

uCosminexus DocumentBroker Version 5 システム導入・運用ガイド

解説・手引書

3021-3-401-10

対象製品

R-1595F-13 uCosminexus DocumentBroker Platform 05-10 (適用 OS : Windows Server 2008 R2 , Windows Server 2012)
R-1L95F-13 uCosminexus DocumentBroker Platform 05-10 (適用 OS : Red Hat Enterprise Linux 6 (x86_64))
R-1595F-43 uCosminexus DocumentBroker Developer 05-10 (適用 OS : Windows Server 2008 R2 , Windows Server 2012 , Windows 7 (x86) , Windows 7 (x64) , Windows 8 (x86) , Windows 8 (x64))
R-1595F-53 uCosminexus DocumentBroker Runtime Library 05-10 (適用 OS : Windows Server 2008 R2 , Windows Server 2012 , Windows 7 (x86) , Windows 7 (x64) , Windows 8 (x86) , Windows 8 (x64))
R-1L95F-53 uCosminexus DocumentBroker Runtime Library 05-10 (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。
Microsoft Office Word は、米国 Microsoft Corporation の商品名称です。
Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft Word は、米国 Microsoft Corporation の商品名称です。
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。
Microsoft および Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
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。
UNIX は、The Open Group の米国ならびに他の国における登録商標です。
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) Office Word	Word	
Microsoft(R) Word		
Windows(R) 8 Pro (32 ビット版)	Windows 8	Windows
Windows(R) 8 Enterprise (32 ビット版)		
Windows(R) 8 Pro (64 ビット版)		
Windows(R) 8 Enterprise (64 ビット版)		
Microsoft(R) Windows(R) 7 Professional 日本語版 (32 ビット版)	Windows 7	
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (32 ビット版)		
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (32 ビット版)		
Microsoft(R) Windows(R) 7 Professional 日本語版 (64 ビット版)		
Microsoft(R) Windows(R) 7 Enterprise 日本語版 (64 ビット版)		
Microsoft(R) Windows(R) 7 Ultimate 日本語版 (64 ビット版)		

製品名称	略称
Microsoft(R) Windows Server(R) 2012 Datacenter 日本語版	Windows Server 2012
Microsoft(R) Windows Server(R) 2012 Standard 日本語版	
Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版	Windows Server 2008 R2
Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版	

発行

2013 年 4 月 3021-3-401-10

著作権

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

All Rights Reserved. Copyright (C) 2012, 2013, Hitachi Solutions, Ltd.

変更内容

変更内容 (3021-3-401-10)

追加・変更内容	変更箇所
入力ストリームを使用した文書操作ができるようになりました。	4.3.1 , 4.3.2
オブジェクトに設定されているユーザ , グループの情報を一括で変更できるようになりました。	4.12 , 7.3 dbjchgusrid
IT Report Utility を利用した保守情報の採取ができるようになりました。	6.4.4
フォルダ一覧を出力できるようになりました。	7.3 dbjlsfldr , 7.4.1
文書一覧を出力できるようになりました。	7.3 dbjlsdoc , 7.4.2
前提 OS に Windows Server 2012 および Windows 8 を追加しました。	-

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示すプログラムプロダクトの環境設定および運用方法について説明したものです。

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

対象読者

このマニュアルは、uCosminexus DocumentBroker のシステムの導入・運用について説明したものです。

uCosminexus DocumentBroker の機能を解説した、マニュアル「DocumentBroker Version 5 概説」の内容を踏まえた上でご利用ください。

- Windows に関する知識
- HiRDB に関する知識
- UNIX に関する知識

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概説

uCosminexus DocumentBroker の概要およびシステム構成について説明しています。

第 2 章 システム導入前の検討

uCosminexus DocumentBroker を利用した文書管理システムの導入前に必要な検討項目について説明しています。

第 3 章 環境設定

uCosminexus DocumentBroker の環境設定について説明しています。

第 4 章 環境設定で必要なファイル

uCosminexus DocumentBroker の環境設定で必要なファイルについて説明しています。

第 5 章 DocumentBroker の起動と終了

uCosminexus DocumentBroker の起動方法および終了方法について説明しています。

第 6 章 運用と障害対策

運用方法および障害対策について説明しています。

第 7 章 コマンドリファレンス

コマンドの文法について説明しています。

付録 A システムクラスおよびシステムプロパティの名称の規則

uCosminexus DocumentBroker のシステムクラスおよびシステムプロパティの名前、表識別子、および列名について説明しています。

付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス

ユーザ管理システムのアクセスルーチンとして DocumentBroker が提供するインターフェースとクラスについて説明しています。

付録 C DocumentBroker Version 5 へのバージョンアップ手順

DocumentBrokerVersion 3 以前のバージョンから、DocumentBrokerVersion 5 へバージョンアップする手順について説明しています。

付録 D このマニュアルの参考情報

関連マニュアルや表記など、参考情報について説明しています。

付録 E 用語解説

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

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。
—	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) [A 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。
< >	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。
...	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ユーザ指定値の内容を表す記号です。

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

構文要素記号	意味
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	概説	1
1.1	DocumentBroker の機能	2
1.1.1	文書の登録機能とバージョン管理機能	2
1.1.2	フォルダ管理機能	3
1.1.3	文書の属性情報の管理機能	4
1.1.4	マルチレンディション管理機能	4
1.1.5	文書間リンク機能	6
1.1.6	リファレンスファイル管理機能	6
1.1.7	レンディションのコンテンツ種別変換機能	7
1.1.8	独立データの管理機能	9
1.1.9	アクセス制御機能	9
1.1.10	検索機能	11
1.1.11	全文検索インデクス一括登録機能	13
1.1.12	レンディション変換機能	13
1.1.13	複数の実行環境機能	13
1.2	DocumentBroker のシステム構成	14
1.2.1	DocumentBroker を使用したシステムの概略	14
2	システム導入前の検討	17
2.1	適用業務の検討	18
2.2	追加するクラスおよびプロパティの検討	19
2.2.1	クラスおよびプロパティの追加	19
2.2.2	サブクラスの追加	20
2.2.3	プロパティの追加	21
2.2.4	全文検索機能付き文書クラスの追加	22
2.2.5	クラスおよびプロパティの追加例	24
2.3	文書空間で使用する文字コード種別の検討	25
2.4	仮想メモリ所要量の見積もり	26
2.5	データベース容量の見積もり	27
2.5.1	データベース容量の見積もり方法	27
2.5.2	ユーザ用 RD エリアの容量の見積もり	28
2.5.3	ユーザ LOB 用 RD エリアの容量の見積もり	32
2.5.4	アクセス制御機能に関連するデータベース容量の見積もり	35
2.6	リファレンスファイル管理機能を使用する場合のファイルシステムのディスク容量の見積もり	37
3	環境設定	39
3.1	環境設定の流れ	40
3.2	環境設定の準備	41

3.2.1	システム管理者の決定	41
3.2.2	利用ユーザおよびグループの決定	41
3.2.3	DocumentBroker の実行環境を作成するディレクトリの決定	41
3.3	インストールとアンインストール (UNIX の場合)	42
3.3.1	インストール	42
3.3.2	アンインストール	43
3.3.3	インストール済みプログラムの一覧表示	43
3.4	インストールとアンインストール (Windows の場合)	45
3.4.1	統合 CD-ROM からのインストール	45
3.4.2	リモートインストール	46
3.4.3	アンインストール	47
3.5	データベースシステムでの環境設定	48
3.5.1	HiRDB の環境設定の準備	48
3.5.2	HiRDB の環境設定	48
3.5.3	HiRDB のユーザ権限の設定	48
3.5.4	HiRDB Text Search Plug-in , HiRDB XML Extension での環境設定	49
3.6	ユーザ管理機能の設定	50
3.6.1	LDAP 対応のディレクトリサービスによるユーザ管理機能を使用する場合の設定	50
3.6.2	ユーザ作成のアクセスルーチンを使用する場合の設定	53
3.7	DocumentBroker の実行環境の作成	54
3.7.1	DocumentBroker 実行環境ディレクトリの作成	54
3.7.2	DocumentBroker の環境変数の設定	54
3.7.3	文書空間の定義	54
3.8	DocumentBroker でデータベースシステムを使用するための設定	60
3.8.1	データベースシステムの設定に必要なファイル	60
3.8.2	データベースシステムの設定手順	60
3.8.3	全文検索機能を使用する場合に出力されるデータベース定義文	62
3.9	アプリケーションサーバの設定	66
3.9.1	アプリケーションサーバの構築	66
3.9.2	セキュリティポリシーの設定	66
3.9.3	保護区リストファイルの設定	66
3.9.4	コンテナ拡張ライブラリの取り込み	67
3.9.5	J2EE サーバの起動	67
3.9.6	リソースアダプタの設定	67
3.10	Java アプリケーションの設定	70
3.10.1	環境変数の設定	70
3.10.2	JAR ファイルの追加	70
3.10.3	データベース情報ファイルの設定	70
3.11	複数の実行環境を構築する場合の設定	71
3.11.1	新規に実行環境を構築する場合	71
3.11.2	実行環境を変更する場合	71
3.11.3	実行環境を削除する場合	71

3.12	リファレンスファイル管理機能を使用する場合の設定	72
3.12.1	コンテンツを管理するための設定	72
3.12.2	ネットワーク上のマシンの共有ディスクでコンテンツを管理する場合の設定	74

