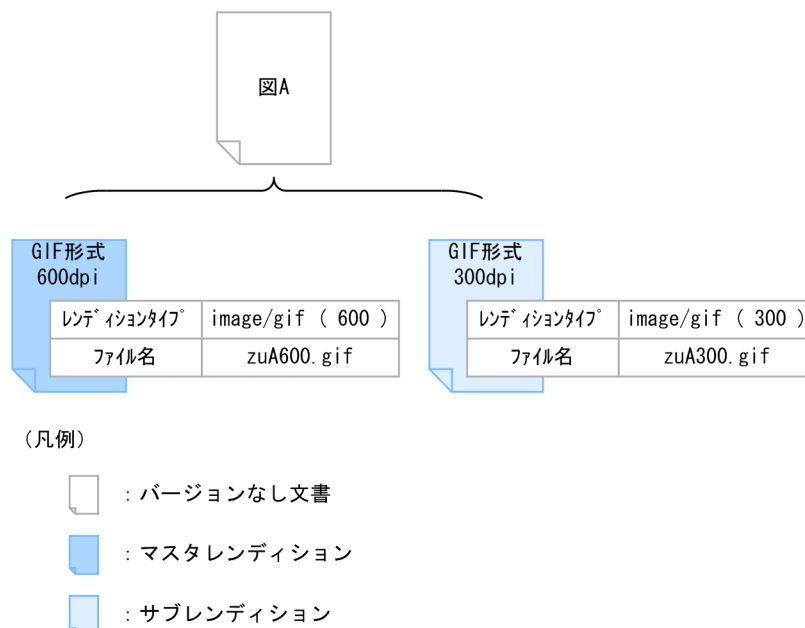
`dbrProp_RenditionStatus` プロパティ
レンディション状態を表すプロパティです。

`dbrProp_ContentType` プロパティ
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。

これらのプロパティは、属性検索条件として指定したり、検索結果一覧の情報として取得したりすることはできません。

レンディションタイプとコンテンツのファイル名を設定したレンディションのプロパティの管理例を次の図に示します。

図 3-10 レンディションのプロパティの管理例



次に、レンディションの各プロパティについて、説明します。

(1) レンディションタイプを表すプロパティ (dbrProp_RenditionType プロパティ)

レンディション管理モデルでは、同じレンディションタイプを重複して登録できません。ただし、"タイプ/サブタイプ(コメント)"などの形式でレンディションタイプを指定できるため、同じ拡張子のファイルでもレンディションタイプを重複させないで登録できます。例えば、図 3-10 の場合、「図 A」というバージョンなし文書のマスタレンディションとサブレンディションに、同じ GIF 形式でも異なる解像度(マスタレンディションは 600dpi、サブレンディションは 300dpi)で保存したコンテンツを登録しています。このため、"image/gif (600)" および "image/gif (300)" のように、コメントまで指定したレンディションタイプを設定して二つの GIF 形式のレンディションを登録しています。

レンディションタイプは文書の作成時、コンテンツの更新時、またはレンディションの追加時に指定します。このとき指定したレンディションタイプが、dbrProp_RenditionType プロパティの値として設定されます。

文書の作成時、コンテンツの更新時、またはレンディションの追加時に、レンディションタイプの指定を省略すると、レンディション定義ファイルの定義に従ってコンテンツのファイルの拡張子から判断して、レンディションタイプを自動的に設定します。レンディション定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。ただし、この例のように拡張子が同じでもファイルの保存状態などによって別々のレンディションに登録したい場合は、ファイルの拡張子からは判断できないので、コメントまで指定したレンディションタイプを明示的に指定してください。

なお、dbrProp_RenditionType プロパティは、STR 型のプロパティです。

(2) ファイル名を表すプロパティ (dbrProp_RetrievalName プロパティ)

レンディションタイプのほかに、各レンディションには、ファイル名を表す dbrProp_RetrievalName プロパティを設定できます。

ファイル名は、文書の作成時、コンテンツの更新時、またはレンディションの追加時に指定します。この

とき指定したファイル名が、dbrProp_RetrievalName プロパティの値として設定されます。図 3-10 の例では、マスタレンディションには zuA600.gif、サブレンディションには zuA300.gif という値が設定されています。なお、dbrProp_RetrievalName プロパティは、STR 型のプロパティです。

3.2.5 レンディション状態を表すプロパティ (dbrProp_RenditionStatus プロパティ)

マルチレンディション文書では、同じ内容を持つ複数のレンディションタイプのコンテンツを、まとめて一つの文書に登録して管理できます。このため、マスタレンディションとサブレンディションは、常に同じ内容にしておく必要があります。マスタレンディションのコンテンツを更新した場合には、サブレンディションのコンテンツも更新する必要があります。

マスタレンディションとサブレンディションの dbrProp_RenditionStatus プロパティの値について説明します。

マスタレンディションの場合

マスタレンディションの dbrProp_RenditionStatus プロパティの値は常に DbjDef.RENDSTATUS_MASTERREND です。

サブレンディションの場合

サブレンディションの dbrProp_RenditionStatus プロパティの値は、メソッドの実行によって遷移します。また、サブレンディションの dbrProp_RenditionStatus プロパティの値は、状態フラグと変換フラグの組み合わせで表されます。

- 状態フラグ

マスタレンディションに対するサブレンディションの更新状態を示す値です。状態フラグの値は、DocumentBroker によって自動的に設定されます。ユーザは状態フラグの値を変更できません。

- 変換フラグ

レンディション変換機能を使用する場合に、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によって自動作成するかどうかを示す値です。また、変換フラグには、レンディション変換でエラーが発生したことを示す値も設定されます。変換フラグの値は、DocumentBroker によって自動的に設定されます。ユーザは変換フラグの値を変更できます。

なお、変換フラグは、レンディション変換機能を使用しない場合は使用しません。

なお、dbrProp_RenditionStatus プロパティは、INT 型のプロパティです。

(1) サブレンディションの状態フラグと変換フラグの値

サブレンディションの状態フラグと変換フラグの値について説明します。

(a) 状態フラグの値

サブレンディションの状態フラグの値は、マスタレンディションまたはサブレンディションのコンテンツの更新時に、DocumentBroker によって自動的に設定されます。サブレンディションの状態フラグの値を次の表に示します。

表 3-1 サブレンディションの状態フラグの値

状態フラグを表す定数	説明
DbjDef.RENDSTATUS_NO_SUBREND	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登録されていません。
DbjDef.RENDSTATUS_SUBREND_EXIST	マスタレンディションとサブレンディションの更新状態が一致しています。

状態フラグを表す定数	説明
DbjDef.RENDSTATUS_MASTERREND_UPDATE	マスタレンディションとサブレンディションの更新状態が不一致です。マスタレンディションのコンテンツは更新されていますが、サブレンディションのコンテンツが更新されていません。

(b) 変換フラグの値

サブレンディションの変換フラグの値は、文書の作成時、コンテンツの更新時またはレンディションの追加時に、コンテンツのファイルパスの指定の有無に応じて DocumentBroker が自動的に設定します。レンディション変換機能によるレンディション変換実行時にエラーが発生した場合には、変換エラーを示す値が設定されます。

変換フラグの値をユーザが変更する場合は、レンディションのプロパティを更新する DbjObj#writeRenditionProperties メソッドを使用します。

サブレンディションの変換フラグの値を次の表に示します。

表 3-2 サブレンディションの変換フラグの値

変換フラグを表す定数	説明
DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED	レンディション変換機能によるレンディション変換の対象になりません。レンディション変換の対象にしないサブレンディションに対して指定してください。
DbjDef.RENDSTATUS_CONVERT_REQUIRED	レンディション変換機能によるレンディション変換の対象になります。レンディション変換の対象にする場合に指定してください。
DbjDef.RENDSTATUS_CONVERT_ERROR	レンディション変換機能で変換エラーが発生しました。レンディション変換機能が設定する値です。

なお、状態フラグの値によって、変換フラグが取る値が異なります。

状態フラグと変換フラグの組み合わせについて、次の表に示します。

表 3-3 状態フラグと変換フラグの組み合わせ

状態フラグを表す定数	変換フラグを表す定数
DbjDef.RENDSTATUS_SUBREND_EXIST	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED DbjDef.RENDSTATUS_CONVERT_REQUIRED
DbjDef.RENDSTATUS_NO_SUBREND	DbjDef.RENDSTATUS_CONVERT_REQUIRED DbjDef.RENDSTATUS_CONVERT_ERROR
DbjDef.RENDSTATUS_MASTERREND_UPDATE	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED DbjDef.RENDSTATUS_CONVERT_REQUIRED DbjDef.RENDSTATUS_CONVERT_ERROR

例えば、サブレンディションの文書のアップロード情報でコンテンツのファイルパスを指定しなかった (null を指定した) 場合、レンディション変換機能によるレンディション変換を要求しない変換フラグ (DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED) を指定することはできません。

また、文書のアップロード情報でコンテンツのファイルパスを指定して更新したいサブレンディションに対しては、マスタレンディションとサブレンディションの更新状態が不一致 (DbjDef.RENDSTATUS_MASTERREND_UPDATE) でも、レンディション変換機能によるレンディション変換を要求しない変換フラグ (DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED) を指定できます。

(2) dbrProp_RenditionStatus プロパティとメソッドの関係

dbrProp_RenditionStatus プロパティの値とメソッドの関係について説明します。

次のメソッドについて説明します。

dbrProp_RenditionStatus プロパティの値を参照するメソッド

dbrProp_RenditionStatus プロパティの値を遷移させるメソッド

dbrProp_RenditionStatus プロパティの変換フラグを変更するメソッド

dbrProp_RenditionStatus プロパティの値によって制限されるメソッド

(a) dbrProp_RenditionStatus プロパティの値を参照するメソッド

dbrProp_RenditionStatus プロパティの値は、次のメソッドによるレンディション一覧の取得時に参照できます。

DbjObj#getRenditionList メソッド

特定のレンディションの dbrProp_RenditionStatus プロパティを参照したい場合は、DbjRenditionList インターフェースおよび DbjRenditionInfo インターフェースを使用してください。

(b) dbrProp_RenditionStatus プロパティの値を遷移させるメソッド

dbrProp_RenditionStatus プロパティの値は、次のメソッドの実行時に、DocumentBroker が自動的に値を設定することによって、遷移します。

DbjObj#changeMasterRendition メソッド

DbjObj#addRendition メソッド

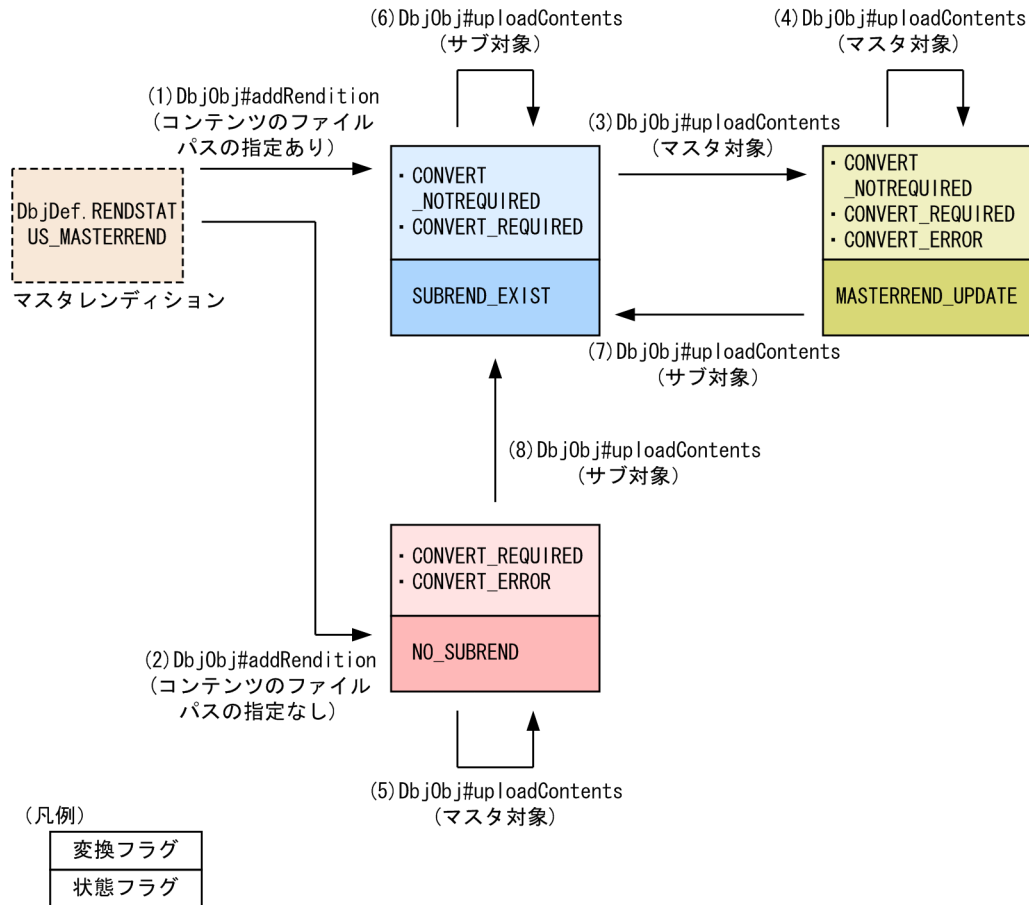
DbjObj#uploadContents メソッド

DbjObj#changeMasterRendition メソッドによる dbrProp_RenditionStatus プロパティの値の遷移については、「(d) dbrProp_RenditionStatus プロパティの値によって制限されるメソッド」を参照してください。

DbjObj#addRendition メソッドおよび DbjObj#uploadContents メソッドによる

dbrProp_RenditionStatus プロパティの値の遷移を、次の図に示します。図中の (1) から (8) の番号は、表 3-4 から表 3-6 の説明中の番号と対応しています。

図 3-11 メソッドによる dbrProp_RenditionStatus プロパティの値の遷移



変換フラグ (上位2バイト)

- CONVERT_NOTREQUIRED : DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED (変換不要)
- CONVERT_REQUIRED : DbjDef.RENDSTATUS_CONVERT_REQUIRED (変換要)
- CONVERT_ERROR : DbjDef.RENDSTATUS_CONVERT_ERROR (変換エラー)

状態フラグ (下位2バイト)

- NO_SUBREND : DbjDef.RENDSTATUS_NO_SUBREND (サブレンディションのコンテンツなし)
- SUBREND_EXIST : DbjDef.RENDSTATUS_SUBREND_EXIST (サブレンディションのコンテンツあり。マスタレンディションの状態と一致)
- MASTERREND_UPDATE : DbjDef.RENDSTATUS_MASTERREND_UPDATE (サブレンディションのコンテンツあり。マスタレンディションの状態と不一致)

マスタ対象 : マスタレンディションに対してメソッドを実行することを示します。
 サブ対象 : サブレンディションに対してメソッドを実行することを示します。

図で示した遷移の詳細を、メソッドごとに説明します。

DbjObj#addRendition メソッド

追加したサブレンディションのプロパティには、コンテンツのファイルパスの指定に応じて、次の表に示す値が設定されます。

表 3-4 DbjObj#addRendition メソッドによって設定される dbrProp_RenditionStatus プロパティの値

コンテンツのファイルパスの指定	変換フラグ	状態フラグ
あり (1) の遷移)	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED	DbjDef.RENDSTATUS_SUBREND_EXIST

3. 文書管理モデル

コンテンツのファイルパスの指定	変換フラグ	状態フラグ
なし ((2) の遷移) (null を指定した場合)	DbjDef.RENDSTATUS_CONVERT_REQUIRED	DbjDef.RENDSTATUS_NO_SUBREND

DbjObj#uploadContents メソッド (マスタレンディションを更新する場合)
メソッドを実行した時点の状態フラグに応じて、次の表に示すように遷移します。

表 3-5 DbjObj#uploadContents メソッドによる dbrProp_RenditionStatus プロパティの値の遷移 (マスタレンディションを更新する場合)

メソッドを実行した時点の状態フラグ	変換フラグの遷移	メソッドを実行したあとの状態フラグ
DbjDef.RENDSTATUS_SUBREND_EXIST ((3) の遷移)	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED (遷移しません)	DbjDef.RENDSTATUS_MASTERREND_UPDATE
	DbjDef.RENDSTATUS_CONVERT_REQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED (遷移しません)	
DbjDef.RENDSTATUS_MASTERREND_UPDATE ((4) の遷移)	DbjDef.RENDSTATUS_CONVERT_ERROR の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED	DbjDef.RENDSTATUS_MASTERREND_UPDATE (遷移しません)
	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED (遷移しません)	
	DbjDef.RENDSTATUS_CONVERT_REQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED (遷移しません)	
DbjDef.RENDSTATUS_NO_SUBREND ((5) の遷移)	DbjDef.RENDSTATUS_CONVERT_ERROR の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED	DbjDef.RENDSTATUS_NO_SUBREND (遷移しません)
	DbjDef.RENDSTATUS_CONVERT_REQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED (遷移しません)	

注 レンディション変換機能で、レンディション変換エラーが起きたあとにマスタレンディションを更新し直す場合は、更新するコンテンツにエラーの要因が残っていないかどうか確認してください。または、マスタレンディションのコンテンツを基に、別のレンディションタイプのファイルをユーザが作成して、サブレンディションのコンテンツとして登録してください。

DbjObj#uploadContents メソッド (サブレンディションを更新する場合)
メソッドを実行した時点の状態フラグに応じて、次の表に示すように遷移します。

表 3-6 DbjObj#uploadContents メソッドによる dbrProp_RenditionStatus プロパティの値の遷移 (サブレンディションを更新する場合)

メソッドを実行した時点の状態フラグ	変換フラグの遷移	メソッドを実行したあとの状態フラグ
DbjDef.RENDSTATUS_SUBREND_EXIST (6) の遷移)	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED (遷移しません)	DbjDef.RENDSTATUS_SUBREND_EXIST (遷移しません)
	DbjDef.RENDSTATUS_CONVERT_REQUIRED の場合： DbjDef.RENDSTATUS_CONVERT_REQUIRED (遷移しません)	
DbjDef.RENDSTATUS_MASTERREND_UPDATE (7) の遷移)	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED	DbjDef.RENDSTATUS_SUBREND_EXIST
DbjDef.RENDSTATUS_NO_SUBREND (8) の遷移)	DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED	DbjDef.RENDSTATUS_SUBREND_EXIST

(c) dbrProp_RenditionStatus プロパティの変換フラグを変更するメソッド

変換フラグの値をユーザが変更する場合は、DbjObj#writeRenditionProperties メソッドを使用します。

設定できる値は次のどちらかです。

DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED

DbjDef.RENDSTATUS_CONVERT_REQUIRED

ただし、次の場合は値を設定できません。

DbjObj#writeRenditionProperties メソッドで変換フラグを設定できない場合

- 状態フラグが DbjDef.RENDSTATUS_NO_SUBREND の場合、変換フラグに DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED は設定できません。
- 状態フラグが DbjDef.RENDSTATUS_NO_SUBREND の場合は、サブレンディションのコンテンツを DbjObj#uploadContents メソッドで更新すると、変換フラグを DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED に遷移させることができます。
- 変換フラグが DbjDef.RENDSTATUS_CONVERT_ERROR の場合、変換フラグに DbjDef.RENDSTATUS_CONVERT_NOTREQUIRED または DbjDef.RENDSTATUS_CONVERT_REQUIRED は設定できません。
- 変換フラグが DbjDef.RENDSTATUS_CONVERT_ERROR の場合は、エラーの要因を取り除いたコンテンツを準備して、マスタレンディションを DbjObj#uploadContents メソッドで再び更新してください。または、マスタレンディションのコンテンツを基に、別のレンディションタイプのファイルをユーザが作成して、DbjObj#uploadContents メソッドでサブレンディションのコンテンツとして登録してください。

(d) dbrProp_RenditionStatus プロパティの値によって制限されるメソッド

状態フラグと変換フラグの組み合わせによっては、DbjObj#changeMasterRendition メソッドの実行が制限されます。制限されるのは、次の場合です。

DbjObj#changeMasterRendition メソッドの実行が制限される場合

- 状態フラグが DbjDef.RENDSTATUS_NO_SUBREND の場合、DbjObj#changeMasterRendition メソッドは実行できません。

3. 文書管理モデル

- 状態フラグが DbjDef.RENDSTATUS_MASTERREND_UPDATE の場合，変換フラグが DbjDef.RENDSTATUS_CONVERT_REQUIRED または DbjDef.RENDSTATUS_CONVERT_ERROR のときは，DbjObj#changeMasterRendition メソッドは実行できません。

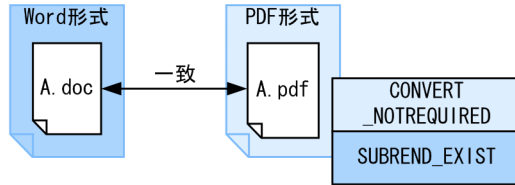
これ以外の組み合わせでは，DbjObj#changeMasterRendition メソッドを実行してマスタレンディションを変更すると，それまでサブレンディションだったレンディションはマスタレンディションに変更されて，そのレンディションの dbrProp_RenditionStatus プロパティの値は DbjDef.RENDSTATUS_MASTERREND に遷移します。

(e) レンディション変換機能によるサブレンディション更新の例

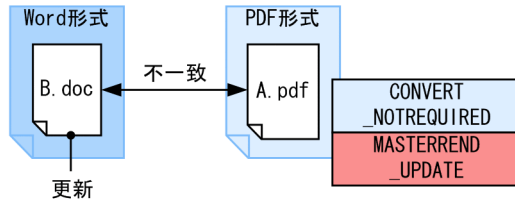
dbrProp_RenditionStatus プロパティの値が遷移する例として，レンディション変換機能によるサブレンディション更新の例を次の図に示します。

図 3-12 レンディション変換機能によるサブレンディション更新の例

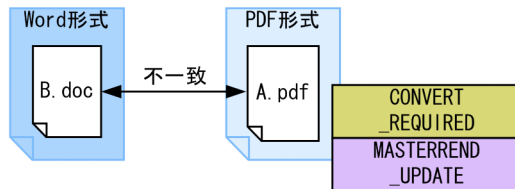
1. マスタレンディションとサブレンディションの内容が一致している。



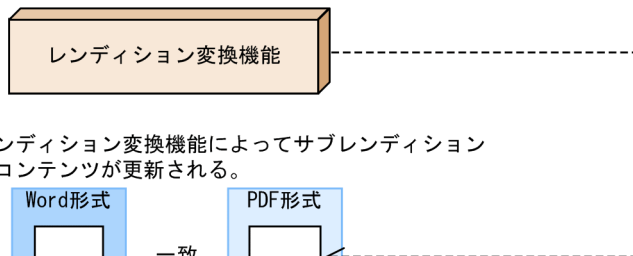
2. マスタレンディションに対してDbjObj#uploadContentsメソッドを実行して、マスタレンディションのコンテンツを更新する。



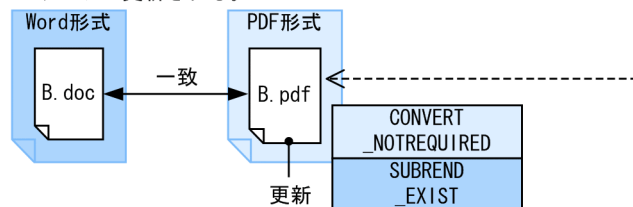
3. サブレンディションに対してDbjObj#writeRenditionPropertiesメソッドを実行して、変換フラグにDbjDef.RENDSTATUS_CONVERT_REQUIREDを設定する。



4. レンディション変換を実行する。



5. レンディション変換機能によってサブレンディションのコンテンツが更新される。



(凡例)

■ : マスタレンディション ■ : サブレンディション

図について説明します。

1. 図中の文書は、マスタレンディションのコンテンツとして Word 形式のファイル「A.doc」、サブレンディションのコンテンツとして PDF 形式のファイル「A.pdf」を持つ、バージョンなし文書です。このとき、マスタレンディションとサブレンディションの内容は一致しています。状態フラグは DbjDef.RENDSTATUS_SUBREND_EXIST です。
2. DbjObj#uploadContents メソッドによって、マスタレンディションのコンテンツを「B.doc」に更新します。このとき、マスタレンディションとサブレンディションの状態が不一致になるため、サブレンディションの状態フラグが DbjDef.RENDSTATUS_MASTERREND_UPDATE に遷移します。
3. DbjObj#writeRenditionProperties メソッドによって、サブレンディションの dbrProp_RenditionStatus プロパティの変換フラグに、

3. 文書管理モデル

DbjDef.RENDSTATUS_CONVERT_REQUIRED を設定します。

4. レンディション変換機能によってレンディション変換を実行します。3. で変換フラグに DbjDef.RENDSTATUS_CONVERT_REQUIRED を設定したサブレンディションは、レンディション変換の対象になります。レンディション変換は、マスタレンディションのコンテンツ「B.doc」を基に実行されます。
5. サブレンディションのコンテンツとして、マスタレンディションの「B.doc」を基にレンディション変換したコンテンツ「B.pdf」が登録されます。このとき、サブレンディションの状態フラグは、DbjDef.RENDSTATUS_SUBREND_EXIST に遷移します。

(3) レンディションのコンテンツ種別を表すプロパティ (dbrProp_ContentType プロパティ)

レンディションごとのコンテンツの種別は、dbrProp_ContentType プロパティによって知ることができます。コンテンツの種別から、レンディションのコンテンツが次のどれかを知ることができます。

シングルファイル文書のコンテンツ

リファレンスファイル文書のコンテンツ

上記以外のコンテンツ

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 dbrProp_ContentType プロパティを設定できません。dbrProp_ContentType プロパティを参照する場合は、DbjObj#getRenditionList メソッドを使用します。

dbrProp_ContentType の値とコンテンツ種別の関係を次の表に示します。

表 3-7 dbrProp_ContentType 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
DbjDef.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。
DbjDef.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。
DbjDef.CONTENTTYPE_OTHER	-1	上記以外のコンテンツを持つオブジェクトです。

3.3 リファレンスファイル文書の管理モデル

この節では、リファレンスファイル文書の管理モデルについて説明します。

3.3.1 リファレンスファイル文書の管理の概要

ここでは、リファレンスファイル文書を管理するための、リファレンスファイル管理機能の概要について説明します。

リファレンスファイル管理機能とは、文書の実体であるコンテンツを任意のディレクトリに格納し、文書のプロパティとコンテンツロケーションだけをデータベースで管理する機能です。

この機能には、次のような特長があります。

- ディスク移行時など、コンテンツ格納ディレクトリを変更する場合、データベースに影響を与えないで、管理するコンテンツ格納先の基点となるディレクトリパス（コンテンツ格納先ベースパス）だけの変更で対応できます。
- データベース容量を削減できます。

リファレンスファイル管理機能では、バージョンなし文書およびバージョン付き文書をリファレンスファイル文書として管理できます。

コンテンツを任意のディレクトリに登録し、そのコンテンツの格納先を、コンテンツロケーションとしてデータベースに登録します。コンテンツロケーションには、ユーザが管理するコンテンツのコンテンツ格納先ベースパスを基点とする相対パス（コンテンツ格納先パス）に登録します。コンテンツは、DocumentBroker の機能で操作します。

リファレンスファイル管理機能を使用する場合、データベースとコンテンツの整合性を考慮する必要があります。このため、1メソッド1トランザクションとすることを前提としています。文書作成を n 件連続で実行し、エラーが発生したときは、ロールバックしたあとにコンテンツだけが存在する状態となります。また、文書削除を n 件連続で実行し、エラーが発生したときは、ロールバックしたあとにコンテンツの存在しない文書が残ります。

なお、リファレンスファイル管理機能では、オブジェクトとコンテンツが不整合となる場合があります。不整合となる場合の内容を、次の表に示します。

表 3-8 オブジェクトとコンテンツが不整合となる場合

機能	エラー発生箇所	不整合の内容	対処
文書の作成	コンテンツに登録したあとのエラー	登録コンテンツが残ります。	再度実行して文書を作成してください。
文書の更新	更新前のコンテンツを削除したあと、更新後のコンテンツに登録するまでのエラー	コンテンツの存在しないオブジェクトが残ります。	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。
	更新後のコンテンツに登録したあとのエラー	コンテンツの存在しないオブジェクト残り、登録コンテンツが残ります。	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	コンテンツを削除したあとのエラー	コンテンツの存在しないオブジェクトが残ります。	再度実行して文書を削除してください。
バージョンのチェックアウト	コンテンツを複製したあとのエラー	複製したコンテンツが残ります。	チェックアウトを取り消し、コンテンツを削除したあと、ロールバックを実行してください。

3. 文書管理モデル

機能	エラー発生箇所	不整合の内容	対処
バージョンの削除	コンテンツを削除したあとのエラー	コンテンツの存在しないオブジェクトが残ります。	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	コンテンツを削除したあとのエラー	コンテンツの存在しないオブジェクトが残ります。	再度実行してチェックアウトを取り消してください。

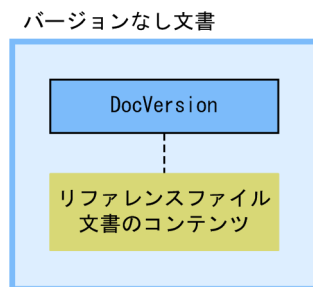
3.3.2 リファレンスファイル文書を構成する DocumentBroker オブジェクト

ここでは、リファレンスファイル文書を構成する DocumentBroker オブジェクトについて説明します。



リファレンスファイル管理機能では、バージョンなし文書およびバージョン付き文書をリファレンスファイル文書として管理できます。

リファレンスファイル文書を管理するバージョンなし文書を構成する DocumentBroker オブジェクトを次の図に示します。

図 3-13 リファレンスファイル文書を管理するバージョンなし文書を構成する DocumentBroker オブジェクト



(凡例)

-  : トップオブジェクト
-  : DocumentBroker オブジェクト

リファレンスファイル文書を管理するバージョンなし文書を構成する DocumentBroker オブジェクトについて説明します。

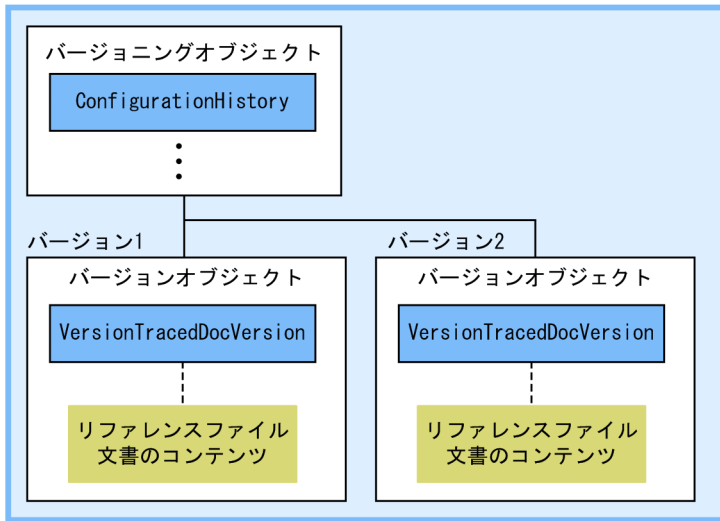
DocVersion オブジェクト

バージョンなし文書のトップオブジェクトです。

次に、バージョン付き文書について説明します。リファレンスファイル文書を管理するバージョン付き文書を構成する DocumentBroker オブジェクトを次の図に示します。

図 3-14 リファレンスファイル文書を管理するバージョン付き文書を構成する DocumentBroker オブジェクト

バージョン付き文書



(凡例)

- : トップオブジェクト
- : DocumentBroker オブジェクト

リファレンスファイル文書を管理するバージョン付き文書を構成する DocumentBroker オブジェクトの詳細については、「3.1.2 バージョン付き文書を構成する DocumentBroker オブジェクト」を参照してください。

3.4 リンクモデル

この節では、文書管理オブジェクト間を関連付けるリンクモデルについて説明します。

3.4.1 リンクによる文書管理の概要

DocumentBroker では、リンクオブジェクトを使用して、複数の文書管理オブジェクト間をリンク付けたり、複数の文書管理オブジェクト間のリンクを一括して削除したりできます。リンクの一覧を取得して、関連付けられている文書管理オブジェクトを検索できます。また、リンクオブジェクトにはプロパティも設定できます。

リンクには次の 3 種類があります。

- 直接型リンク
- 参照型リンク
- 文書間リンク

リンクモデルでは、リンクを設定するリンク元の文書管理オブジェクトをリンク元オブジェクト、リンクが設定されるリンク先の文書管理オブジェクトをリンク先オブジェクトといいます。なお、フォルダを使用したリンクの場合は、リンク元オブジェクトであるフォルダを上位オブジェクト、リンク先オブジェクトである文書またはフォルダを下位オブジェクトといいます。

リンクオブジェクトは、文書管理オブジェクトの作成時に、同時に既存の文書管理オブジェクトとのリンクを設定した場合に作成されます。また、既存の文書管理オブジェクトからほかの文書管理オブジェクトに対してリンクを設定した場合にも作成されます。リンクを解除すると、リンクオブジェクトは削除されます。また、リンクオブジェクトを削除することによって、リンクを解除することもできます。

3.4.2 リンクオブジェクトを構成する DocumentBroker オブジェクト

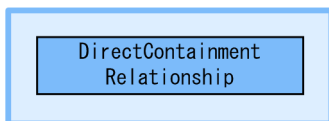
ここでは、リンクオブジェクトを構成する DocumentBroker オブジェクトについて説明します。

(1) 直接型リンクおよび参照型リンク

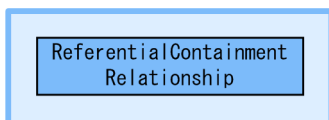
ここでは、フォルダを使用した関連付けの場合の、直接型リンクおよび参照型リンクについて説明します。直接型リンクおよび参照型リンクを表すリンクオブジェクトを構成する DocumentBroker オブジェクトを次の図に示します。

図 3-15 直接型リンクおよび参照型リンクを表すリンクオブジェクトを構成する DocumentBroker オブジェクト

リンクオブジェクト（直接型リンク）




リンクオブジェクト（参照型リンク）



（凡例）

 : DocumentBrokerオブジェクト

 : トップオブジェクト

各リンクオブジェクトを構成する DocumentBroker オブジェクトについて説明します。

DirectContainmentRelationship オブジェクト

直接型リンクを表すリンクオブジェクトのトップオブジェクトです。トップオブジェクトクラスは、`dmaClass_DirectContainmentRelationship` クラスまたはそのサブクラスです。

ReferentialContainmentRelationship オブジェクト

参照型リンクを表すリンクオブジェクトのトップオブジェクトです。トップオブジェクトクラスは、`dmaClass_ReferentialContainmentRelationship` クラスまたはそのサブクラスです。

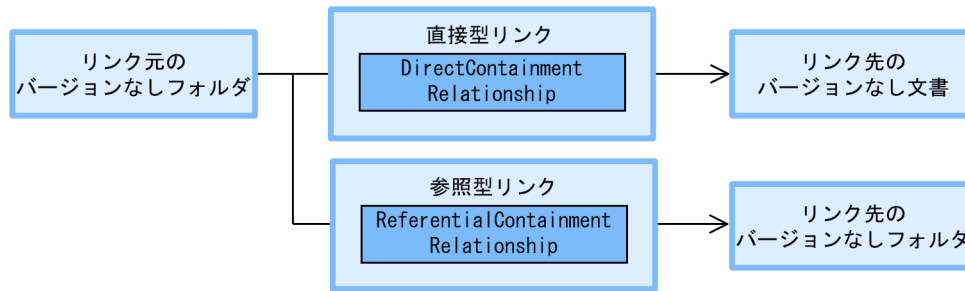
これらの DocumentBroker オブジェクトの識別にはリンク識別子を使用します。上位オブジェクトまたは下位オブジェクトとの関連付けを表すリンクオブジェクトの一覧の取得時に、リンク識別子を取得できません。

これらのリンクオブジェクトは、文書またはフォルダ（下位オブジェクト）の作成時に、同時にリンク付ける既存のフォルダ（上位オブジェクト）を指定した場合に作成されます。また、既存のフォルダ（上位オブジェクト）からほかのフォルダまたは文書（下位オブジェクト）に対してリンクを設定した場合にも作成されます。リンクオブジェクトは、上位オブジェクトまたはリンクオブジェクト自身のインターフェースで操作します。また、リンクオブジェクトには、プロパティも設定できます。

上位オブジェクトまたは下位オブジェクトを削除すると、設定されていたリンクオブジェクトも削除されます。ただし、リンクの種類によってはリンクオブジェクトが削除されない場合もあります。詳細については「(3) リンクの解除、リンク元文書およびリンク先文書の削除」を参照してください。

直接型リンクおよび参照型リンクは、フォルダと文書、フォルダとフォルダを関連付けます。直接型リンクおよび参照型リンクの設定例を次の図に示します。

図 3-16 直接型リンクおよび参照型リンクの設定例



(凡例)

- : DocumentBrokerオブジェクト
- : トップオブジェクト
- > : リンクによる関連付け

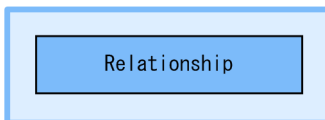
この例では、直接型リンクおよび参照型リンクを使用して、バージョンなしフォルダからほかの文書管理オブジェクトを関連付けています。

(2) 文書間リンク

ここでは、文書間を関連付ける場合の、文書間リンクについて説明します。文書間リンクを表すリンクオブジェクトを構成する DocumentBroker オブジェクトを次の図に示します。

図 3-17 文書間リンクを表すリンクオブジェクトを構成する DocumentBroker オブジェクト

リンクオブジェクト (文書間リンク)



(凡例)

- : DocumentBrokerオブジェクト
- : トップオブジェクト

文書間リンクを表すリンクオブジェクトを構成する DocumentBroker オブジェクトについて説明します。

Relationship オブジェクト

文書間リンクを表すリンクオブジェクトのトップオブジェクトです。トップオブジェクトクラスは、edmClass_Relationship クラスです。この DocumentBroker オブジェクトの識別にはリンク識別子を使用します。リンク元文書またはリンク先文書との関連付けを表すリンクオブジェクトの一覧の取得時に、リンク識別子を取得できます。

文書間リンクを表すリンクオブジェクトは、文書 (リンク元文書) の作成時に、同時に既存の文書 (リンク先文書) に対して文書間リンクを設定した場合に作成されます。また、既存の文書 (リンク元文書) からほかの文書 (リンク先文書) に対して文書間リンクを設定した場合にも作成されます。リンクオブジェクトは、リンク元文書またはリンクオブジェクト自身のインターフェースで操作します。また、リンクオブジェクトには、プロパティも設定できます。

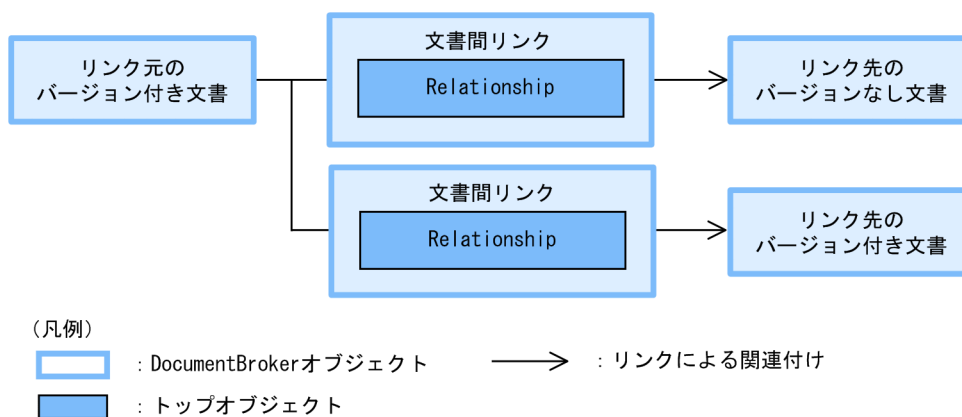
なお、文書間リンクを表すリンクオブジェクトは、リンク元文書に從属する DocumentBroker オブジェクトです。このため、リンク元文書を削除すると、設定されていたリンクオブジェクトも削除されますが、リンク先文書を削除しても、リンク元文書に設定されたリンクオブジェクトは削除されません。したがって、リンクを設定している文書を削除した場合は、リンク元文書からのリンクを解除することをお勧めします。

文書間リンクを表すリンクオブジェクトは、文書を表す次の DocumentBroker オブジェクト間の関連付けに使用できます。また、バージョン付き文書の 1 バージョンに対してリンクを設定することもできます。

- バージョンなし文書
- バージョン付き文書

文書間リンクの設定例を次の図に示します。

図 3-18 文書間リンクの設定例



この例では、文書間リンクを使用して、バージョン付き文書からほかの文書を関連付けています。

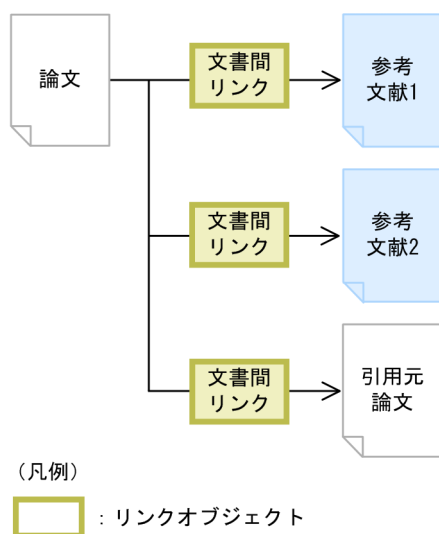
3.4.3 リンクによる文書管理

ここでは、文書間リンクを設定した文書管理について説明します。直接型リンク、参照型リンクを設定した文書管理については、「3.5 バージョンなしフォルダによる管理モデル」を参照してください。

(1) 文書間リンクの設定

文書間リンクを設定した文書の例を次の図に示します。

図 3-19 文書間リンクを設定した文書の例



この例では、「論文」をリンク元文書、「参考文献 1」、「参考文献 2」および「引用元論文」をリンク先文書として文書間リンクを設定しています。

3. 文書管理モデル

なお、設定したリンクオブジェクトは、リンク元文書が管理します。したがって、リンクオブジェクトのプロパティを設定したり、リンクを解除したりする場合は、リンク元文書またはリンクオブジェクト自身のインターフェースで操作します。

(2) 文書間リンクをたどる参照

文書間リンクによって関連付けられた文書は、双方向に参照できます。例えば、図 3-18 の場合に、リンク元文書「論文」からリンク先文書「参考文献 1」、「参考文献 2」および「引用元論文」の一覧を取得して参照できます。また、リンク先文書「参考文献 1」からリンク元文書「論文」を参照できます。

リンク元文書またはリンク先文書を参照する場合は、リンク元文書またはリンク先文書との関連付けを表すリンクオブジェクトの一覧を取得します。リンクオブジェクトをたどってリンク元文書またはリンク先文書を参照します。また、リンクオブジェクト取得時には、次の情報も取得できます。

- リンク元文書またはリンク先文書のオブジェクト種別
- リンク識別子
- リンク元文書またはリンク先文書のプロパティ
- リンクオブジェクトのプロパティ

また、一覧取得時には、すでにリンク先文書が削除されているリンクオブジェクトも取得できます。

(3) リンクの解除、リンク元文書およびリンク先文書の削除

ここでは、リンクの解除と、リンク元文書およびリンク先文書の削除の関係について説明します。

(a) リンクの解除

リンク元文書からリンクを解除するか、リンクオブジェクト自身のインターフェースでリンクオブジェクトを削除します。

(b) リンク元文書の削除

リンク元文書を削除すると、設定されていたリンクオブジェクトも削除されます。また、バージョン付き文書の 1 バージョンに当たるバージョンなし文書がリンク元文書の場合に、バージョン付き文書からそのバージョンを削除したときも、バージョンなし文書に設定していたリンクは解除されて、リンクオブジェクトも削除されます。ただし、リンク先文書は削除されません。

(c) リンク先文書の削除

リンク先文書を削除した場合、設定されていたリンクオブジェクトは削除されません。したがって、リンク元文書には、リンク先文書が削除されているリンクオブジェクトも残ります。この場合、リンク元文書からリンクオブジェクトを削除してください。

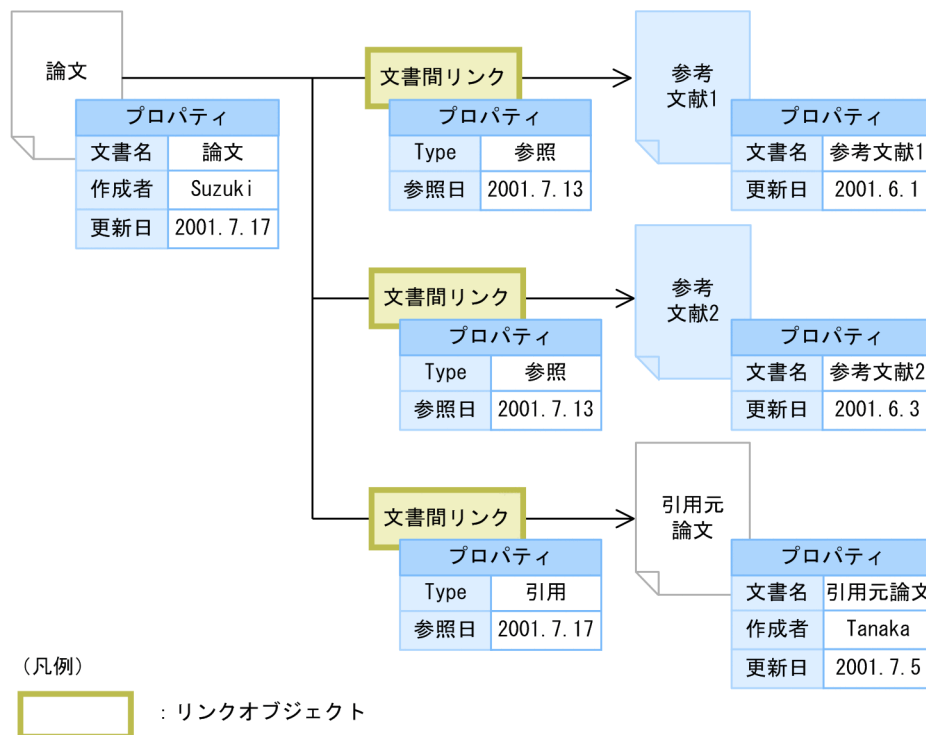
リンク先文書との関連付けを表すリンクオブジェクトの一覧を取得すれば、リンク先文書が削除されていないかを確認できます。

(4) 文書間リンクのプロパティの設定

リンク元文書、リンク先文書および文書のリンクには、ユーザ定義プロパティを設定して管理できます。リンクのプロパティとは、リンクオブジェクトに設定するプロパティです。リンクオブジェクトのプロパティは、リンク元文書やリンク先文書を分類したり、リンク元文書やリンク先文書のプロパティと組み合わせた管理に使用したりできます。リンクオブジェクトのプロパティは、リンク元文書またはリンクオブジェクト自身のインターフェースを使用して設定します。また、リンクオブジェクトのプロパティは検索に使用できます。検索では、ユーザ定義プロパティのほか、文書間リンクを表す Relationship オブジェクトのプロパティとして設定されている、DocumentBroker が規定したプロパティ (dmaProp_ または edmProp_ で始まる識別子を持つプロパティ) も使用できます。

プロパティを設定した文書間リンクの例を次の図に示します。

図 3-20 プロパティを設定した文書間リンクの例



この例では、リンクオブジェクトにユーザ定義プロパティとして、「Type」と「参照日」を設定しています。例えば、リンクオブジェクトのプロパティ「Type」によって、リンク先文書を「参照」と「引用」に分類したりできます。また、リンクオブジェクトのプロパティ「参照日」とリンク先文書のプロパティ「更新日」を比較して、参照後に文書が更新されたかどうかを確認したりできます。また、これらのプロパティを使用して、「リンクオブジェクトの『Type』が『参照』であり、『文書名』が『論文』である文書を検索する」というような edmSQL 検索を実行することもできます。edmSQL 検索については、「4. 検索機能」を参照してください。edmSQL 検索に指定する edmSQL の文法については、マニュアル「DocumentBroker Version 5 リファレンス API」を参照してください。

3.5 バージョンなしフォルダによる管理モデル

この節では、バージョンなしフォルダによる管理モデルについて説明します。

3.5.1 バージョンなしフォルダによる文書管理の概要

バージョンなしフォルダでは、複数の文書を目的に応じて一つにまとめたり、一つの文書を複数の分類に関連付けたりして文書を管理できます。また、バージョンなしフォルダを別のバージョンなしフォルダに関連付けて管理できます。

フォルダによる文書管理では、二つの文書管理オブジェクトを想定して、一方を「包含するオブジェクト」、もう一方を「包含されるオブジェクト」として考えます。このマニュアルでは、オブジェクトを包含するオブジェクトを上位オブジェクト、オブジェクトに包含されるオブジェクトを下位オブジェクトといいます。

例えば、バージョンなしフォルダに文書を格納することを考えた場合、バージョンなしフォルダが上位オブジェクト、文書が下位オブジェクトに相当します。また、バージョンなしフォルダの下位にバージョンなしフォルダを作成することを考えた場合、上位のバージョンなしフォルダが上位オブジェクト、下位のバージョンなしフォルダが下位オブジェクトに相当します。フォルダによる文書管理は、このような上位オブジェクトと下位オブジェクトとのリンク関係で表せます。リンクを解除すると、二つの文書管理オブジェクトの関連付けがなくなります。

バージョンなしフォルダでは、次の2種類のリンクで上位オブジェクトと下位オブジェクトをリンク付けます。

直接型リンク

上位オブジェクトと下位オブジェクトを $1:n$ (n は 1 以上の整数) の関係でリンク付けて、文書管理オブジェクトをまとめて管理できます。なお、上位のフォルダを下位以下のフォルダの下位のオブジェクトとして関連付けることはできません。

参照型リンク

上位オブジェクトと下位オブジェクトを $n:m$ (n, m は 1 以上の整数) の関係でリンク付けて、文書管理オブジェクトを分類して管理できます。

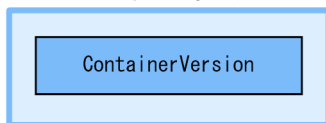
なお、バージョンなしフォルダの直接型リンクおよび参照型リンクでは、文書管理オブジェクト間をリンクオブジェクトによってリンク付けます。リンクオブジェクトについては、「3.4 リンクモデル」を参照してください。

3.5.2 バージョンなしフォルダを構成する DocumentBroker オブジェクト

ここでは、バージョンなしフォルダ構成する DocumentBroker オブジェクトについて説明します。バージョンなしフォルダを構成する DocumentBroker オブジェクトを次の図に示します。


図 3-21 バージョンなしフォルダを構成する DocumentBroker オブジェクト

バージョンなしフォルダ



(凡例)

 : DocumentBroker オブジェクト

 : トップオブジェクト

バージョンなしフォルダのトップオブジェクトクラスは、`dmaClass_Container` クラス、`edmClass_ContainerVersion` クラスまたはそのサブクラスです。

バージョンなしフォルダを構成する DocumentBroker オブジェクトについて説明します。

ContainerVersion オブジェクト

バージョンなしフォルダのトップオブジェクトです (トップオブジェクトクラスは、`edmClass_ContainerVersion` クラスまたはそのサブクラス)。

Container オブジェクト

バージョンなしフォルダのトップオブジェクトです (トップオブジェクトクラスは、`dmaClass_Container` クラス、または `edmClass_ContainerVersion` クラス以外の `dmaClass_Container` クラスのサブクラス)。

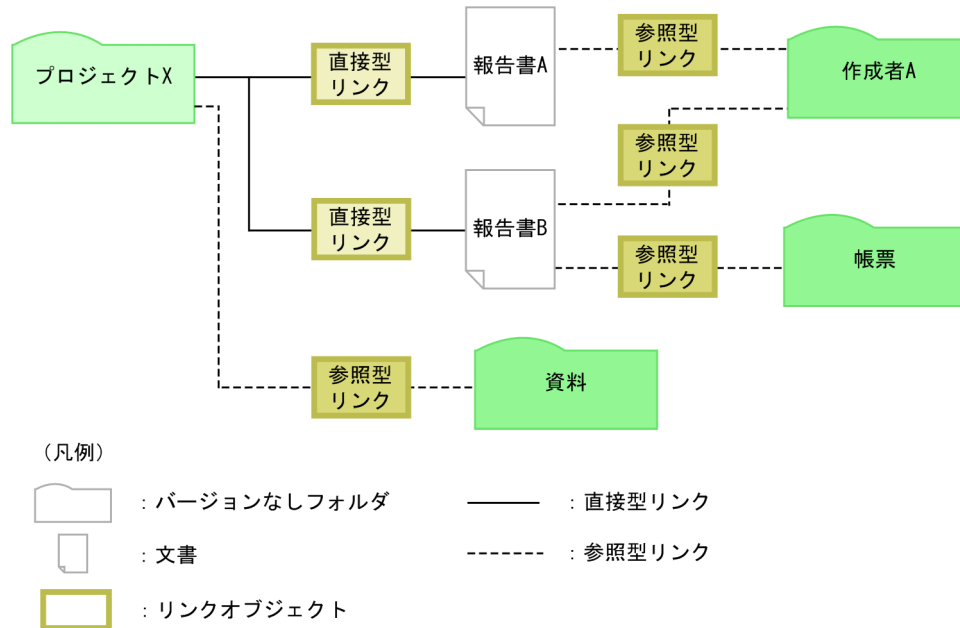
3.5.3 バージョンなしフォルダによる文書管理

ここでは、バージョンなしフォルダによる文書管理について説明します。

(1) リンクの設定

バージョンなしフォルダから文書およびフォルダに、直接型リンクおよび参照型リンクを設定した例を次の図に示します。

図 3-22 バージョンなしフォルダによるリンクの設定例



この例では、バージョンなしフォルダ「プロジェクト X」は、直接型リンクで文書をリンク付け、参照型リンクでフォルダをリンク付けています。直接型リンクでは、上位オブジェクトと下位オブジェクトの関係は 1:n になるため、例えば、文書「報告書 A」または「報告書 B」は、フォルダ「プロジェクト X」と別のフォルダに対して、直接型リンクで同時にリンク付けることはできません。このため、文書「報告書 A」または「報告書 B」は、参照型リンクを使用して、フォルダ「作成者 A」または「帳票」にリンク付けています。

(2) リンクをたどる文書管理オブジェクトの参照

リンクによって関連付けられた上位オブジェクトと下位オブジェクトは、双方向に参照できます。例えば、図 3-21 の場合は、フォルダ「プロジェクト X」からリンクを設定している文書「報告書 A」、「報告書 B」およびフォルダ「資料」を参照できます。また、文書「報告書 A」からフォルダ「プロジェクト X」および「作成者 A」を参照できます。

上位オブジェクトまたは下位オブジェクトを参照する場合は、上位オブジェクトまたは下位オブジェクトとの関連付けを表すリンクオブジェクトの一覧を取得します。リンクオブジェクトをたどって上位オブジェクトまたは下位オブジェクトを参照します。また、リンクオブジェクト取得時には、次の情報も取得できます。

- 上位オブジェクトまたは下位オブジェクトのオブジェクト種別
- リンク識別子
- 上位オブジェクトまたは下位オブジェクトのプロパティ
- リンクオブジェクトのプロパティ

(3) リンクの解除、上位オブジェクトおよび下位オブジェクトの削除

ここでは、リンクの解除と、上位オブジェクトおよび下位オブジェクトの削除の関係について説明します。

(a) リンクの解除

上位オブジェクトからリンクを解除するか、リンクオブジェクト自身のインターフェースでリンクオブジェクトを削除します。

(b) 上位オブジェクトの削除

上位オブジェクトを削除すると、設定されていたリンクオブジェクトも削除されます。

(c) 下位オブジェクトの削除

下位オブジェクトを削除すると、設定されていたリンクオブジェクトも削除されます。ただし、リンクの種類によってはリンクオブジェクトが削除されない場合もあります。詳細については「3.4.3(3) リンクの解除、リンク元文書およびリンク先文書の削除」を参照してください。

(4) バージョンなしフォルダのプロパティの設定

バージョンなしフォルダおよびバージョンなしフォルダのリンクには、ユーザ定義プロパティを設定して管理できます。リンクのプロパティとは、リンクオブジェクトに設定するプロパティです。フォルダのプロパティとして「作成者」などを設定しておくことで、フォルダの検索で使用できます。また、リンクオブジェクトのプロパティは、検索に使用したり、重要度などの値に応じてソートして、バージョンなしフォルダの下位オブジェクトに順序性を持たせたりできるので便利です。リンクオブジェクトのプロパティは、バージョンなしフォルダを指定して、上位オブジェクトまたはリンクオブジェクト自身のインターフェースを使用して設定します。

3.6 独立データ管理モデル

この節では、独立データ管理モデルについて説明します。

独立データは、独立したデータを扱う文書管理オブジェクトです。単なる表として扱いたいデータなどは独立データとして管理できます。独立データには、ユーザ定義プロパティを設定できます。

独立データは、文書のようにフォルダにリンク付けたり、バージョンを管理したり、フォルダのように上位・下位の階層構造で管理したりすることはできません。

3.6.1 独立データを構成する DocumentBroker オブジェクト


独立データを構成する DocumentBroker オブジェクトを次の図に示します。


図 3-23 独立データを構成する DocumentBroker オブジェクト

独立データ



(凡例)

 : DocumentBrokerオブジェクト

 : トップオブジェクト

独立データを構成する DocumentBroker オブジェクトについて説明します。

IndependentPersistence オブジェクト

独立データのトップオブジェクトです。トップオブジェクトクラスは、`edmClass_IndependentPersistence` クラスまたはそのサブクラスです。

3.7 属性管理モデル

この節では、属性管理モデルについて説明します。

DocumentBroker で管理する文書管理オブジェクトは、プロパティの値を DocumentBroker やユーザが設定することによって作成されます。文書管理オブジェクトはプロパティに設定された値によって区別されます。文書管理オブジェクトのプロパティの値を参照してその状態（作成者はだれか、作成日はいつかなど）を確認したり、プロパティをキーにして検索したりできます。

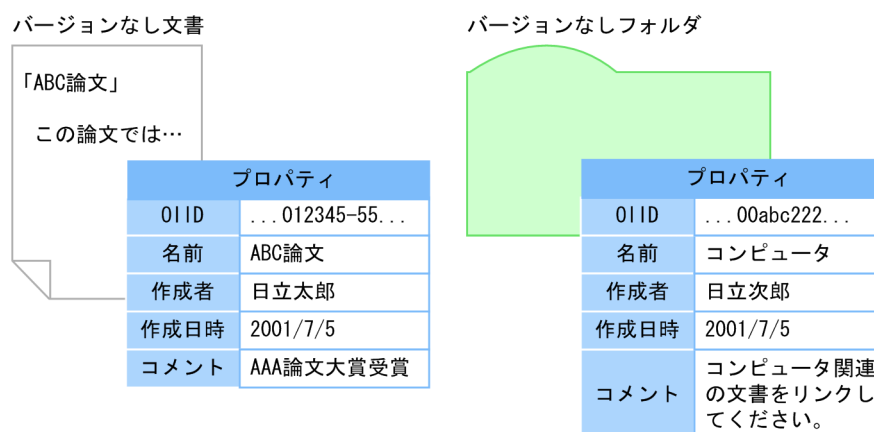
3.7.1 プロパティの管理例

ここでは、文書管理オブジェクトのプロパティとして、ユーザが設定できるプロパティの管理例を示します。

(1) バージョンなし文書およびバージョンなしフォルダのプロパティの管理例

バージョンなし文書およびバージョンなしフォルダのプロパティの管理例を次の図に示します。なお、バージョンを持たない独立データおよびパブリック ACL も、同様にプロパティを管理できます。

図 3-24 バージョンなし文書およびバージョンなしフォルダのプロパティの管理例

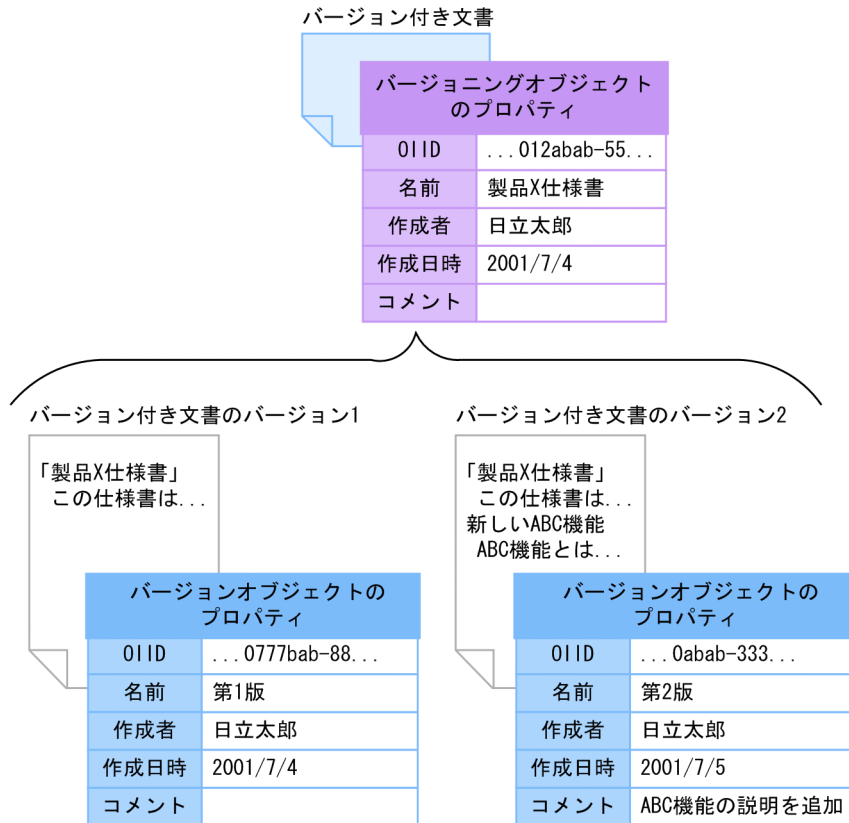


この例では、バージョンなし文書やバージョンなしフォルダには、文書管理オブジェクトの名前 (userProp_Name)、作成者 (userProp_Creator)、作成日時 (userProp_CreateTime) およびコメント (userProp_Comment) などを表すプロパティをユーザ定義プロパティとして追加定義しています。OIID を表すプロパティ (dmaProp_OIID) は DocumentBroker が規定したプロパティであり、文書管理オブジェクト登録時に DocumentBroker が設定します。このようにプロパティの値を設定しておくことによって、文書管理オブジェクトを名前で区別したり、設定されている値をキーに目的の文書管理オブジェクトを検索したりできます。例えば、「『作成日時』が『2001年7月以降』で、『作成者』は『日立太郎』の、『名前』に『論文』を含むバージョンなし文書」のような条件で検索できます。OIID は、各文書管理オブジェクトにユニークに与えられる ID であるため、文書管理オブジェクトを特定して DocumentBroker に操作を要求する場合に使用します。

(2) バージョン付きオブジェクトのプロパティの管理例

バージョン付きオブジェクトのプロパティの管理例を、バージョン付き文書の例で次の図に示します。

図 3-25 バージョン付き文書のプロパティの管理例



バージョン付きオブジェクトでは、複数バージョンの共通のプロパティをバージョンオブジェクトのプロパティとして、各バージョン固有のプロパティをバージョンオブジェクトのプロパティとして設定できます。この例では、バージョン付き文書の共通の名称として「名前」(userProp_Name)に「製品 X 仕様書」という値を設定しています。各バージョンには、バージョン固有の名称として「名前」(userProp_Name)に「第 1 版」、「第 2 版」という値を設定し、「第 2 版」には「コメント」(userProp_Comment)として「第 1 版」からの変更内容を設定して管理しています。

(3) VARRAY 型のプロパティの管理例

VARRAY 型のプロパティの管理例を、バージョンなし文書の例で次の図に示します。

図 3-26 VARRAY 型のプロパティの管理例

バージョンなし文書

「製品X販売企画」
この企画書は...