4

環境設定に必要なファイル	77	
4.1	ファイルの種類	78
4.1.1	プロパティファイル	78
4.1.2	コマンドで使用するファイル	79
4.2	動作環境定義ファイル (docbconf.properties)	81
4.2.1	動作環境定義ファイルの概要	81
4.2.2	動作環境定義ファイルの記述形式	82
4.3	文書空間構成定義ファイル (docspace.properties)	85
4.3.1	文書空間構成定義ファイルの概要	85
4.3.2	文書空間構成定義ファイルの記述形式	87
4.3.3	文書空間構成定義ファイルの記述例	96
4.3.4	文書空間構成定義ファイルの注意事項	99
4.4	ユーザ権限定義ファイル (userperm.properties)	102
4.4.1	ユーザ権限定義ファイルの概要	102
4.4.2	ユーザ権限定義ファイルの記述形式	102
4.4.3	ユーザ権限定義ファイルの記述例	103
4.5	レンディション定義ファイル (mime.properties)	104
4.6	メタ情報初期化ファイル (docbmetainit.properties)	105
4.6.1	メタ情報初期化ファイルの記述形式	105
4.7	データベース情報ファイル (databaseinfo.properties)	106
4.7.1	データベース情報ファイルの記述形式	106
4.8	メタ情報ファイル	107
4.9	定義情報ファイル	108
4.9.1	定義情報ファイルの概要	108
4.9.2	サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則	108
4.9.3	定義情報ファイルの記述形式	109
4.9.4	定義情報ファイルの記述例	112
4.9.5	定義情報ファイルの記述例 (VariableArray 型のプロパティを追加する場合)	113
4.9.6	定義情報ファイルの記述例 (全文検索機能を使用する文書クラスを追加する場合)	115
4.9.7	定義情報ファイルの記述例 (文字列型プロパティに対する全文検索機能を使用する場合)	115
4.10	RD エリア定義情報ファイル	118
4.10.1	RD エリア定義情報ファイルの概要	118
4.10.2	RD エリア定義情報ファイルの記述形式	118
4.10.3	RD エリア定義情報ファイルの記述例	119
4.10.4	RD エリア定義情報ファイルの注意事項	120
4.11	インデクス情報ファイル	121
4.11.1	インデクス情報ファイルの概要	121

4.11.2	インデクス名をデータベース定義の名称に使用する場合の規則	121
4.11.3	インデクス情報ファイルの記述形式	121
4.11.4	インデクス情報ファイルの記述例	122
4.12	変更情報ファイル	123
4.12.1	変更情報ファイルの概要	123
4.12.2	変更情報ファイルの記述規則	123
4.12.3	ユーザ識別子変更情報ファイル	124
4.12.4	グループ識別子変更情報ファイル	125
4.12.5	プロパティ変更情報ファイル	126

5

DocumentBroker の起動と終了	129
5.1 DocumentBroker の起動方法	130
5.2 DocumentBroker の終了方法	131

6

運用と障害対策	133
6.1 データベースの運用	134
6.1.1 バックアップとリストア	134
6.1.2 ジャーナルファイルの運用	137
6.1.3 レプリケーション	137
6.2 クラス定義の変更	139
6.2.1 クラスの定義を変更する場合について	139
6.2.2 クラスの追加	139
6.2.3 クラスの削除	140
6.2.4 クラスの定義の変更	141
6.3 リファレンスファイル管理機能を使用する場合の運用	143
6.3.1 コンテンツの格納先ディレクトリの運用方法	143
6.3.2 ファイルシステムのバックアップとリストア	144
6.4 障害対策	145
6.4.1 障害が発生した時に DocumentBroker が出力する情報	145
6.4.2 障害要因の特定	146
6.4.3 障害への対処	146
6.4.4 IT Report Utility を利用した保守情報の採取	147

7

コマンドリファレンス	149
7.1 コマンドの種類	150
7.1.1 コマンド一覧	150
7.2 コマンドの形式	151
7.3 コマンドの文法	153
7.4 コマンドが出力するファイル	166
7.4.1 dbjlsfldr コマンドが出力するファイル	166

7.4.2 dbjlsdoc コマンドが出力するファイル	167
------------------------------	-----

付録 171

付録 A システムクラス, システムプロパティおよびデータベース定義の名称規則	172
付録 A.1 システムクラスの名称定義の規則	172
付録 A.2 システムプロパティの名称の規則	172
付録 A.3 データベース定義の名称定義の規則	175
付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス	176
付録 B.1 jp.co.Hitachi.soft.docbroker.auth.DbjAuthModule インターフェイス	177
付録 B.2 jp.co.Hitachi.soft.docbroker.auth.DbjAuthUserInfo インターフェイス	179
付録 B.3 jp.co.Hitachi.soft.docbroker.auth.DbjAuthException クラス	180
付録 C DocumentBroker Version 5 へのバージョンアップ手順	181
付録 D このマニュアルの参考情報	188
付録 D.1 関連マニュアル	188
付録 D.2 このマニュアルでの表記	189
付録 D.3 DocumentBroker のマニュアルで使用する略語	190
付録 D.4 KB (キロバイト) などの単位表記について	191
付録 E 用語解説	192

索引 205

1

概説

この章では , DocumentBroker の概要について説明します。

1.1 DocumentBroker の機能

1.2 DocumentBroker のシステム構成

1.1 DocumentBroker の機能

この節では、DocumentBroker の機能について説明します。

DocumentBroker の文書管理システムで提供する機能の一覧を次の表に示します。

表 1-1 DocumentBroker の文書管理システムで提供する機能の一覧

項番	分類	機能	機能の実現に必要な関連製品とユティリティ製品	参照先
1	文書管理	文書の登録機能	-	1.1.1
2		バージョン管理機能	-	
3		フォルダ管理機能	-	1.1.2
4		文書の属性情報の管理機能	-	1.1.3
5		マルチレンディション管理機能	-	1.1.4
6		文書間リンク機能	-	1.1.5
7		リファレンスファイル管理機能	-	1.1.6
8		レンディションのコンテンツ種別変換機能	-	1.1.7
9	独立データの管理	独立データの管理機能	-	1.1.8
10	アクセス制御	アクセス制御機能	LDAP 対応のディレクトリサービス	1.1.9
11	検索	属性検索	-	1.1.10
12		全文検索	<ul style="list-style-type: none"> • HiRDB Text Search Plug-in, または HiRDB XML Extension 	
13		概念検索	<ul style="list-style-type: none"> • HiRDB Text Search Plug-in, または HiRDB XML Extension • HiRDB Text Search Plug-in Conceptual Extension 	
14		文字列型プロパティに対する全文検索	<ul style="list-style-type: none"> • HiRDB Text Search Plug-in, または HiRDB XML Extension 	
15	変換・登録	全文検索インデクス一括登録機能	DocumentBroker Text Search Index Loader	1.1.11
16		レンディション変換機能	-	1.1.12
17	環境構築	複数の実行環境機能	-	1.1.13

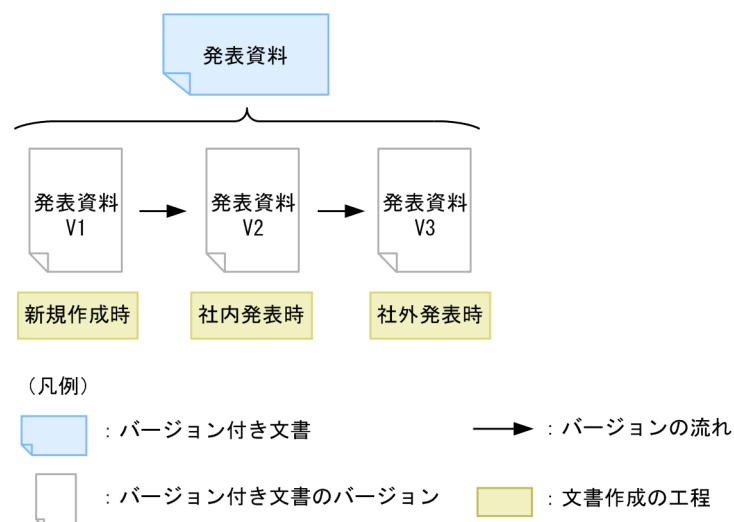
1.1.1 文書の登録機能とバージョン管理機能

文書をデータベースに登録して一元管理できます。また、文書を登録する際には、必要に応じて「Version1」や「Version2」などの版（バージョン）を付けて、文書の履歴を管理できます。文書のバージョンを管理することで、過去のある時点での文書を復元したり、その文書を流用して新たに文書を作成したりできます。一つのバージョンだけを持ち、バージョンを管理しない文書を、バージョンなし文書と

います。複数のバージョンを持ち、バージョンを管理する文書を、バージョン付き文書といいます。

バージョン付き文書の管理例を次の図に示します。

図 1-1 バージョン付き文書の管理例



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。

文書のバージョンを管理するときには、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いきます。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。チェックアウト中はチェックアウトしたユーザ以外は文書を上書きできないため、ほかのユーザによる文書の二重更新を防止できます。

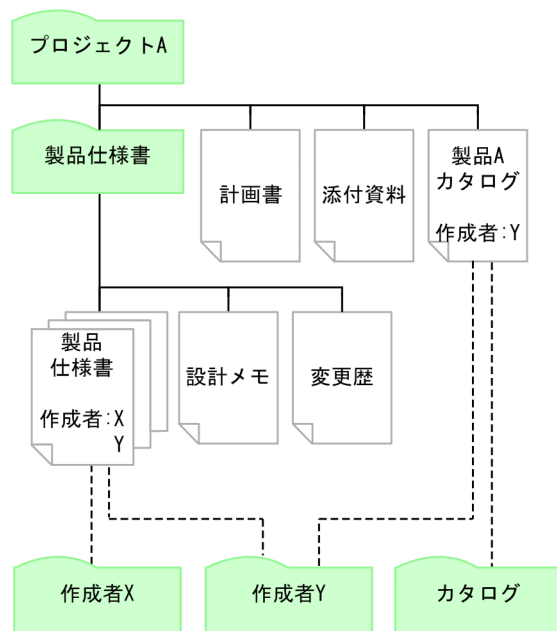
1.1.2 フォルダ管理機能

文書をまとめて格納するフォルダや、文書を分類するフォルダを利用して文書を管理できます。文書をまとめたり、分類したりするフォルダに相当するオブジェクトを、フォルダといいます。

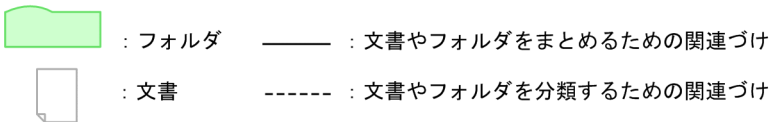
フォルダには、複数の文書またはフォルダを関連づけられます。フォルダを使用すると、複数の文書を目的に応じて一つにまとめて管理したり、一つの文書を複数の観点から分類して管理したりできます。また、フォルダとフォルダを関連づけることで、フォルダや分類に階層を持たせることもできます。

フォルダによる文書の管理例を次の図に示します。

図 1-2 フォルダによる文書の管理例



(凡例)



この図では、「プロジェクトA」というフォルダで、このプロジェクトに関連する文書をまとめて管理しています。「製品仕様書」は、製品仕様書、設計メモおよび変更歴で構成されます。したがって、「プロジェクトA」フォルダの下位に「製品仕様書」というフォルダを作成して、関連する文書をまとめて格納して管理しています。