プロパティ	
OID	...0777bab-88...
名前	第1版
作成者	日立太郎
作成日時	2001/7/9
コメント	

執筆者	所属	社員番号	氏名
	営業	95012	日立太郎
	企画	97018	東京花子
	制作	98005	横浜次郎

この例では、VARRAY 型のプロパティとして「執筆者」というプロパティを定義して管理しています。VARRAY 型のプロパティでは、複数のプロパティを 1 組として、その組を複数個管理できます。要素となる複数のプロパティのデータ型は同一でなくてもかまいません。この例では、「所属」、「社員番号」および「氏名」という三つのプロパティを定義して、それらを 1 組にして「執筆者」プロパティの要素とし、複数人の「執筆者」の情報を管理しています。このとき、「所属」のデータ型が STR 型で、「社員番号」のデータ型が INT 型でもかまいません。

「所属」、「社員番号」および「氏名」のように VARRAY 型のプロパティの要素となるプロパティのことを VARRAY 型のプロパティの構成要素のプロパティといいます。

(4) LDAP 対応のディレクトリサービスとの連携を使用したプロパティの管理例

ユーザアプリケーションプログラムを開発する際に、JNDI を使用して LDAP 対応のディレクトリサービスと連携すれば、LDAP 対応のディレクトリサービスで管理されている情報を直接取得できます。

例えば、GUI に文書一覧などを表示する際、文書の所有者の情報をユーザ識別子で表示しても、人物を特定するのが困難です。この場合、ユーザ識別子でなく漢字の氏名などを表示するとわかりやすくなります。また、属性検索でも、所有者を検索条件にして検索する場合、ユーザ識別子を検索条件に指定するよりも、漢字の氏名を検索条件に指定する方がわかりやすくなります。

このような場合に、JNDI を使用して LDAP 対応のディレクトリサービスと連携していれば、ユーザ識別子やグループ識別子の値を検索条件にして、対応する漢字の氏名などのわかりやすい名前を、LDAP 対応のディレクトリサービスから検索して取得できます。ユーザ識別子やグループ識別子が設定されるプロパティについては、LDAP 対応のディレクトリサービスからユーザにわかりやすい名前を検索して、その名前を設定するプロパティを対応付けて定義および管理すると便利です。

LDAP 対応のディレクトリサービスとの連携については、ご使用になる LDAP 対応のディレクトリサービスのマニュアルを参照してください。

3.8 アクセス制御モデル

この節では、DocumentBroker でのアクセス制御モデルについて説明します。DocumentBroker では、ユーザがログインするときに生成されるユーザ情報や文書管理オブジェクトに設定されたアクセス制御情報などからユーザのアクセス権を判定してアクセスを制御します。

次のようにアクセスが制御されます。

1. ユーザが文書空間にログインします。
2. ログイン時にユーザ情報が生成されます。
ユーザ情報には、ログインユーザを識別する情報や、ログインユーザのユーザ権限や特権の有無についての情報が設定されます。
3. ログインユーザが操作を要求するとユーザ情報が参照されます。
 - ユーザ情報に特権を示す情報を保持している場合、アクセス権の判定は成功したものととして、要求が処理されます。
 - ユーザ権限として、要求した操作を許可する情報を保持している場合、要求が処理されます。
4. 特権または要求した操作を許可するユーザ権限がユーザ情報に保持されていない場合、操作対象の文書管理オブジェクトに設定されているアクセス制御情報が参照されます。
 - 操作対象の文書管理オブジェクトに、要求した操作をそのユーザに許可するアクセス制御情報が設定されている場合、要求が処理されます。
 - 操作対象の文書管理オブジェクトに、要求した操作をそのユーザに許可するアクセス制御情報が設定されていない場合、要求は処理されず、エラーが返却されます。

なお、ログイン時のユーザ認証に必要な情報、およびアクセス権の判定に必要なユーザ情報を取得する場合は、DocumentBroker を介してユーザ管理システムにアクセスします。この場合にどのユーザ管理システムを使用するかについては、DocumentBroker に定義されています。

ユーザ情報、特権およびユーザ権限の詳細については、「3.8.1 アクセス権の判定に使用される情報」を参照してください。

3.8.1 アクセス権の判定に使用される情報

ここでは、アクセス権の判定に使用される情報について説明します。

DocumentBroker では、ユーザのアクセス権の判定に次の情報を使用します。

- ユーザ情報
- 文書管理オブジェクトごとのアクセス制御情報

(1) ユーザ情報

DocumentBroker では、アクセス権の判定に、ユーザ情報を使用します。ユーザ情報は、文書空間にログインしたときに DocumentBroker のセッションオブジェクトに関連付けられて保持されます。ユーザ情報は、次の情報で構成されます。

- ユーザ識別子
- 所属グループ
- 特権
- ユーザ権限

それぞれについて説明します。

(a) ユーザ識別子

ユーザが文書空間にログインしたときに使用したユーザ管理システムから取得される、ログインユーザを識別する情報です。設定される値は、使用したユーザ管理システムによって異なります。ユーザ管理システムが LDAP 対応のディレクトリサービスの場合は、DocumentBroker の文書空間構成定義ファイルに定義した属性の値が設定されます。

DocumentBroker の文書空間構成定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。また、LDAP 対応のディレクトリサービスについては、ご使用になる LDAP 対応のディレクトリサービスのマニュアルを参照してください。

(b) 所属グループ

ユーザが文書空間にログインしたときに使用したユーザ管理システムから取得される、ログインユーザが所属するグループの数と、各グループを識別するためのグループ識別子の一覧情報です。設定される値は、使用したユーザ管理システムによって異なります。ユーザ管理システムが LDAP 対応のディレクトリサービスの場合は、DocumentBroker の文書空間構成定義ファイルに定義した属性の値が設定されます。

DocumentBroker の文書空間構成定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。また、LDAP 対応のディレクトリサービスについては、ご使用になる LDAP 対応のディレクトリサービスのマニュアルを参照してください。

(c) 特権

アクセス権の判定を受けないですべての文書管理オブジェクトに対して自由にアクセスできる権限のことを特権といいます。DocumentBroker では、すべての文書管理オブジェクトや、各文書管理オブジェクトに設定されたアクセス制御情報の保守を担当する、特権を持つユーザを設定できます。このユーザをセキュリティ管理者といいます。セキュリティ管理者は、DocumentBroker の文書空間構成定義ファイルに定義しておきます。ログイン時に文書空間構成定義ファイルが参照され、ログインするユーザがセキュリティ管理者の場合、ユーザ情報に「特権のあるセキュリティ管理者」を示す値が設定されます。ログインユーザがセキュリティ管理者でない場合は「特権なし」を示す値が設定されます。

セキュリティ管理者の定義方法については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

(d) ユーザ権限

ユーザ権限とは、DocumentBroker のユーザ権限定義ファイルに定義した情報を基に、ユーザ、グループまたはすべてのユーザに対して設定できる権限です。ユーザ権限定義ファイルには、すべての文書管理オブジェクトに対するアクセス権を定義します。

ユーザ権限には次の 2 種類があります。

- オブジェクト作成権限
- オブジェクト操作権限

オブジェクト作成権限

オブジェクト作成権限とは、文書空間にオブジェクトを作成する権利を任意のユーザまたはグループに対して許可するアクセス制御情報です。ログイン時にユーザ権限定義ファイルが参照されて、ログインユーザにオブジェクト作成権が設定されている場合、ユーザ情報に「オブジェクトを作成する権利」を示すパーミッションが設定されます。

オブジェクト作成権限の定義方法については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

オブジェクト操作権限

オブジェクト操作権限とは、文書空間内のすべてのオブジェクトに対してどの範囲の操作を許可するかを定義するアクセス制御情報です。オブジェクト操作権限は、任意のユーザまたはグループごとに設定できます。例えば、「ユーザ A にプロパティを参照する権利を与える」と定義しておく、「ユーザ A」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。ログイン時にユーザ権限定義ファイルが参照されて、ログインするユーザにオブジェクト操作権限としてプロパティ参照権が定義されていた場合、ユーザ情報に「プロパティを参照する権利」を示すパーミッションが設定されます。

オブジェクト操作権限の設定については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

(2) 文書管理オブジェクトごとのアクセス制御情報

文書管理オブジェクトに対するアクセスを制御するための情報です。文書管理オブジェクトごとに「どのユーザまたはグループに」、「どのような操作を許可するか」を表すアクセス制御情報を設定しておくことで、各ユーザまたはグループに対して文書管理オブジェクトごとのアクセス権を設定できます。文書管理オブジェクトのアクセス制御情報の種類については、「3.8.3 文書管理オブジェクトごとのアクセス制御情報の種類」を参照してください。

アクセス制御情報を設定できる文書管理オブジェクトのオブジェクト種別は、次のとおりです。

- バージョンなし文書
- バージョン付き文書
- バージョン付き文書のバージョン（バージョンなし文書）
- バージョンなしフォルダ
- パブリック ACL
- 独立データ

3.8.2 アクセス制御情報に設定できるパーミッションの種類

ここでは、アクセス制御情報に設定できるパーミッションの種類について説明します。

「どのような操作を許可するか」を表す値をパーミッションといいます。例えば、ある文書に対して、「ユーザ A に、プロパティの参照だけを許可する」、「ユーザ B に、プロパティの参照およびコンテンツの更新を許可する」のように設定して、許可する操作の範囲を限定できます。

(1) パーミッションの種類

アクセス制御情報として設定するパーミッションの種類を次の表に示します。

表 3-9 パーミッションの種類

パーミッションの用途	パーミッションの種類
個々の文書管理オブジェクトに対する操作を許可する	基本パーミッション 組み合わせパーミッション
アクセス制御情報に対する操作を許可する	アクセス制御情報変更権
文書管理オブジェクトの作成を許可する	オブジェクト作成権

各パーミッションについて説明します。

基本パーミッションと組み合わせパーミッション

個々の文書管理オブジェクトに対する操作を許可するパーミッション（ACFlag および ACL に指定するパーミッション）です。パーミッションは複数指定できます。例えば、複数の基本パーミッションを指定したり、複数の組み合わせパーミッションを指定したり、基本パーミッションと組み合わせ

パーミッションを混在させて指定したりできます。この場合、パーミッションとして設定される内容は、指定したパーミッションの論理和になります。

基本パーミッションの詳細については、「(2) 基本パーミッション」を参照してください。組み合わせパーミッションの詳細については、「(3) 組み合わせパーミッション」を参照してください。また、ACFlag および ACL については、「3.8.3 文書管理オブジェクトごとのアクセス制御情報の種類」を参照してください。

アクセス制御情報変更権

アクセス制御情報に対する操作を許可するパーミッションです。セキュリティ ACL だけに指定します。アクセス制御情報変更権およびセキュリティ ACL については、「3.8.4 アクセス制御情報の管理」を参照してください。

オブジェクト作成権

文書管理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。ユーザ権限のオブジェクト作成権限を定義するときに、ユーザ権限定義ファイルに指定します。パーミッションを表す定数は「DbjDef.PERM_CREATE」です。ユーザ権限定義ファイルで定義するパーミッションについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

なお、パーミッションを個々の文書管理オブジェクトに設定する方法については、「3.8.3 文書管理オブジェクトごとのアクセス制御情報の種類」を参照してください。

(2) 基本パーミッション

基本パーミッションは、文書管理オブジェクトを操作する範囲を限定するために、DocumentBroker が提供する基本的なパーミッションです。基本パーミッションは、ユーザ権限のオブジェクト操作権限の定義およびオブジェクトごとのアクセス制御情報の定義で指定します。基本パーミッションの種類を次の表に示します。

表 3-10 基本パーミッションの種類

パーミッションの種類	説明
基本プロパティ参照権 (DbjDef.PERM_PRIM_READ_PROPS)	文書管理オブジェクトのプロパティを参照する権利を与える基本パーミッションです。 このパーミッションを設定されたユーザは、文書やフォルダのOID、ユーザ定義プロパティを参照できます。プロパティを指定した属性検索を実行する場合も、このパーミッションが必要です。また、このパーミッションを設定されないユーザに対しては、その文書管理オブジェクトの存在もわからないようにできます。 このパーミッションには、フォルダに設定されたリンクオブジェクトのプロパティを参照したり、フォルダとリンク付けられている文書管理オブジェクトのプロパティを参照したりする権利も含まれません。
基本プロパティ更新権 (DbjDef.PERM_PRIM_WRITE_PROPS)	文書管理オブジェクトのプロパティを更新する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。 このパーミッションを設定されたユーザは、文書やフォルダに設定されたユーザ定義プロパティ、そのほかの読み取り専用以外のプロパティを更新できます。
基本コンテンツ参照権 (DbjDef.PERM_PRIM_READ_CONTENTS)	文書のコンテンツを参照する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。 全文検索インデクスを使用した検索を実行する場合は、このパーミッションが必要です。
基本コンテンツ更新権 (DbjDef.PERM_PRIM_WRITE_CONTENTS)	文書のコンテンツを更新する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。 全文検索インデクスを作成および削除する場合は、このパーミッションが必要です。

パーミッションの種類	説明
基本リンク権 (DbjDef.PERM_PRIM_LINK)	リンクに関する操作を実行する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。リンクに関する操作とは、リンク（直接型リンク、参照型リンク）を設定および解除する権利です。このパーミッションには、リンクオブジェクトを削除する権利、リンクオブジェクトのプロパティを設定および更新する権利も含まれます。
基本バージョン管理権 (DbjDef.PERM_PRIM_VERSION)	バージョン管理に関する操作を実行する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。バージョン管理に関する操作とは、バージョン付きオブジェクトのバージョンを追加および削除する権利です。
基本削除権 (DbjDef.PERM_PRIM_DELETE)	文書管理オブジェクトを削除する権利を与える基本パーミッションです。基本プロパティ参照権を含みます。

注 「DbjDef.PERM_」で始まる定数は、パーミッションを表す定数です。

基本パーミッションでは、それぞれのパーミッションは独立であり、包含関係はありません。ただし、基本プロパティ参照権は、すべての基本パーミッションに含まれます。

基本パーミッションは、複数組み合わせで使用できます。また、組み合わせパーミッションを使用すると、複数の基本パーミッションを指定した場合と同じパーミッションが設定できます。この場合、パーミッションとして設定される内容は、指定したパーミッションの論理和になります。組み合わせパーミッションについては、「(3) 組み合わせパーミッション」を参照してください。

基本パーミッションの設定例

基本パーミッションの設定例を示します。

プロパティの参照を可能にする場合

次のパーミッションを設定します。

- 基本プロパティ参照権

フォルダの階層をたどったり、フォルダ名、文書名などの文書管理オブジェクトのプロパティの値を参照したりできます。

プロパティとコンテンツの参照を可能にする場合

次のパーミッションを設定します。

- 基本コンテンツ参照権

フォルダの階層をたどったり、フォルダ名、文書名などの文書管理オブジェクトのプロパティの値を参照したり、コンテンツをダウンロードしたりできます。

プロパティとコンテンツの更新を可能にする場合

次のパーミッションを設定します。

- 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

文書またはフォルダのプロパティとコンテンツを参照可能にした場合の操作に加えて、プロパティ、コンテンツの更新が可能になります。

削除を可能にする場合

次のパーミッションを設定します。

- 基本削除権

文書またはフォルダのプロパティの参照と削除ができます。使用されていない文書またはフォルダを削除してデータベースを保守するユーザなどに与えるパーミッションです。

リンクを可能にする場合

次のパーミッションを設定します。

- 基本リンク権

フォルダに文書，またはほかのフォルダをリンク付けたり，文書間リンクを設定したりできます。あるユーザがフォルダ体系を決定し，別のユーザには，そのフォルダ体系に従って文書をリンク付けたり，文書間リンクを設定させたりしたい場合，文書のリンク付けや文書間リンクの設定だけをするユーザには基本リンク権だけを設定します。

バージョンアップを可能にする場合

バージョン付きオブジェクトのバージョンオブジェクトに次のパーミッションを設定します。

- 基本バージョン管理権

また，バージョン付きオブジェクトのバージョンオブジェクトに次のパーミッションを設定します。

- 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

バージョン付きオブジェクトをバージョンアップし，追加したバージョンのバージョン名を設定したり，バージョン付き文書の場合はコンテンツを更新したりできます。

なお，文書管理オブジェクトごとにアクセス制御情報を設定する場合，文書管理オブジェクトのオブジェクト種別によって，アクセス制御情報に設定できるパーミッションの種類が異なります。パーミッションを設定する対象オブジェクトと基本パーミッションの対応を次の表に示します。

表 3-11 パーミッションを設定する対象オブジェクトと基本パーミッションの対応

基本パーミッションの種類	オブジェクト種別			
	D	VD	F	IP
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	×		×	×
基本削除権				

(凡例)

D：バージョンなし文書およびバージョン付き文書のバージョンオブジェクト

VD：バージョン付き文書

F：バージョンなしフォルダのバージョンオブジェクト

IP：独立データ

：設定できることを示します。

×：設定できないことを示します。

(3) 組み合わせパーミッション

組み合わせパーミッションは，基本パーミッションを組み合わせで定義されたパーミッションです。ある文書管理オブジェクトに対するパーミッションを設定する場合に，一般的に同時に指定すると考えられる複数の基本パーミッションが，まとめて一つのパーミッションとして指定できるように定義されています。組み合わせパーミッションの種類を次の表に示します。

3. 文書管理モデル

表 3-12 組み合わせパーミッションの種類

パーミッションの種類	説明
プロパティ参照権 (DbjDef.PERM_READ_PROPS)	文書管理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。 このパーミッションは、文書管理オブジェクトのプロパティを参照したり、階層をたどったりする操作を許可するユーザに対して設定します。
プロパティ更新権 (DbjDef.PERM_WRITE_PROPS)	文書管理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。 このパーミッションは、文書管理オブジェクトのプロパティを参照して、そのプロパティの値の更新を許可するユーザに対して設定します。
参照権 (DbjDef.PERM_READ)	文書管理オブジェクトを参照する権利を与えるパーミッションです。 このパーミッションは、文書管理オブジェクトのプロパティを参照した上で、コンテンツの参照も許可するユーザに対して設定します。
参照更新権 (DbjDef.PERM_READ_WRITE)	文書管理オブジェクトを参照および更新する権利を与えるパーミッションです。 このパーミッションは、文書管理オブジェクトのプロパティやコンテンツの参照および更新を許可するユーザに対して設定します。
削除権 (DbjDef.PERM_DELETE)	文書管理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。 このパーミッションは、文書管理オブジェクトのプロパティを参照した上で、その文書管理オブジェクトの削除も許可するユーザに対して設定します。 使用されていない文書を削除してデータベースを保守するユーザなどに設定します。
リンク権 (DbjDef.PERM_LINK)	文書やフォルダのリンクを設定する権利を与えるパーミッションです。 このパーミッションは、文書やフォルダの名前などは変更させないで、文書やフォルダのリンクの設定を許可するユーザに対して設定します。 すでにフォルダの体系が決定されている場合に、あるユーザにその体系に従って文書を格納させたいときなどに設定します。
バージョン権 (DbjDef.PERM_VERSION)	バージョンを管理する権利を与えるパーミッションです。 バージョンの管理を許可するユーザに対して設定します。ただし、バージョン付きオブジェクトをバージョンアップする場合などには、バージョン管理の対象になる文書またはフォルダに対してこのパーミッションを与えると同時に、個々のバージョンに対応する文書またはフォルダに対して参照更新権が必要です。
フルコントロール (DbjDef.PERM_FULL_CONTROL)	文書管理オブジェクトに対するすべての操作を実行する権利を与えるパーミッションです。 このパーミッションは、文書管理オブジェクトの参照、更新および削除などのすべての操作を許可するユーザに対して設定します。

注 「DbjDef.PERM_」で始まる定数は、パーミッションを表す定数です。

組み合わせパーミッションと基本パーミッションの対応を次の表に示します。

表 3-13 組み合わせパーミッションと基本パーミッションの対応

組み合わせパーミッションの種類	対応する基本パーミッションの種類						
	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	基本削除権
プロパティ参照権		-	-	-	-	-	-
プロパティ更新権			-	-	-	-	-
参照権		-		-	-	-	-
参照更新権					-	-	-
削除権		-	-	-	-	-	
リンク権		-	-	-		-	-

組み合わせパーミッションの種類	対応する基本パーミッションの種類						
	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	基本削除権
バージョン権					-		-
フルコントロール							

(凡例)

- ：組み合わせパーミッションに対応する基本パーミッションです。
- ：組み合わせパーミッションに対応しない基本パーミッションです。

組み合わせパーミッションを使用すると、一つの組み合わせパーミッションを指定することによって、複数の基本パーミッションを指定した場合と同じパーミッションが設定できます。

例えば、あるユーザに対して文書のコンテンツの更新を許可したい場合、運用によっては、「検索によって文書を取得し、既存のコンテンツの内容を参照した上で、修正、更新する」という操作の流れが考えられます。また、「コンテンツを更新する場合には、更新した日時を表すプロパティを更新する」という操作があるとします。この場合、基本パーミッションを使用すると、「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」の3種類のパーミッションをそのユーザに設定する必要があります。これに対して、組み合わせパーミッションを使用すると、「参照更新権」を指定するだけで、3種類の基本パーミッションを指定した場合と同じパーミッションが設定できます。

また、組み合わせパーミッションと基本パーミッションを組み合わせで指定することもできます。例えば、「参照更新権」を与えたユーザに対して、「文書管理オブジェクトを削除する権利も与えたい」という場合は、組み合わせパーミッションの「参照更新権」と、基本パーミッションの「基本削除権」をあわせて指定できます。この場合、パーミッションとして設定される内容は、指定したパーミッションの論理和になります。

3.8.3 文書管理オブジェクトごとのアクセス制御情報の種類

ここでは、アクセス制御情報のうち、文書管理オブジェクトごとに設定するアクセス制御情報について説明します。文書管理オブジェクトごとに設定するアクセス制御情報の種類を次に示します。

- 所有者
- プライマリグループ
- ACFlag
- ACL (ローカル ACL, パブリック ACL, セキュリティ ACL)

文書管理オブジェクトごとのアクセス制御情報は、ACFlag または ACL として設定します。文書管理オブジェクトごとに設定されたアクセス制御情報を管理するための情報も、ACL として設定します。また、文書管理オブジェクトごとに設定する所有者およびプライマリグループのアクセス権は、ACFlag として設定します。

ACFlag として設定する情報、ACL として設定する情報について説明します。

(1) ACFlag として設定する情報

ACFlag は、文書管理オブジェクトの所有者、プライマリグループおよびすべてのユーザという区切りで、それぞれにパーミッションを設定できるアクセス制御情報です。

ACFlag は、文書管理オブジェクトのプロパティとして直接設定します。したがって、ACL を使用したアクセス権の判定に比べて、高速なアクセス権の判定ができるのが特長です。ACFlag は、次の三つのプロ

パティで表されます。

- dbrProp_OwnerPermission プロパティ
所有者のアクセス権を設定するプロパティです。
- dbrProp_PrimaryGroupPermission プロパティ
プライマリグループに所属するユーザのアクセス権を設定するプロパティです。
- dbrProp_EveryonePermission プロパティ
すべてのユーザのアクセス権を設定するプロパティです。

プロパティの詳細については、「3.8.5(2) アクセス制御情報を設定するためのプロパティ」を参照してください。

ACFlag によってアクセス権を設定できる所有者、プライマリグループおよびすべてのユーザについて説明します。

所有者

アクセス制御機能を使用している場合、文書管理オブジェクトには必ず所有者が存在します。所有者とは、文書管理オブジェクトに設定されているアクセス制御情報を変更できるユーザです。文書管理オブジェクト作成時には、作成したユーザが所有者として登録されます。所有者以外のユーザがアクセス制御情報を変更できるようにしたい場合、まず所有者によってアクセス制御情報を変更する権利を与える必要があります。また、所有者は、所有者自身を変更できます。

ACFlag では、アクセス制御情報と所有者の変更以外に、所有者が実行できる操作を設定できます。

所有者は、dbrProp_OwnerId プロパティに設定されたユーザ識別子によって表されます。

dbrProp_OwnerId プロパティの値は、文書管理オブジェクト作成時に設定されます。所有者またはセキュリティ管理者は、dbrProp_OwnerId プロパティの値を変更することによって、所有者を変更できます。なお、所有者がアクセス制御情報やセキュリティ ACL の値を参照する場合は、基本プロパティ参照権が必要です。セキュリティ ACL については、「3.8.4 アクセス制御情報の管理」を参照してください。

プライマリグループ

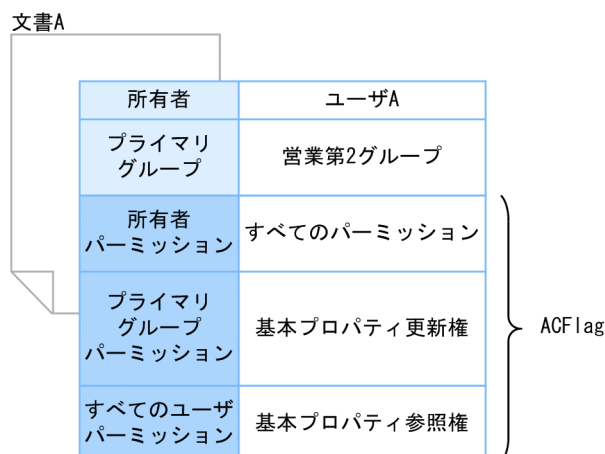
ACFlag でパーミッションを設定してアクセス権を与える一つのグループです。プライマリグループは、dbrProp_PrimaryGroupId プロパティに設定されたグループ識別子によって表されます。文書管理オブジェクト作成時にこのプロパティに設定される値は、ユーザが文書空間にログインしたときに使用したユーザ管理システムによって異なります。ユーザ管理システムが LDAP 対応のディレクトリサービスの場合は、文書管理オブジェクト作成時には設定されません。文書管理オブジェクトに頻繁にアクセスするグループのグループ識別子をユーザが設定します。ユーザ作成のアクセスルーチンの場合は、文書管理オブジェクト作成時に所有者が所属するグループ（ユーザ作成のアクセスルーチンのグループ識別子）が設定されます。

すべてのユーザ

所有者、プライマリグループに所属するユーザ以外のすべてのユーザです。

ACFlag の概要を次の図に示します。

図 3-27 ACFlag の概要



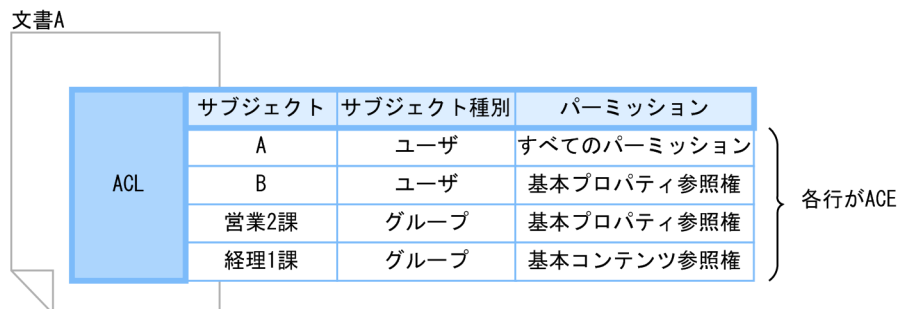
この例では、「ユーザ A」が作成した「文書 A」を ACFlag によってアクセス制御する場合のパーミッションの設定方法を示しています。このように ACFlag を設定すると、所有者である「ユーザ A」にすべてのアクセス権、プライマリグループに設定されている「営業第 2 グループ」に所属するユーザに基本プロパティ更新権（基本プロパティ参照権を含む）、そのほかのすべてのユーザに基本プロパティ参照権だけを設定するというアクセス制御が可能になります。

(2) ACL として設定する情報

ACL は、任意のユーザまたは任意のグループごとにパーミッションを設定してアクセス権を与えるためのリストです。

ACL の概要を次の図に示します。

図 3-28 ACL の概要



この例のように、ACL には、複数の ACE（アクセス制御エレメント）を設定できます。ACE は 64 個まで設定できます。個々の ACE には「アクセス権を与えるユーザまたはグループ」と「アクセスの範囲を決定するパーミッション」を対にして設定します。アクセス権を与えるユーザまたはグループのことを特にサブジェクトといいます。この例では、サブジェクトとしてユーザ A、ユーザ B、営業 2 課および経理 1 課を指定して、それぞれにパーミッションを設定しています。

ACL を使用すると、複数のユーザやグループに対して個別にパーミッションを設定してアクセス権を与えられますが、アクセス権の判定には、ACFlag の場合よりも時間がかかります。

DocumentBroker では、目的に応じて次の 2 種類の ACL を使用して、アクセス制御情報を設定できます。

- ローカル ACL

3. 文書管理モデル

- パブリック ACL

また、次の ACL を使用して、アクセス制御情報に対する操作を許可するアクセス制御情報変更権を管理できます。

- セキュリティ ACL

ここでは、ローカル ACL およびパブリック ACL について説明します。アクセス制御情報変更権およびセキュリティ ACL については、「3.8.4 アクセス制御情報の管理」を参照してください。

(a) ローカル ACL

文書管理オブジェクトごとのアクセス権を設定するための ACL です。個々の文書管理オブジェクトの `dbrProp_ACL` プロパティとして設定します。図 3-27 で示したように、ローカル ACL は、ACE を表すオブジェクトを構成要素とした、VARRAY 型のプロパティです。`dbrProp_ACL` プロパティについては、「3.8.5(2) アクセス制御情報を設定するためのプロパティ」を参照してください。

(b) パブリック ACL

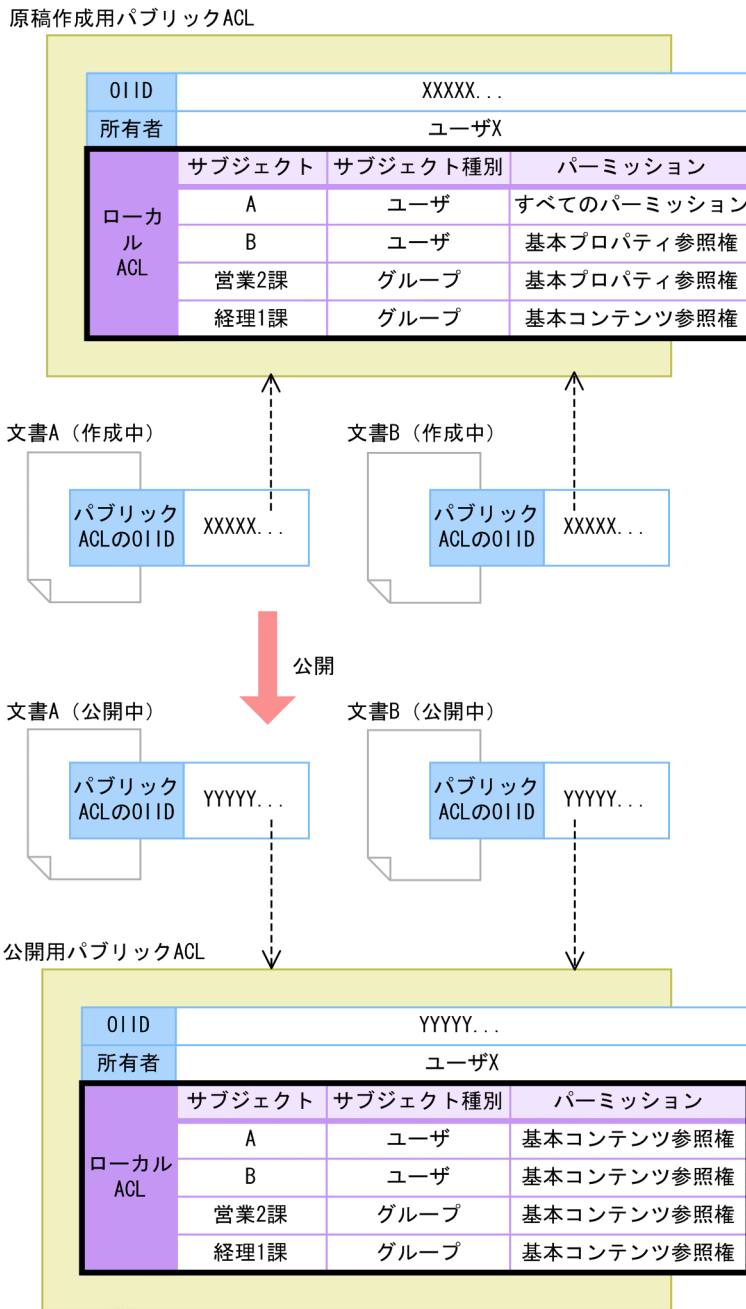
アクセス制御情報の一括管理に使用できる ACL です。パブリック ACL を使用すると、複数の文書管理オブジェクト間でアクセス制御情報を共有できます。

パブリック ACL は、ACL を共有するために使用する独立した文書管理オブジェクトです。共有したい ACL をパブリック ACL に設定して、文書やフォルダなどのほかの文書管理オブジェクトから参照させます。これによって、パブリック ACL に設定した ACL の値を、複数の文書管理オブジェクトに一括して適用させることができます。パブリック ACL は一つの文書管理オブジェクトに対して 10 個まで設定できます。

なお、文書やフォルダを表す文書管理オブジェクトからパブリック ACL を参照することをバインドといいます。また、パブリック ACL のバインドを解除することをアンバインドといいます。

パブリック ACL の概要を次の図に示します。

図 3-29 パブリック ACL の概要



(凡例)

----> : バインド

この例では、作成中の段階には文書 A と文書 B から「原稿作成用パブリック ACL」をバインドさせて、原稿を公開する段階になった時点で、「公開用パブリック ACL」をバインドさせるように切り替えてアクセス制御しています。

パブリック ACL は、複数の文書管理オブジェクトに同じ ACL を設定して、一括してアクセス制御する場合に便利です。また、パブリック ACL のローカル ACL の設定を更新すると、そのパブリック ACL をバインドしているすべての文書管理オブジェクトに更新が反映されるため、グループ名やグループを構成するメンバの変更が頻繁にある場合の保守が容易になります。

パブリック ACL は文書管理オブジェクトとして作成します。パブリック ACL の文書管理オブジェクトクラスは、パブリック ACL クラスです。パブリック ACL として、複数の文書管理オブジェクト間で共有するために設定するアクセス制御情報は、パブリック ACL の `dbrProp_ACL` プロパティとして設定します。

パブリック ACL にユーザ定義プロパティを設定して、検索で使用できます。パブリック ACL のプロパティについては、「3.8.5(4) パブリック ACL のプロパティ」を参照してください。

なお、パブリック ACL には、自身をアクセス制御するための `ACFlag` やローカル ACL はありません。パブリック ACL に対するアクセス権を次の表に示します。

表 3-14 パブリック ACL に対するアクセス権

実行できる操作	対 象
作成	文書空間にオブジェクト作成権限を持つユーザ
参照	すべてのユーザ
ローカル ACL とユーザ定義プロパティの変更	セキュリティ ACL でアクセス制御情報変更権が与えられているユーザ、パブリック ACL の所有者およびセキュリティ管理者
セキュリティ ACL の変更	パブリック ACL の所有者およびセキュリティ管理者
削除	パブリック ACL の所有者およびセキュリティ管理者

3.8.4 アクセス制御情報の管理

ここでは、アクセス制御情報の管理について説明します。

DocumentBroker では、`ACFlag` や `ACL` を使用して、文書管理オブジェクトに対するユーザのアクセスを制御します。文書管理オブジェクトに対してアクセス権を保持しないユーザでも、`ACFlag` や `ACL` を変更すればアクセス権を取得できます。このため、`ACFlag` や `ACL` は、適切な権利を持つユーザだけが変更でき、それ以外のユーザからは変更されないように制御する必要があります。アクセス制御情報を変更するための権利をアクセス制御情報変更権といいます。DocumentBroker では、デフォルトの状態では、文書管理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アクセス制御情報変更権を持っていますが、所有者およびセキュリティ管理者が必要に応じて、アクセス制御情報変更権を任意のユーザやグループに対して与えることもできます。

(1) アクセス制御情報へのアクセス権の設定

文書管理オブジェクトごとに「ユーザ A およびグループ B に」、「アクセス制御情報を変更する権利を与える」のように設定して、その文書管理オブジェクトのアクセス制御情報を変更する権利を任意のユーザまたはグループに対して与えられます。アクセス制御情報変更権を設定・更新できるのは、その文書管理オブジェクトの所有者およびセキュリティ管理者だけです。ただし、所有者が、設定されている値を参照する場合は、基本プロパティ参照権が必要です。

(a) アクセス制御情報変更権 (DbjDef.PERM_CHANGE_PERM)

アクセス制御情報変更権は、アクセス制御情報に対する操作を許可するパーミッションです。アクセス制御情報変更権は、セキュリティ ACL に設定します。

アクセス制御情報変更権を与えられたユーザまたはグループは、基本プロパティ更新権を与えられていなくても、文書管理オブジェクトの `ACFlag` の値の更新、ローカル ACL の値の更新および文書管理オブジェクトのパブリック ACL へのバインド・アンバインドの権利を持ちます。ただし、値を参照する場合は、基本プロパティ参照権が必要です。なお、バインドされているパブリック ACL の内容を更新する権利は、パブリック ACL の所有者、パブリック ACL のセキュリティ ACL でアクセス制御情報変更権を設

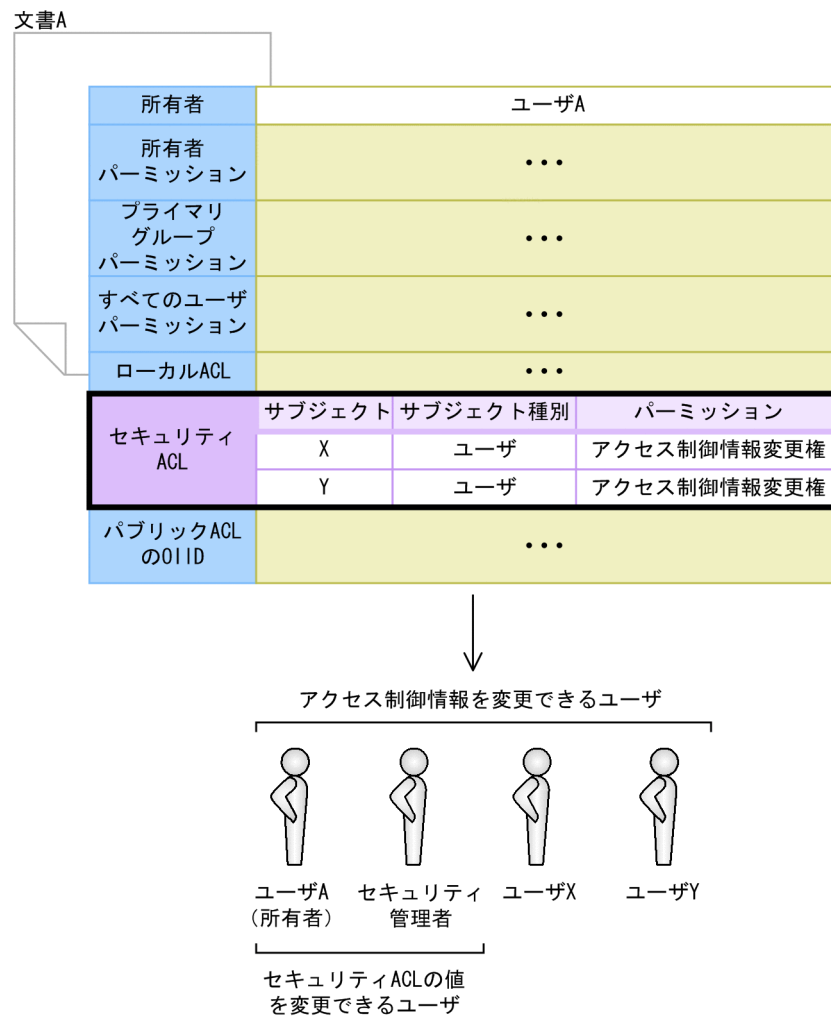
定されているユーザおよびセキュリティ管理者にあります。

(b) セキュリティ ACL

文書管理オブジェクトに設定されている ACFlag の更新，ローカル ACL の各 ACE の値の更新およびパブリック ACL へのバインド・アンバインドを実行するためのパーミッション（アクセス制御情報変更権）を設定する ACL です。セキュリティ ACL は，ACE を表すオブジェクトを構成要素とした，VARRAY 型のプロパティです。個々の文書管理オブジェクトの dbrProp_SACL プロパティとして設定します。dbrProp_SACL プロパティについては、「3.8.5(2) アクセス制御情報を設定するためのプロパティ」を参照してください。

セキュリティ ACL の ACE のパーミッションとして指定できるのはアクセス制御情報変更権だけです。また，セキュリティ ACL を設定できるのは，その文書管理オブジェクトの所有者およびセキュリティ管理者だけです。セキュリティ ACL を使用するとオブジェクトの所有者以外にもアクセス制御情報変更権を与えることができます。セキュリティ ACL の概要を次の図に示します。

図 3-30 セキュリティ ACL の概要



(凡例)

... : セキュリティACLに設定されたユーザが変更できるようになる値

この例では，文書Aの所有者であるユーザAがセキュリティACLを使用して，ユーザXとユーザYにも

3. 文書管理モデル

アクセス制御情報変更権を与えています。このため、文書 A の所有者であるユーザ A、セキュリティ管理者、ユーザ X およびユーザ Y の 4 人が、文書 A のアクセス制御情報を変更できます。ただし、セキュリティ ACL の値を変更できるのは、文書 A の所有者であるユーザ A およびセキュリティ管理者の 2 人だけです。

なお、パブリック ACL のセキュリティ ACL でアクセス制御情報変更権が設定されているユーザまたはグループは、パブリック ACL 内のローカル ACL の設定・更新、およびユーザ定義プロパティの値の設定・更新ができます。ただし、値を参照する場合は、基本プロパティ参照権が必要です。

(c) セキュリティ管理者の設定

DocumentBroker では、アクセス権の判定を受けずに文書空間のすべてのオブジェクトに対して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や、個々の文書管理オブジェクトに設定されたアクセス制御情報の保守を担当するユーザを設定できます。このユーザをセキュリティ管理者といいます。セキュリティ管理者は、DocumentBroker の文書空間構成定義ファイルに定義しておきます。文書空間構成定義ファイルの定義方法については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

3.8.5 アクセス制御モデルで使用するプロパティ

ここでは、アクセス制御モデルで使用するプロパティについて説明します。

(1) ユーザ情報を表すプロパティ

アクセス制御モデルでは、ユーザがログインした際に生成されるユーザ情報からログインユーザを識別したり、DocumentBroker で定義されているログインユーザが保持するユーザ権限の情報を取得したりできます。ユーザ情報には、次のものがあります。

ユーザ識別子

所属グループ

- 所属するグループの数
- 所属するグループのグループ識別子の一覧

特権

ユーザ権限

各情報の詳細については、「3.8.1(1) ユーザ情報」を参照してください。それぞれのユーザ情報には、プロパティの名前が付けられています。ユーザ情報を表すプロパティの一覧を次の表に示します。

表 3-15 ユーザ情報を表すプロパティの一覧

意味	プロパティの名前	データ型
ユーザ識別子	dbrProp_UserId	STR 型
所属するグループの数	dbrProp_GroupCount	INT 型
所属するグループのグループ識別子の一覧	dbrProp_GroupList	STRLIST 型
特権	dbrProp_UserPrivilege	INT 型
ユーザ権限	dbrProp_UserPermission	INT 型

(2) アクセス制御情報を設定するためのプロパティ

個々の文書管理オブジェクトにアクセス制御情報を設定するためのプロパティを次の表に示します。なお、

VARRAY 型のプロパティについては、構成要素のプロパティを「VARRAY 型のプロパティの名前：構成要素のプロパティの名前」と表記します。

表 3-16 アクセス制御情報を設定するためのプロパティの一覧

意味		プロパティの名前	データ型	制限値	値を更新できるユーザ
所有者のユーザ識別子		dbrProp_OwnerId	STR 型	1 ~ 254 バイト	所有者 ³ セキュリティ管理者
プライマリグループのグループ識別子		dbrProp_PrimaryGroupId	STR 型	0 ~ 254 バイト	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
ACFlag の所有者のパーミッション		dbrProp_OwnerPermission	INT 型	4 バイト ³	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
ACFlag のプライマリグループのパーミッション		dbrProp_PrimaryGroupPermission	INT 型	4 バイト ³	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
ACFlag のすべてのユーザのパーミッション		dbrProp_EveryonePermission	INT 型	4 バイト ³	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
ローカル ACL	サブジェクト	dbrProp_ACL:dbrProp_Subject	STR 型	ACE は 64 個まで	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
	サブジェクト種別	dbrProp_ACL:dbrProp_SubjectType	INT 型		
	パーミッション	dbrProp_ACL:dbrProp_Permission	INT 型		
セキュリティ ACL	サブジェクト	dbrProp_SACL:dbrProp_Subject	STR 型	ACE は 64 個まで	所有者 ¹ セキュリティ管理者
	サブジェクト種別	dbrProp_SACL:dbrProp_SubjectType	INT 型		
	パーミッション	dbrProp_SACL:dbrProp_Permission	INT 型		
バインドしているパブリック ACL の個数		dbrProp_PublicACLCount	INT 型	10 個	- ⁴
バインドしているパブリック ACL の OIID のリスト	バインドしているパブリック ACL の OIID	dbrProp_PublicACLIds:dbrProp_ACLIdElem	STR 型	- ⁵	所有者 ¹ セキュリティ管理者 DbjDef.PERM_CHANG E_PERM ^{1 2}
ログインユーザのパーミッション		dbrProp_UserPermission	INT 型	-	- ⁴

(凡例) - : 該当しないことを示します。

注 1

値を更新する場合、基本プロパティ更新権は必要ありません。ただし、値を参照する場合は、基本プロパティ参照権が必要です。

注 2

セキュリティ ACL でアクセス制御情報変更権 (DbjDef.PERM_CHANGE_PERM) を与えられているユーザを示します。

注 3

パーミッションを表す定数を指定します。パーミッションに指定する値については、「3.8.2 アクセス制御情報に設定できるパーミッションの種類」を参照してください。

注 4

読み取り専用のプロパティです。

注 5

一つの文書管理オブジェクトにバインドできるパブリック ACL は 10 個までになります。また、個々のパブリック ACL で設定できる ACE は 64 個までになります。

各プロパティの詳細について説明します。

所有者のユーザ識別子 (dbrProp_OwnerId)

文書管理オブジェクト作成時に、作成したユーザのユーザ識別子が設定されるプロパティです。このプロパティに設定されているユーザ識別子のユーザには、所有者として、ACFlag でアクセス権を与えることができます。また、所有者は、基本プロパティ更新権の有無に関係なく、その文書管理オブジェクトのアクセス制御情報およびアクセス制御情報変更権を設定・更新できます。ただし、値を参照する場合は、基本プロパティ参照権が必要です。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。また、属性検索条件にも指定できます。例えば、「A さんが所有する文書を検索する」という条件で検索できます。ただし、検索条件に指定する場合は、「edmProp_OwnerId」という名前で指定してください。

このプロパティの値を更新できるのは、所有者およびセキュリティ管理者だけです。所有者およびセキュリティ管理者以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

プライマリグループのグループ識別子 (dbrProp_PrimaryGroupId)

ACFlag でアクセス権を与えるグループのグループ識別子を設定するプロパティです。設定できるグループは一つだけです。このプロパティに設定されているグループ識別子のグループは、その文書管理オブジェクトに対して「ACFlag のプライマリグループのパーミッション」で設定された範囲のアクセス権を保持します。ユーザ管理システムが LDAP 対応のディレクトリサービスの場合は、オブジェクトを作成したあとに、その文書管理オブジェクトの所有者またはセキュリティ管理者がユーザ情報のグループ識別子の中から一つ選択して、設定してください。ユーザ作成のアクセスルーチンの場合は、文書管理オブジェクトを作成するときにユーザ作成のアクセスルーチンに記述されているグループ識別子が設定されます。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。また、属性検索条件にも指定できます。例えば、「プライマリグループが A グループの文書を検索する」という条件で検索できます。ただし、検索条件に指定する場合は、「edmProp_PrimaryGroupId」という名前で指定してください。

このプロパティの値を更新できるのは、所有者およびセキュリティ管理者だけです。所有者およびセキュリティ管理者以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

ACFlag の所有者のパーミッション (dbrProp_OwnerPermission)

「所有者のユーザ識別子」に設定されているユーザに対して与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。文書管理オブジェクト作成時には、文書空間構成定義ファイルに設定されているデフォルトのパーミッションが設定されます。文書空間構成定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索条件には指定できません。

このプロパティの値を更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

ACFlag のプライマリグループのパーミッション (dbrProp_PrimaryGroupPermission)

「プライマリグループのグループ識別子」に設定されているグループに対して与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。文書管理オブジェクト作成時には、文書空間構成定義ファイルに設定されているデフォルトのパーミッションが設定されます。文書空間構成定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

「プライマリグループのグループ識別子」が設定されていない場合は、このプロパティに値を設定しても無視されます。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索条件には指定できません。

このプロパティの値を更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

ACFlag のすべてのユーザのパーミッション (dbrProp_EveryonePermission)

すべてのユーザに対して与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。文書管理オブジェクト作成時には、文書空間構成定義ファイルに設定されているデフォルトのパーミッションが設定されます。文書空間構成定義ファイルについては、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索条件には指定できません。

このプロパティの値を更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

ローカル ACL (dbrProp_ACL)

ローカル ACL を設定するプロパティです。ローカル ACL は「サブジェクト」、「サブジェクト種別」および「パーミッション」という三つの要素を持つ ACE を複数管理する VARRAY 型のプロパティです。ローカル ACL の構成要素である ACE のプロパティについて説明します。

サブジェクト (dbrProp_ACL:dbrProp_Subject)

ローカル ACL の ACE でアクセス権を与えるユーザの識別子、グループの識別子またはシステムサブジェクトを表す定数を設定するプロパティです。

サブジェクト種別 (dbrProp_ACL:dbrProp_SubjectType)

ローカル ACL の ACE でアクセス権を与えるユーザまたはグループの種別を設定するプロパティ

です。

パーミッション (dbrProp_ACL:dbrProp_Permission)

「サブジェクト」に設定したユーザまたはグループに与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。

これらのプロパティに設定する値の詳細については、「(3) アクセス制御情報の構成要素のプロパティ」を参照してください。

基本プロパティ参照権を持つユーザは、これらのプロパティの値を参照できます。ただし、属性検索条件には指定できません。また、これらのプロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、これらのプロパティの値を更新できません。

セキュリティ ACL (dbrProp_SACL)

セキュリティ ACL を設定するプロパティです。セキュリティ ACL は「サブジェクト」、「サブジェクト種別」および「パーミッション」という三つの要素を持つ ACE を複数管理する VARRAY 型のプロパティです。セキュリティ ACL の構成要素である ACE のプロパティについて説明します。

サブジェクト (dbrProp_SACL:dbrProp_Subject)

セキュリティ ACL の ACE でアクセス制御情報変更権のパーミッションを設定してアクセス制御情報へのアクセス権を与えるユーザの識別子、グループの識別子またはシステムサブジェクトを表す定数を設定するプロパティです。

サブジェクト種別 (dbrProp_SACL:dbrProp_SubjectType)

セキュリティ ACL の ACE でアクセス制御情報変更権のパーミッションを設定してアクセス制御情報へのアクセス権を与えるユーザまたはグループの種別を設定するプロパティです。

パーミッション (dbrProp_SACL:dbrProp_Permission)

「サブジェクト」に指定したユーザまたはグループに与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。このプロパティに設定できる値は、アクセス制御情報変更権を表す定数「DbjDef.PERM_CHANGE_PERM」です。

「サブジェクト」および「サブジェクト種別」に設定する値の詳細については「(3) アクセス制御情報の構成要素のプロパティ」を参照してください。

基本プロパティ参照権を持つユーザは、これらのプロパティの値を参照できます。ただし、属性検索条件には指定できません。また、これらのプロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、これらのプロパティの値を更新できません。

バインドしているパブリック ACL の個数 (dbrProp_PublicACLCount)

文書管理オブジェクトがバインドしているパブリック ACL の個数が設定されるプロパティです。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索条件には指定できません。

バインドしているパブリック ACL の OIID のリスト (dbrProp_PublicACLIds)

文書管理オブジェクトがバインドしているパブリック ACL の OIID のリストを表す VARRAY 型のプロパティです。構成要素であるパブリック ACL の OIID のプロパティについて説明します。

バインドしているパブリック ACL の OIID (dbrProp_PublicACLIds:dbrProp_ACLIdElem)

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索

条件には指定できません。

このプロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、このプロパティの値を更新できません。

ログインユーザのパーミッション (dbrProp_UserPermission)

文書管理オブジェクトに対するログインユーザのアクセス権の範囲を定めるパーミッションが設定されるプロパティです。文書管理オブジェクトの ACFlag, ローカル ACL およびバインドしているパブリック ACL のアクセス制御情報から、文書管理オブジェクトに操作要求を出した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アクセス権の範囲を定めるパーミッションの論理和が求められて設定されます。

(3) アクセス制御情報の構成要素のプロパティ

ここでは、アクセス制御情報のうち、ACL の構成要素である ACE のプロパティについて説明します。

ACE のプロパティを次の表に示します。

表 3-17 ACE のプロパティ

意味	プロパティの名前	データ型	制限値
サブジェクト	dbrProp_Subject	STR 型	サブジェクト種別に応じて制限値が異なります。 <ul style="list-style-type: none"> ユーザ識別子の場合 1 ~ 254 (単位はバイト) グループ識別子の場合 1 ~ 254 (単位はバイト) システムサブジェクトの場合 1 ~ 254 バイト
サブジェクト種別	dbrProp_SubjectType	INT 型	4 バイト
パーミッション	dbrProp_Permission	INT 型	4 バイト

(a) サブジェクト (dbrProp_Subject)

次のどれかを指定します。

ユーザ識別子

ユーザ単位でパーミッションを設定する場合に指定します。

グループ識別子

グループ単位でパーミッションを設定する場合に指定します。

システムサブジェクトを表す定数

システムサブジェクトに対してパーミッションを設定する場合に指定します。

システムサブジェクトは、DocumentBroker によって設定された情報に対してパーミッションを設定する場合に指定します。このサブジェクトは、主にパブリック ACL に指定して使用します。

システムサブジェクトの定数と意味を次の表に示します。

表 3-18 システムサブジェクトの定数と意味

定数	意味
DbjDef.SYSSUBJECT_SELF	対象の文書管理オブジェクトの所有者を表すサブジェクトです。 <ul style="list-style-type: none"> パブリック ACL の ACL に設定した場合、そのパブリック ACL をバインドしている文書やフォルダを表す文書管理オブジェクトの所有者を示します。 文書やフォルダを表す文書管理オブジェクトのローカル ACL およびセキュリティ ACL に設定した場合は、その文書管理オブジェクトの所有者を示します。 パブリック ACL のセキュリティ ACL に設定した場合は、パブリック ACL の所有者を示します。
DbjDef.SYSSUBJECT_EVERYONE	すべてのユーザを表すサブジェクトです。

(b) サブジェクト種別 (dbrProp_SubjectType)

「サブジェクト」に指定した内容に応じて、種別を示す定数を指定します。

ユーザ識別子を指定した場合

定数「DbjDef.SUBJECTTYPE_USR」を指定します。

グループ識別子を指定した場合

定数「DbjDef.SUBJECTTYPE_GRP」を指定します。

システムサブジェクトを表す定数を指定した場合

定数「DbjDef.SUBJECTTYPE_SYS」を指定します。

(c) パーミッション (dbrProp_Permission)

「サブジェクト」に指定したユーザまたはグループに対するパーミッションを表す定数を指定します。

パーミッションに指定する値については、「3.8.2 アクセス制御情報に設定できるパーミッションの種類」を参照してください。

(4) パブリック ACL のプロパティ

パブリック ACL のプロパティを次の表に示します。なお、VARRAY 型のプロパティについては、構成要素のプロパティを「VARRAY 型のプロパティの名前：構成要素のプロパティの名前」と表記します。

表 3-19 パブリック ACL のプロパティの一覧

意味		プロパティの名前	データ型	制限値	値を更新できるユーザ
オブジェクトの識別子		dmaProp_OIID	STR 型	-	- 1
所有者のユーザ識別子		dbrProp_OwnerId	STR 型	1 ~ 254 バイト	所有者 セキュリティ管理者
ローカル ACL	サブジェクト	dbrProp_ACL:dbrProp_Subject	STR 型	ACE は 64 個まで	所有者 セキュリティ管理者 DbjDef.PERM_CHANGE_PERM ²
	サブジェクト種別	dbrProp_ACL:dbrProp_SubjectType	INT 型		
	パーミッション	dbrProp_ACL:dbrProp_Permission	INT 型		
セキュリティ ACL	サブジェクト	dbrProp_SACL:dbrProp_Subject	STR 型	ACE は 64 個まで	所有者 セキュリティ管理者
	サブジェクト種別	dbrProp_SACL:dbrProp_SubjectType	INT 型		

意味	プロパティの名前	データ型	制限値	値を更新できるユーザ
パーミッション	dbrProp_SACL:dbrProp_Permission	INT 型		
パブリック ACL をバインドしている文書管理オブジェクトの個数	dbrProp_BindObjectCount	INT 型	-	- 1
ログインユーザのパーミッション	dbrProp_UserPermission	INT 型	-	- 1
ユーザ定義プロパティ	(ユーザ定義プロパティの識別子)	-	-	所有者 セキュリティ管理者 DbjDef.PERM_CHANGE_PERM E_PERM ²

(凡例) - : 該当しないことを示します。

注 1

読み取り専用のプロパティです。

注 2

セキュリティ ACL でアクセス制御情報変更権 (DbjDef.PERM_CHANGE_PERM) を与えられているユーザを示します。

各プロパティの詳細について説明します。

オブジェクトの識別子 (dmaProp_OIID)

パブリック ACL を識別するための OIID 文字列が設定されるプロパティです。パブリック ACL 作成時に、DocumentBroker が値を設定します。この OIID を使用して、パブリック ACL をほかの文書管理オブジェクトからバインドさせたり、すでに作成されているパブリック ACL を指定してプロパティの値を設定・参照したりできます。また、属性検索条件にも指定できます。

所有者のユーザ識別子 (dbrProp_OwnerId)

パブリック ACL を作成したユーザのユーザ識別子が設定されるプロパティです。このプロパティにユーザ識別子が設定されているユーザは、アクセス権の有無に関係なく、次の操作ができます。

- パブリック ACL のアクセス制御情報の設定・更新
- アクセス制御情報変更権の設定・更新
- ユーザ定義プロパティの設定・更新
- パブリック ACL の削除

このプロパティの値は、属性検索条件にも指定できます。例えば、「Aさんが所有するパブリック ACL を検索する」という条件で検索できます。ただし、検索条件に指定する場合は、「edmProp_OwnerId」という名前で指定してください。

このプロパティの値を設定・更新できるのは、所有者およびセキュリティ管理者だけです。

ローカル ACL (dbrProp_ACL)

複数の文書管理オブジェクト間で共有したいアクセス制御情報を設定するプロパティです。ローカル ACL は「サブジェクト」、「サブジェクト種別」および「パーミッション」という三つの要素を持つ ACE を複数管理する VARRAY 型のプロパティです。ローカル ACL の構成要素である ACE のプロパティについて説明します。

サブジェクト (dbrProp_ACL:dbrProp_Subject)

アクセス権を与えるユーザの識別子、グループの識別子またはシステムサブジェクトを表す定数を設定するプロパティです。

サブジェクト種別 (dbrProp_ACL:dbrProp_SubjectType)

アクセス権を与えるユーザまたはグループの種別を設定するプロパティです。

パーミッション (dbrProp_ACL:dbrProp_Permission)

「サブジェクト」に指定したユーザまたはグループに与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。

これらのプロパティに設定する値の詳細については、「(3) アクセス制御情報の構成要素のプロパティ」を参照してください。

基本プロパティ参照権を持つユーザは、これらのプロパティの値を参照できます。ただし、属性検索条件には指定できません。また、これらのプロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、これらのプロパティの値を更新できません。

セキュリティ ACL (dbrProp_SACL)

パブリック ACL のローカル ACL およびユーザ定義プロパティの値を更新する権利を設定するプロパティです。このプロパティでアクセス制御情報変更権を与えられたユーザまたはグループは、パブリック ACL 内のローカル ACL の設定・更新およびユーザ定義プロパティの設定・更新ができます。セキュリティ ACL は「サブジェクト」、「サブジェクト種別」および「パーミッション」という三つの要素を持つ ACE を複数管理する VARRAY 型のプロパティです。セキュリティ ACL の構成要素である ACE のプロパティについて説明します。

サブジェクト (dbrProp_SACL:dbrProp_Subject)

パブリック ACL のローカル ACL およびユーザ定義プロパティの値を更新する権利を与えるユーザの識別子、グループの識別子またはシステムサブジェクトを表す定数を設定するプロパティです。

サブジェクト種別 (dbrProp_SACL:dbrProp_SubjectType)

パブリック ACL を更新する権利を与えるユーザまたはグループの種別を設定するプロパティです。

パーミッション (dbrProp_SACL:dbrProp_Permission)

「サブジェクト」に指定したユーザまたはグループに与えるアクセス権の範囲を定めるパーミッションを設定するプロパティです。このプロパティに設定できる値は、アクセス制御情報変更権を表す定数「DbjDef.PERM_CHANGE_PERM」です。アクセス制御情報変更権を設定されたユーザまたはグループは、パブリック ACL のローカル ACL の設定・更新およびユーザ定義プロパティの設定・更新ができます。

「サブジェクト」および「サブジェクト種別」に設定する値の詳細については、「(3) アクセス制御情報の構成要素のプロパティ」を参照してください。

基本プロパティ参照権を持つユーザは、これらのプロパティの値を参照できます。ただし、属性検索条件には指定できません。また、これらのプロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザ以外のユーザは、基本プロパティ更新権があっても、これらのプロパティの値を更新できません。

パブリック ACL をバインドしている文書管理オブジェクトの個数 (dbrProp_BindObjectCount)

パブリック ACL をバインドしている文書管理オブジェクトの個数が設定されるプロパティです。

基本プロパティ参照権を持つユーザは、このプロパティの値を参照できます。ただし、属性検索条件には指定できません。

ログインユーザのパーミッション (dbrProp_UserPermission)

パブリック ACL に対するログインユーザのアクセス権の範囲を定めるパーミッションが設定されるプロパティです。セキュリティ ACL から、パブリック ACL に操作要求を出した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アクセス権の範囲を定めるパーミッションの論理和が求められて設定されます。

ユーザ定義プロパティ

パブリック ACL クラスには、ユーザ定義プロパティを追加定義できます。

ユーザ定義プロパティは、属性検索条件としても指定できます。

ユーザ定義プロパティの値を設定・更新できるのは、所有者、セキュリティ管理者およびセキュリティ ACL でアクセス制御情報変更権を設定されているユーザです。

3.8.6 オブジェクトの操作とアクセス制御情報の関係

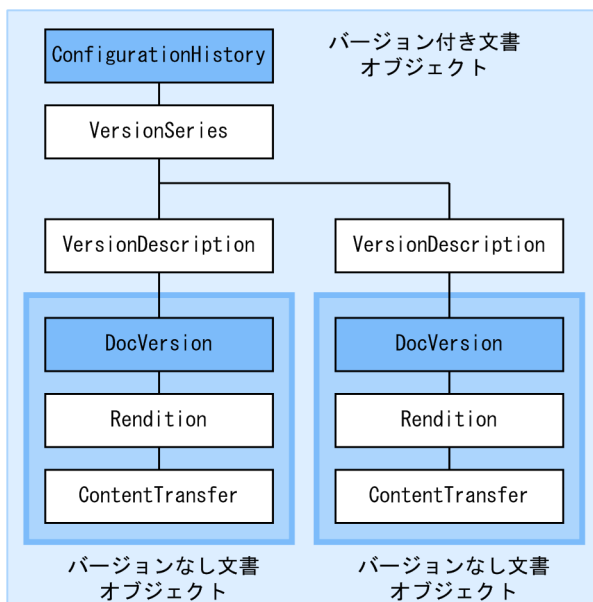
ここでは、オブジェクトの操作とアクセス制御情報の関係について説明します。

(1) バージョン管理に関する操作

バージョン管理をしている文書の場合、一つのバージョン付き文書は、複数のバージョンなし文書によって構成されています。一つのバージョン付き文書を操作する場合、バージョン付き文書のトップオブジェクトに設定されたパーミッションと、バージョンなし文書のトップオブジェクトに設定されたパーミッションが必要となることがあります。

バージョン付きオブジェクトとバージョンなしオブジェクトに設定されているアクセス制御情報について、次の図に示します。

図 3-31 バージョン付きオブジェクトとバージョンなしオブジェクトに設定されているアクセス制御情報



(凡例)

: アクセス制御情報が設定されているオブジェクト (トップオブジェクト)

ここでは、バージョン付き文書をバージョン付きオブジェクト、バージョンなし文書をバージョンなしオブジェクトとして、これら二つのオブジェクトにパーミッションが必要な操作について説明します。

3. 文書管理モデル

バージョンなし文書オブジェクトは、バージョン付き文書オブジェクトの一つのバージョンに相当します。ユーザ A がバージョン付き文書オブジェクトの 1 バージョンを削除しようとした場合、まず、バージョン付き文書オブジェクトに対して基本バージョン管理権が必要です。しかし、バージョンを削除することは実際には一つのバージョンなし文書オブジェクトを削除することと同じこととなります。したがって、この場合は、バージョン付き文書オブジェクトの基本バージョン管理権のほかに、バージョンなし文書オブジェクトの基本オブジェクト削除権が必要になります。

バージョン付きオブジェクトを操作する場合に必要なパーミッション

バージョン付きオブジェクトに対してバージョン管理に関する操作をする時、操作の種類によってはバージョンなしオブジェクトに対してもアクセス権が必要な場合があります。

メソッドをコールするユーザがバージョン付きオブジェクトに対して必要なパーミッションとバージョンなしオブジェクトに対して必要なパーミッションを次の表に示します。

表 3-20 バージョン付きオブジェクトとバージョンなしオブジェクトに必要なパーミッション（バージョン付きオブジェクトを操作する場合）

バージョン管理に関する操作（メソッド）	バージョン付きオブジェクトに必要なパーミッション	バージョンなしオブジェクトに必要なパーミッション
バージョンチェックアウト • checkOut	基本バージョン管理権 (PRIM_VERSION)	文書オブジェクトの場合： 基本コンテンツ参照権 (PRIM_READ_CONTENTS ¹)
バージョンチェックイン • checkIn	基本バージョン管理権 (PRIM_VERSION)	-
バージョンチェックアウトのキャンセル • cancelCheckOut	基本バージョン管理権 (PRIM_VERSION)	-
バージョンの削除 • deleteVersion	基本バージョン管理権 (PRIM_VERSION)	基本オブジェクト削除権 (PRIM_DELETE)
バージョン情報の一覧取得 • getVersionObjList	基本プロパティ参照権 (PRIM_READ_PROPS)	基本プロパティ参照権 ² (PRIM_READ_PROPS)
バージョン識別子を指定したバージョンなしオブジェクトの操作	基本プロパティ参照権 (PRIM_READ_PROPS)	バージョンなしオブジェクトに直接メソッドをコールした場合に必要なパーミッション

（凡例）

- : パーミッションは必要ありません。

注 1

カレントバージョンに対応するバージョンなしオブジェクトの場合に必要です。

注 2

バージョンなしオブジェクトのプロパティを取得しない場合は不要です。また、バージョンなしオブジェクトのプロパティを取得する場合に、基本プロパティ参照権が設定されていないオブジェクトについては、一覧として取得されません。

バージョンなしオブジェクトを操作する場合に必要なパーミッション

バージョン付きオブジェクトの 1 バージョンに相当するバージョンなしオブジェクトを操作する時、操作の種類によってはバージョン付きオブジェクトに対してもアクセス権が必要な場合があります。

メソッドをコールするユーザがバージョン付きオブジェクトに対して必要なパーミッションとバージョンなしオブジェクトに対して必要なパーミッションを次の表に示します。

表 3-21 バージョン付きオブジェクトとバージョンなしオブジェクトに必要なパーミッション（バージョンなしオブジェクトを操作する場合）

バージョンなしオブジェクトに対する操作（メソッド）	バージョン付きオブジェクトに必要なパーミッション	バージョンなしオブジェクトに必要なパーミッション
オブジェクトの削除 • removeObject	基本バージョン管理権 (PRIM_VERSION)	基本オブジェクト削除権 (PRIM_DELETE)
自オブジェクトを1バージョンとして包含しているバージョン付きオブジェクトの一覧取得 • getVersioningInfo	基本プロパティ参照権 (PRIM_READ_PROPS)	基本プロパティ参照権 (PRIM_READ_PROPS)

注

バージョン付きオブジェクトのプロパティを取得しない場合は不要です。また、バージョン付きオブジェクトのプロパティを取得する場合に、基本プロパティ参照権が設定されていないオブジェクトについては、一覧として取得されません。

バージョンなしオブジェクトに対して表 3-21 に示した以外のメソッドをコールする場合、バージョン付きオブジェクトのパーミッションは必要ありません。

チェックアウトによってコピーされたオブジェクトのパーミッション
checkOut メソッドをコールすると、カレントバージョンに該当するバージョンなしオブジェクトがコピーされて、仮のバージョンに該当するバージョンなしオブジェクトが作成されます。
このとき、仮のバージョンに該当するバージョンなしオブジェクトの所有者およびプライマリグループには、コピー元のオブジェクトに設定されていた所有者およびプライマリグループが設定されます。
また、仮のバージョンに該当するバージョンなしオブジェクトには、コピー元のオブジェクトに設定されていた ACFlag およびローカル ACL が設定され、チェックアウトしたオブジェクトからバインドしていたパブリック ACL がバインドされます。

(2) 文書間リンクに関する操作

文書間リンクを設定している文書間では、メソッドによってはリンク元文書とリンク先文書に対して適切なパーミッションが与えられている必要があります。

メソッドをコールするユーザがリンク元文書とリンク先文書に対して必要なパーミッションを次の表に示します。

表 3-22 リンク元文書とリンク先文書に対して必要なパーミッション

リンクに関係する操作（メソッド）	リンク元文書に必要なパーミッション	リンク先文書に必要なパーミッション
リンクの作成 • link	基本リンク権 (PRIM_LINK)	基本プロパティ参照権 (PRIM_READ_PROPS)
リンク情報の取得 • getRelList	基本プロパティ参照権 (PRIM_READ_PROPS)	基本プロパティ参照権 (PRIM_READ_PROPS)
リンクの削除 • unlink • unlinkByLinkId	基本リンク権 (PRIM_LINK)	-
リンクプロパティの設定 • writeProperties	基本リンク権 (PRIM_LINK)	-

(凡例)

- : パーミッションは必要ありません。

注

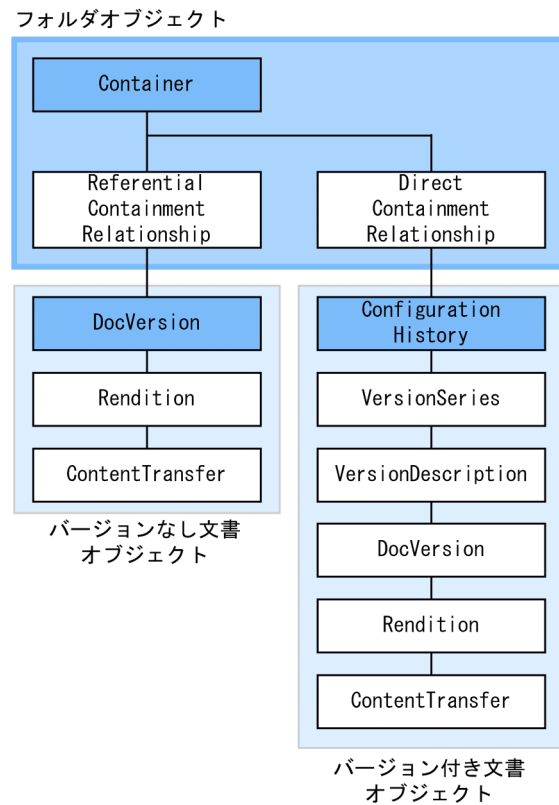
リンク先文書のプロパティを取得しない場合は不要です。

(3) フォルダへの関連づけに関する操作

直接型や参照型のリンクによってフォルダと関連づけている文書やフォルダでは、フォルダを表すオブジェクトと関連づけられるフォルダまたは文書を表す二つのオブジェクトに対して、適切なパーミッションを与えられている必要があります。

例えば、フォルダオブジェクトに対して、バージョンなし文書オブジェクトおよびバージョン付き文書オブジェクトとのリンクを設定する場合に必要なアクセス制御情報が設定されているオブジェクトは、次の図のようになります。

図 3-32 フォルダとフォルダに関連づけられているオブジェクトに設定されているアクセス制御情報



(凡例)

: コンテインメントを設定する場合に必要なアクセス制御情報が設定されているオブジェクト

フォルダを操作する場合に必要なパーミッション

フォルダを操作する時、操作の種類によってはフォルダに関連づけられている文書またはフォルダに対してもアクセス権が必要な場合があります。

メソッドをコールするユーザがフォルダに対して必要なパーミッションとフォルダに関連づけられる文書またはフォルダに対して必要なパーミッションを次の表に示します。

表 3-23 フォルダとフォルダと関連づけられている文書またはフォルダに対して必要なパーミッション
(フォルダを操作する場合)

リンクに係る操作 (メソッド)	フォルダに必要なパーミッション	フォルダに関連づけられるオブジェクトに必要なパーミッション
フォルダとオブジェクトの関連づけ • link	基本リンク権 (PRIM_LINK)	基本プロパティ参照権 (PRIM_READ_PROPS)
フォルダとオブジェクトの関連づけの解除 • unlink • unlinkByLinkId	基本リンク権 (PRIM_LINK)	-
リンクのプロパティの参照 • readProperties	基本プロパティ参照権 (PRIM_READ_PROPS)	-
リンクのプロパティの更新 • writeProperties	基本リンク権 (PRIM_LINK)	-
フォルダと関連づけられている要素の一覧取得 • getChildList	基本プロパティ参照権 (PRIM_READ_PROPS)	基本プロパティ参照権 (PRIM_READ_PROPS)