また、「作成者X」フォルダ、「作成者Y」フォルダおよび「カタログ」フォルダは、文書を分類するためのフォルダです。「プロジェクトA」に関連する文書、「プロジェクトA」の製品仕様書を構成する文書」というまとめ方とは別の視点で、文書をまとめています。したがって、製品Aカタログや製品仕様書がプロジェクトAに関連する文書であることを知らないユーザでも、文書を分類しているフォルダからたどることで、文書を見つけやすくなります。

1.1.3 文書の属性情報の管理機能

文書やフォルダなどのオブジェクト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。プロパティには、DocumentBroker によってあらかじめ定義されているプロパティ（システムプロパティ）と、ユーザが任意に追加定義するプロパティ（ユーザプロパティ）があります。例えば、文書を管理する場合、ユーザプロパティとして、「文書名」や「作成日時」などの標準的な属性情報だけでなく、「顧客名」や「競合他社名」などの業務に応じた属性情報も一緒に管理できます。

文書やフォルダにプロパティを定義すると、プロパティをキーにしてオブジェクトを検索したり、プロパティの値を参照してオブジェクトの状態を確認したりできます。

1.1.4 マルチレンディション管理機能

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、その形式を表すレンディションタイプ (MIME 形式) の情報を合わせて、レンディションといいます。

バージョンなし文書とバージョン付き文書には、1個または複数のレンディションを登録して管理できます。一つの文書に対して同じ内容を表す複数のレンディションを登録して管理する機能のことを、マルチレンディション管理機能といいます。また、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文書は、一つの文書の内容を、対応するアプリケーションごとの複数の形式に変換した場合などに使用できます。

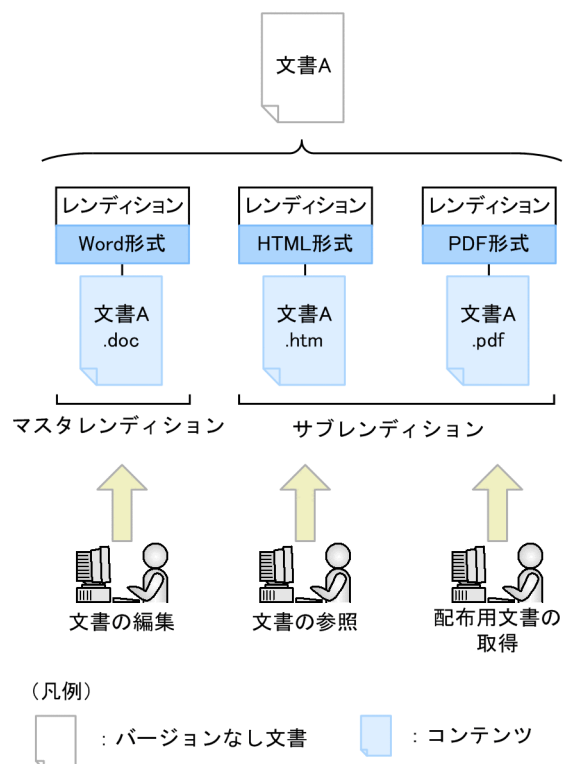
マルチレンディション管理機能では、次の2種類のレンディションを区別して扱います。

- 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。
- サブレンドーション
マスタレンディション以外のレンディションです。

なお、このマニュアルでは、2種類のレンディションを合わせて、レンディションと呼びます。レンディションの種類によって説明が異なる場合は、それぞれ「マスタレンディション」、「サブレンドーション」と記述して区別します。

マルチレンディション文書の管理例を次の図に示します。

図 1-3 マルチレンディション文書の管理例



この図では、マルチレンディション文書のコンテンツとして、同じ内容を表す複数の形式のファイルを登録しています。文書を編集するユーザは Word 形式のファイル、Word がインストールされていないマシンで文書を参照するユーザは HTML 形式のファイル、配布用の文書を取得したいユーザは PDF 形式のファイルをそれぞれダウンロードできます。このように、マルチレンディション文書では、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、適切なレンディションを指定して文書を利用

できます。

また、マルチレンディション管理機能とアクセス制御機能を組み合わせて、マルチレンディション文書を管理できます。例えば、Word 形式のファイルと PDF 形式のファイルを登録して、文書を更新できるユーザには Word 形式のファイルを参照させて、文書の参照だけできるユーザには PDF 形式のファイルを参照させるというように、ユーザに設定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。アクセス制御機能については、「1.1.9 アクセス制御機能」を参照してください。

1.1.5 文書間リンク機能

文書と文書を関連づけて管理する機能を、文書間リンクといいます。

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 は、二つの文書を文書間リンクという概念を利用して関連づけます。この場合、文書間リンクを設定した文書をリンク元文書、文書間リンクを設定された文書をリンク先文書といいます。

一つの文書間リンクでは、一つのリンク元文書と一つのリンク先文書を関連づけられます。また、一つの文書に対して、複数の文書間リンクを設定できます。一つの文書に対して複数の文書から文書間リンクを設定することもできます。

1.1.6 リファレンスファイル管理機能

リファレンスファイル管理機能では、文書の実体であるコンテンツを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。

ファイルシステムには、次のどれかを使用してください。

- DocumentBroker が存在するマシン上のローカルディスクまたは共有ディスク
- UNIX の場合
ネットワークファイルシステム（NFS）を利用してアクセス可能な共有ディスク
- Windows の場合

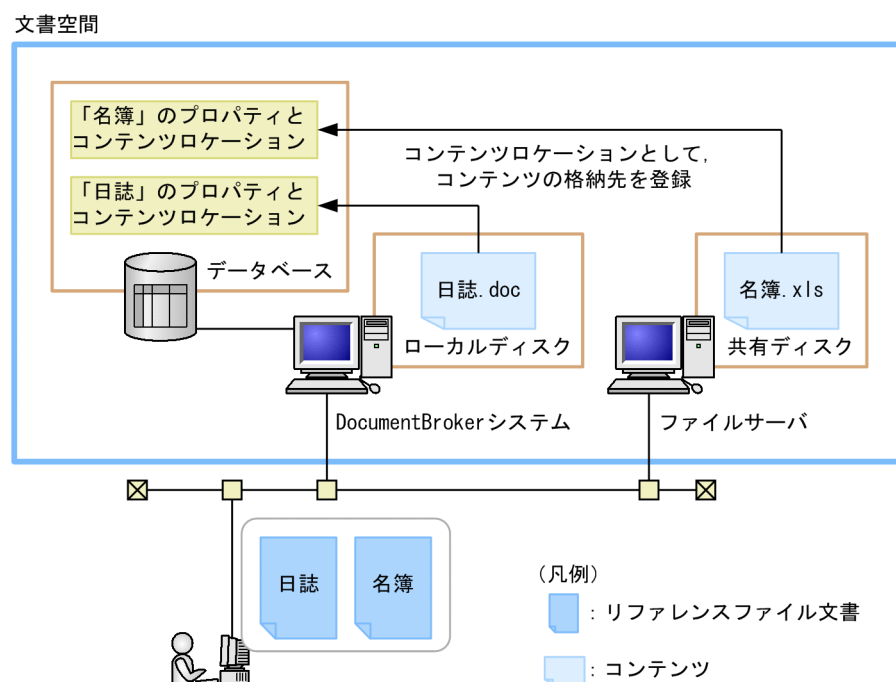
DocumentBroker が存在するマシンから UNC 形式のパスを使用して接続可能な共有ディスク

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。このため、コンテンツの格納先を移行する場合などは、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。

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。バージョンなし文書およびバージョン付き文書は、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 1-4 リファレンスファイル文書の管理例



この図では、ファイルシステムとして、DocumentBroker が存在するマシンのローカルディスクを利用してリファレンスファイル文書を管理する場合と、ファイルサーバの共有ディスクを利用してリファレンスファイル文書を管理する場合の例を示しています。ファイルシステムにコンテンツを格納し、データベースにプロパティとコンテンツロケーションを登録して管理しています。このように、リファレンスファイル管理機能では、データベースにコンテンツを格納しないので、コンテンツの格納先を選択したり、コンテンツの格納先のディスクを移行したりと、柔軟にコンテンツの格納先を管理できます。

また、ファイルシステムをネットワーク上のマシンの共有ディスクとする場合は、コンテンツ格納先ベースパスを、UNC 形式のパスで指定してコンテンツを操作します。

1.1.7 レンディションのコンテンツ種別変換機能

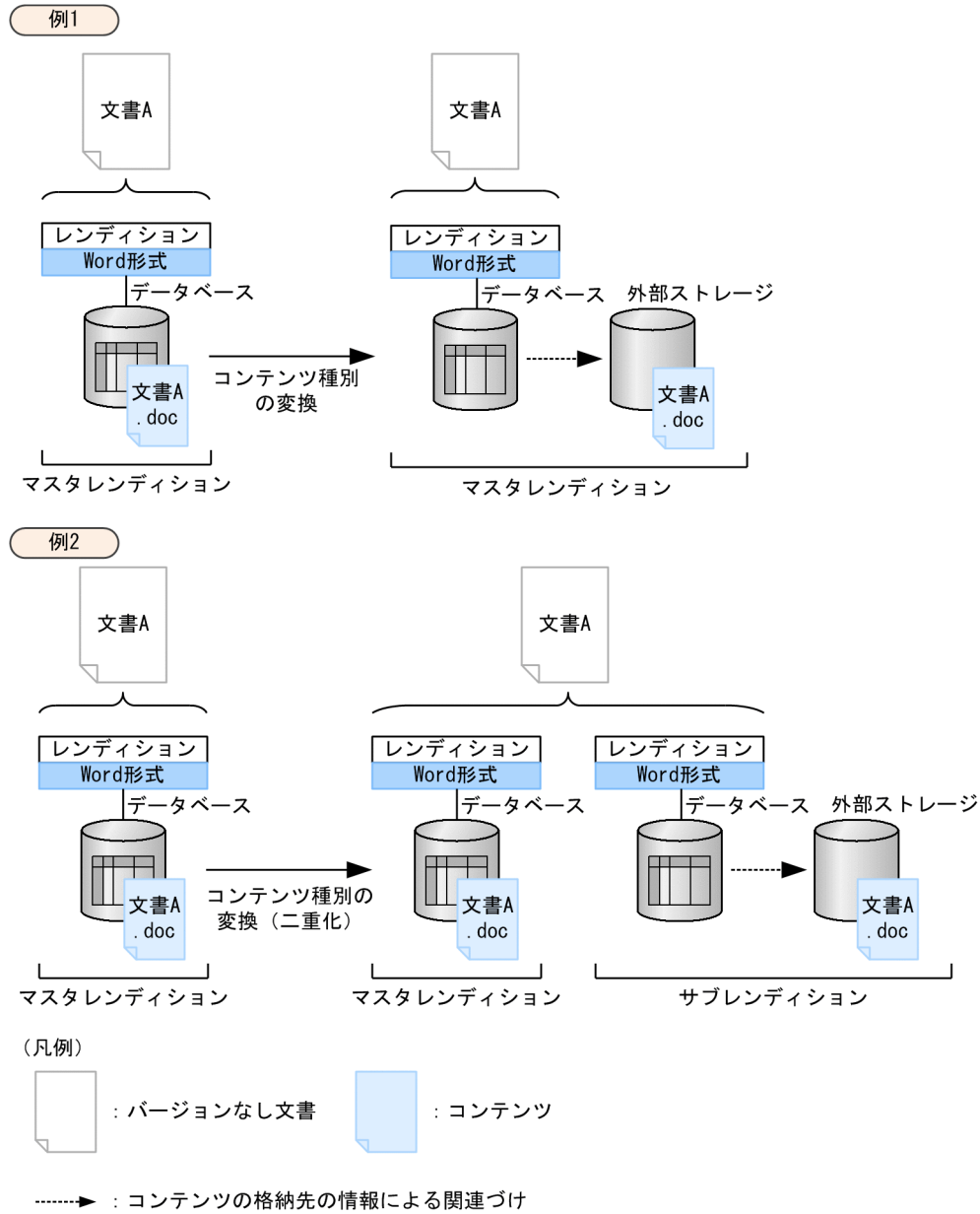
レンディションのコンテンツ種別変換機能は、レンディションのコンテンツの格納先（コンテンツ種別）を変換する機能です。