(凡例)

- : パーミッションは必要ありません。

注

フォルダと関連づけられるオブジェクトのプロパティを取得しない場合は不要です。また、フォルダと関連づけられるオブジェクトのプロパティを取得する場合に、基本プロパティ参照権が設定されていないオブジェクトについては、一覧として取得されません。

フォルダと関連づけられるオブジェクトを操作する場合に必要なパーミッション

フォルダと関連づけられている文書またはフォルダを操作する時、操作の種類によってはフォルダに対してもアクセス権が必要な場合があります。

メソッドをコールするユーザがフォルダに対して必要なパーミッションとフォルダと関連づけられる文書またはフォルダに対して必要なパーミッションを次の表に示します。

表 3-24 フォルダとフォルダに関連づけられている文書またはフォルダに対して必要なパーミッション
(フォルダと関連づけられるオブジェクトを操作する場合)

フォルダに関連づけられているオブジェクトに対する操作 (メソッド)	フォルダに必要なパーミッション	フォルダに関連づけられている文書またはフォルダに必要なパーミッション
自オブジェクトを包含要素として関連づけられているフォルダの一覧取得 • getParentList	基本プロパティ参照権 (PRIM_READ_PROPS)	基本プロパティ参照権 (PRIM_READ_PROPS)

注

フォルダであるオブジェクトのプロパティを取得しない場合は不要です。また、フォルダであるオブジェクトのプロパティを取得する場合に、基本プロパティ参照権が設定されていないオブジェクトについては、一覧として取得されません。

フォルダに関連づけられているオブジェクトに対して表 3-24 に示した以外のメソッドがコールされた場合、フォルダであるオブジェクトのパーミッションは必要ありません。

(4) 一覧取得メソッドを使用する操作

一覧取得メソッドとは、get ~ List メソッドなどのメソッドです。クラスライブラリのオブジェクトがほかのクラスライブラリのオブジェクトとリンクやバージョン管理機能によって関連づけられている場合に、その関連づけられているオブジェクトの一覧を取得します。

一覧取得メソッドには、メソッドで設定されたデフォルトの情報だけを取得する以外に、明示的に指定した接続先オブジェクトのプロパティを取得できるメソッドがあります。これらのメソッドで、接続先のオブジェクトのプロパティを取得する場合は、接続先オブジェクトに基本プロパティ参照権が必要です。このパーミッションが設定されていない場合は、一覧に取得されません。なお、デフォルトの情報だけを取得する場合は、接続先のオブジェクトにパーミッションは必要ありません。これは、ユーザプログラムの内部処理に使うことを想定しています。システムを使用するエンドユーザに対してプロパティを表示して文書一覧を表示する場合などには、接続先のオブジェクトのプロパティを取得するメソッドを使用して、アクセス権があるものだけを表示するといった運用方法をお勧めします。

また、readProperties メソッドを使用して、該当するプロパティを指定した場合も、接続先オブジェクトの個数は取得できます。ただし、このメソッドでは、接続先オブジェクトに対するパーミッションに関係なく、接続されているすべてのオブジェクトの個数が取得されます。

(5) マルチレンディション管理に関する操作

マルチレンディション管理をしている文書の場合、レンディションの操作は、そのレンディションを含むバージョンなし文書オブジェクトのトップオブジェクトにに対して設定されているアクセス権に従って制御されます。

3.8.7 アクセス制御モデルの運用例

ここでは、アクセス制御情報の使い分けとアクセス制御モデルの運用例を示します。

(1) アクセス制御情報の使い分け

ここでは、これまでに説明してきたアクセス制御情報の使い分けの考え方について説明します。アクセス権の判定に使用されるアクセス制御情報の処理順序や特長を考慮して、組織に合わせたアクセス制御モデルを構築してください。アクセス制御情報の処理順序と特長を次の表に示します。

表 3-25 アクセス制御情報の処理順序と特長

種類	処理順序	特長
ユーザ権限	1	<ul style="list-style-type: none"> アクセス権の判定で最初に参照される情報なので、判定が高速です。 あるユーザに対して、文書空間内で同一の操作を許可する場合に使用すると便利です。
ACFlag	2	<ul style="list-style-type: none"> 所有者、プライマリグループの情報と組み合わせて判定されます。 ユーザ権限の次に参照される情報なので、ローカル ACL やパブリック ACL に比べて早い時点で判定されます。 所有者、プライマリグループまたはすべてのユーザに対して許可する操作だけ設定できます。
パブリック ACL	3	<p>複数の文書管理オブジェクトに対して共通のアクセス制御情報（ローカル ACL と同様にユーザまたはグループ単位で個別に設定できる）が設定できます。したがって、アクセス権を一括して管理でき、アクセス権の保守が容易になります。</p> <p>例えば、「企画部」というグループに対して参照を許可するパブリック ACL を作成して、複数の文書管理オブジェクトからこのパブリック ACL をバインドしていたとします。この場合、途中で「企画部」に対して更新も許可するような運用に変更するときは、パブリック ACL で「企画部」に対して更新を許可するパーミッションを追加すれば、このパブリック ACL をバインドしているすべての文書管理オブジェクトに変更を反映できます。</p>
ローカル ACL	4	<p>文書管理オブジェクト固有のアクセス制御情報をユーザまたはグループ単位で個別に設定できます。個々の文書管理オブジェクトに、状況に応じて柔軟にアクセス制御情報を設定できます。</p>

(2) ACFlag を使用したアクセス制御の運用例

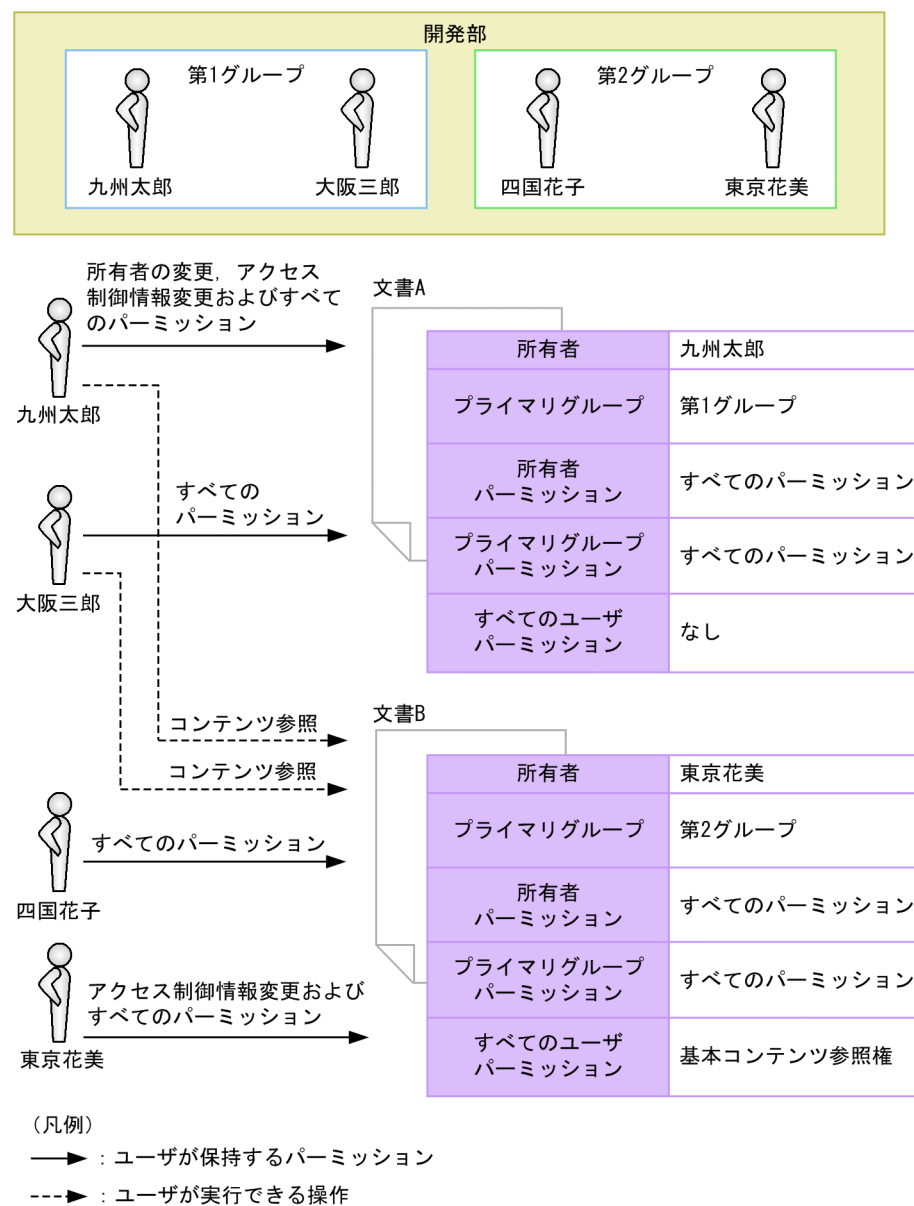
ACFlag を使用したアクセス制御の運用例について説明します。

なお、この例では、次のことを前提としています。

- すべてのユーザは文書を作成できます。
- ユーザは、所属するグループのユーザが作成した文書に対してすべてのパーミッションを組み合わせたアクセス権を与えられます。
- 文書の作成者は、文書の所有者としてアクセス制御情報変更権を保持し、作成した文書のアクセス権を設定・更新して管理します。

ACFlag を使用したアクセス制御の運用例を次の図に示します。

図 3-33 ACFlag を使用したアクセス制御の運用例



この例では、文書のライフサイクルの段階に応じて次のようにアクセスを制御することが考えられます。

3. 文書管理モデル

文書が完成するまでの段階

文書 A のように所有者が ACFlag のすべてのユーザパーミッションを「なし」に設定しておき、所属グループ内だけに公開して作業します。

文書が完成した段階

文書 B のように所有者が ACFlag のすべてのユーザパーミッションに「基本コンテンツ参照権」を設定して、ほかのグループにも完成した文書の内容の参照を許可します。

(3) ACFlag およびローカル ACL を使用した運用例

ACFlag を使用したアクセス制御モデルで、常にグループのグループリーダーを所有者に設定して運用する運用例と、ローカル ACL もあわせて設定する運用例について説明します。

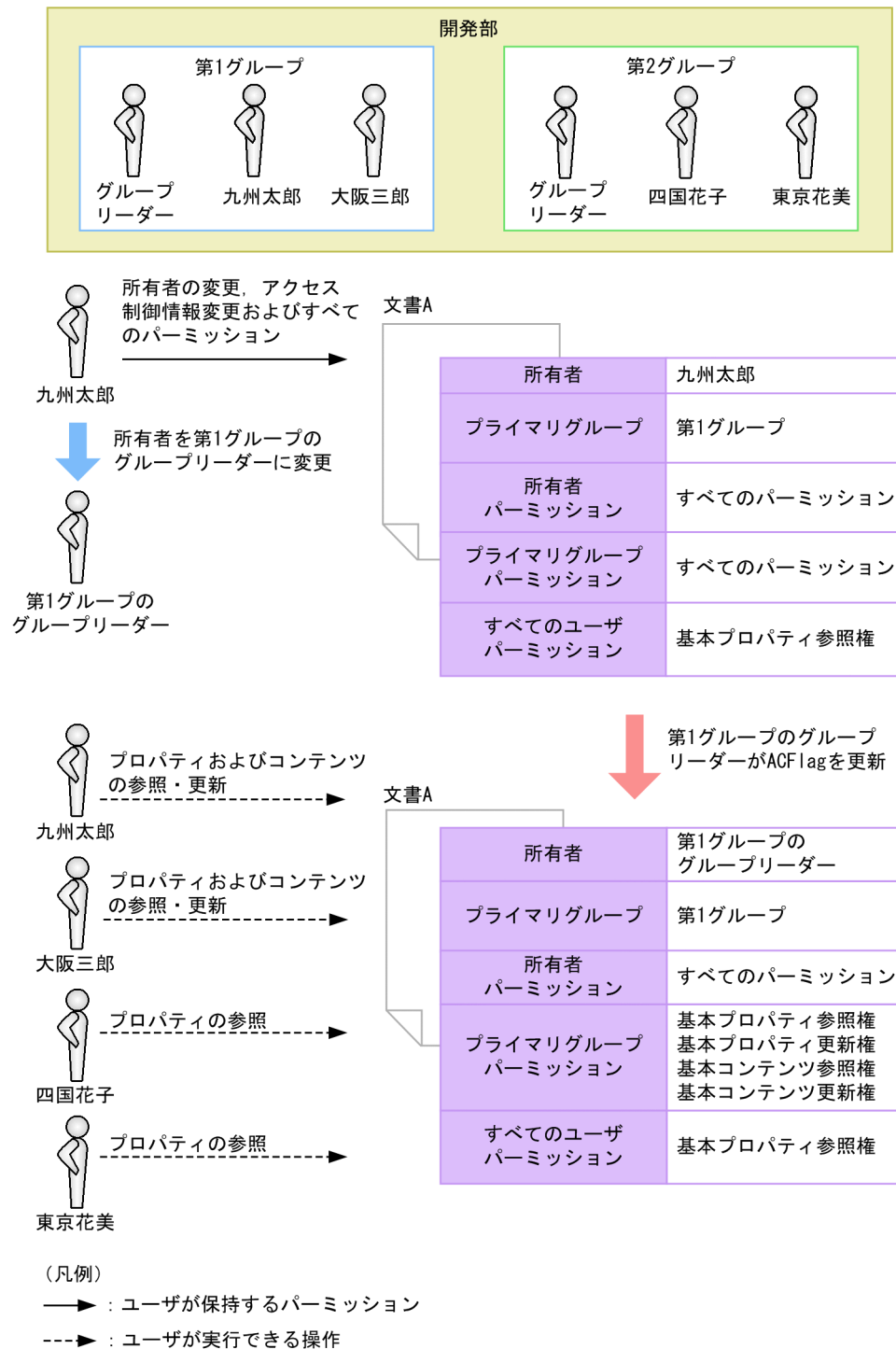
グループリーダーに管理を任せる ACFlag を使用した運用例

なお、この例では、次のことを前提としています。

- すべてのユーザは文書を作成、参照および更新できます。
- 文書の作成者は、作成後に文書の所有者をグループリーダーに変更します。
- グループリーダーはすべてのパーミッションを組み合わせたアクセス権を保持し、作成された文書を管理します。
- グループリーダー以外のユーザは作成者であっても文書を削除したり、フォルダにリンク付けたり、バージョンを管理したりすることはできません。

グループリーダーに管理を任せる ACFlag を使用した運用例を次の図に示します。

図 3-34 グループリーダーに管理を任せる ACFlag を使用した運用例



この例では、グループリーダーの判断で、ライフサイクルに応じた適切なパーミッションを設定して文書を管理しています。

ローカル ACL を併用した運用例

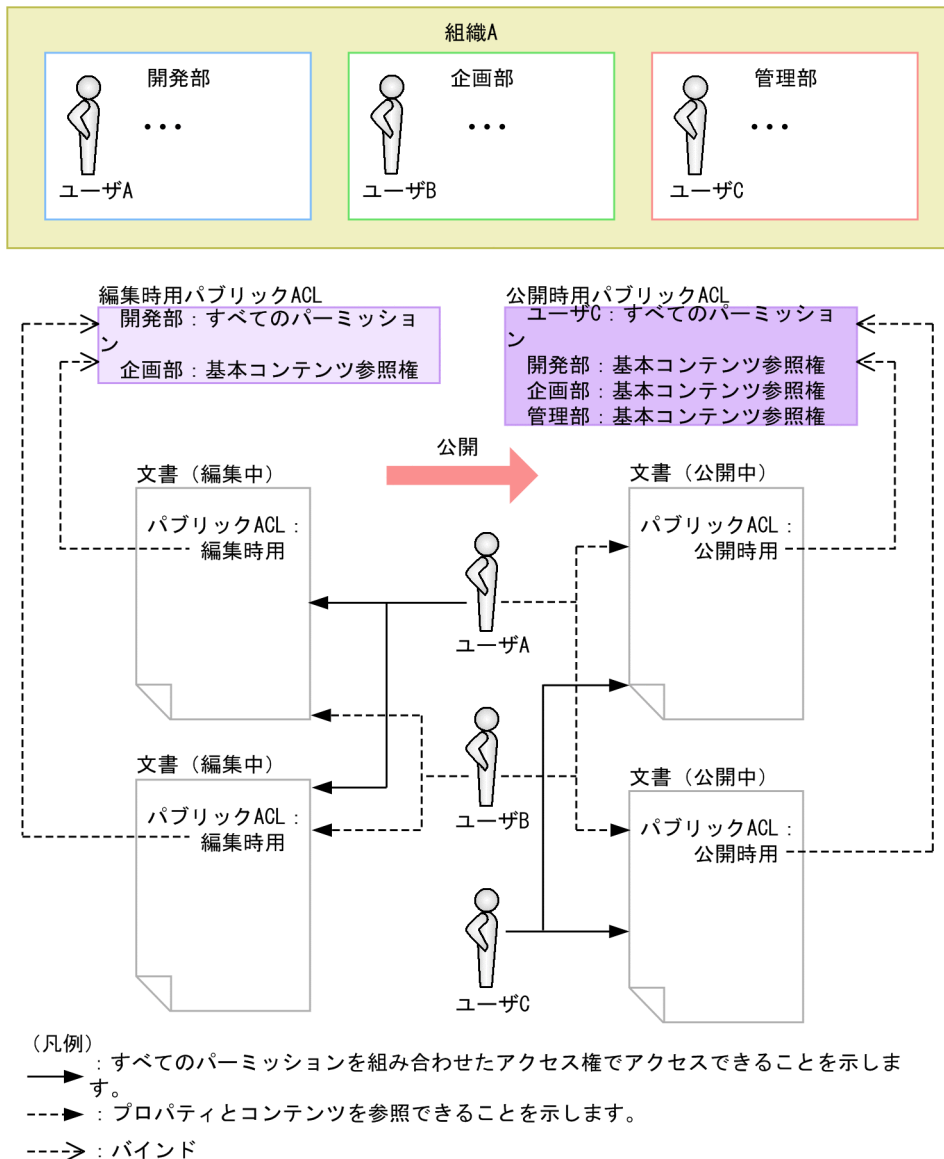
例えば、この例の運用上、文書 A について、「一定期間だけ他部署の部長（ユーザ X）にもコンテンツを参照させる必要ができた」という場合があったとします。ACFlag では特定ユーザに対するアクセス情報は設定できません。したがって、この場合は、ローカル ACL に「ユーザ X にコンテンツの

参照権を与える」と設定する運用が考えられます。

(4) パブリック ACL を使用した運用例

パブリック ACL を使用した運用例について説明します。パブリック ACL は、複数の文書やフォルダに対してアクセス制御情報を共通して設定したい場合に使用します。パブリック ACL を使用した運用例を次の図に示します。

図 3-35 パブリック ACL を使用した運用例



例えば、組織 A で、「『開発部』」のユーザが作成した文書は、「『開発部』」内のすべてのユーザがすべてのパーミッションを組み合わせたアクセス権を与えられて更新や削除ができるようにして、「『企画部』」内のすべてのユーザがコンテンツの参照だけできるようにしたい」という場合があったとします。このとき、パブリック ACL を一つ作成して、「グループ『開発部』」に対してすべてのパーミッションを組み合わせたアクセス権を与える」という ACE と「グループ『企画部』」に対してコンテンツの参照権を与える」という ACE を設定します。「『開発部』」が作成したすべての文書に対してこのパブリック ACL を設定すれば、すべての文書に同じアクセス制御情報が設定できます。

なお、パブリック ACL は複数の文書管理オブジェクトで共有されますので、作成時に明確な運用方法を決めてからアクセス制御情報を設定してください。複数の文書管理オブジェクトからパブリック ACL をバインドしている場合に、特定の文書管理オブジェクトのアクセス制御情報を変更する必要があるときは、そのパブリック ACL の内容を変更するのではなく、アクセス制御情報を変更する必要がある文書管理オブジェクトだけ別のパブリック ACL にバインドし直すという運用をお勧めします。

例えば、「文書が完成するまでの文書の編集段階では『開発部』にすべてのパーミションを組み合わせたアクセス権を与え、『企画部』にコンテンツの参照だけを許可したいが、一度公開したあとでは『管理部』内の特定のユーザ（ユーザ C）以外には更新させないようにしたい」というような場合は、公開時にパブリック ACL の内容を変更するのではなく、編集時用のパブリック ACL と、公開時用のパブリック ACL をそれぞれ作成しておく運用が考えられます。ある文書を公開したら、その文書にバインドするパブリック ACL を、編集時用から公開時用にバインドし直すことで、アクセス制御情報を変更できます。

(5) セキュリティ ACL を使用した運用例

アクセス制御情報は、デフォルトでは、その文書管理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設定・更新できます。セキュリティ ACL を使用すると、文書やフォルダに設定された ACFlag の更新、ローカル ACL の更新およびパブリック ACL のバインド・アンバインドの権利を所有者およびセキュリティ管理者以外のユーザにも与えることができます。

例えば、パブリック ACL を使用した運用で、文書のライフサイクルの段階に応じてパブリック ACL をバインドし直すという運用をしている場合、文書管理オブジェクトの所有者以外のユーザにパブリック ACL のバインド・アンバインドの管理を任せるために、セキュリティ ACL を使用することが考えられます。例えば、「『開発部』が作成した文書を、『管理課』の特定ユーザ（ユーザ Y）が、公開時に『公開時パブリック ACL』をバインドして公開する」という場合、開発部が作成する文書のセキュリティ ACL に「ユーザ Y にアクセス制御情報変更権を与える」と設定しておきます。

3.9 排他制御モデル

この節では、DocumentBroker での排他制御について説明します。

DocumentBroker では、HiRDB のロック機能を使用して文書管理オブジェクトに対する排他制御を実現します。このため、文書管理オブジェクトに設定されるロックは、一つのトランザクション内だけで有効です。通常、一つのメソッドが一つのトランザクションに対応し、メソッド実行時に、DocumentBroker が文書管理オブジェクトに対してロックを設定します。また、ユーザがトランザクションの開始と終了を指定することもできます。文書管理オブジェクトに設定されるロック種別は、使用するインターフェースおよびメソッドによって異なりますが、ロック種別をユーザが明示的に指定してオブジェクトにアクセスすることもできます。

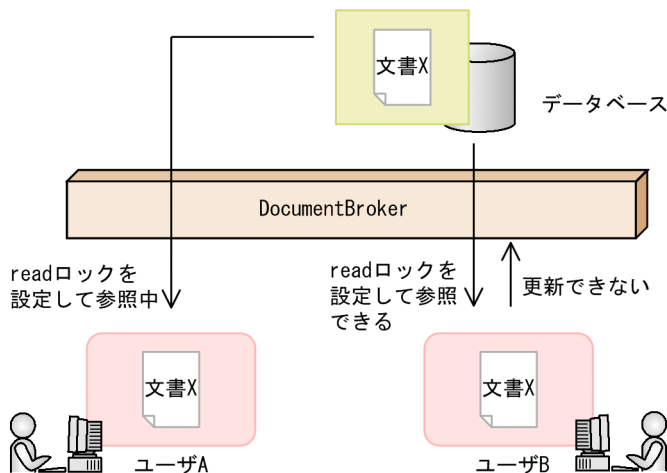
3.9.1 ロック種別

DocumentBroker では、次に示すロック種別があります。

read ロック

文書管理オブジェクトを参照する場合に設定されるロックです。あるユーザが read ロックを設定して取り出した文書管理オブジェクトに対して、ほかのユーザは write ロックを設定してその文書管理オブジェクトを更新、削除することはできません。ただし、read ロックを設定してその文書管理オブジェクトを参照することはできます。read ロックを設定した文書管理オブジェクトの操作を次の図に示します。

図 3-36 read ロックを設定した文書管理オブジェクトの操作

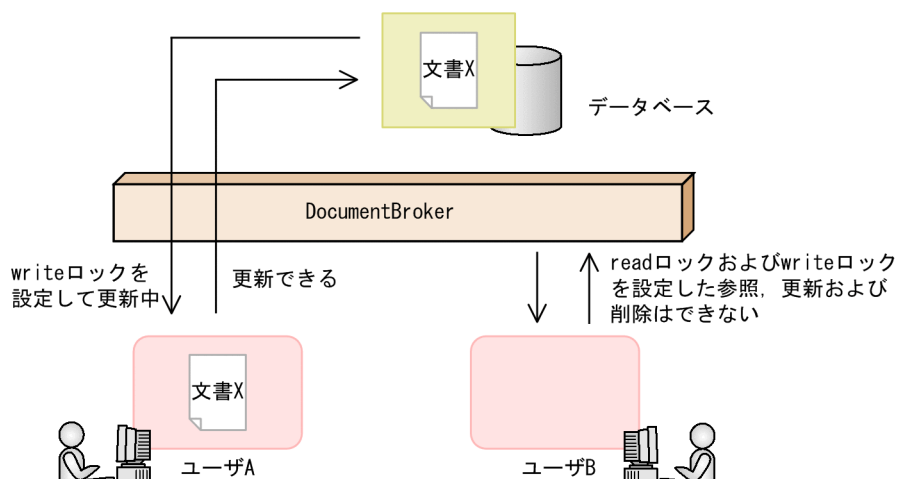


ユーザ A がほかのユーザより先に文書 X に read ロックを設定して参照している場合、ほかのユーザは文書 X に read ロックを設定して参照することはできますが、write ロックを設定して更新および削除することはできません。

write ロック

文書管理オブジェクトを更新または削除する場合に設定されるロックです。あるユーザが write ロックを設定して取り出した文書管理オブジェクトに対して、ほかのユーザは read ロックまたは write ロックを設定できません。したがって、write ロックを設定した以外のユーザが、read ロックまたは write ロックを設定してその文書管理オブジェクトを参照、更新または削除することはできません。write ロックを設定した文書管理オブジェクトの操作を次の図に示します。

図 3-37 write ロックを設定した文書管理オブジェクトの操作



ユーザ A がほかのユーザより先に文書 X に write ロックを設定して更新または削除しようとしている場合、ほかのユーザは文書 X に read ロックを設定して参照したり、write ロックを設定して更新および削除したりすることはできません。

3.9.2 ロックによる排他制御

あるユーザ（ユーザ A）が、ある文書管理オブジェクト（文書 X）に対してロックを設定している状態を、「ユーザ A が文書 X のロックを取得している」と表現します。DocumentBroker では、同じ文書管理オブジェクトに対して、あるユーザ（ユーザ A）がすでに取得しているロックと、ほかのユーザ（ユーザ B）が取得するロックが競合した場合、ユーザ B は、ユーザ A が取得しているロック種別によって、次の表で示すように排他制御されます。

表 3-26 ロックが競合した場合のロックの関係

ユーザ A が取得したロック		read	write
ユーザ B が取得できるロック	read		×
	write	×	×

（凡例）

：ロックを設定するメソッドを実行できます。

×：ロックを設定できないため、実行するメソッドは、ロックが解除されるまで待ち状態（WAIT モード）になります。

3.9.3 ロックが設定されているオブジェクトに対する接続

DocumentBroker では、WAIT モードでメソッドを実行します。例えば、ユーザ A がある文書管理オブジェクトのロックを取得しようとした場合に、すでにユーザ B がその文書管理オブジェクトのロックを取得していたとします。このとき、ロックが解除されるまで、ユーザ A が実行しようとしたメソッドは待ち状態となります。ただし、タイムアウトやデッドロックが発生した場合は、処理を終了します。なお、ロックは次の時点まで継続します。

- セッションの切断
- トランザクションの終了

なお、WAIT モードでのタイムアウト時間は HiRDB に設定します。HiRDB のタイムアウト時間の設定については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

3.9.4 ロック種別の変更

メソッド実行時に文書管理オブジェクトに設定されるロック種別は、使用するインターフェースおよびメソッドによって異なります。参照系メソッドを実行する場合、デフォルトの設定では、文書管理オブジェクトに対して read ロックが設定されます。DocumentBroker では、このロック種別を write ロックに変更して文書管理オブジェクトにアクセスできます。

3.9.5 ロックに関する注意事項

ここでは、ロックに関する注意事項について説明します。

(1) ロックが設定される範囲

DocumentBroker では、メソッド実行時に、文書管理オブジェクトを構成する DocumentBroker オブジェクトのトップオブジェクトに対してロックが設定されます。また、トップオブジェクト以外の DocumentBroker オブジェクトについては、メソッドの処理に必要なオブジェクトに対してロックが設定されます。このため、次の場合、トップオブジェクト以外の DocumentBroker オブジェクトに対するロックを、ほかのユーザが取得できてしまうことがあります。

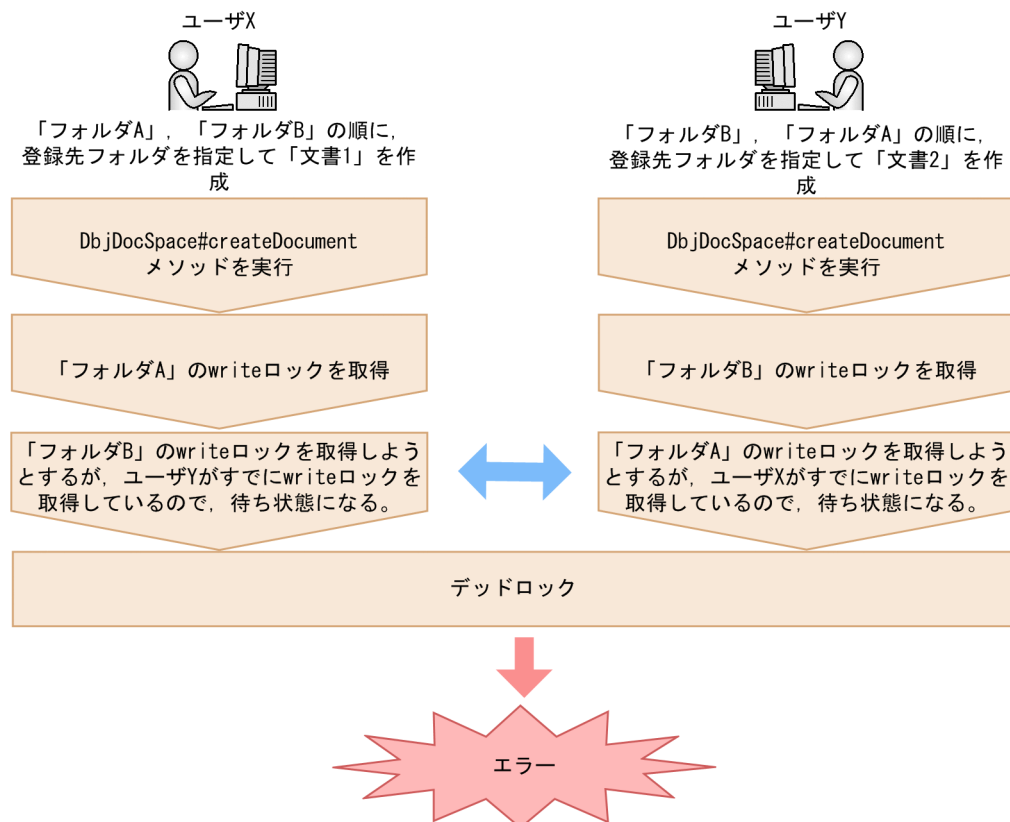
文書管理オブジェクトを構成する DocumentBroker オブジェクトのロックを取得していても、ほかのユーザが、DbjDocSpace#executeSearch メソッドを使用して検索結果に対してロックを取得すると、トップオブジェクト以外の DocumentBroker オブジェクトのロックを取得できてしまう場合があります。

バージョンングオブジェクトに対してロックを取得していても、ほかのユーザは、バージョンオブジェクトのロックを取得できてしまう場合があります。

(2) 複数のオブジェクトに write ロックを設定するメソッドの使用上の注意

複数の文書管理オブジェクトに write ロックを設定することがあるメソッドを使用して、複数の文書管理オブジェクトを操作する場合、パラメタの指定によっては、デッドロックが発生することがあります。デッドロックの例を次の図に示します。

図 3-38 デッドロックの例



この例では、操作対象の文書管理オブジェクトとして二つのバージョンなしフォルダ「フォルダ A」と「フォルダ B」があります。ユーザ X が、バージョンなし文書「文書 1」の作成と同時に、その文書の登録先フォルダを「フォルダ A」、「フォルダ B」の順に指定して、バージョンなし文書を作成するメソッド (`DbjDocSpace#createDocument` メソッド) を実行したとします。このとき、ユーザ Y も、バージョンなし文書「文書 2」の作成と同時に、その文書の登録先フォルダを「フォルダ B」、「フォルダ A」の順に指定して同じメソッドを実行したとします。この場合、お互いに途中で待ち状態 (WAIT モード) のままになってデッドロックが発生し、デッドロックが検知された時点でエラーになることがあります。

デッドロックを発生しにくくするには、次のような方法が考えられます。例えば、文書やフォルダなどの文書管理オブジェクトの一覧を取得する場合に、文書管理オブジェクトを一定の順序にソートしておいて、更新系メソッドに指定するバージョン識別子またはリンク識別子を、一定の順序で指定します。これによって、図 3-36 の例のような指定順が原因であるデッドロックが発生する可能性を軽減できます。

4

検索機能

この章では、DocumentBroker の検索機能である、edmSQL を使用して文書管理オブジェクトを検索する方法の概要について説明します。DocumentBroker では、SQL に準じた構文規則を持つ edmSQL を使用してオブジェクトを検索できます。この検索では、プロパティの値を指定した検索や、検索タームを指定した検索が実行できます。

-
- 4.1 検索の概要
 - 4.2 検索の種類
 - 4.3 検索の実行方法の種類
 - 4.4 検索対象になる文書管理オブジェクト
 - 4.5 全文検索の対象になる文書の作成
-

4.1 検索の概要

この節では、DocumentBroker で提供する検索機能の概要について説明します。

4.1.1 DocumentBroker での検索

DocumentBroker で提供している検索機能には、次の二種類があります。

検索条件を指定するメソッドを実行する方法

edmSQL という SQL に準じた構文で検索条件を指定するメソッドを実行する方法です。取得したいオブジェクトに関して知っている情報（プロパティの値など）を基に、文書管理オブジェクトを検索できます。検索結果として、文書管理オブジェクトの任意のプロパティが取得できます。OID を表すプロパティを取得することもできます。

一覧を取得するメソッドを実行する方法

一つの文書管理オブジェクトから、関連があるほかの文書管理オブジェクトの一覧を取得するメソッドを実行する方法です。例えば、フォルダからリンク付けている文書の一覧を取得したり、バージョン付きオブジェクトのバージョンオブジェクトからバージョンオブジェクトの一覧を取得したりして、文書管理オブジェクトを検索します。

これらの検索によって、検索結果として取得した文書管理オブジェクトを操作するためのインターフェースが取得できます。

なお、この章では、主に検索条件を指定するメソッドを実行する方法について説明します。

4.1.2 検索条件と検索結果

検索を実行する場合に指定する検索条件と、取得する検索結果について説明します。

(1) 検索条件

検索条件を指定するメソッドの検索条件は、edmSQL 文として指定します。

edmSQL とは、文書管理オブジェクトを検索するための手段として DocumentBroker が提供する検索用言語です。SQL の文法に関する知識を活用して、文書管理オブジェクトの検索ができます。なお、これ以降、このマニュアルでは、edmSQL を使用した検索を、edmSQL 検索といいます。また、edmSQL に従って記述した条件式を、edmSQL 文といいます。

edmSQL には、次のような特長があります。

文書管理で使用する概念に基づいた検索条件を指定できます。

edmSQL では、DocumentBroker の文書管理で使用する文書管理オブジェクトを構成する DocumentBroker クラスのクラス名やプロパティ名を使用して、検索条件を指定できます。

データベース標準問い合わせ言語である SQL に基づいた文法です。

edmSQL の文法は、標準的な SQL の文法に基づいています。SQL の文法に関する知識を活用して edmSQL 文を作成できます。

(2) 検索結果

ここでは、次の項目について説明します。

検索結果集合

検索結果キャッシュ

検索結果取得情報

(a) 検索結果集合

検索条件を指定するメソッドを実行すると、検索結果集合が作成され、メソッドの戻り値として返却されます。検索結果集合は、検索条件に一致する文書管理オブジェクトの集合です。

検索結果集合は、要素がプロパティ値である、行と列の二次元データとして表すことができます。さらに、DocumentBroker では、この二次元データにメタデータを付けて表すことができます。メタデータとは、列のプロパティ値のデータ型、および列名の集合です。

DocumentBroker で扱う検索結果集合を次の図に示します。

図 4-1 検索結果集合

edmSQL文のSELECT句に指定したプロパティの個数分の列数

メタデータ	dmaProp_OIID	SO. Name	Date	...
	STR型	STR型	INT型	...
検索条件に一致した 文書空間オブジェクト の個数分の行数	dma:///07a17522 -a626-...	“日立太郎”	20010205	...
	dma:///07a43721 -d727-...	“鈴木一郎”	20010423	...
	dma:///06b18572 -57d6-...	“田中花子”	20000227	...
	dma:///17e18357 -7657-...	“日立太郎”	20010513	...