DocumentBroker では、文書の実体であるコンテンツをデータベースや DocumentBroker が接続可能なファイルシステムの任意のディレクトリなどに格納して管理できます。これらの格納先で管理されるコンテンツを、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できます。

コンテンツ種別を、シングルファイル文書（データベースに登録されている一つのファイルをコンテンツとして持つ文書）から、リファレンスファイル文書（DocumentBroker から接続可能なファイルシステムの任意のディレクトリに格納されているファイルをコンテンツとして持つ文書）に変換したコンテンツは、磁気テープなどのストレージへのバックアップや移動など、コンテンツ管理の柔軟な運用を行うことができます。

レンディションのコンテンツ種別変換機能を使用して、コンテンツ種別を変換する例を次の図に示します。

図 1-5 レンディションのコンテンツ種別変換の例



コンテンツ種別の変換は、文書のマルチレンディション管理機能を使用して、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。

例2では、シングルファイル文書のコンテンツを、リファレンスファイル文書のコンテンツに変換し、レンディションを追加します。これをコンテンツの二重化と呼びます。コンテンツを二重化する場合、追加するレンディションのレンディションタイプにコメントを付与して格納先を分類します。なお、コンテンツを二重化したあとにレンディションを操作する場合は、マルチレンディション管理機能を使用します。

(1) 変換できるコンテンツ種別

レンディションのコンテンツ種別変換機能で変換できるコンテンツ種別を次の表に示します。

表 1-2 レンディションのコンテンツ種別変換機能で変換できるコンテンツ種別

変換前のコンテンツ種別	変換後のコンテンツ種別
シングルファイル文書	リファレンスファイル文書
リファレンスファイル文書	シングルファイル文書

レンディションのコンテンツ種別は、レンディションのコンテンツ種別を表すプロパティを参照することで判別できます。プロパティの詳細については、マニュアル「DocumentBroker Version 5 概説」を参照してください。

(2) 注意事項

コンテンツ種別を変換して、サブレンディションとして追加するときには、追加するレンディションに対してレンディションコメントを付与します。コンテンツの変換前、または変換後のどちらかのレンディションに対してコメントを付与できませんが、変換したコンテンツであることが明確になるように変換後のコンテンツに対して、コンテンツ種別を表すコメントを付けることをお勧めします。

ただし、DocumentBroker のユティリティ製品を使用した運用で、変換後のコンテンツをマスタレンディションとする場合などでは、変換前のレンディションに対してコメントを付与できません。

レンディションのコンテンツ種別変換機能を使用する場合に、DocumentBroker のユティリティ製品を使用するときの注意事項を次に示します。

- DocumentBroker Text Search Index Loader
レンディションタイプを前方一致で扱います。レンディションのコンテンツ種別変換機能を使用した運用では、レンディション定義ファイル中で該当するレンディションタイプを前方一致で定義します。ただし、DocumentBroker Text Search Index Loader では、リファレンスファイル文書のテキスト抽出はできません。

1.1.8 独立データの管理機能

文書やフォルダなどのほかのオブジェクトの体系に関係しない、独立したデータを管理するオブジェクトのことを独立データといいます。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。

独立データは、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できません。

1.1.9 アクセス制御機能

DocumentBroker では、DocumentBroker が管理するメモリ空間（文書空間）にユーザがログインするときに生成されるユーザ情報や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。文書空間にログインするときのユーザ認証、オブジェクトへのアクセス制御は、ユーザ管理システムとの連携によって実現します。

(1) ユーザ認証

DocumentBroker では、LDAP 対応のディレクトリサービスと連携したユーザ認証機能を提供しています。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。このとき、ログインしたユーザに関する情報を取得して、ユーザ情報を生成します。

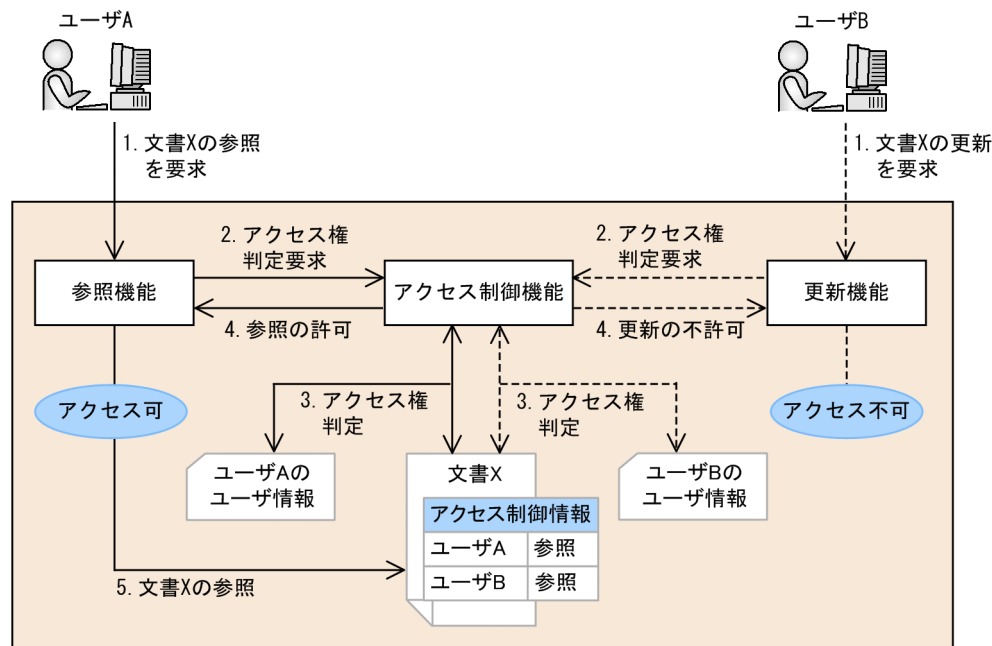
(2) オブジェクトに対するアクセス制御

DocumentBroker で管理しているオブジェクトに対して、アクセス権を設定できます。例えば、文書に対してアクセス権を設定すると、アクセスを許可されていないユーザが誤って文書を更新してしまうようなことはありません。また、特定の人やグループだけに文書の参照や編集を許可するような運用もできます。

DocumentBroker が提供するアクセス制御機能は、ユーザやグループに対してアクセス権を与えたり、アクセス権を変更したりできます。例えば、文書の作成、審査、承認といった作業工程に応じてアクセス権を設定できます。これによって、文書のライフサイクル全体を管理できます。

文書にアクセスする場合の、アクセス制御の概要を次の図に示します。

図 1-6 アクセス制御の概要



(凡例)

——> : ユーザAの参照要求の処理

-----> : ユーザBの更新要求の処理

DocumentBroker のアクセス制御は、アクセス制御対象オブジェクトへアクセスする場合、次に示す二つの情報をアクセス制御機能によって判定し、アクセス制御を実行します。

ログインしたユーザのユーザ情報

アクセス制御対象オブジェクトのアクセス制御情報

文書の操作を要求した場合、操作要求の内容とユーザ情報を比較してアクセス権を判定します。その結果、アクセス権があるときは、要求された文書の操作を実行します。しかし、アクセス権がないときは、「アクセス権なし」というエラーが返却されます。図 1-6 では、ユーザ A が文書の参照を要求しています。

DocumentBroker は、ユーザ A のログインによって作成されたユーザ情報と文書 X に設定されているアクセス制御情報とを比較します。その結果、ユーザ A に参照権があると判定されて、ユーザ A は文書 X を参照できます。しかし、ユーザ B が文書 X の更新を要求した場合、DocumentBroker は、アクセス権なしのエラーを返却しています。これは、ユーザ B には、文書 X を参照する権利だけが与えられているためです。

なお、LDAP 対応のディレクトリサービスと連携してアクセス制御機能を使用する場合、アクセス制御機

能で使用するユーザ情報は、LDAP 対応のディレクトリサービスに管理されている情報を基に生成されます。ただし、運用中のユーザ管理システムが DocumentBroker のアクセス制御機能に適合していない場合、ユーザが作成したアクセスルーチンを組み込むことによって、そのユーザ管理システムを DocumentBroker のアクセス制御機能として使用できます。

1.1.10 検索機能

DocumentBroker で使用できる検索機能について説明します。

(1) 属性検索機能

オブジェクトに設定されているプロパティの値を条件にした検索を実行できる、属性検索機能を提供します。プロパティの値を基に、文書、フォルダ、インデクスに相当するオブジェクトなどが検索できます。例えば、文書名と著者がプロパティとして設定されている場合に、「文書名が『報告書』であり、著者が『日立太郎』である文書を検索する」というような検索ができます。

(2) 文書に対する全文検索機能

文書の内容に含まれるキーワード（検索ターム）を条件にした検索を実行できる、全文検索機能を提供します。

DocumentBroker は、次のプログラムを利用して全文検索機能を実現します。

HiRDB Text Search Plug-in , または HiRDB XML Extension

さらに、次のプログラムを利用することによって、HiRDB Text Search Plug-in , または HiRDB XML Extension の機能を拡張した検索機能を実現します。

HiRDB Text Search Plug-in Conceptual Extension (概念検索の場合)

全文検索機能では、文書中の任意の検索タームを条件にできます。例えば、「テキスト内に『Document』と『Text』という単語が含まれる文書を検索する」という場合に使用できます。

次に、全文検索機能の基本機能と拡張機能について説明します。なお、このマニュアルでは、これらの検索機能を総称して、全文検索といいます。それぞれの検索に固有の説明をする場合は、それぞれの機能名で説明します。

基本機能

HiRDB Text Search Plug-in , または HiRDB XML Extension の機能を使用した検索機能です。

- 同義語・異表記展開検索

検索タームとして「コンピュータ」を指定した場合、「PC」や「パソコン」といった検索タームの同義語に相当する単語を含む文書も検索できます（同義語展開検索）。

アルファベットの大文字と小文字、かたかなの異表記および半角と全角の文字列を展開して、検索できます。例えば、「ski」を検索タームに指定する場合、「ski」、「SKI」、「Ski」などで表記されている文書も検索できます（異表記展開検索）。

- 近傍条件検索

二つの検索タームについて、それらの距離や出現順序関係を指定した検索ができます。例えば、「『最新』と『情報』の間の文字数が 20 文字ちょうどの文書を検索する。ただし、『最新』と『情報』はどちらが先に出現してもよい」という指定ができます。

拡張機能

ほかのプログラムの機能によって HiRDB Text Search Plug-in , または HiRDB XML Extension の機能を拡張した検索機能です。

- 概念検索

全文検索では、指定する検索タームに適した具体的な単語が思いつかない場合に、検索したい文書に類似した概念を持つ文章（種文章）を検索条件とした検索もできます。これを、概念検索といいます。概念検索では、検索条件として指定した文書からその文書の特徴づける単語（特徴ターム）が抽出され、それを検索タームとした検索が実行されます。例えば、「『...リサイクルは、資源の有効活用に貢献するだけでなく、ゴミの減量化にも役立ちます...』という文章に近い概念の文書を検索する」という場合に使用できます。概念検索機能を使用するためには、HiRDB Text Search Plug-in、または HiRDB XML Extension の機能のほかに、HiRDB Text Search Plug-in Conceptual Extension が必要です。

（3）文字列型プロパティに対する全文検索機能

オブジェクトに設定されている文字列型プロパティの値に含まれるキーワード（検索ターム）を条件にした検索を実行できる、全文検索機能を提供します。全文検索機能を使用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。

DocumentBroker は、次のプログラムを利用して文字列型プロパティに対する全文検索機能を実現します。

HiRDB Text Search Plug-in、または HiRDB XML Extension

HiRDB Text Search Plug-in、または HiRDB XML Extension を利用すると、次に示す機能を使用できます。

- 同義語・異表記展開検索
検索タームとして「コンピュータ」を指定した場合、「PC」や「パソコン」といった検索タームの同義語に相当する単語をプロパティの値に含む文書も検索できます（同義語展開検索）。
アルファベットの大文字と小文字、かたかなの異表記および半角と全角の文字列を展開して、検索できます。例えば、「ski」を検索タームに指定する場合、「ski」、「SKI」、「Ski」などで表記されている文字列をプロパティの値に含む文書も検索できます（異表記展開検索）。
- 近傍条件検索
二つの検索タームについて、それらの距離や出現順序関係を指定した検索ができます。例えば、「『最新』と『情報』の間の文字数が 20 文字ちょうどの値をプロパティに含む文書を検索する。ただし、『最新』と『情報』はどちらが先に出現してもよい」という指定ができます。

文字列型プロパティに対する全文検索機能に対応した DocumentBroker クラスライブラリのインターフェースおよびメソッドについては、マニュアル「DocumentBroker Version 5 概説」を参照してください。

（4）検索条件の論理演算機能

検索条件同士の論理積や論理和などを求める機能も提供します。

AND 検索

検索条件同士の論理積を求められます。例えば、「著者が『日立太郎』で、文書中に『コンピュータ』という文字列を含む文書を検索する」というような場合に使用できます。

OR 検索

検索条件同士の論理和を求めることができます。例えば、「作成者が『日立太郎』である文書、または作成者の所属が『日立製作所』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。

NOT 検索

検索条件との不一致を求めることができます。例えば、「作成者が『日立太郎』ではない文書を検索する」というような場合に使用できます。

1.1.11 全文検索インデクス一括登録機能

DocumentBroker に登録されている文書からテキスト情報を抽出し、全文検索インデクスとして一括登録する機能です。この機能を利用する場合は、DocumentBroker Text Search Index Loader を使用します。DocumentBroker Text Search Index Loader の詳細については、マニュアル「DocumentBroker Text Search Index Loader Version 3」を参照してください。

なお、リファレンスファイル文書の場合、全文検索インデクス一括登録機能は使用できません。

1.1.12 レンディション変換機能

マルチレンディション管理機能を利用して文書を管理する場合に、バッチ処理でレンディション変換を実行する機能です。次のような場合に、登録済みのマスタレンディションのファイルを基に、別のレンディションタイプのファイルを作成して、サブレンディションのコンテンツとして登録すると便利です。

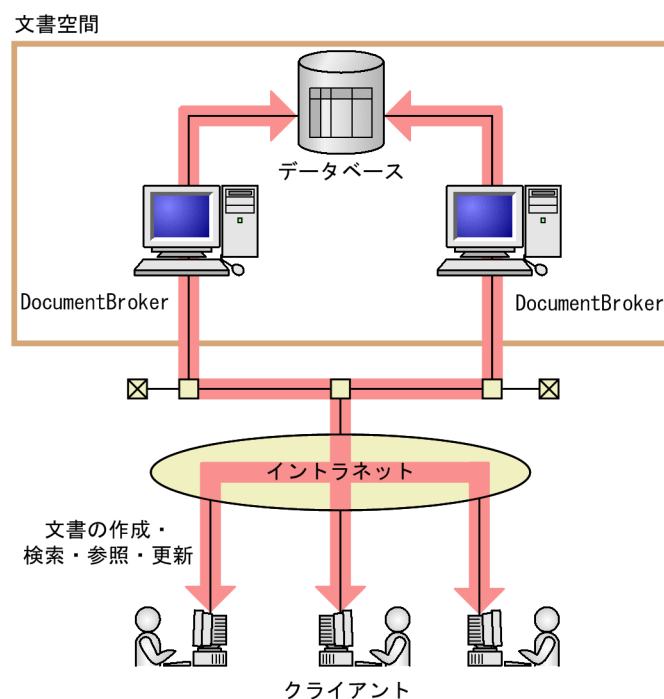
- マスタレンディションだけにファイルが登録されている場合
- サブレンディションのファイルの登録後にマスタレンディションのファイルが更新された場合

なお、この機能についてはサポート時期をご確認ください。

1.1.13 複数の実行環境機能

使用ユーザ数や単位時間当たりのトランザクション数の増加などによるシステム負荷を軽減するために、一つのデータベースに複数の DocumentBroker の実行環境を配置したシステム構成を構築できます。これによって、システム負荷が複数のサーバに分散されるため、システム全体の処理能力が向上します。複数の DocumentBroker の実行環境を配置したシステム構成を次の図に示します。

図 1-7 複数の DocumentBroker の実行環境を配置したシステム構成



実行環境を識別する実行環境識別子を使用して、文書を管理できます。これによって、複数の実行環境から、同じ文書空間にアクセスする運用形態を構築できます。

1.2 DocumentBroker のシステム構成

この節では、DocumentBroker のシステム構成について説明します。

1.2.1 DocumentBroker を使用したシステムの概略

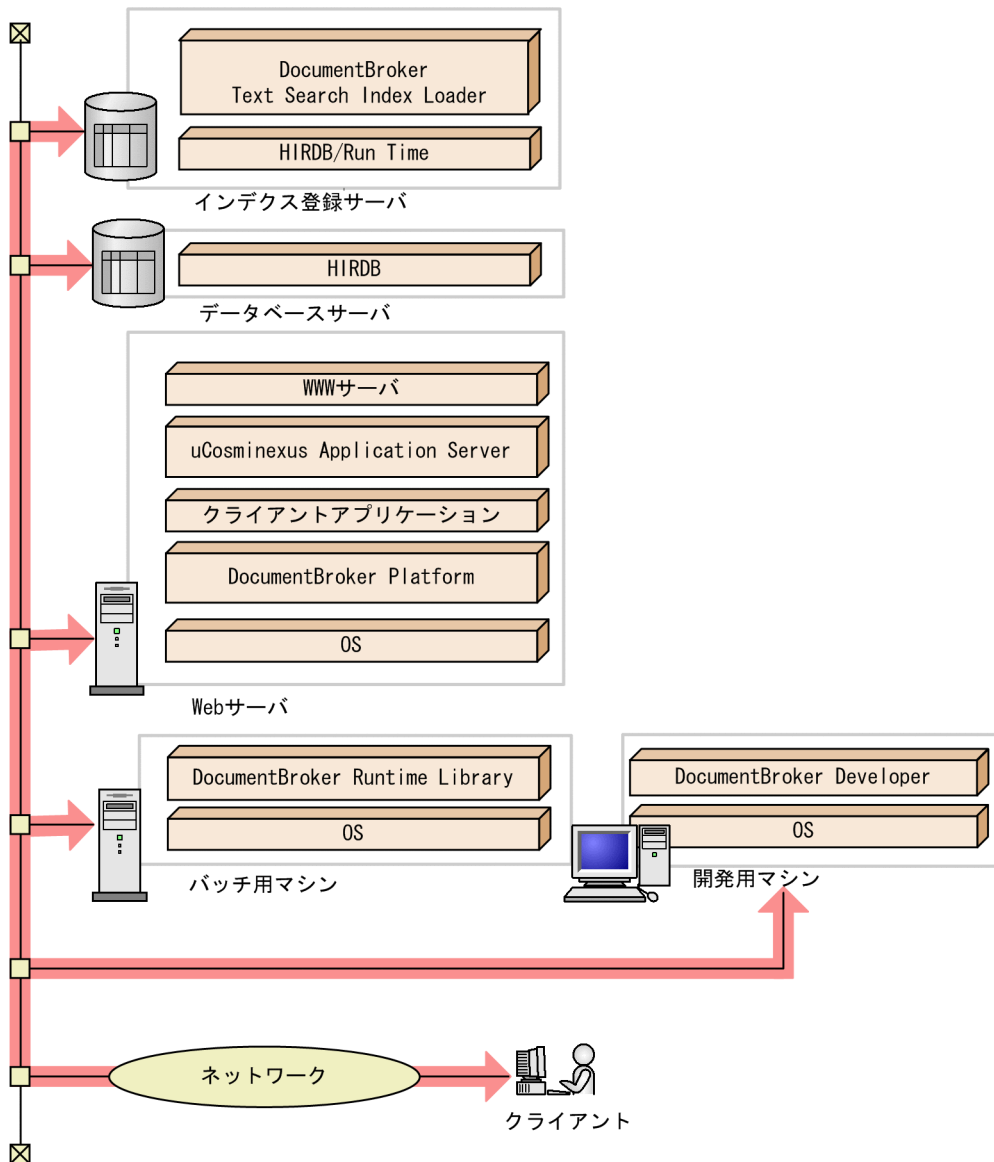
ここでは、DocumentBroker を使用したシステムの概略について説明します。