なお、メタデータとして列名（プロパティ名）が付けられるのは、名前付き検索結果の場合です。名前付き検索結果の詳細については、「4.3.4 名前付き検索結果を取得する検索」を参照してください。

検索結果集合は、検索実行メソッドの戻り値として返却される、DbjResultSet インターフェースを使用し操作します。DbjResultSet インターフェースでは、検索結果集合を、各行を要素とするリストとして扱います。さらに、行の各要素を、各列の値を要素とするリストとして扱います。

検索結果集合の行は、カーソル行を移動して特定します。各行の列の要素は、列の値を要素とするリストに対して、列インデクスを指定することによって特定します。列インデクスは、0 から始まる整数値であり、edmSQL文のSELECT句に指定した順に、0, 1, ...n と割り当てられます。

検索結果集合の要素を特定する例を次に示します。

図 4-2 検索結果集合の要素の特定

1. カーソル行を取得したい行に移動する。

	dma:///07a17522 -a626-...	“日立太郎”	20010205	...
カーソル行 →	dma:///07a43721 -d727-...	“鈴木一郎”	20010423	...
	dma:///06b18572 -57d6-...	“田中花子”	20000227	...
	dma:///17e18357 -7657-...	“日立太郎”	20010513	...



2. カーソル行の列を要素とするリストを取得する。

dma:///07a43721 -d727-...	“鈴木一郎”	20010423	...
------------------------------	--------	----------	-----



3. 取得する要素の列インデックスを指定する。

列インデクス [1]

dma:///07a43721 -d727-...	“鈴木一郎”	20010423	...
------------------------------	--------	----------	-----

(b) 検索結果キャッシュ

検索結果は、いったんキャッシュに保存して、複数回に分けて取得することができます。このキャッシュを検索結果キャッシュといいます。検索結果キャッシュは、検索条件を指定するメソッドおよび一覧を取得するメソッドで使用できます。検索結果キャッシュを使用するための情報は、検索結果取得情報に設定します。

検索結果キャッシュを使用すると、1回の検索でキャッシュから取得する検索結果の行数を指定したり、キャッシュ上の取得開始位置を指定して任意の行から検索結果を取得したりできます。検索結果キャッシュはキャッシュ名によって識別されます。このため、キャッシュ名が異なる複数の検索結果キャッシュを保持できます。キャッシュ名を指定しないと、ユーザアプリケーションプログラムが検索結果を取得し終わった時点で、検索結果キャッシュは無効になります。また、キャッシュ名は、同一セッション内だけで有効です。

なお、検索結果キャッシュから検索結果を取得できるのは、キャッシュ名が同じ検索結果キャッシュに対して、同じ検索条件で検索を実行する場合だけです。キャッシュを使用する検索では、次の条件を満たす場合、検索条件が同じであると判断されます。

同じセッション内で、同じキャッシュ名の検索結果キャッシュに対して検索を実行しているログアウトすると、検索結果キャッシュは破棄されます。

検索結果キャッシュに保持されているキャッシュキーと検索結果取得情報に設定されているキャッシュキーが一致する

1回目の検索では、キャッシュキーには、定数 DbjDef.INITIAL_KEY を指定します。2回目以降は、前回の検索結果と一緒に取得したキャッシュキーを指定します。なお、DbjDef.INITIAL_KEY を指定すると、必ずキャッシュキーが不一致になるため、検索結果キャッシュは破棄されて新しい検索が実行されます。

検索結果キャッシュを作成した時と同じメソッドで検索を実行している

例えば、一覧を取得メソッドで取得した検索結果キャッシュに対して検索条件を指定するメソッドを実行した場合などには、検索結果は破棄されます。

(c) 検索結果取得情報

検索結果の取得方法は、検索結果取得情報として指定します。検索結果取得情報には、検索結果集合からユーザアプリケーションプログラムに返却する検索結果の最大件数、取得開始位置と件数、並び順などが指定できます。また、検索結果キャッシュを使用する場合には、キャッシュに取得する最大件数、キャッシュ名およびキャッシュキーも指定できます。検索結果取得情報は、検索条件を指定するメソッドおよび一覧を取得するメソッドで指定できます。

取得する検索結果の並び順は、コンパレータを使用して指定します。コンパレータを設定した検索結果取得情報を指定して検索を実行すると、コンパレータによってソートされた検索結果が取得できます。また、検索結果キャッシュに対して検索を実行する場合は、検索結果キャッシュの内容が並べ替えられます。並べ替えは、検索結果キャッシュに対して、異なるコンパレータを設定した検索結果取得情報を指定して検索を実行する時に実行されます。例えば、A というコンパレータでソートしていた検索結果キャッシュに対して、B というコンパレータを設定した検索結果取得情報を指定して検索を実行すると、検索結果キャッシュの内容が並べ替えられます。ただし、コンパレータとして null 値を指定した場合は、前と同じコンパレータを指定したものとして処理されます。このため、並べ替えは実行されません。

4.2 検索の種類

ここでは、edmSQL で実行できる検索の種類について説明します。edmSQL では、次に示す検索が実行できます。

edmSQL では、次の 2 種類の検索が実行できます。

属性検索

全文検索

全文検索機能付き文字列型プロパティを使用した全文検索

これらの検索は、組み合わせて指定することもできます。ここでは、それぞれの検索の概要について説明します。

4.2.1 属性検索

属性検索とは、文書管理オブジェクトの一つまたは複数個のプロパティの値を指定して実行する検索です。例えば、「文書名」がプロパティとして設定されている文書に対して、「文書名が『SQL 入門』である文書を探す」などの検索が実行できます。複数個のプロパティの値を指定する場合は、論理条件を指定して、「文書名が『SQL 入門』で、かつ、出版社が『ABC 出版』である文書を探す」という検索もできます。

検索の対象として、VARRAY 型のプロパティも指定できます。例えば、著者が複数人存在する可能性がある文書を管理するために、「著者」を VARRAY 型のプロパティで管理して、それぞれの「名前」をその要素のプロパティとして管理している場合、「『田中』という名前の著者が含まれる文書を探す」などの条件を指定した検索が実行できます。

4.2.2 全文検索

ここでは、全文検索について説明します。

(1) 全文検索とは

全文検索とは、登録したコンテンツを対象に、キーワードとなる文字列（検索ターム）を一つまたは複数個指定して実行する検索です。例えば、「文書のコンテンツ中に『コンピュータ』という文字列が含まれる文書を探す」などの検索が実行できます。

なお、全文検索には、次の 2 種類の検索が含まれます。

検索タームを指定する全文検索（狭義の全文検索）

HiRDB Text Search Plug-in、または HiRDB XML Extension の機能で実行できる、コンテンツに対して検索タームを指定して実行できる全文検索です。

概念検索

HiRDB Text Search Plug-in、または HiRDB XML Extension の関連プログラムである HiRDB Text Search Plug-in Conceptual Extension の機能で実行できる、検索タームを文章（種文章）から抽出して実行する全文検索です。この検索では、検索条件に指定した文章に近い概念を持つ（同じ単語が多く含まれる）文書が検索できます。

このマニュアルでは、これらの二つの検索を合わせて、「全文検索」といいます。それぞれの検索に特有の説明をする場合は、「検索タームを指定する全文検索」または「概念検索」と区別して説明します。

全文検索の対象にする文書は、全文検索機能付き文書クラスを基に作成されている必要があります。全文検索機能付き文書クラスの作成方法については、「4.5.2 全文検索インデックスの作成」を参照してください

い。

なお、edmSQL では、全文検索の次の機能を実現するための関数を提供しています。

プレーンテキストに対する全文検索（検索タームを指定する全文検索）

プレーンテキストに対する概念検索

検索結果にハイライトタグの埋め込みをして SGML 形式で出力する全文検索

ここでは、これらの検索の概要と、使用する edmSQL の関数について説明します。また、全文検索では、HiRDB Text Search Plug-in、または HiRDB XML Extension の機能によって、検索タームや種文章をそのまま指定する検索だけでなく、同義語や異表記を同時に検索したり、検索タームに重みを指定したりすることもできます。これらの全文検索で指定できる機能の概要についても説明します。

(2) 全文検索の種類

ここでは、全文検索の種類について説明します。

(a) プレーンテキストに対する全文検索（検索タームを指定する全文検索）

プレーンテキストとは、構造を持たないテキスト形式の文書のことです。文書のコンテンツに対して、一つまたは複数の検索タームを指定した検索が実行できます。複数の検索タームを指定する場合、それらの論理条件も指定できます。

例えば、「文書のコンテンツに『コンピュータ』または『データベース』という文字列を含む文書」を検索したりできます。また、属性検索条件と組み合わせて、「文書のタイトル（プロパティ）が『データベース入門』であり、文書のコンテンツに『コンピュータ』という文字列を含む文書」のような検索も実行できます。

この検索には、contains 関数を使用します。

(b) プレーンテキストに対する概念検索

全文検索実行時に、特定の検索タームを基に検索する以外に、「ある文章に似た内容の文書を検索したい」というような検索が実行できます。例えば、「リサイクルを含む環境問題」について書かれた文書を探したい場合に、「ゴミのリサイクル」に関する文章を基に検索が実行できます。

概念検索は、edmSQL などの検索条件に種文章を指定することで実行できます。種文章として指定できる容量は、5 メガバイト以内です。

ここでは、種文章に「...ゴミのリサイクルによって、資源が有効活用され、ゴミが減量化されます。...」という文字列を指定した場合の、概念検索の実行の流れを説明します。

1. HiRDB Text Search Plug-in Conceptual Extension の機能によって、種文章から特徴タームが抽出されます。

特徴タームとは、種文章に指定した文章から、漢字、かたかなおよびアルファベットを抽出した単語です。

この例の場合は、「ゴミ」、「リサイクル」、「資源」、「有効活用」、「減量化」というような単語が抽出されます。

2. HiRDB Text Search Plug-in Conceptual Extension によって、1. の特徴タームに優先順位が与えられた検索用特徴タームが選出されます。

検索用特徴タームの選出基準については、HiRDB Text Search Plug-in、または HiRDB XML

Extension の環境定義ファイルに指定します。詳細は、マニュアル「HiRDB Text Search Plug-in」、または「HiRDB XML Extension」を参照してください。概念検索では、この検索用特徴タームを基に、検索が実行されます。

3. 検索用特徴タームを検索タームとして、全文検索が実行されます。さらに、種文章との適合度順（検索用特徴タームを多く含む文章順）に、スコア値が設定されて、検索結果が出力されます。

なお、概念検索条件に指定する種文章は、?パラメータで指定する必要があります。この検索には、concept_with_score 関数を使用します。

(c) 検索結果にハイライトタグの埋め込みをして SGML 形式で出力する検索

全文検索実行時に、検索結果として検索タームがハイライト表示される SGML 形式のファイルを出力します。例えば、「ネットワーク」という検索タームが含まれるテキスト形式の文書を検索する場合に、検索結果と同時に「」というハイライト表示用タグが埋め込まれた SGML 形式のファイルを出力できます。なお、検索結果として取得するハイライト表示用の文書は、検索対象である文書とは異なる文書として作成されます。

次に、検索対象文書とハイライト表示用の文書の例を示します。

検索対象文書

```
<!DOCTYPE body SYSTEM "DOC.dtd">
<body>
<p>
連絡事項
<note>
ネットワークの変更については、添付資料を参照してください。
</body>
```

検索結果として取得するハイライト表示用の文書

```
<!DOCTYPE body SYSTEM "DOC.dtd">
<body>
<p>
連絡事項
<note>
<STRONG>ネットワーク</STRONG>の変更については、添付資料を参照してください。
</body>
```

(3) 全文検索で指定できる機能

edmSQL では、全文検索条件として、HiRDB Text Search Plug-in、または HiRDB XML Extension の規則に従った検索条件が指定できます。

HiRDB Text Search Plug-in、および HiRDB XML Extension では、単純文字列としての検索タームおよび種文章のほか、検索タームを展開したり、重みを設定したりする指定ができます。

edmSQL では、これらの指定は全文検索を実行する関数の引数として指定します。これらの関数は、HiRDB Text Search Plug-in、または HiRDB XML Extension で提供されている関数と対応しています。指定方法については、マニュアル「HiRDB Text Search Plug-in」、または「HiRDB XML Extension」を参照してください。

ここでは、全文検索条件に指定できる機能の概要について説明します。

(a) 検索タームの異表記を検索対象にした検索（異表記展開指定）

異なる表記方法で表記された検索タームを含む文書も検索条件に指定できる検索です。

異表記展開を指定した場合は、検索タームに指定したかたかなの異表記を含む文書を検索したり、全角アルファベットの大きい文字・小さい文字の異表記を含む文書を検索したり、アルファベットの全角・半角の異表記を含む文書を検索したりできます。例えば、検索ターム「バイオリン」を検索する場合に、「ヴァイオリン」のように表記されている文書も検索できます。また、検索ターム「Ski」を検索する場合に、「ski」や「S k i」のように表記されている文書も検索できます。

異表記展開検索では、データベースで決められた規則に従って、検索タームの異表記を含む文書を検索します。

(b) 検索タームの同義語を検索対象にした検索（同義語展開指定）

検索タームと同じ意味を持つ単語（同義語）を含む文書も検索条件に指定できる検索です。

例えば、検索ターム「Ski」を検索する場合に、「スキー」を含む文書も一緒に検索できます。同義語展開検索では、あらかじめ作成してある同義語辞書を基に、検索タームの同義語を含む文書を検索します。

(c) 複数の検索ターム間の距離条件や出現順序を指定した検索（近傍条件指定）

複数の検索タームを指定した場合に、検索タームが出現する距離（文字数）を検索条件に指定できる検索です。

例えば、『最新』と『技術』という検索タームを含み、これらの二つの単語が10文字以内に存在する文書などが検索できます。この場合には、「最新印刷技術」や「最新の技術」などの文字列を含む文書が検索されます。

(d) 検索タームに重みを付ける検索（重み指定）

検索タームに重み（重要度）を付ける検索です。重みを付けた検索とは、『インターネット』および『データベース』が含まれる文書を検索したい。ただし、『インターネット』をより多く含む文書から取り出したいので、『インターネット』に『5』、『データベース』に『2』の重みを付けて検索する」のように指定する検索です。

4.2.3 全文検索機能付き文字列型プロパティを使用した全文検索

ここでは、全文検索機能付き文字列型プロパティを使用した全文検索について説明します。なお、全文検索機能付き文字列型プロパティを使用するには、HiRDB Text Search Plug-in、または HiRDB XML Extension が必要です。

(1) 全文検索機能付き文字列型プロパティとは

全文検索が実行できる文字列型プロパティのことを、全文検索機能付き文字列型プロパティといいます。全文検索機能付き文字列型プロパティを使用すると、edmSQL で次に示す検索条件を指定して文書の検索ができます。

1. 指定した文字列がプロパティの値に含まれているか
2. 指定した文字列を異表記展開した文字列がプロパティの値に含まれているか
3. 指定した文字列を同義語展開した文字列がプロパティの値に含まれているか
4. 指定した文字列の距離（文字数）が指定の距離内にあるか

また、これらの検索条件を AND または OR で結ぶことができます。

(2) 全文検索機能付き文字列型プロパティの全文検索

edmSQL を使用して全文検索機能付き文字列型プロパティの値を取得する場合は、SELECT 句に extracts 関数を指定してください。

また、全文検索機能付き文字列型プロパティを全文検索する場合は、WHERE 句に contains 関数を指定してください。

全文検索機能付き文字列型プロパティを使用した全文検索の例を次に示します。

(例)

4. 検索機能

全文検索機能付き文字列型プロパティ `usrProp_DocSummary` に対応する文書中に、「コンピュータ」を同義語展開した文字列があるかどうかを検索し、文字列があればそのテキストデータを抽出します。

edmSQL の指定例

```
SELECT extracts(usrProp_DocSummary)
FROM   usrClass_PropTextSearch
WHERE  contains(usrProp_DocSummary, '{SYNONYM(myDic, "コンピュータ")}') is
true
```

注 `usrClass_PropTextSearch` は `dmaClass_DocVersion` クラスのサブクラスです。

! 注意事項

全文検索機能付き文字列型プロパティは、`extracts` 関数または `contains` 関数の第一引数に指定してください。それ以外の個所には指定できません。

(3) 全文検索機能付き文字列型プロパティに対応しているインターフェースおよびメソッド

全文検索機能付き文字列型プロパティに対応しているインターフェースおよびメソッドを次の表に示します。

表 4-1 全文検索機能付き文字列型プロパティに対応しているインターフェースおよびメソッド

対応しているインターフェース	対応しているメソッド
DbjDocSpace インターフェース	<ul style="list-style-type: none">• <code>createDocument</code>• <code>createFolder</code>• <code>createIndependentData</code>• <code>createVrDocument</code>• <code>createVrFolder</code>• <code>executeSearch</code>
DbjObj インターフェース	<ul style="list-style-type: none">• <code>cancelCheckOut</code>• <code>checkIn</code>• <code>checkOut</code>• <code>getDCRParent</code>• <code>readProperties</code>• <code>uploadContents</code>• <code>writeProperties</code>• <code>writeRenditionProperties</code>

4.3 検索の実行方法の種類

ここでは、検索の実行方法の種類について説明します。

edmSQL 検索では、検索条件に ? パラメタを指定して実行できます。また、検索実行時には、検索結果集合の取得方法や、検索結果に対する排他制御、アクセス制御などの設定ができます。

4.3.1 ? パラメタを使用した検索

業務によっては、検索条件のうち、一部の定数値だけを変更して繰り返し同じ検索を実行したい場合があります。このような場合に、? パラメタを使用できます。

例えば、文書の作成日がプロパティとして設定されている場合に、同じ検索条件で、作成日の値だけを「20010603」、「20010604」、「20010605」と変更して検索したい場合があります。このような、検索条件の特定の個所だけ変更して繰り返し検索を実行する場合に、このパラメタが使用できます。この検索では、検索条件式の作成日の値には「?」を設定しておきます。実行するたびに変更する値については別のパラメタとして指定します。これによって、複数の検索に対して一つの検索条件式を繰り返し使用でき、2 回目以降の検索にかかる処理時間を短縮できます。このパラメタを、? パラメタといいます。また、概念検索で検索条件に指定する種文章は、必ず ? パラメタを使用して指定します。

4.3.2 ロック指定検索

検索を実行する場合に、検索結果として取得する文書管理オブジェクトに対してロックを設定することができます。この検索をロック指定検索といいます。これによって、検索結果集合に含まれる文書管理オブジェクトに対して、ほかのユーザが参照または更新することを防ぐことができます。ロック指定検索を実行すると、検索結果として取得する文書管理オブジェクト全体に、指定した種別のロックが設定されます。ただし、ロック指定検索は、ロックを指定しない場合に比べて処理に時間がかかります。必要に応じて使い分けてください。

なお、ロックを指定しない形式のメソッドを実行した場合、またはロック種別に DbjDef.LOCK_NONE を指定してメソッドを実行した場合は、検索結果集合に含まれる文書管理オブジェクトにロックは設定されません。

ロックについての詳細は、「3.9 排他制御モデル」を参照してください。

4.3.3 アクセス制御機能付き検索

アクセス制御機能を使用している環境の場合、検索結果として取得する文書管理オブジェクトのアクセス制御情報を有効にするか無効にするかを選択できます。アクセス制御情報は、検索対象の文書管理オブジェクトごとに設定されています。また、アクセス制御情報を有効にするか無効にするかの選択は、検索ごとに指定できます。

検索結果集合のアクセス制御情報を有効にする検索を、アクセス制御機能付き検索といいます。

アクセス制御機能付き検索を実行すると、アクセス制御機能を使用しない検索に比べて、処理に時間がかかります。このため、検索結果集合のアクセス制御情報を有効にするかどうかは、業務に応じて決定してください。

また、アクセス制御機能付き検索を実行する場合、次の制限があります。

アクセス制御の対象になる文書管理オブジェクトは、主問い合わせの検索対象である文書管理オブジェクトだけです。したがって、副問い合わせの検索対象である文書管理オブジェクトには、アクセス制御

は適用されません。

アクセス制御は、文書管理オブジェクトを対象に実行されます。この場合は、オブジェクトの識別性に基いた検索になります。したがって、重複排除のようなオブジェクトのプロパティ値に基づく問い合わせや、集合関数のような統計演算をする検索を実行することはできません。

アクセス制御情報の詳細については、「3.8 アクセス制御モデル」を参照してください。

4.3.4 名前付き検索結果を取得する検索

検索結果集合を取得する時に、検索結果集合を表す表に、列名を付けることができます。列名が付いた検索結果集合を、名前付き検索結果といいます。また、列名が付いていない検索結果集合を、名前なし検索結果といいます。なお、名前なし検索を取得した場合も、あとから列名を設定して、名前付き検索結果に変更できます。

名前付き検索結果は、検索結果集合をプロパティ値集合にマッピングする場合に使用できます。例えば、検索結果集合の1行目をプロパティ値集合として取得して、ほかの文書管理オブジェクトのプロパティを更新するために使用したりできます。また、名前付き検索結果を取得すると、検索結果から列名を取得したり、列の値を取得する場合などに列インデクスではなく列名を指定したりすることができます。

名前付き検索結果の列名には、edmSQL文のSELECT句の項目名が設定されます。例えば、SELECT句に、一つ目の項目として「dmaProp_OIID」を、二つ目の項目として「usrProp_Author」というプロパティ名を指定した場合は、取得する検索結果集合の列インデクス [0] の列に「dmaProp_OIID」、列インデクス [1] の列に「usrProp_Author」という列名が設定されます。

なお、名前付き検索結果を取得する検索を実行する場合、SELECT句の選択項目はedmSQL文に直接記述しないで、選択項目のリストをedmSQL文とは別に作成する必要があります。edmSQL文のSELECT句には、「\$」を指定します。

4.3.5 検索結果キャッシュと検索結果取得情報を指定した検索

同じ検索条件の検索を複数回繰り返す場合、検索結果キャッシュを使用することができます。検索結果キャッシュを使用する検索を、キャッシュ検索といいます。また、検索結果キャッシュを使用しない検索は、キャッシュなし検索といいます。

キャッシュ検索では、1回目の検索で取得した検索結果集合を、いったんキャッシュに保管します。ユーザアプリケーションプログラムには、ユーザアプリケーションプログラムで指定した件数だけ返却します。2回目以降の検索では、検索結果を、検索結果キャッシュからユーザアプリケーションプログラムに返却します。

例えば、検索結果が1,000件ある検索を実行する時に、「画面に20件ずつ表示したいので、検索結果は20件ずつ取得したい」という指定ができます。この場合、1回目の検索で1,000件分の検索結果集合がキャッシュに保存され、ユーザアプリケーションプログラムには、そのうち先頭から20件分の検索結果だけが返却されます。2回目以降の検索では、検索結果キャッシュからユーザアプリケーションプログラムに返却されます。「検索結果の21件目から20件を取得したい」という指定をして検索すれば、検索結果キャッシュの21件目から40件目の検索結果が、ユーザアプリケーションプログラムに返却されます。

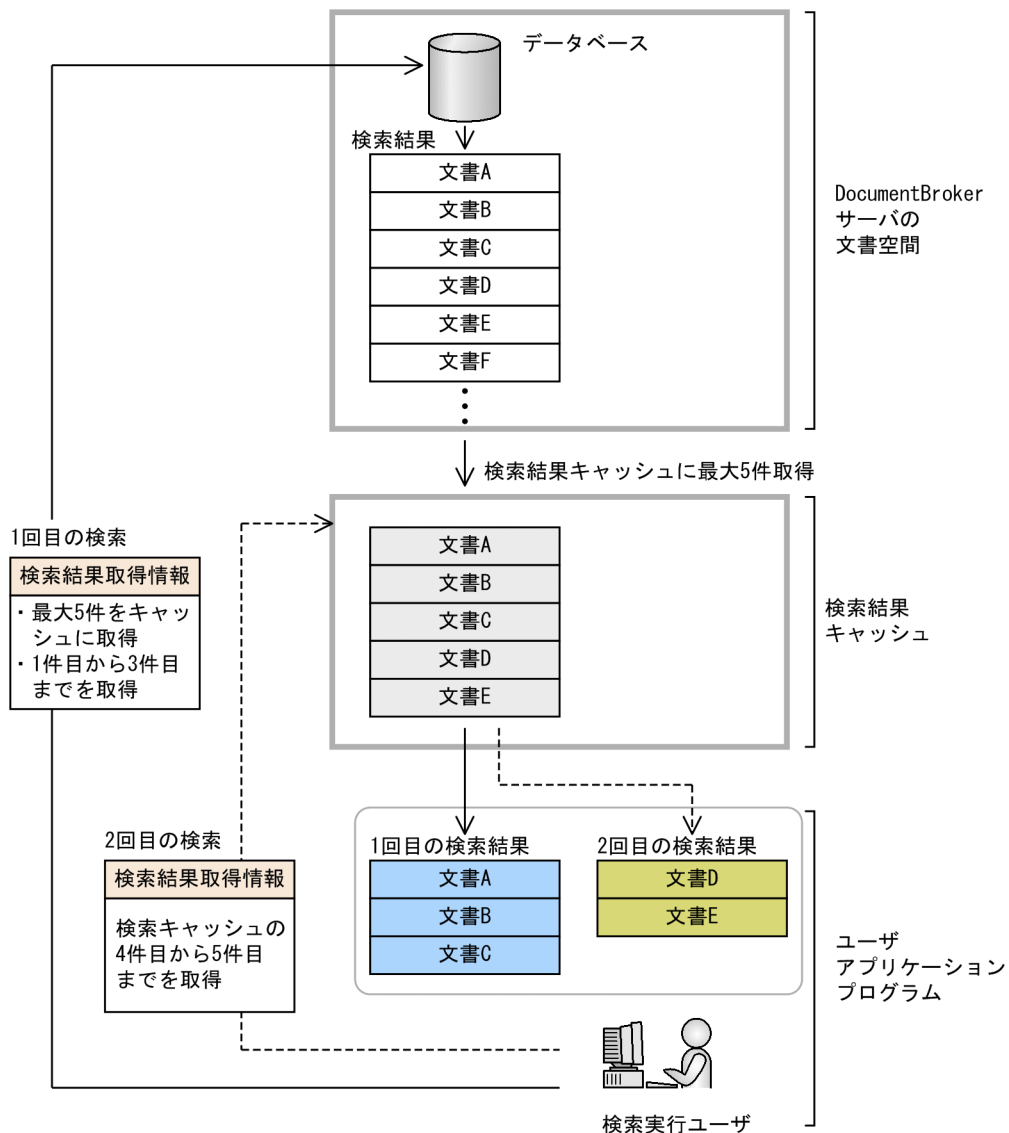
また、検索結果が10,000件ある検索を実行した場合に、「検索結果としては1,000件までしか必要ない」という場合は、検索結果キャッシュとして取得する件数を1,000件までに限定することもできます。

検索結果キャッシュは、キャッシュ名を付けて管理します。キャッシュ名を指定しない場合、検索結果はキャッシュに保持されません。また、検索結果キャッシュはキャッシュ名によって識別されるため、一度に複数の検索結果キャッシュを保持できます。

検索結果キャッシュを使用する場合、「検索結果の何件目から何件目までを取得する」という情報は、検索結果取得情報として指定します。検索結果取得情報に指定した件数分の検索結果が、検索結果キャッシュからユーザアプリケーションプログラムに返却されます。また、キャッシュに取得する件数も、検索結果取得情報に指定します。

検索結果キャッシュと検索結果取得情報を指定した検索の例を、次の図に示します。

図 4-3 検索結果キャッシュと検索結果取得情報を指定した検索の例



(凡例)

→ : 1回目の検索処理の流れ

---> : 2回目以降の検索処理の流れ

また、一覧を取得するメソッドを実行する場合にも、検索結果キャッシュを作成して、検索結果取得情報を指定できます。これによって、例えば、「フォルダに関連付けている文書の一覧を10件ずつ取得する」というような処理が実現できます。

4.4 検索対象になる文書管理オブジェクト

この節では、edmSQL 検索の対象になる文書管理オブジェクトと、検索できるプロパティの詳細について説明します。

4.4.1 検索対象になるクラス

検索対象は、edmSQL 文の FROM 句に指定する項目です。

edmSQL の検索条件には、検索する文書管理オブジェクトの構成要素である DocumentBroker クラスを検索対象として指定します。DocumentBroker クラスは、データベースの表と対応しています。

DocumentBroker クラスを指定すると、文書管理に存在する、その DocumentBroker クラスを構成要素とするすべての文書管理オブジェクトが、検索の対象になります。

例えば、DocumentBroker クラスの usrClass_DocVersion クラス (dmaClass_DocVersion クラスのサブクラス) を検索対象に指定すると、その DocumentBroker 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、バージョン付き文書およびバージョンなし文書 (またはバージョン付き文書の 1 バージョン) がすべて検索対象になります。

検索対象には、複数の DocumentBroker クラスを結合して指定することもできます。edmSQL では、次の種類の結合ができます。

- 内部結合 (INNER JOIN)
- 左外部結合 (LEFT OUTER JOIN)

4.4.2 検索結果として取得できるプロパティ

検索を実行すると、検索結果として、検索条件に合った文書管理オブジェクトのプロパティが取得できません。

取得できるプロパティは、検索対象に指定したクラスに定義されている、DocumentBroker が規定したプロパティ (dmaProp_ または edmProp_ で始まるプロパティ) およびユーザが定義したプロパティなどです。

検索結果として取得するプロパティは、edmSQL 文の SELECT 句に指定します。これによって、SELECT 句に指定したプロパティを列の要素とする検索結果集合が取得できます。データベース上のデータである検索結果は、データ型ごとに対応する DocumentBroker のデータ型に変換されます。これらの値は、副問い合わせの評価値として、述語や演算子にも使用できます。

4.4.3 検索条件に指定できるプロパティ

FROM 句に指定した検索対象に対して、どのような条件を満たす文書管理オブジェクトを取得するかは、検索条件として指定します。また、複数のクラスを検索対象として指定する場合に、どのような条件で結合するかの指定も、検索条件として指定します。

検索条件は、edmSQL 文の WHERE 句や FROM 句の ON 節に指定できます。検索条件には、検索条件の対象になるデータ型に従って、四則演算などを指定するための演算、比較演算子などを指定するための述語、全文検索などを指定するための関数を指定します。

4.5 全文検索の対象になる文書の作成

この節では、全文検索の対象になる文書の作成方法について説明します。

DocumentBroker で全文検索の対象になるのは、次のように作成された文書です。

全文検索に必要なプロパティを持ったクラス（全文検索機能付き文書クラス）を基に作成された文書
文書の内容に対応するテキストデータを基に作成された全文検索用のインデクス（全文検索インデクス）を持つ文書

文書のレンディションタイプによって、全文検索の実行が制限されることはありません。ただし、全文検索用のインデクスの作成方法は、文書のレンディションタイプによって異なります。レンディションタイプごとの全文検索インデクスの作成方法については、「4.5.2 全文検索インデクスの作成」で説明します。

なお、全文検索インデクスは、全文検索に必要なプロパティの一つとして管理されます。

4.5.1 全文検索対象文書の作成手順

全文検索の対象となる文書は、次の手順で作成します。

1. 全文検索機能付き文書クラスを作成します。
全文検索機能付き文書クラスは、DocumentBroker クラスの dmaClass_DocVersion クラスのサブクラスとして作成します。作成方法の詳細については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」の「2.2.4 全文検索機能付き文書クラスの追加」を参照してください。
2. 文書のアップロード情報を作成します。
文書のアップロード情報には、文書のコンテンツとして登録するファイルのほか、全文検索用のインデクスの作成に使用するファイルも指定します。コンテンツとして登録するファイルのレンディションタイプによっては、自動的にインデクスを作成することもできます。詳細は、「4.5.2 全文検索インデクスの作成」を参照してください。
3. 1. で作成した全文検索機能付き文書クラスを指定して、バージョンなし文書またはバージョン付き文書を作成します。メソッド実行時には、2. で作成した文書のアップロード情報も指定します。
バージョンなし文書を作成する場合は、文書のトップオブジェクトクラスを指定する引数に、全文検索機能付き文書クラスのクラス名を指定してメソッドを実行します。
バージョン付き文書を作成する場合は、バージョンオブジェクトのトップオブジェクトクラスを指定する引数に、全文検索機能付き文書クラスのクラス名を指定してメソッドを実行します。