DocumentBroker は、分散オブジェクト環境での文書管理基盤フレームワークとして、文書管理に関するさまざまな機能を提供します。これによって、用途に合わせた各種業務システムを構築できます。

(1) DocumentBroker のシステムの構成

DocumentBroker のシステムの構成を次の図に示します。

図 1-8 DocumentBroker のシステムの構成



DocumentBroker は、業務に対応したクライアントアプリケーションを構築し、それを実行することで DocumentBroker の機能を利用します。クライアントアプリケーションを実行し、DocumentBroker 上でデータベースに格納されているオブジェクトを操作することで、文書の参照や更新などの文書管理が実現します。

2

システム導入前の検討

この章では、DocumentBroker を利用した文書管理システムの導入前に必要な検討項目について説明します。

2.1 適用業務の検討

2.2 追加するクラスおよびプロパティの検討

2.3 文書空間で使用する文字コード種別の検討

2.4 仮想メモリ所要量の見積もり

2.5 データベース容量の見積もり

2.6 リファレンスファイル管理機能を使用する場合のファイルシステムのディスク容量の見積もり

2.1 適用業務の検討

DocumentBroker は、基幹業務での文書管理システムの構築を支援します。この文書管理システムによって膨大な量の文書として蓄積されているデータを扱うことができるとともに、文書管理を基幹業務プロセスの一環として組み込むことで業務の効率を高めます。この節では、DocumentBroker を使用した文書管理システムの導入時に検討していただきたいことについて説明します。

適用業務の分析

文書管理システムを適用する基幹業務の性質についての分析が必要です。検索機能を充実させたシステムが必要な業務か、手書きの書類を大量にデータとして登録する業務か、文書のライフサイクル管理が必要な業務か、異なる業務間の連携が必要な業務か、などといった基幹業務の性質について分析してください。

文書に付加する情報の検討

対象の文書の分析結果を基にして、管理する文書に付加する情報を検討します。この作業では、管理する文書を種類ごとに分類して、それぞれに必要な付加情報を検討します。例えば、製品の設計ドキュメントを管理する場合は、「設計者名」、「開発完了予定日」、「特許情報」などの情報をあわせて管理することを検討します。なお、DocumentBroker では、「文書」は「クラス」に相当し、「付加情報」は「プロパティ」に相当します。したがって、ここでは、管理する文書の種類ごとにサブクラスを作成するとともに、文書の種類ごとに必要な付加情報については、各サブクラスにプロパティを追加していきます。

文書の管理方法の検討

DocumentBroker では、文書の管理方法によって、使用するオブジェクトが異なります。例えば、文書のバージョンを管理する場合はバージョン付き文書を、フォルダ機能を使用して文書をまとめて管理する場合はフォルダを使用して、フォルダに具体的でわかりやすい名称を付けるためのプロパティを追加することが考えられます。関連づけの種類ごとに、サブクラスを追加して、そのサブクラスを基にフォルダを作成していくような運用も考えられます。このように、文書の管理方法に合わせて、作成するオブジェクトについて検討し、管理方法に合わせたクラスおよびプロパティを効果的に定義する必要があります。

文書管理に使用するオブジェクトとオブジェクトの構成については、マニュアル「DocumentBroker Version 5 概説」を参照してください。

また、クラスおよびプロパティの追加については、「2.2 追加するクラスおよびプロパティの検討」を参照してください。

文書空間で使用する文字コード種別の分析

文書管理システムで使用する文字コード種別を分析し、文書空間で使用する文字コード種別を選択します。

文字コード種別は、文書の付加情報やファイル名に使用する文字コード種別に応じて、Shift-JIS または UTF-8 を選択してください。

文書空間で使用する文字コード種別の検討については、「2.3 文書空間で使用する文字コード種別の検討」を参照してください。

2.2 追加するクラスおよびプロパティの検討

この節では、クラスとプロパティの追加の概要、サブクラスとプロパティを追加できるクラス、全文検索機能を使用するために作成する全文検索機能付き文書クラスの追加について説明します。また、具体的な例を示してクラスとプロパティの追加について説明します。

DocumentBroker では、文書の管理方法に合わせてサブクラスを作成したり、クラスにプロパティを追加したりできます。この節の内容を参照して、サブクラスを作成とプロパティの追加について検討してください。なお、実際にサブクラスを作成して、クラスにプロパティを追加するためには、定義情報ファイルを作成する必要があります。定義情報ファイルの作成については、「3.8.1 データベースシステムの設定に必要なファイル」、および「4.9 定義情報ファイル」を参照してください。

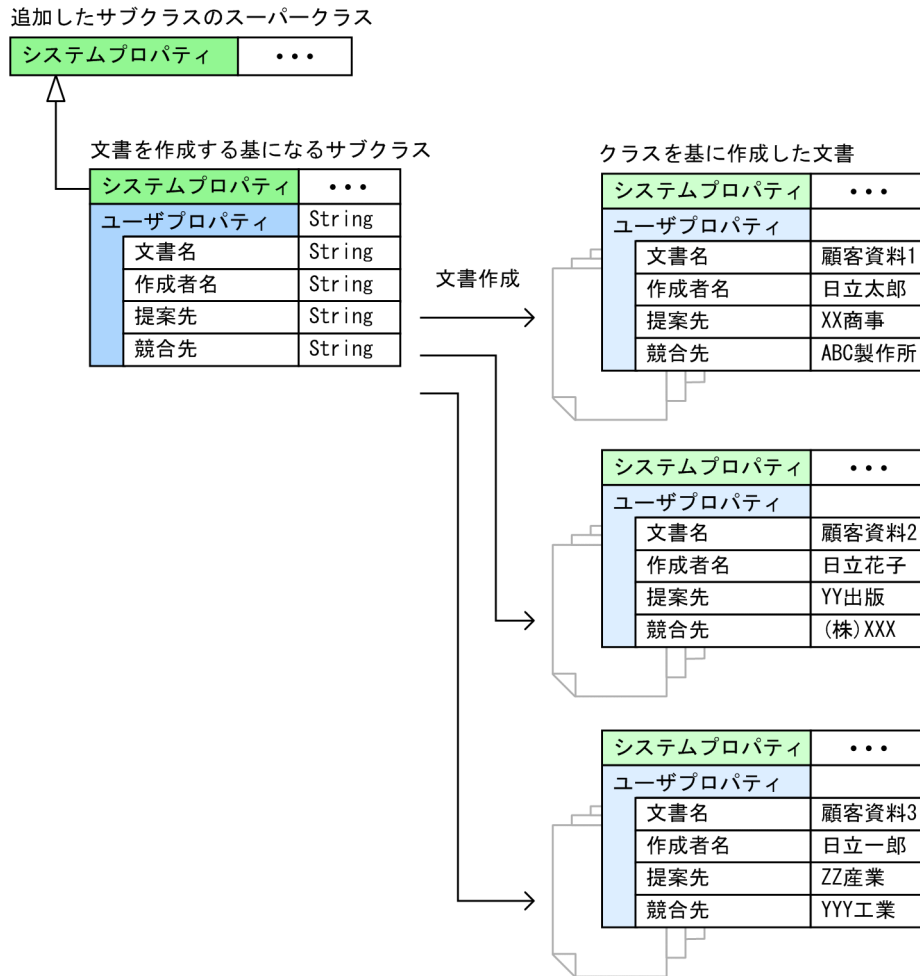
2.2.1 クラスおよびプロパティの追加

DocumentBroker で扱うオブジェクトは、オブジェクト指向に基づいて作成されるため、あるクラスを基に作成されるオブジェクトは、そのクラスのインスタンスとして扱われ、クラスからすべてのプロパティを継承します。すなわち、あるクラスを基に作成したオブジェクトは、同じ種類のオブジェクトであり、同じプロパティが付加されることとなります。

幾つかのクラスは、文書の種類や管理方法に合わせてサブクラスを追加できます。DocumentBroker が提供するクラスやユーザが追加したサブクラスには、必要に応じてプロパティを追加できます。プロパティには、文書に共通的に追加すると便利な情報やその文書に固有の情報など、ユーザがオブジェクトの管理情報として保持したい情報を設定できます。また、追加したプロパティをキーにしてオブジェクトを検索できます。オブジェクトをどのように管理するか、どのように検索するかを考慮して、必要なサブクラスを作成し、管理情報として利用できるプロパティや検索のキーになるようなプロパティを追加して、プロパティの値を設定する必要があります。

DocumentBroker で管理する文書はその種類ごとに分類できます。例えば、ある製品の購入を勧めるためのプレゼンテーション資料を管理する場合、プレゼンテーション資料作成用のサブクラスを定義して、そのサブクラスに対して、必要なプロパティを追加するような運用が考えられます。プレゼンテーション資料作成用に定義したクラスと文書の作成例を次の図に示します。

図 2-1 プレゼンテーション資料作成用に定義したクラスと文書の作成例



(凡例)

→ : クラスの継承関係を示す

システムプロパティ : DocumentBrokerによって定義されているプロパティ

ユーザプロパティ : ユーザが任意に追加定義するプロパティ

この図では、プレゼンテーション資料作成用に追加定義したサブクラスに、次に示すプロパティを追加しています。

- 文書管理に必要な一般的な管理情報
「文書名」と「作成者名」という String 型のプロパティを追加しています。
- プレゼンテーション資料としての付加情報
プレゼンテーション先の顧客名を管理する「提案先」というプロパティと競合する会社名を管理する「競合先」というプロパティを追加しています。

このクラスを基に作成した文書（プレゼンテーション資料）は、共通のプロパティを持ちます。プロパティに値を入れることによって、それぞれの文書の内容をプロパティから判別したり、文書の検索性を向上させたりできます。

2.2.2 サブクラスの追加

DocumentBroker が提供しているクラスに対して、必要に応じてサブクラスを追加できます。ここでは、

サブクラスを追加できるクラスとサブクラスを追加する場合の注意事項について説明します。

(1) サブクラスを追加できるクラス

サブクラスを追加できるクラスを次の表に示します。

表 2-1 サブクラスを追加できるクラス

項番	クラス分類	クラス
1	バージョンなし文書	<ul style="list-style-type: none"> • dmaClass_DocVersion クラス • edmClass_VersionTracedDocVersion クラス • edmClass_VersionTracedComponentDocVersion クラス
2	バージョン付き文書	dmaClass_ConfigurationHistory クラス
3	バージョンなしフォルダ	<ul style="list-style-type: none"> • dmaClass_Container クラス • edmClass_ContainerVersion クラス
4	独立データ	edmClass_IndPersistence クラス
5	可変長配列	edmClass_Struct クラス

(2) サブクラスを追加する場合の注意事項

- 追加するサブクラスの名称には「dmaClass」および「edmClass」で始まる文字列は使用できません。
- サブクラスの名称は重複しないようにしてください。
- 追加したサブクラスはスーパークラスのすべてのプロパティを継承します。

2.2.3 プロパティの追加

DocumentBroker が提供するクラスとユーザが定義したクラスに対して、プロパティを追加できます。例えば、dmaClass_DocVersion クラスに「文書名」のプロパティを追加するような場合です。ここでは、プロパティを追加できるクラス、プロパティのデータ型、プロパティの基本単位、およびプロパティを追加する場合の注意事項について説明します。

(1) プロパティを追加できるクラス

プロパティを追加できるクラスは次のとおりです。

- dmaClass_DirectContainmentRelationship クラス
- dmaClass_ReferentialContainmentRelationship クラス
- edmClass_PublicACL クラス
- edmClass_Relationship クラス
- ユーザが追加したサブクラス

(2) データ型の対応

追加できるプロパティのデータ型と HiRDB で対応するデータ型について次の表に示します。

表 2-2 追加できるプロパティのデータ型と HiRDB で対応するデータ型

追加できるプロパティのデータ型	HiRDB で対応するデータ型
String	MVARCHAR
Integer32	INTEGER
Boolean	
VariableArray	要素のデータ型に準ずる

追加できるプロパティのデータ型	HiRDB に対応するデータ型
FreeWordIndex	FREWORD

(3) プロパティを追加する場合の注意事項

- 追加するプロパティの名称には「dmaProp」、「edmProp」および「dbrprop」で始まる文字列は使用できません。

2.2.4 全文検索機能付き文書クラスの追加

ここでは、全文検索機能を利用するための全文検索機能付き文書クラスの追加について説明します。

構造を持たないテキスト形式の文書（プレーンテキスト）に対して全文検索を実行するためには、全文検索の対象となるサブクラスを追加する必要があります。全文検索の対象となるサブクラスのことを、全文検索機能付き文書クラスといいます。全文検索機能付き文書クラスとは、全文検索に必要な機能を持ったプロパティを追加した dmaClass_DocVersion クラスのサブクラスのことです。

全文検索のうち、概念検索の対象になるクラスについては、全文検索用の機能を持ったプロパティのほかに、概念検索用の機能を持ったプロパティも追加する必要があります。概念検索に必要な機能を持ったプロパティを追加した dmaClass_DocVersion クラスのサブクラスのことを特に、概念検索機能付き文書クラスといいます。

全文検索の対象になるオブジェクトを作成する場合は、全文検索機能付き文書クラスを、次に示すオブジェクトの構成要素として指定します。

- バージョンなし文書
- バージョン付き文書

全文検索機能付き文書クラスに追加するプロパティは、使用する検索機能によって異なります。ここでは、プレーンテキストの文書を対象として、次に示す文書クラスの定義について説明していきます。

- 全文検索機能付き文書クラス
- 概念検索機能付き文書クラス

(1) 全文検索機能付き文書クラスの定義

ここでは、検索タームを指定する全文検索機能を使用する文書クラスの追加定義に必要なプロパティについて説明します。

(a) 追加するプロパティ

全文検索機能を使用する文書クラスを作成する場合、dmaClass_DocVersion クラスのサブクラスとして、次に示すプロパティを追加した文書クラスを作成する必要があります。

- 全文検索インデクス用プロパティ
- edmProp_DocLength プロパティ
- edmProp_ContentIndexStatus プロパティ

全文検索インデクス用プロパティを次に示します。

- edmProp_TextIndex
- edmProp_ConceptTextIndex

(b) 必要に応じて追加するプロパティ

全文検索機能を使用する文書クラスの定義には、次に示すプロパティのうち、必要なプロパティを追加し

てください。

1. edmProp_TextIndex プロパティ

検索エンジンに対して、全文検索用の登録情報を参照することを示すためのプロパティです。プレーンテキストの登録情報を参照する場合は、このプロパティを追加します。

2. edmProp_ConceptTextIndex プロパティ

検索エンジンに対して、全文検索用または概念検索用の登録情報を参照することを示すためのプロパティです。プレーンテキストの登録情報を参照する場合は、このプロパティを追加します。

3. edmProp_DocLength プロパティ

検索した文書の長さを示すプロパティです。バイト単位で示します。

4. edmProp_ContentIndexStatus プロパティ

全文検索機能を使用する文書クラスに、全文検索インデクスの登録状態を示します。

文書の管理方法に合わせて、プロパティを追加してください。

全文検索機能を使用する文書クラスを定義する場合

- 1.、2 のプロパティのうち、どちらか一つを必ず追加してください。また、3. および 4. のプロパティを必ず追加してください。

(2) 概念検索機能付き文書クラスの定義

ここでは、概念検索機能を使用する文書クラスの追加定義に必要なプロパティについて説明します。

(a) 追加するプロパティ

概念検索機能を使用する文書クラスを作成する場合、dmaClass_DocVersion クラスのサブクラスとして、次に示すプロパティを追加した文書クラスを作成する必要があります。

- 全文検索インデクス用プロパティ（概念検索機能用）
- edmProp_DocLength プロパティ
- edmProp_ContentIndexStatus プロパティ

全文検索インデクス用プロパティ（概念検索機能用）を次に示します。

- edmProp_ConceptTextIndex

(b) 必要に応じて追加するプロパティ

概念検索機能を使用する文書クラスの定義には、次に示すプロパティのうち、必要なプロパティを追加してください。

1. edmProp_ConceptTextIndex プロパティ

検索エンジンに対して、概念検索用の登録情報を参照することを示すためのプロパティです。プレーンテキストの登録情報を参照する場合は、このプロパティを追加します。

2. edmProp_DocLength プロパティ

検索した文書の長さを示すプロパティです。バイト単位で示します。

3. edmProp_ContentIndexStatus プロパティ

概念検索機能を使用する文書クラスに、概念検索インデクスの登録状態を示すプロパティです。

文書の管理方法に合わせて、プロパティを追加してください。

概念検索機能を使用する文書クラスを追加する場合

1. のプロパティを必ず追加してください。また、2. および 3. のプロパティを必ず追加してください。

2.2.5 クラスおよびプロパティの追加例

ここでは、具体的な例を示して、クラスとプロパティの追加について説明します。

文書を作成、管理するために追加するクラス（`dmaClass_DocVersion` クラスのサブクラス）とプロパティについて、例に沿って説明します。

（例）

「問題点管理ドキュメント」を管理するためにクラスを追加します。この例で扱う文書の詳細情報を次の表に示します。

表 2-3 文書の詳細情報

文書の種類	概要	付加情報	管理方法
問題点管理ドキュメント	製品の開発時の問題点を管理するための文書	<ul style="list-style-type: none"> 問題点分類 問題点発生日時 検討結果 解決日時 	直接型のリンクを使用して管理する。

この文書を管理するためのクラスを次のように設計します。追加するサブクラスとスーパークラスの対応を次の表に示します。

表 2-4 追加するサブクラスとスーパークラスの対応

管理する文書	追加するサブクラス	スーパークラス
問題点管理ドキュメント	<code>usrClass_ProbInfoDoc</code>	<code>dmaClass_DocVersion</code>

問題点管理ドキュメントは、直接型のリンクを使用して管理します。したがって、`dmaClass_DocVersion` クラスのサブクラスとして `usrClass_ProbInfoDoc` クラスを追加して管理します。

次に、文書に対して付加する情報を各クラスのプロパティとして追加します。文書の付加情報とプロパティの対応を次の表に示します。

表 2-5 付加情報とプロパティの対応

クラス名	付加情報	プロパティ名	データ型
<code>usrClass_ProbInfoDoc</code>	問題点分類	<code>usrProp_Category</code>	String 型
	問題点発生日時	<code>usrProp_DateOfProb</code>	Integer32 型
	検討結果	<code>usrProp_Status</code>	Boolean 型
	解決日時	<code>usrProp_DateOfResolve</code>	Integer32 型

なお、追加するサブクラスは、全文検索機能付き文書クラスとして追加します。したがって、この表に示したプロパティのほかに、全文検索用のプロパティを追加する必要があります。全文検索機能付き文書クラスに追加するプロパティについては、「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。

このように、管理する文書の種類や付加情報などから必要なクラスとプロパティを決定してから、その設計情報を基に必要なクラスおよびプロパティを追加してください。

2.3 文書空間で使用する文字コード種別の検討

この節では、文書空間で使用する文字コード種別について説明します。

DocumentBroker の文書空間では、次のどちらかの文字コード種別を使用できます。一つの文書空間では複数の文字コード種別を使用できません。

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

選択した文字コード種別は、DocumentBroker およびデータベースで必ず一致させてください。異なる文字コード種別を設定した場合の動作は保証しません。

また、日本語および英語以外の言語も使用する場合は UTF-8 を選択してください。なお、JIS X0213:2004 の規格には Shift-JIS で扱えない文字が含まれています。そのため、日本語だけを使用する場合でも、UTF-8 を選択することを推奨します。