4.5.2 全文検索インデクスの作成

ここでは、全文検索インデクスの作成方法について説明します。全文検索インデクスとは、全文検索機能付き文書クラスに追加した全文検索インデクス用プロパティに格納される、全文検索の対象になるテキストデータです。

全文検索インデクスは、どのレンディションタイプを持つ文書に対して作成するかによって、作成方法が異なります。

(1) 全文検索インデクスの作成時点

全文検索インデクスは、文書作成時や更新時など、文書のコンテンツを登録する時に同時に作成します。

(2) レンディションタイプと全文検索インデクスの作成方法との対応

DocumentBroker では、文書のレンディションタイプに関係なく、全文検索インデクスを持つすべての文

書に対して全文検索を実行できます。ただし、全文検索インデクスの作成方法は、レンディションタイプによって異なります。

レンディションタイプごとの全文検索インデクスの作成方法について説明します。なお、文書がマルチレンディション文書である場合には、マスタレンディションのコンテンツとして登録するファイルのレンディションタイプによって、全文検索インデクスの作成方法が異なります。

特定のレンディションタイプを持つテキストファイルの全文検索インデクスの作成

特定のレンディションタイプとは、次のレンディションタイプです。

- 先頭が「text/」で始まるレンディションタイプ
例えば、「text/plain」や「text/xml」などです。

これらのレンディションタイプを持つ文書の場合、文書の登録時に、登録した文書のコンテンツから全文検索インデクスが作成できます。

特定のレンディションタイプを持たない文書の全文検索インデクスの作成

前に示したレンディションタイプ以外の形式を持つ文書です。ビットマップなどのイメージデータや、Wordなどのアプリケーションプログラムで作成した文書などが該当します。

これらの文書の場合、登録した文書のコンテンツから全文検索インデクスを作成することはできません。この場合は、ユーザアプリケーションプログラムなどで別に作成した全文検索インデクス作成用のテキストデータを文書のコンテンツと同時に登録して、全文検索インデクスを作成する必要があります。

4.5.3 全文検索インデクスの削除

全文検索インデクスだけを削除することはできません。文書管理オブジェクトを削除すると、全文検索インデクスも一緒に削除されます。

付録

付録 A 文書管理オブジェクトのプロパティ一覧

付録 B このマニュアルの参考情報

付録 C 用語解説

付録 A 文書管理オブジェクトのプロパティ一覧

ここでは、DocumentBroker で扱うことができる文書管理オブジェクトのプロパティについて説明します。文書管理オブジェクトのプロパティには、文書管理オブジェクトクラスごとに定義されたプロパティと、DocumentBroker クラスごとに定義されたプロパティがあります。

付録 A.1 文書管理オブジェクトクラスのプロパティ一覧

ここでは、文書管理オブジェクトクラスごとに設定されているプロパティ（dbrProp_ で始まるプロパティ名を持つプロパティ）の一覧を示します。

次に示す項目を説明しています。

プロパティ一覧

文書管理オブジェクト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。

プロパティ名	データ型	編集可能
dbrProp_X	STR 型	

プロパティの一覧表の各項目について説明します。

プロパティ名

プロパティの名前です。

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。

編集可能

編集できるプロパティかまたは読み取り専用のプロパティかを記述しています。

：プロパティの値を更新するメソッドなどで編集できるプロパティであることを示します。

-：読み取り専用のプロパティであることを示します。

(1) バージョンなし文書クラスのプロパティ

バージョンなし文書クラスのプロパティの一覧を次の表に示します。

表 A-1 バージョンなし文書クラスのプロパティ

プロパティ名	データ型	編集可能
dbrProp_ContainersCount	INT 型	-
dbrProp_HeadRelationsCount	INT 型	-
dbrProp_ParentCount	INT 型	-
dbrProp_RenditionStatus	INT 型	-
dbrProp_RenditionType	STR 型	-
dbrProp_RetrievalName	STR 型	
dbrProp_TailRelationsCount	INT 型	-

アクセス制御機能を使用している場合は、表 A-1 のプロパティに加えて、アクセス制御情報を表すプロパティも定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

(2) バージョン付き文書クラスのプロパティ

バージョンングオブジェクトのプロパティの一覧を次の表に示します。

表 A-2 バージョン付き文書クラスのプロパティ (バージョンングオブジェクトのプロパティ)

プロパティ名	データ型	編集可能
dbrProp_ContainersCount	INT 型	-
dbrProp_CurrentVersion	STR 型	-
dbrProp_HeadRelationsCount	INT 型	-
dbrProp_ParentCount	INT 型	-
dbrProp_RenditionType	STR 型	-
dbrProp_RetrievalName	STR 型	-
dbrProp_TailRelationsCount	INT 型	-
dbrProp_VersionsCount	INT 型	-

アクセス制御機能を使用している場合は、表 A-2 のプロパティに加えて、アクセス制御情報を表すプロパティも定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

バージョンオブジェクトのプロパティの一覧を次の表に示します。

表 A-3 バージョン付き文書クラスのプロパティ (バージョンオブジェクトのプロパティ)

プロパティ名	データ型	編集可能
dbrProp_ContainersCount	INT 型	-
dbrProp_HeadRelationsCount	INT 型	-
dbrProp_ParentCount	INT 型	-
dbrProp_RenditionStatus	INT 型	-
dbrProp_RenditionType	STR 型	-
dbrProp_RetrievalName	STR 型	-
dbrProp_TailRelationsCount	INT 型	-

アクセス制御機能を使用している場合は、表 A-3 のプロパティに加えて、アクセス制御情報を表すプロパティも定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

(3) バージョンなしフォルダクラスのプロパティ

バージョンなしフォルダのプロパティを次の表に示します。

表 A-4 バージョンなしフォルダクラスのプロパティ

プロパティ名	データ型	編集可能
dbrProp_ChildrenCount	INT 型	-
dbrProp_ContaineesCount	INT 型	-
dbrProp_ContainersCount	INT 型	-
dbrProp_ParentCount	INT 型	-

アクセス制御機能を使用している場合は、表 A-4 のプロパティに加えて、アクセス制御情報を表すプロパティも定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

また、トップオブジェクトクラスが、dmaClass_Container クラスなど、edmClass_ContainerVersion クラスまたはそのサブクラス以外の場合の、バージョンなしフォルダクラスのプロパティの一覧を次の表に示します。

表 A-5 バージョンなしフォルダクラスのプロパティ (トップオブジェクトクラスが edmClass_ContainerVersion クラスまたはそのサブクラス以外の場合)

プロパティ名	データ型	編集可能
dbrProp_ChildrenCount	INT 型	-
dbrProp_ContainersCount	INT 型	-
dbrProp_ContaineesCount	INT 型	-
dbrProp_ParentCount	INT 型	-

アクセス制御機能を使用している場合は、表 A-5 のプロパティに加えて、アクセス制御情報を表すプロパティも定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

(4) 独立データクラスのプロパティ

アクセス制御機能を使用していない場合、独立データクラスのプロパティはありません。

アクセス制御機能を使用している場合は、アクセス制御情報を表すプロパティが定義されます。詳細については、「(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ」を参照してください。

(5) パブリック ACL クラスのプロパティ

パブリック ACL クラスのプロパティの一覧を次の表に示します。

表 A-6 パブリック ACL クラスのプロパティ

プロパティ名	データ型	編集可能
dbrProp_ACL	VARRAY 型	1
dbrProp_BindObjectCount	INT 型	-
dbrProp_OwnerId	STR 型	2
dbrProp_SACL	VARRAY 型	2
dbrProp_UserPermission	INT 型	-

注 1 編集は、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。

注 2 編集は、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。

(6) アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ

アクセス制御機能を使用する場合に、パブリック ACL 以外のすべての文書管理オブジェクトに追加されるプロパティの一覧を次の表に示します。

表 A-7 アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ

プロパティ名	データ型	編集可能
dbrProp_ACL	VARRAY 型	1
dbrProp_EveryonePermission	INT 型	1
dbrProp_OwnerId	STR 型	2
dbrProp_OwnerPermission	INT 型	1
dbrProp_PrimaryGroupId	STR 型	1
dbrProp_PrimaryGroupPermission	INT 型	1
dbrProp_PublicACLCount	INT 型	-
dbrProp_PublicACLIds	VARRAY 型	1
dbrProp_SACL	VARRAY 型	2
dbrProp_UserPermission	INT 型	-

注 1 編集は、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。

注 2 編集は、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。

付録 A.2 DocumentBroker のクラスのプロパティ一覧

ここでは、プロパティの定義元になる DocumentBroker のクラスのプロパティのうち、DocumentBroker で使用できる主なプロパティの一覧を示します。

クラスごとに、次に示す項目を説明しています。

対応する文書管理オブジェクトクラス

説明する DocumentBroker のクラスと対応する文書管理オブジェクト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

プロパティの一覧表の各項目について説明します。

プロパティ名

プロパティの名前です。

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。

なお、「オブジェクトリファレンス」は、DocumentBroker オブジェクトへのリファレンスを表すデータ型のプロパティです。このプロパティは、edmSQL 文の中だけで使用できます。

DocumentBroker のメソッドで扱うことはできません。

なお、DocumentBroker クラスに定義された dmaProp_ または edmProp_ で始まるプロパティには、DocumentBroker のメソッドで直接編集するプロパティはありません。

(1) dmaClass_ConfigurationHistory クラスのプロパティ

dmaClass_ConfigurationHistory クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス

- バージョン付き文書クラス

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 DocumentBroker クラスです。文書をバージョン管理するとき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 A-8 dmaClass_ConfigurationHistory クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Parent	オブジェクトリファレンス
dmaProp_ParentContainer	オブジェクトリファレンス
dmaProp_PrimaryVersionSeries	オブジェクトリファレンス

(2) dmaClass_Container クラスのプロパティ

dmaClass_Container クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス

バージョンなしフォルダクラス

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 DocumentBroker 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 A-9 dmaClass_Container クラスのプロパティ一覧

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Parent	オブジェクトリファレンス
dmaProp_ParentContainer	オブジェクトリファレンス

(3) dmaClass_DirectContainmentRelationship クラスのプロパティ

dmaClass_DirectContainmentRelationship クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
リンクオブジェクトクラス

クラスの説明

dmaClass_DirectContainmentRelationship クラスは、一つの文書管理内でのオブジェクト間の直接型リンクを規定する DocumentBroker クラスで、dmaClass_ContainmentRelationship クラスのサブクラスです。

直接型リンクでは、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。

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 A-10 dmaClass_DirectContainmentRelationship クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Head	オブジェクトリファレンス
dmaProp_Tail	オブジェクトリファレンス

(4) dmaClass_DocVersion クラスのプロパティ

dmaClass_DocVersion クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
バージョンなし文書クラス

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 DocumentBroker 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、ほかの文書と関連付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 A-11 dmaClass_DocVersion クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス

プロパティ名	データ型
dmaProp_Parent	オブジェクトリファレンス
dmaProp_ParentContainer	オブジェクトリファレンス
dmaProp_CurrentOfSeriesCount	INT 型
edmProp_RenditionsCount	INT 型

(5) dmaClass_ReferentialContainmentRelationship クラスのプロパティ

dmaClass_ReferentialContainmentRelationship クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
リンクオブジェクトクラス

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リンクを規定する DocumentBroker 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 A-12 dmaClass_ReferentialContainmentRelationship クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Head	オブジェクトリファレンス
dmaProp_Tail	オブジェクトリファレンス

(6) edmClass_ContainerVersion クラスのプロパティ

edmClass_ContainerVersion クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
バージョンなしフォルダクラス

クラスの説明

edmClass_ContainerVersion 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 DocumentBroker 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_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 A-13 edmClass_ContainerVersion クラス

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Parent	オブジェクトリファレンス
dmaProp_ParentContainer	オブジェクトリファレンス
dmaProp_CurrentOfSeriesCount	INT 型

(7) edmClass_ContentSearch クラスのプロパティ

edmClass_ContentSearch クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
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 DocumentBroker 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_ConceptTextIndex 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 A-14 edmClass_ContentSearch クラスのプロパティの一覧

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
edmProp_TextIndex	STR 型
edmProp_ConceptTextIndex	STR 型
edmProp_DocLength	INT 型
edmProp_ContentIndexStatus	INT 型

(8) edmClass_IndependentPersistence クラスのプロパティ

edmClass_IndependentPersistence クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
独立データクラス

クラスの説明

edmClass_IndependentPersistence クラスは、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 DocumentBroker のオブジェクトとして扱います。

DocumentBroker で、この DocumentBroker 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します。なお、複数のクラスに同じ GUID を定義することはできません。
- 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します。
- edmClass_IndependentPersistence クラスで定義しているプロパティは、すべて継承します。削除はできません。

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。

BOOL 型

INT 型

STR 型

VARRAY 型

プロパティ値の設定および取得に関する規則

DocumentBroker またはユーザアプリケーションプログラムが定義する GUID か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 A-15 edmClass_IndependentPersistence クラスのプロパティの一覧

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス

(9) edmClass_PublicACL クラスのプロパティ

edmClass_PublicACL クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス

パブリック ACL 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 A-16 edmClass_PublicACL クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス

プロパティ名	データ型
edmProp_OwnerId	STR 型

(10) edmClass_Relationship クラスのプロパティ

edmClass_Relationship クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
リンクオブジェクトクラス

クラスの説明

edmClass_Relationship クラスは、文書間リンクによってオブジェクト間の関連付けを設定するための DocumentBroker 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 A-17 edmClass_Relationship クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Head	オブジェクトリファレンス
dmaProp_Tail	オブジェクトリファレンス
edmProp_RelationType	INT 型

(11) edmClass_VersionTracedDocVersion クラスのプロパティ

edmClass_VersionTracedDocVersion クラスのプロパティについて説明します。

対応する文書管理オブジェクトクラス
バージョンなし文書クラス

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 DocumentBroker 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 A-18 edmClass_VersionTracedDocVersion クラスのプロパティ

プロパティ名	データ型
dmaProp_OIID	STR 型
dmaProp_This	オブジェクトリファレンス
dmaProp_Parent	オブジェクトリファレンス

プロパティ名	データ型
dmaProp_ParentContainer	オブジェクトリファレンス
dmaProp_CurrentOfSeriesCount	INT 型
edmProp_RenditionsCount	INT 型

付録 A.3 各プロパティの説明

ここでは、それぞれのプロパティに設定される値について説明します。

(1) dbrProp_ACL プロパティ

ACL (アクセス制御リスト) を表す ACE (アクセス制御エレメント) のリストです。

(2) dbrProp_BindObjectCount プロパティ

このプロパティを持つパブリック ACL をバインドしているオブジェクトの数です。

(3) dbrProp_ChildrenCount プロパティ

直接型リンクで包含しているオブジェクトの個数です。

(4) dbrProp_ContaineesCount プロパティ

参照型リンクで包含しているオブジェクトの個数です。

(5) dbrProp_ContainersCount プロパティ

参照型リンクでオブジェクトを包含しているフォルダの個数です。

(6) dbrProp_CurrentVersion プロパティ

カレントバージョンのバージョン識別子です。

(7) dbrProp_EveryonePermission プロパティ

すべてのユーザに与えられるパーミッションを表す定数です。

(8) dbrProp_HeadRelationsCount プロパティ

文書間リンクでリンク付けているリンク先文書の個数です。

(9) dbrProp_OwnerId プロパティ

オブジェクトの所有者のユーザ識別子です。

(10) dbrProp_OwnerPermission プロパティ

オブジェクトの所有者に与えられるパーミッションを表す定数です。

(11) dbrProp_ParentCount プロパティ

直接型リンクでオブジェクトを包含しているフォルダの個数です。

(12) dbrProp_PrimaryGroupId プロパティ

プライマリグループのグループ識別子です。

(13) dbrProp_PrimaryGroupPermission プロパティ

プライマリグループに与えられるパーミッションを表す定数です。

(14) dbrProp_PublicACLCount プロパティ

バインドしているパブリック ACL の数です。

(15) dbrProp_PublicACLIds プロパティ

バインドしているパブリック ACL の OIID のリストです。

(16) dbrProp_RenditionStatus プロパティ

レンディションの状態を表す 16 進数です。

(17) dbrProp_RenditionType プロパティ

レンディションタイプを表す文字列です。

(18) dbrProp_RetrievalName プロパティ

登録されているファイルのファイル名を表す文字列です。

(19) dbrProp_TailRelationsCount プロパティ

文書間リンクでリンク付けられているリンク元文書の個数です。

(20) dbrProp_SACL プロパティ

セキュリティ ACL を表す ACE (アクセス制御エレメント) のリストです。

(21) dbrProp_VersionsCount プロパティ

オブジェクトが管理しているバージョンの個数です。

(22) dbrProp_UserPermission プロパティ

オブジェクトに設定される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です。

(23) dmaProp_CurrentOfSeriesCount プロパティ

最新バージョンとしてチェックインしたバージョン付き文書のバージョンオブジェクト (VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト) に対して与えられる番号です。

DocumentBroker 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、次の値が設定されません。

表 A-19 DocumentBroker で設定される dmaProp_CurrentOfSeriesCount プロパティの値

VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクトの状態	dmaProp_CurrentOfSeriesCount プロパティに設定される値
バージョン付き文書のカレントバージョンに対応する VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト	1

VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクトの状態		dmaProp_CurrentOf SeriesCount プロパティに設定される値
チェックアウト中の VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト		-2
チェックイン後の VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト	チェックインによって追加された VersionTracedDocVersion オブジェクトまたは 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 VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト	0
バージョン管理の対象でないバージョンなし文書に含まれる VersionTracedDocVersion オブジェクトまたは DocVersion オブジェクト		-1

(24) dmaProp_Head プロパティ

関連付けられた二つのオブジェクトのうち、Head (リンク先) 側のオブジェクトへのオブジェクトリファレンスです。例えば、リンクでは、包含される側のオブジェクトです。

このプロパティは、リンクのプロパティです。DocumentBroker では、リンク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。

(25) dmaProp_OIID プロパティ

オブジェクトを識別する識別子です。

(26) dmaProp_Parent プロパティ

このオブジェクトを直接型リンクで包含している Parent に相当する DocumentBroker オブジェクトを識別する DirectContainmentRelationship オブジェクトへのオブジェクトリファレンスです。

(27) dmaProp_ParentContainer プロパティ

DirectContainmentRelationship オブジェクトを介してこの DocumentBroker オブジェクトを包含している Container オブジェクトへのオブジェクトリファレンスです。

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。

このプロパティを使用して探索を実行できます。

(28) 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 オブジェクトへのオブジェクトリファレンスです。

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ときに値が設定されます。なお、VersionSeries オブジェクトとは、バージョン管理で使用する DocumentBroker オブジェクトです。

(29) dmaProp_RenditionType プロパティ

文書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
```

DocumentBroker では、次の文書管理オブジェクトクラスオブジェクトの dbrProp_RenditionType プロパティとして操作します。

- バージョンなし文書クラス
- バージョン付き文書クラス

(30) dmaProp_Tail プロパティ

関連付けられた二つのオブジェクトのうち、Tail (リンク元) 側のオブジェクトへのオブジェクトリファレンスです。例えば、リンクでは、包含する側のオブジェクトです。

このプロパティは、リンクのプロパティです。DocumentBroker では、リンク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。

(31) dmaProp_This プロパティ

オブジェクト自身を表すオブジェクトリファレンスです。

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。

DocumentBroker では、リンクの 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、検索実行時に検索対象クラスを結合する場合などに利用します。

(32) 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、全文検索エンジンである HiRDB Text Search Plug-in 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使用されるプロパティです。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。

(33) edmProp_ContentIndexStatus プロパティ

全文検索インデックスの登録状態を示す値です。次の表に示す値が設定されます。

表 A-20 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、全文検索インデックスは未登録です。
2	文書および全文検索インデッ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 以上	DocumentBroker Text Search Index Loader が使用する値が設定されます。

(34) 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。

(35) 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。

DocumentBroker 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。

(36) 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する値です。0 以下の値は設定できません。

(37) 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。

(38) 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、全文検索エンジンである HiRDB Text Search Plug-in 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使用されるプロパティです。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。

付録 B このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 B.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

DocumentBroker のマニュアル

- uCosminexus DocumentBroker Version 5 リファレンス API (3021-3-403)
DocumentBroker クラスライブラリの詳細、インターフェースの詳細、およびメソッドの文法について知りたい場合に参照してください。
- uCosminexus DocumentBroker Version 5 システム導入・運用ガイド (3021-3-401)
データベースに HiRDB を使用して、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。
- uCosminexus DocumentBroker Version 5 メッセージ (3021-3-405)
DocumentBroker が出力するメッセージについて知りたい場合に参照してください。
- uCosminexus DocumentBroker Version 5 サンプル Web アプリケーション (3021-3-404)
DocumentBroker Developer が提供しているサンプル Web アプリケーションの機能と使用方法、およびサンプル Web アプリケーションを参考にした Web アプリケーションの開発方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)
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)

- ノンストップデータベース HiRDB Version 9 UAP 開発ガイド (3020-6-456) ¹
- HiRDB XML 拡張機能 HiRDB XML Extension Version 9 (3020-6-480) ²
- HiRDB 全文検索プラグイン HiRDB Text Search Plug-in Version 9 (3020-6-481) ³

注 1

このマニュアルでは「HiRDB UAP 開発ガイド」と表記しています。

注 2

このマニュアルでは「HiRDB XML Extension」と表記しています。

注 3

このマニュアルでは「HiRDB Text Search Plug-in」と表記しています。

関連製品のマニュアル (その他)

- Hitachi Web Server (3000-3-350)

付録 B.2 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
uCosminexus Application Server Version 9	Cosminexus
uCosminexus Developer Version 9	
uCosminexus DocumentBroker Developer Version 5	DocumentBroker
uCosminexus DocumentBroker Platform Version 5	
uCosminexus DocumentBroker Runtime Library Version 5	
uCosminexus DocumentBroker Developer Version 5	DocumentBroker Developer
uCosminexus DocumentBroker Platform Version 5	DocumentBroker Platform
uCosminexus DocumentBroker Runtime Library Version 5	DocumentBroker Runtime Library
uCosminexus DocumentBroker Text Search Index Loader Version 3	DocumentBroker Text Search Index Loader
HiRDB Server Version 9	HiRDB
HiRDB Text Search Plug-in Conceptual Extension Version 7	HiRDB Text Search Plug-in Conceptual Extension
HiRDB Text Search Plug-in Version 9	HiRDB Text Search Plug-in
HiRDB XML Extension Version 9	HiRDB XML Extension
Red Hat Enterprise Linux 6 (x86_64)	Linux

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 が提供する Java 言語対応のクラスライブラリを DocumentBroker クラスライブラリと表記します。
- Linux を UNIX と表記することがあります。
- Java™ を Java と表記します。
- Enterprise JavaBeans™ を Enterprise JavaBeans と表記します。
- JavaBeans™ を JavaBeans と表記します。
- Java Naming and Directory Interface™ を JNDI と表記します。
- JavaServer Pages™ を JSP と表記します。
- Java™ Servlet を Servlet と表記します。
- Java™ 2 Platform, Enterprise Edition を Java 2 Platform, Enterprise Edition または J2EE と表記します。

付録 B.3 DocumentBroker のマニュアルで使用する略語

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	Access Control Element
ACFlag	Access Control Flag
ACL	Access Control List
API	Application Programming Interface
DMA	Document Management Alliance
DN	Distinguished Name
DTD	Document Type Definition

英略語	英字での表記
GIF	Graphics Interchange Format
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
IANA	Internet Assigned Numbers Authority
ID	Identifier
JIS	Japanese Industrial Standards
LDAP	Lightweight Directory Access Protocol
MIME	Multipurpose Internet Mail Extensions
OIID	Object Instance Identifier
OS	Operating System
PC	Personal Computer
PDF	Portable Document Format
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
UNC	Universal Naming Convention
UTF-8	8-bit UCS Transformation Format
WWW	World Wide Web
XML	Extensible Markup Language

付録 B.4 KB (キロバイト) などの単位表記について

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

付録 C 用語解説

DocumentBroker で使用する用語について説明します。

ここで説明していない DocumentBroker の用語については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

(記号)

?パラメタ

edmSQL 文中のパラメタ値を、検索条件を指定するときには固定しないにおいて、検索実行時にアプリケーションから値を設定するために使用するパラメタです。edmSQL 文中の ? パラメタによって値を渡す個所には、定数を指定する代わりに「?」を指定しておきます。

DocumentBroker では、DbjQParam インターフェースのサブインターフェースによって、? パラメタの値を設定できます。

(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 条件

検索条件同士の論理積を求める結合条件です。例えば、「著者が『日立太郎』で、文書のコンテンツ中に『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です。

ContainerVersion オブジェクト

DocumentBroker クラスの edmClass_ContainerVersion クラスまたはそのサブクラスを基に作成された DocumentBroker オブジェクトです。

Container オブジェクト

DocumentBroker クラスの dmaClass_Container クラス、または edmClass_ContainerVersion クラス以外の dmaClass_Container クラスのサブクラスを基に作成された DocumentBroker オブジェクトです。

dmaClass_ConfigurationHistory クラス

バージョン管理に使用する DocumentBroker クラスです。

dmaClass_ConfigurationHistory クラスまたはそのサブクラスは、バージョンングオブジェクトのトップオブジェクトクラスになります。

dmaClass_Container クラス

オブジェクトをディレクトリに格納するイメージでリンクしてまとめて管理したり、オブジェクトに分類を付けるイメージでリンクして関連付けて管理したりできるフォルダを表す DocumentBroker クラスです。

dmaClass_Container クラスまたはそのサブクラスは、構成管理できないバージョンなしフォルダのトップオブジェクトクラスになります。

dmaClass_DocVersion クラス

文書を表す DocumentBroker クラスです。

dmaClass_DocVersion クラスまたはそのサブクラスは、バージョンなし文書のトップオブジェクトクラスになります。また、構成管理できないバージョン付き文書のバージョンオブジェクトのトップオブジェクトクラスになります。

DocumentBroker オブジェクト

DocumentBroker クラスを基に作成されたオブジェクトです。

DocumentBroker クラス

DocumentBroker のオブジェクトモデルに基づいたクラス、DocumentBroker で定義しているクラスおよびそのサブクラスです。DocumentBroker の文書管理オブジェクトクラスのプロパティは、DocumentBroker クラスに定義します。

DocVersion オブジェクト

DocumentBroker クラスの dmaClass_DocVersion クラス、または edmClass_VersionTracedDocVersion クラス以外の dmaClass_DocVersion クラスのサブクラスを基に作成された DocumentBroker オブジェクトです。

DocumentBroker クラスライブラリ

DocumentBroker にアクセスするユーザアプリケーションプログラムを Java 言語で作成するために、DocumentBroker が提供しているインターフェースです。

edmClass_ContainerVersion クラス

オブジェクトのまとまりをバージョン管理できるフォルダを表す DocumentBroker クラスです。オブジェクトをディレクトリに格納するイメージでリンクしてまとめて管理したり、オブジェクトに分類を付けるイメージでリンクして関連付けて管理したりできるフォルダを表す DocumentBroker クラスです。edmClass_ContainerVersion クラスまたはそのサブクラスは、バージョンなしフォルダのトップオブジェクトクラスになります。

edmClass_IndependentPersistence クラス

独立データを表す DocumentBroker クラスです。

edmClass_IndependentPersistence クラスまたはそのサブクラスは、独立データのトップオブジェクトクラスになります。

edmClass_PublicACL クラス

パブリック ACL を表す DocumentBroker クラスです。文書やフォルダなどのオブジェクトとは独立に作成・参照でき、複数の文書やフォルダなどのオブジェクトから共有できるアクセス制御情報を表すクラスです。パブリック ACL のトップオブジェクトクラスになります。

edmClass_VersionTracedDocVersion クラス

文書を表す DocumentBroker クラスです。

edmClass_VersionTracedDocVersion クラスまたはそのサブクラスは、バージョン付き文書のバージョンオブジェクトのトップオブジェクトクラスになります。

edmSQL

文書管理オブジェクトを検索する場合に使用する、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す。

edmSQL 検索

検索条件に、SQL の文法に基づいた文法で記述できる edmSQL 文を指定して実行する検索のことで。

getter メソッド

パラメタクラスのインターフェースで扱うオブジェクトのプロパティの値を取得するためのメソッドです。

GUID

Globally Unique Identifier の略です。DocumentBroker のクラス、プロパティなどに与えるユニークな識別子です。GUID は、「X」を 0 ~ 9, a ~ f (小文字), および A ~ F (大文字) 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。

MIME 形式

MIME::text/plain など、コンテンツのレンディションタイプを表す形式です。なお、DocumentBroker では、「MIME::」は省略して指定します。

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。データベースに格納された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、各オブジェクトに与えるユニークな識別子です。

OR 条件

検索条件同士の論理和を求める結合条件です。例えば、「作成者が『日立太郎』である文書管理オブジェクトか、作成者の所属が『日立製作所』である文書管理オブジェクトを検索する」という場合、「文書のコンテンツ中に『コンピュータ』という文字列を含むか、『インターネット』という文字列を含む文書を検索する」という場合に使用できます。

Proxy オブジェクト

文書管理オブジェクトを操作する場合に使用する代理オブジェクトです。DocumentBroker では、文書管理オブジェクトを操作するとき、文書管理オブジェクトを直接操作するのではなく、Proxy オブジェクトを操作することで間接的に操作します。

Proxy オブジェクトはメモリ空間に存在します。不要になった場合は、Java のガベージコレクションによって削除されます。

setter メソッド

パラメタクラスのインターフェースで扱うオブジェクトのプロパティの値を設定するためのメソッドです。

VARRAY 型

文書管理オブジェクトのプロパティのデータ型の一つです。VARRAY 型のプロパティは、プロパティのデータ型に従った複数の値を可変長な一次元配列として持ちます。

VersionTracedDocVersion オブジェクト

DocumentBroker クラスの edmClass_VersionTracedDocVersion クラスまたはそのサブクラスを基に作成された DocumentBroker オブジェクトです。

W3C (World Wide Web Consortium)

HTML や WWW に関する技術の標準化を推進する非営利団体です。

(ア行)

アクセス権

文書管理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 の文書空間での文書管理オブジェクトの作成や、管理されている文書やフォルダなどの文書管理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文書管理オブジェクトを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実行する際に、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

文書管理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の文書管理オブジェクトに設定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の値を変更する権利を含みます。

アクセス制御フラグ (ACFlag : Access Control Flag)

文書管理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。

アンバインド

パブリック ACL とのバインドを解除することです。

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全文検索のオプションです。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。

インターフェース

DocumentBroker の機能を実行するためのメソッドが定義されている、クラスの仕様です。DocumentBroker では、オブジェクトを操作するとき、まず、インターフェースを取得して、そのインターフェースで定義されているメソッドを実行するという手順で操作します。

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です。

永続プロパティ

データベースに存在するプロパティです。

オーナーオブジェクトプロパティ

リンク Proxy オブジェクトのプロパティです。ターゲットリンクオブジェクトによってリンクが設定されている文書管理オブジェクトのうち、リンク元である文書管理オブジェクトの Proxy オブジェクトが設定されます。

オブジェクト作成権

オブジェクト作成権限で、文書管理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に指定します。

オブジェクト作成権限

文書管理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、文書管理オブジェクトを作成するメソッドを実行できます。

オブジェクト種別

文書管理オブジェクトの種類を表す定数です。オブジェクト種別には、バージョンなし文書、バージョン付き文書、バージョンなしフォルダ、独立データおよびパブリック ACL があります。

オブジェクト種別プロパティ

Proxy オブジェクトのプロパティです。ターゲットオブジェクトのオブジェクト種別が設定されます。

オブジェクト操作権限

文書空間内のすべての文書管理オブジェクトを、与えられた権限の範囲で操作する権利です。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基本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文書管理オブジェクトのプロパティを参照できます。

オブジェクトリファレンス

DocumentBroker オブジェクトへのリファレンスを示す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。DocumentBroker のメソッドでこの値を使用することはできませんが、`edmSQL` 文中では使用することができます。

(力行)

下位オブジェクト

フォルダを使用した文書管理をする場合に、フォルダからリンクを設定されている、リンク先オブジェクトです。例えば、文書をフォルダに格納する管理をしている場合は、文書が下位オブジェクトです。

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。

可変長配列

VARRAY 型のプロパティの値です。DocumentBroker では、DbjVArray インターフェースで扱います。

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・更新するときに使用します。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。

カレントバージョン

バージョン付き文書が持つ複数のバージョンの中の、最新バージョンのことです。

基本コンテンツ更新権

基本パーミッションの一つで、バージョンなし文書またはバージョン付き文書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。また、全文検索インデックスを作成、削除する権利も含みます。

基本コンテンツ参照権

基本パーミッションの一つで、バージョンなし文書またはバージョン付き文書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。

基本削除権

基本パーミッションの一つで、文書管理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文書管理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文書管理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場合に使用します。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ます）。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、文書管理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、文書管理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。フォルダに対して設定すると、フォルダのリンクをたどることも許可します。

基本リンク権

基本パーミッションの一つで、リンクを設定・解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。

キャッシュキー

キャッシュ付き検索を実行する場合に、2 回目以降の検索条件が前回の検索条件と同じかどうかを判断するためのキーです。検索条件が同じ場合、検索結果キャッシュに対して検索が実行されます。検索条件が異なる場合、検索結果キャッシュは破棄されて、文書空間に対して検索が再度実行されます。

キャッシュ検索

検索結果キャッシュを使用する検索です。文書空間に対して検索を実行したときに、取得した検索結果をキャッシュに保存しておき、同じ検索条件の 2 回目以降の検索では検索結果キャッシュから検索結果を取得します。同じ検索条件で複数回検索を実行する場合に、検索結果取得処理が高速になります。例えば、10,000 件検索結果がある場合に、これを検索結果キャッシュとして保存しておき、ユーザアプリケーションプログラムではこれを 100 件ずつ取得する、という検索ができます。

キャッシュ名

検索結果キャッシュの名称です。検索結果キャッシュは、キャッシュ名によって識別されます。

近傍条件検索

全文検索条件に複数の検索タームを指定した場合に、検索ターム間の距離を条件にして検索する方法です。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場合に使用します。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。

クラスディスクリプション

DocumentBroker クラスに関するメタ情報です。DocumentBroker クラスごとに定義されています。

継承

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。

検索結果キャッシュ

キャッシュ付き検索実行時に作成される検索結果のキャッシュです。キャッシュ付き検索では、例えば、10,000 件の検索結果がある場合に、これを検索結果キャッシュとして保存しておき、ユーザアプリケーションプログラムではこれを 100 件ずつ取得する、という検索ができます。

検索結果集合

検索実行時に作成される、検索条件に一致する文書管理オブジェクトの集合です。要素がプロパティ値である、行と列の二次元データに、列名や列のデータ型などのメタデータを付けたものとして表されます。検索結果集合は、DbjResultSet インターフェースで扱います。

検索結果取得情報

キャッシュ付き検索実行時に、検索結果を何件検索結果キャッシュとして保存するか、また、検索結果キャッシュから何件ユーザアプリケーションプログラムに返却するかを指定する情報です。DbjFetchInfo インターフェースで扱います。

更新系メソッド

操作対象になるオブジェクトの状態を変化させるメソッドです。文書のコンテンツの更新や、文書やフォルダのプロパティの更新を実行するメソッドなどが該当します。

コレクション

java.util.Collection インターフェースの仕様に従うオブジェクトです。要素オブジェクトの集合を表します。

コンテンツ

文書のデータ部分を指します。DocumentBroker で規定されているコンテンツ管理モデルに従って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）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を示します。

コンテンツ情報

文書のコンテンツ管理モデルで管理されているコンテンツに関する情報です。ファイル名およびレンディションタイプが管理されています。DbjContentInfo インターフェースで扱います。

(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。

サブインターフェース

あるインターフェースから派生するインターフェースのことです。上位インターフェースで定義されているメソッドを継承します。

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

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

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。

参照型リンク

一つのオブジェクトから複数のバージョンなしフォルダを親として関連付けるリンクの種別です。一つの文書に対して複数の分類を付けるイメージでの管理を実現します。

参照系メソッド

操作対象になるオブジェクトの状態を変化させないメソッドです。文書のコンテンツの参照や、文書やフォルダのプロパティの参照を実行するメソッドなどが該当します。

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。

サンプル Web アプリケーション

DocumentBroker がサンプルとして提供しているアプリケーションプログラムです。DocumentBroker を使用して開発されたコンポーネントで構成されています。

ユーザアプリケーションプログラムを開発する際に、アプリケーションプログラムのアーキテクチャを参考にできます。また、DocumentBroker のクラス、インターフェース、メソッドなどの具体的な使用方法について参考にできます。

システム管理者

DocumentBroker の運用、管理および保守をするユーザです。DocumentBroker の実行環境を設定することができます。

上位オブジェクト

フォルダを使用した文書管理をする場合のリンク元オブジェクトです。つまり、フォルダのことです。

所有者

文書管理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アクセス権を与えられます。所有者に設定されているユーザは、その文書管理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、その文書管理オブジェクトに対して実行できます。また、その文書管理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。

スーパーインターフェース

あるインターフェースのインターフェース定義に使用されたインターフェースです。派生したインターフェースでは、スーパーインターフェースで定義されているメソッドを継承します。

スーパークラス

あるクラスのクラス定義に使用されたクラスを、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

文書管理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、アクセス権判定を受けることなく、すべての文書管理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文書管理オブジェクトを保守するユーザです。文書空間構成定義ファイルに定義します。

文書空間構成定義ファイル

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、および文書管理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。

セッション

文書空間に接続している間のことです。文書空間に接続することを、セッションの確立といいます。文書空間との接続を解除することを、セッションの切断といいます。

セッションオブジェクト

セッションを確立する機能とセッション内のトランザクションを制御する機能を持つ、DocumentBroker のオブジェクトです。DbjSession インターフェースの機能を実行できます。

セット

java.util.Set インターフェースの仕様に従うオブジェクトです。重複のない、要素オブジェクトの集合を表します。

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

文書を全文検索するために、データベースに登録するインデクスです。全文検索の対象になるテキストデータに対応する edmProp_TextIndex プロパティ、edmProp_ConceptTextIndex プロパティに相当します。

全文検索機能付き文書クラス

dmaClass_DocVersion クラスのサブクラスに全文検索に必要なプロパティを追加したサブクラスです。プロパティの追加は、ユーザが行います。

バージョンなし文書作成時のトップオブジェクトクラスの DocumentBroker クラス名として、またはバージョン付き文書作成時のバージョンオブジェクトのトップオブジェクトクラスの DocumentBroker クラス名として指定します。

edmSQL で全文検索を実行するときには、この DocumentBroker クラスを検索対象として FROM 句に指定します。

属性検索

文書管理オブジェクトのプロパティを対象にした検索です。例えば、「『文書名』が『X』で『作成者』が『A』の文書」のような条件を設定して検索する方法です。

(夕行)

ターゲット OIID プロパティ

Proxy オブジェクトのプロパティです。ターゲットオブジェクトの OIID が設定されます。

ターゲットオブジェクト

Proxy オブジェクトが対象にする文書管理オブジェクトのことです。

ターゲットオブジェクトプロパティ

リンク Proxy オブジェクトのプロパティです。ターゲットリンクオブジェクトが関連付けている二つの文書管理オブジェクトのうち、リンク先オブジェクトの Proxy オブジェクトが設定されます。

例えば、フォルダと文書間の直接型リンクをターゲットリンクオブジェクトとするリンク Proxy オブジェクトの場合、ターゲットオブジェクトプロパティには文書の Proxy オブジェクトが設定されます。

ターゲットバージョン

バージョン付きオブジェクトの複数のバージョンオブジェクトのうち、操作の対象になるバージョンです。

ターゲットバージョン識別子プロパティ

Proxy オブジェクトのプロパティです。ターゲットバージョンであるバージョンオブジェクトのバージョン識別子が設定されます。

ターゲットプロパティ値集合プロパティ

Proxy オブジェクトのプロパティです。文書管理オブジェクトからロードしたプロパティ値集合が設定されます。

ターゲットリンクオブジェクト

リンク Proxy オブジェクトが対象にするリンクオブジェクトのことです。

ターゲットリンク識別子プロパティ

リンク Proxy オブジェクトのプロパティです。ターゲットリンクオブジェクトのリンク識別子が設定されます。

種文章

概念検索の検索条件に指定する文章です。概念検索では、種文章を特徴付ける単語が種文章から抽出され、さらに抽出された特徴タームの中から種文章の概念を表す（実際の検索に使用する）タームが選出されます。このタームが、検索タームとして使用されます。

チェックアウト

バージョン付き文書にバージョンを追加するために、仮のバージョンを追加することです。

チェックアウト情報

バージョン付きオブジェクトのチェックアウトに関する情報です。チェックアウトしているか、チェックアウトしている場合はだれがチェックアウトしているか、また、チェックアウトしたときに設定された仮のバージョン識別子は何かについての情報です。DbjCheckOutInfo インターフェースで扱います。

チェックイン

バージョン付き文書をチェックアウトして追加した仮のバージョンを最新バージョンとして確定することです。

直接型リンク

一つのフォルダを親として複数のオブジェクトを関連付けるリンクの種別です。ディレクトリにファイルを格納するイメージでの管理を実現します。

定数定義クラス

DocumentBroker のメソッドで指定する定数が定義されている DbjDef クラスのことです。

ディレクトリサービス

ネットワーク上にあるユーザや組織の情報などの資源とその属性を記憶し、検索できるようにしたシステムです。

DocumentBroker では、Active Directory や Oracle Directory Server Enterprise Edition などの製品を使用した LDAP 対応のディレクトリサービスと連携できます。

LDAP 対応のディレクトリサービスとして使用できる製品の詳細については、マニュアル「DocumentBroker Version 5 システム導入・運用ガイド」を参照してください。

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。

独立データ

独立したデータを表すオブジェクトです。プロパティだけを持つことができる文書管理オブジェクトです。

edmClass_IndependentPersistence クラスまたはそのサブクラスを基に作成した DocumentBroker オブジェクトをトップオブジェクトとする文書管理オブジェクトです。

特権

アクセス制御機能を利用した文書空間で、アクセス権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。文書空間構成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、ログイン時に文書空間構成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

文書管理オブジェクトを構成する DocumentBroker オブジェクトのうち、代表的な DocumentBroker オブジェクトです。例えば、バージョンなし文書の場合は、DocVersion オブジェクトです。

また、バージョン付きオブジェクトの場合、バージョンングオブジェクトのトップオブジェクトとバージョンオブジェクトのトップオブジェクトがあります。例えば、バージョン付き文書の場合は、ConfigurationHistory オブジェクトと VersionTracedDocVersion オブジェクト（または DocVersion オブジェクト）がトップオブジェクトです。

トップオブジェクトクラス

トップオブジェクトの基になる DocumentBroker クラスです。文書管理オブジェクトのプロパティの定義元になります。

トランザクション

文書管理オブジェクトの処理単位です。トランザクション単位で、文書管理オブジェクトへの操作を確定または取り消すことができます。

トランザクションの単位は、ユーザが明示的に指定できます。指定しなかった場合は、メソッド単位でトランザクションが分割されます。

トレースクラス

DocumentBroker を使用してユーザアプリケーションプログラムを開発・運用する上で必要なトレース情報を出力するメソッドを提供しています。

トレース情報

DocumentBroker で発生した障害の原因を追求するための情報です。

(ナ行)

名前付き検索結果

列名が付いている検索結果集合です。列名には、edmSQL 文で SELECT 句に指定した項目の名前（プロパティ名など）が設定されます。

名前なし検索結果

列名が付いていない検索結果集合です。名前なし検索結果は、あとから列名を追加して、名前付き検索結果集合にすることができます。

(ハ行)

バージョンングオブジェクト

バージョン付きオブジェクトの一連のバージョンを統括する文書管理オブジェクトです。DocumentBroker オブジェクトの ConfigurationHistory オブジェクトをトップオブジェクトとする文書管理オブジェクトです。

バージョンオブジェクト

バージョン付きオブジェクトの個々のバージョンを表す文書管理オブジェクトです。

バージョン付き文書の場合は、DocumentBroker オブジェクトの VersionTracedDocVersion オブジェクト（または DocVersion オブジェクト）をトップオブジェクトとする文書管理オブジェクトです。

バージョン管理

文書管理オブジェクトを更新する場合に、古いオブジェクトを残して新しい状態のオブジェクトを追加して、一連の履歴を管理することです。

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。

バージョン識別子

複数のバージョンを持つ文書管理オブジェクトの、バージョンを識別するための識別子です。特定のバージョンを指定して操作するときに使用します。この識別子は DocumentBroker によってバージョンごとに設定される識別子であり、バージョンオブジェクトの OIID とは異なります。

バージョン付き文書

複数のバージョンを保持できる文書を表す文書管理オブジェクトです。

バージョンなしフォルダ

バージョン管理しないフォルダを表す文書管理オブジェクトです。

バージョンなし文書

バージョン管理しない文書を表す文書管理オブジェクトです。

パーミッション

文書管理オブジェクトの作成、プロパティ参照、コンテンツ更新などの実行できる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。

バインド

文書やフォルダからパブリック ACL を参照することです。

パブリック ACL

文書管理オブジェクトとして存在するアクセス制御情報です。複数の文書管理オブジェクトで共有できます。

パラメタクラス

DocumentBroker 固有のデータをメソッドで受け渡す場合に使用するインターフェースの総称です。

ファクトリオブジェクト

DbjFactory インターフェースの機能を実行するためのオブジェクトです。パラメタクラスのオブジェクトおよびセッションオブジェクトが作成できます。

ファクトリクラス

パラメタクラスのオブジェクトの生成、セッションオブジェクトの生成、および文書空間メタ情報アクセスインターフェースの取得を実行する、クラスならびにインターフェースです。

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。

フラッシュ

Proxy オブジェクトのターゲットプロパティ値集合プロパティに設定した値を、文書管理オブジェクトのプロパティに反映することです。

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。文書管理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

オブジェクトに関する付加情報です。プロパティに設定される値をプロパティ値といいます。

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。

プロパティ値集合

プロパティ名とプロパティ値の対応を要素として持つ集合です。DbjPropSet インターフェースで扱います。

DbjPropSet インターフェースは、java.util.Map インターフェースを継承しています。キーをプロパティ名、値をプロパティ値とするマップとして表されます。

Proxy オブジェクトにロードしたプロパティの値を操作したり、文書管理オブジェクトに設定するためのプロパティの値を設定したりするときに使用します。

プロパティディスクリプション

文書管理オブジェクトのプロパティについてのメタ情報です。プロパティごとに定義されています。

文書

`dmaClass.DocVersion` クラスまたはそのサブクラスをトップオブジェクトクラスとする文書管理オブジェクトです。コンテンツを保持できます。

文書管理クラス

`DocumentBroker` の文書管理モデルに基づいた管理を実現するための機能を提供するインターフェースの総称です。

文書空間

`DocumentBroker` オブジェクトモデルを実装するリポジトリです。

文書空間アクセスオブジェクト

セッションを確立した文書空間を、概念的なオブジェクトとして扱うためのオブジェクトです。`DbjDocSpace` インターフェースの機能を実行できます。

文書管理オブジェクト

`DocumentBroker` の文書管理で使用する、文書空間に存在するオブジェクトの総称です。
バージョンなし文書、バージョン付き文書、バージョンなしフォルダ、独立データおよびパブリック ACL があります。

文書空間識別子

セッションを確立する文書空間を特定するための識別子です。GUID 文字列で表されます。

文書のアップロード情報

文書のコンテンツを更新するときに、コンテンツとして登録するファイルのパス名やファイル名、レンディションタイプなどを設定します。`DbjUploadInfo` インターフェースで扱います。

(マ行)

マップ

`java.util.Map` インターフェースの仕様に従うオブジェクトです。キーと値が対応付けられた要素オブジェクトの集合を表します。

メソッド

データを操作するために定義されている方法です。`DocumentBroker` では、インターフェースおよびクラスごとにメソッドが定義されています。

メタクラス

文書空間のメタ情報を扱うためのインターフェースの総称です。

メッセージ情報

`DocumentBroker` のメソッドを実行してエラーが発生したときに取得できるエラーメッセージです。

(ヤ行)

ユーザ管理システム

`DocumentBroker` にログインするユーザのユーザ名や所属グループなどの情報を管理しているシステムです。LDAP 対応のディレクトリサービスなどが使用できます。

なお、ログイン時の認証に必要な情報、およびアクセス制御機能に必要な情報を取得する場合は、`DocumentBroker` を介してユーザ管理システムにアクセスします。この場合にどのユーザ管理システムと連携するかについては、`DocumentBroker` で定義されています。

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、アクセス権判定に使用されます。

ユーザ定義プロパティ

ユーザが業務に応じて追加するプロパティです。

(ラ行)

ライブラリ情報取得クラス

DocumentBroker のバージョン情報を返却するメソッドを提供するクラスです。

リスト

java.util.List インターフェースの仕様に従うオブジェクトです。順序付けのある要素オブジェクトの集合を表します。

リンク Proxy オブジェクト

リンクオブジェクトを操作する場合に使用する代理オブジェクトです。

リンク Proxy オブジェクトは、メモリ空間に存在します。不要になった場合は、Java のガベージコレクションによって削除されます。

リンクオブジェクト

文書管理オブジェクトの関連付けに使用するオブジェクトです。要素は、dmaClass_Relationship クラスのサブクラスを基に作成された DocumentBroker オブジェクトです。文書空間およびデータベースに存在する永続オブジェクトです。

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。

リンク先オブジェクト

リンクを設定する先になる文書管理オブジェクトです。

リンク識別子

文書管理オブジェクト間（フォルダとフォルダ間またはフォルダと文書間）のリンク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する場合に使用します。

この識別子は関連付けをしたときに DocumentBroker によって設定される識別子です。同じフォルダに、同じオブジェクトを 2 度リンク付けした場合は、それぞれ異なるリンク識別子が設定されます。

リンク種別

リンクの種類です。直接型リンク、参照型リンク、文書間リンクがあります。

リンク設定情報

リンクの設定に関する情報です。リンク種別、リンクオブジェクトのプロパティ、リンク先になるオブジェクトに関する情報などがあります。DbjSetLinkInfo インターフェースのサブインターフェースで表されます。なお、DbjSetLinkInfo インターフェースのサブインターフェースは、リンク種別ごとに存在します。

リンクプロパティ値集合プロパティ

リンク Proxy オブジェクトのプロパティです。リンクオブジェクトからロードしたプロパティ値集合が設定されます。

リンク元オブジェクト

リンクを設定する元になる文書管理オブジェクトです。

例外クラス

DocumentBroker で発生する例外のうち、DocumentBroker 固有の例外を扱うクラスの総称です。java.lang.Exception クラスまたは java.lang.Error クラスを継承しています。

列名

検索結果集合の列に付ける名前です。edmSQL 文の SELECT 句に指定したプロパティ名が設定されます。

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併せた概念です。

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、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します。DocumentBroker では、レンディションタイプとして、MIME 形式を指定することを推奨しています。

レンディション定義ファイル

ファイルの拡張子とレンディションタイプ (MIME 形式) の対応を定義するファイルです。DocumentBroke がコンテンツとして登録するファイルの拡張子によってレンディションタイプ (MIME 形式) を自動的に判別して設定する場合に参照されます。

ローカル ACL

文書管理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。VARRAY 型のプロパティとして設定されます。

ロード

文書管理オブジェクトのプロパティを、Proxy オブジェクトのターゲットプロパティ値集合プロパティに読み込むことです。

ログアウト

文書空間とのセッションを切断することです。

ログイン

文書空間とのセッションを確立することです。ログインするときには、ユーザ識別子とパスワードによる認証処理も実行されます。

ロック指定検索

検索結果集合として取得した文書管理オブジェクトに、指定した種別のロックを設定する検索です。

ロック種別

排他制御を実行するために設定する、ロックの種類です。read ロックと write ロックがあります。

ロック種別プロパティ

Proxy オブジェクトのプロパティです。その Proxy オブジェクトのインターフェースのメソッドで設定するロック種別が設定されます。

(ワ行)

ワイルドカード

文字の代わりに指定する記号です。検索する単語の一部しかわからない場合、わかっている部分にワイルドカードを付けて検索条件のキーワードとして指定します。

- 以上 -

索引

記号

- ? パラメタ 119, 144
- ? パラメタを使用した検索 119

A

- ACE 79, 144
- ACFlag 144
- ACFlag およびローカル ACL を使用した運用例 100
- ACFlag として設定する情報 77
- ACFlag を使用したアクセス制御の運用例 99
- ACL 144
- ACL として設定する情報 79
- AND 条件 144
- API (Application Programming Interface) 144

C

- ConfigurationHistory オブジェクト 35
- ContainerVersion オブジェクト 63, 144
- Container オブジェクト 63, 144

D

- dbrProp_ACL:dbrProp_Permission 85, 90
- dbrProp_ACL:dbrProp_Subject 85, 90
- dbrProp_ACL:dbrProp_SubjectType 85, 90
- dbrProp_ACL プロパティ 136
- dbrProp_BindObjectCount 91
- dbrProp_BindObjectCount プロパティ 136
- dbrProp_ChildrenCount プロパティ 136
- dbrProp_ContaineesCount プロパティ 136
- dbrProp_ContainersCount プロパティ 136
- dbrProp_ContentType プロパティ 52
- dbrProp_CurrentVersion プロパティ 136
- dbrProp_EveryonePermission 85
- dbrProp_EveryonePermission プロパティ 136
- dbrProp_GroupCount 84
- dbrProp_GroupList 84
- dbrProp_OwnerId 85, 90
- dbrProp_OwnerId プロパティ 136
- dbrProp_OwnerPermission 85
- dbrProp_OwnerPermission プロパティ 136
- dbrProp_ParentCount プロパティ 136
- dbrProp_PrimaryGroupId 85
- dbrProp_PrimaryGroupId プロパティ 136
- dbrProp_PrimaryGroupPermission 85
- dbrProp_PrimaryGroupPermission プロパティ 137

- dbrProp_PublicACLCount 85
- dbrProp_PublicACLCount プロパティ 137
- dbrProp_PublicACLIds:dbrProp_ACLIdElem 85
- dbrProp_PublicACLIds プロパティ 137
- dbrProp_RenditionStatus プロパティ 44
- dbrProp_RenditionType プロパティ 43, 137
- dbrProp_RetrievalName プロパティ 43, 137
- dbrProp_SACL:dbrProp_Permission 85, 91
- dbrProp_SACL:dbrProp_Subject 85, 90
- dbrProp_SACL:dbrProp_SubjectType 85, 90
- dbrProp_SACL プロパティ 137
- dbrProp_TailRelationsCount プロパティ 137
- dbrProp_UserId 84
- dbrProp_UserPermission 84, 85, 91
- dbrProp_UserPermission プロパティ 137
- dbrProp_UserPrivilege 84
- dbrProp_VersionsCount プロパティ 137
- dmaClass_ConfigurationHistory クラス 144
- dmaClass_ConfigurationHistory クラスのプロパティ 130
- dmaClass_Container クラス 144
- dmaClass_Container クラスのプロパティ 130
- dmaClass_DirectContainmentRelationship クラスのプロパティ 131
- dmaClass_DocVersion クラス 145
- dmaClass_DocVersion クラスのプロパティ 131
- dmaClass_ReferentialContainmentRelationship クラスのプロパティ 132
- dmaProp_CurrentOfSeriesCount プロパティ 137
- dmaProp_Head プロパティ 138
- dmaProp_OIID 90
- dmaProp_OIID プロパティ 138
- dmaProp_ParentContainer プロパティ 138
- dmaProp_Parent プロパティ 138
- dmaProp_PrimaryVersionSeries プロパティ 138
- dmaProp_RenditionType プロパティ 138
- dmaProp_Tail プロパティ 139
- dmaProp_This プロパティ 139
- DocumentBroker オブジェクト 145
- DocumentBroker クラス 145
- DocumentBroker クラスライブラリ 145
- DocumentBroker クラスライブラリの特長 2
- DocumentBroker で実現する機能 4
- DocumentBroker とは 2
- DocumentBroker のオブジェクト 15
- DocumentBroker の概要 1
- DocumentBroker のクラス 17

DocumentBroker のクラスのプロパティ一覧 129
 DocumentBroker の前提プログラム 8
 DocumentBroker の操作で使用する概念 14
 DocumentBroker の特長 2
 DocumentBroker を使用する場合のシステム構成 8
 DocVersion オブジェクト 36, 38, 54, 145

E

edmClass_ContainerVersion クラス 145
 edmClass_ContainerVersion クラスのプロパティ 132
 edmClass_ContentSearch クラスのプロパティ 133
 edmClass_IndependentPersistence クラス 145
 edmClass_IndependentPersistence クラスのプロパティ 133
 edmClass_PublicACL クラス 145
 edmClass_PublicACL クラスのプロパティ 134
 edmClass_Relationship クラスのプロパティ 135
 edmClass_VersionTracedDocVersion クラス 145
 edmClass_VersionTracedDocVersion クラスのプロパティ 135
 edmProp_ConceptTextIndex プロパティ 139
 edmProp_DocLength プロパティ 140
 edmProp_OwnerId プロパティ 140
 edmProp_RelationType プロパティ 140
 edmProp_RenditionsCount プロパティ 140
 edmProp_TextIndex プロパティ 140
 edmSQL 110, 145
 edmSQL 検索 110, 145
 edmSQL文 110

G

getter メソッド 145
 GUID 146

I

IndependentPersistence オブジェクト 66

J

Java の環境 10

L

LDAP 対応のディレクトリサービスとの連携を使用したプロパティの管理例 69

M

MIME 形式 21, 146

N

NOT 条件 146

O

OIID 146
 OR 条件 146

P

Proxy オブジェクト 146

R

read ロック 104

S

setter メソッド 146

V

VARRAY 型 146
 VARRAY 型のプロパティの管理例 68
 VersionTracedDocVersion オブジェクト 35, 146

W

W3C 146
 WAIT モード 105
 write ロック 104
 WWW ブラウザ 11

あ

アクセス権 27, 146
 アクセス権の判定に使用される情報 70
 アクセス制御 27
 アクセス制御エレメント (ACE:Access Control Element) 146
 アクセス制御機能 6, 147
 アクセス制御機能付き検索 119, 147
 アクセス制御機能を使用する場合に追加される DocumentBroker 固有のプロパティ 128
 アクセス制御情報 27, 147
 アクセス制御情報に設定できるパーミッションの種類 72
 アクセス制御情報の管理 82
 アクセス制御情報の構成要素のプロパティ 89
 アクセス制御情報の使い分け 98
 アクセス制御情報へのアクセス権の設定 82
 アクセス制御情報変更権 82, 147
 アクセス制御情報を設定するためのプロパティ 84

アクセス制御フラグ (ACFlag:Access Control Flag) 147
 アクセス制御モデル 70
 アクセス制御モデルで使用するプロパティ 84
 アクセス制御モデルの運用例 98
 アクセス制御リスト (ACL:Access Control List) 147
 アンバインド 80, 147

い

異表記展開 116
 異表記展開検索 147
 インターフェース 147

え

永続オブジェクト 147
 永続プロパティ 147

お

オーナーオブジェクトプロパティ 147
 オブジェクト作成権 73, 147
 オブジェクト作成権限 71, 147
 オブジェクト種別 148
 オブジェクト種別プロパティ 148
 オブジェクト操作権限 71, 148
 オブジェクトとデータベースの関係 16
 オブジェクトの操作とアクセス制御情報の関係 93
 オブジェクトリファレンス 148
 重み 117
 重み指定 117

か

下位オブジェクト 56, 62, 148
 下位オブジェクトの削除 65
 概念検索 114, 148
 可変長配列 148
 仮のバージョン識別子 148
 カレントバージョン 33, 148

き

基本コンテンツ更新権 73, 148
 基本コンテンツ参照権 73, 148
 基本削除権 74, 148
 基本バージョン管理権 74, 148
 基本パーミッション 73, 149
 基本パーミッションの設定例 74
 基本プロパティ更新権 73, 149
 基本プロパティ参照権 73, 149

基本リンク権 74, 149
 キャッシュキー 112, 149
 キャッシュ検索 120, 149
 キャッシュなし検索 120
 キャッシュ名 112, 120, 149
 近傍条件 117
 近傍条件検索 149

く

組み合わせパーミッション 75, 149
 クラスディスクリプション 149
 グループ識別子 71

け

継承 149
 検索結果 110
 検索結果キャッシュ 112, 120, 150
 検索結果キャッシュと検索結果取得情報を指定した検索 120
 検索結果集合 111, 150
 検索結果取得情報 113, 121, 150
 検索結果として取得できるプロパティ 122
 検索条件 110
 検索条件に指定できるプロパティ 122
 検索ターム 114
 検索タームを指定する全文検索 114
 検索対象になるクラス 122
 検索対象になる文書管理オブジェクト 122
 検索の実行方法の種類 119
 検索の種類 114
 検索用特徴ターム 115

こ

更新系メソッド 150
 コレクション 150
 コンテンツ 37, 150
 コンテンツ格納先パス 53
 コンテンツ格納先ベースパス 5, 53
 コンテンツ種別 52, 150
 コンテンツ情報 150
 コンテンツロケーション 5
 コンパレータ 113

さ

削除権 76, 150
 サブインターフェース 150
 サブクラス 150
 サブジェクト 79, 91, 92, 150

サブジェクト種別 92, 150
サブレンディション 37
参照型リンク 25, 62, 150
参照系メソッド 151
参照権 76, 151
参照更新権 76, 151
サンプル Web アプリケーション 3, 151

し

システム管理者 151
システムサブジェクト 89
上位オブジェクト 56, 62, 151
上位オブジェクトの削除 65
状態フラグ 44
状態フラグの値 44
所属グループ 71
所有者 78, 151
処理の流れ 12

す

スーパーインターフェース 151
スーパークラス 151

せ

セキュリティ ACL 83, 151
セキュリティ ACL を使用した運用例 103
セキュリティ管理者 71, 84, 151
セキュリティ管理者およびセキュリティ運用者の設定
84
セキュリティ定義ファイル 71
セッション 151
セッションオブジェクト 152
セット 152
全文検索 114, 152
全文検索インデクス 123, 152
全文検索インデクスの削除 124
全文検索インデクスの作成 123
全文検索インデクスの作成時点 123
全文検索機能付き文書クラス 123, 152
全文検索機能付き文字列型プロパティを使用した全文
検索 117
全文検索対象文書の作成手順 123
全文検索の種類 115
全文検索の対象になる文書の作成 123

そ

属性管理モデル 67
属性検索 114, 152

た

ターゲット OIID プロパティ 152
ターゲットオブジェクト 152
ターゲットオブジェクトプロパティ 152
ターゲットバージョン 152
ターゲットバージョン識別子プロパティ 152
ターゲットプロパティ値集合プロパティ 152
ターゲットリンクオブジェクト 152
ターゲットリンク識別子プロパティ 153
種文章 114, 153

ち

チェックアウト 32, 153
チェックアウト情報 153
チェックイン 32, 153
直接型リンク 25, 62, 153

て

定数定義クラス 153
ディレクトリサービス 10, 153
デッドロック 106

と

同義語展開 117
同義語展開検索 153
独立データ 16, 26, 153
独立データ管理モデル 66
独立データクラスのプロパティ 128
独立データを管理する機能 6
独立データを構成する DMA オブジェクト 66
特権 71, 153
トップオブジェクト 15, 153
トップオブジェクトクラス 15, 154
トランザクション 154
トレースクラス 154
トレース情報 154

な

名前付き検索結果 120, 154
名前付き検索結果を取得する検索 120
名前なし検索結果 120, 154

は

バージョンングオブジェクト 17, 20, 154
バージョンオブジェクト 17, 20, 154
バージョン管理 20, 32, 154
バージョン管理機能 4

- バージョン管理権 154
- バージョン管理の概要 32
- バージョン管理モデル 32
- バージョン権 76
- バージョン識別子 36, 154
- バージョン付きオブジェクト 17
- バージョン付きオブジェクトクラス 18
- バージョン付きオブジェクトとバージョンなしオブジェクトの関係 33
- バージョン付きオブジェクトのプロパティの管理例 67
- バージョン付き文書 16, 19, 155
- バージョン付き文書クラスのプロパティ 127
- バージョン付き文書による文書管理 36
- バージョン付き文書を構成する DocumentBroker オブジェクト 34
- バージョンなしオブジェクト 17
- バージョンなしオブジェクトクラス 18
- バージョンなしフォルダ 16, 25, 155
- バージョンなしフォルダクラスのプロパティ 127
- バージョンなしフォルダによる管理モデル 62
- バージョンなしフォルダによる文書管理 63
- バージョンなしフォルダによる文書管理の概要 62
- バージョンなしフォルダを構成する DocumentBroker オブジェクト 62
- バージョンなし文書 16, 19, 155
- バージョンなし文書およびバージョンなしフォルダのプロパティの管理例 67
- バージョンなし文書クラスのプロパティ 126
- バージョンの指定方法 33
- バージョンの追加 32
- パーミッション 72, 155
- パーミッションの種類 72
- 排他制御 29
- 排他制御機能 7
- 排他制御モデル 104
- バインド 80, 155
- パブリック ACL 16, 80, 155
- パブリック ACL クラスのプロパティ 128
- パブリック ACL のプロパティ 90
- パブリック ACL を使用した運用例 102
- パラメタクラス 155

ふ

- ファクトリオブジェクト 155
- ファクトリクラス 155
- フォルダ 16, 25
- フォルダオブジェクトクラス 18

- 複数のオブジェクトに write ロックを設定するメソッドの使用上の注意 106
- プライマリグループ 78, 155
- フラッシュ 155
- フルコントロール 76, 155
- プレーンテキスト 115
- プロパティ 155
- プロパティ更新権 76, 155
- プロパティ参照権 76, 155
- プロパティ値集合 155
- プロパティディスクリプション 156
- プロパティの管理例 67
- プロパティの説明 136
- 文書 16, 19, 155
- 文書オブジェクトクラス 18
- 文書管理オブジェクト 156
- 文書管理オブジェクトクラスと DocumentBroker クラスの対応 17
- 文書管理オブジェクトクラスのプロパティ一覧 126
- 文書管理オブジェクトごとのアクセス制御情報 72
- 文書管理オブジェクトごとのアクセス制御情報の種類 77
- 文書管理オブジェクトと DocumentBroker オブジェクトの関係 15
- 文書管理オブジェクトの種類 16
- 文書管理オブジェクトのプロパティ一覧 126
- 文書管理クラス 156
- 文書管理モデル 31
- 文書間リンク 23
- 文書間リンクの設定 59
- 文書間リンクをたどる参照 60
- 文書空間 156
- 文書空間 (データベース) に接続する機能 4
- 文書空間アクセスオブジェクト 156
- 文書空間構成定義ファイル 151
- 文書空間識別子 156
- 文書のアップロード情報 156
- 文書やフォルダを検索する機能 6
- 文書を作成, 管理する機能 4

へ

- 変換フラグ 44
- 変換フラグの値 45

ま

- マスタレンディション 37
- 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の更新 41
- マップ 156

マルチレンディション機能 5

め

メソッド 156
メタクラス 156
メッセージ情報 156

ゆ

ユーザ管理システム 156
ユーザ権限 71, 157
ユーザ権限定義ファイル 157
ユーザ識別子 71
ユーザ情報 27, 70, 157
ユーザ情報を表すプロパティ 84
ユーザ定義プロパティ 157

ら

ライブラリ情報取得クラス 157

り

リスト 157
リファレンスファイル管理 22
リファレンスファイル管理機能 5
リファレンスファイル文書 22
リファレンスファイル文書の管理の概要 53
リファレンスファイル文書の管理モデル 53
リファレンスファイル文書を構成する
DocumentBroker オブジェクト 54
リンク 16, 23
リンク Proxy オブジェクト 157
リンクオブジェクト 23, 157
リンクオブジェクトを構成する DocumentBroker オ
ブジェクト 56
リンク権 76, 157
リンク先オブジェクト 56, 157
リンク先文書 23
リンク先文書の削除 60
リンク識別子 57, 58, 157
リンク種別 157
リンク設定情報 157
リンクによる文書管理 59
リンクによる文書管理の概要 56
リンクの解除 60, 64
リンクプロパティ値集合プロパティ 158
リンクモデル 56
リンク元オブジェクト 56, 158
リンク元文書 23
リンク元文書の削除 60

リンクをたどる文書管理オブジェクトの参照 64

れ

例外クラス 158
列名 158
連携できるプログラム 10
レンディション 21, 158
レンディション管理 21
レンディション管理の概要 37
レンディション管理モデル 37
レンディション状態を表すプロパティ 44
レンディションタイプ 158
レンディション定義ファイル 158
レンディションによる文書管理 39
レンディションの検索 42
レンディションのコンテンツの管理 37
レンディションの削除 41
レンディションの参照 41
レンディションの種類 37
レンディションの追加 40
レンディションの登録 40
レンディションのプロパティ 42
レンディションを管理する文書を構成する
DocumentBroker オブジェクト 38

ろ

ローカル ACL 80, 158
ロード 158
ログアウト 158
ログイン 158
ロックが設定されているオブジェクトに対する接続
105
ロックが設定される範囲 106
ロック指定検索 119, 158
ロック種別 104, 158
ロック種別の変更 106
ロック種別プロパティ 158
ロックに関する注意事項 106
ロックによる排他制御 105

わ

ワイルドカード 159