(1) データベースの文字コード種別の設定

データベース (HiRDB) の文字コード種別は、データベースの環境構築時に HiRDB で設定します。HiRDB の動作環境の設定コマンドの詳細は、マニュアル「HiRDB コマンドリファレンス」を参照してください。

また、文字コードに UTF-8 を使用する場合、全文検索インデクスに UCS-2 を利用するか UCS-4 を利用するかを検討してください。UCS-4 を使用する場合、インデクス定義文にプラグインオプションを追加する必要があります。詳細は、マニュアル「HiRDB Text Search Plug-in」、またはマニュアル「HiRDB XML Extension」を参照してください。

(2) 文書空間の文字コード種別の設定

文書空間の文字コード種別は、DocumentBroker の環境構築時に設定します。メタ情報の登録コマンド (dbjinitmeta) 実行時に、メタ情報初期化ファイルの DocSpaceCharacterSet エントリに指定します。

メタ情報の登録コマンドについては、「7.3 コマンドの文法」を参照してください。

2.4 仮想メモリ所要量の見積もり

ここでは、DocumentBroker の仮想メモリ所要量の見積もりについて説明します。

仮想メモリ所要量は、次に示す計算式で算出してください。なお、仮想メモリが不足すると、動作が不安定になり異常終了するおそれがあります。仮想メモリには必要量を確保してください。

計算式（単位：メガバイト）

$$40 + a + b$$

- $a = 4 \times \text{MIN}(c, d) \times e$

- $b = ((500 + f \times 2) \times g \times h) \div 1,048,576$

表 2-6 仮想メモリの計算式の変数に設定する値

変数	設定する値
c	文書空間で管理する最大文書サイズ（単位：メガバイト）
d	文書空間構成定義ファイルの BinaryAccessSize プロパティの値（単位：メガバイト）
e	文書空間で管理する同時作業文書数
f	検索で取得するプロパティ長の合計（単位：バイト）
g	検索ヒット件数
h	同時検索実行数

2.5 データベース容量の見積もり

ここでは、DocumentBroker のデータベース容量の見積もりについて説明します。

2.5.1 データベース容量の見積もり方法

HiRDB でデータベースシステムを構築するために必要なデータベース容量として、次の容量を確保してください。

データベース容量 = RD エリアの容量 + システムファイルの容量

RD エリアは、HiRDB が管理する表およびインデクスを格納する論理的なエリアのことです。

DocumentBroker で使用する RD エリアを次の表に示します。

表 2-7 DocumentBroker で使用する RD エリア

分類	RD エリアの種類	説明
システム用 RD エリア	マスタディレクトリ用 RD エリア	データベースシステムの情報を格納します。
	データディレクトリ用 RD エリア	
	データディクショナリ用 RD エリア	
システム LOB 用 RD エリア	データディクショナリ LOB 用 RD エリア	
レジストリ用 RD エリア	レジストリ用 RD エリア	HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した全文検索機能を使用する場合に、HiRDB Text Search Plug-in, または HiRDB XML Extension のレジストリ情報を格納します。
	レジストリ LOB 用 RD エリア	
ユーザ用 RD エリア	ユーザ表用 RD エリア	DocumentBroker で扱う文書やフォルダのデータを格納します。
	ユーザインデクス用 RD エリア	
ユーザ LOB 用 RD エリア	コンテンツ格納用 RD エリア	文書のコンテンツを格納します。
	SGMLTEXT データ格納用 RD エリア	HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した全文検索機能を使用する場合に、コンテンツのテキストデータ、テキストデータのインデクスを格納します。
	n-gram インデクス情報格納用 RD エリア	
	全文検索機能付き文字列型プロパティのインデクス格納用 RD エリア	HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した文字列型プロパティに対する全文検索機能を使用する場合に、全文検索機能付き文字列型プロパティのデータのインデクスを格納します。

このマニュアルでは、DocumentBroker で扱う文書やフォルダのデータを格納する、次の RD エリアの容量の算出方法について説明します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア

ほかの RD エリアの容量およびシステムファイルの容量の算出方法については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。なお、レジストリ用 RD エリアの容量の算出方法については、マニュアル「HiRDB Text Search Plug-in」、またはマニュアル「HiRDB XML Extension」も参照してください。

2.5.2 ユーザ用 RD エリアの容量の見積もり

ここでは、DocumentBroker で使用するユーザ用 RD エリアの容量の見積もりについて説明します。なお、アクセス制御機能で使用する容量も加算して、RD エリアの容量を見積もりする必要があります。アクセス制御機能を使用する場合の容量の算出方法については、「2.5.4 アクセス制御機能に関連するデータベース容量の見積もり」を参照してください。

(1) DocumentBroker で使用するユーザ用 RD エリア

DocumentBroker で管理する文書やフォルダは、DocumentBroker オブジェクトとして、データベースに格納されて管理されます。DocumentBroker オブジェクトは、複数のオブジェクトを包含した形で表現されています。したがって、ユーザ用 RD エリアの容量を見積もりする場合、DocumentBroker オブジェクトに包含されるオブジェクトの構成を考慮して、容量を算出しておく必要があります。

DocumentBroker で使用するユーザ用 RD エリアを次の表に示します。

表 2-8 DocumentBroker で使用するユーザ用 RD エリア

ユーザ用 RD エリアの種類	説明
ユーザ表用 RD エリア	次の情報を格納します。 <ul style="list-style-type: none"> システムクラス/ユーザクラスに対応する表 DocumentBroker のメタ情報
ユーザインデクス用 RD エリア	次の情報を格納します。 <ul style="list-style-type: none"> システムクラス/ユーザクラスに対応する表のインデクス DocumentBroker のメタ情報のインデクス

ユーザ用 RD エリアには、表とそのインデクス、および DocumentBroker のメタ情報とそのインデクスを格納します。これらの容量は、文書やフォルダなどの数、定義されているプロパティなどから算出します。また、バージョン付き文書の場合は管理するバージョンの数、マルチレンディション文書の場合はレンディションの数など、文書の構成を考慮する必要があります。

(2) ユーザ用 RD エリアが使用するデータベースリソースの所要量

ここでは、ユーザ用 RD エリアの容量を見積もるための目安として、DocumentBroker のクラス（表）ごとに、各レコードのデータサイズ、および各表に格納するレコード数の見積もり方法を説明した一覧を示します。この一覧を参照して、どのように RD エリアの容量を見積もるかについては、「(3) ユーザ用 RD エリアの容量の見積もり方法」で説明します。

ユーザ用 RD エリア分のデータベースリソースの所要量を次の表に示します。

表 2-9 ユーザ用 RD エリア分のデータベースリソースの所要量

種別	表名	列数	データサイズ(バイト)	レコード数見積もり方法	備考
文書	dmaClass_DocVersion ¹	11	573	バージョン付き文書ごとの平均バージョン数×バージョン付き文書数+バージョンなし文書数 ²	-

種別	表名	列数	データサイズ(バイト)	レコード数見積もり方法	備考
	dmaClass_Rendition	6	380	(バージョンごとのRendition数 - 1) × (バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数 ²)	Rendition が一つの場合は、DocVersionの表に含まれます。
	dmaClass_ContentReference	8	4949	バージョンごとのRendition数 × (バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数 ²)	リファレンスファイル管理機能を使用する場合にだけ必要なクラス。
	dmaClass_ContentTransfer	6	605	バージョンごとのRendition数 × (バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数 ²)	-
	edmClass_VT DocVersion ¹	11	573	バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数 ²	-
フォルダ	dmaClass_Container ¹	5	176	バージョンなしフォルダ数	-
	edmClass_ContainerVersion ¹	6	180	バージョンなしフォルダ数	-
	edmClass_VT Container ¹	5	176	0	-
バージョン	dmaClass_ConfigHistory ¹	6	228	バージョン付き文書数	-
	dmaClass_VersionSeries	11	656	バージョン付き文書数	バージョン付き文書ごとのVersionSeries数は一つです。
	dmaClass_VersionDescription	6	180	バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数	-
リンク	dmaClass_DC Relationship	4	172	DirectContainment Relationship オブジェクトを利用した関連するオブジェクトの個数	-

2. システム導入前の検討

種別	表名	列数	データサイズ(バイト)	レコード数見積もり方法	備考
	dmaClass_RC Relationship	4	172	ReferentialContainmentRelationship オブジェクトを利用した関連するオブジェクトの個数	-
	edmClass_VTC Relationship	7	280	0	-
	edmClass_Relationship	5	176	文書間リンクに使用している Relationship オブジェクトの個数	-
	edmClass_VT Relationship	13	496	0	-
独立データ	edmClass_Ind Persistence ₁	3	72	独立データオブジェクト数	-
全文検索	全文検索機能付き文書クラス	14	606	バージョン付き文書ごとの平均バージョン数×バージョン付き文書数+バージョンなし文書数 ²	全文検索用のデータを格納する列を追加した文書クラス。
システム用	edmClass_CompoDocVersion ¹	11	573	バージョン付き文書ごとの平均バージョン数×バージョン付き文書数+バージョンなし文書数 ²	-
	edmClass_VT CompoDocVersion ¹	11	573	バージョン付き文書ごとの平均バージョン数×バージョン付き文書数+バージョンなし文書数 ²	-
OIID	edmClass_OIID	3	20	実行環境の数	-
繰り返し列	edmClass_Struct ¹	3	84	VariableArray 型プロパティの配列に格納する平均要素数×VariableArray 型プロパティを定義するオブジェクト数	基本単位が VariableArray 型のプロパティの配列の要素を定義する場合のクラス。
メタ情報	EDMS_META_CLASSDESC	4	-	28 + a	a : 定義するユーザクラスの数
	EDMS_META_CLASSDESC_PROP	4	-	241 + a × 23+b	a : 定義するユーザクラスの数 b : 各クラスに追加するユーザプロパティの合計数
	EDMS_META_PROPDESC	9	-	79 + a	a : 定義するユーザプロパティの数
	EDMS_META_SYSINFO	2	-	11	-
	EDMS_META_ENVID	3	-	実行環境の数	-

注 1

サブクラスごとに算出する必要があります。

注 2

バージョンなし文書は、バージョン付き文書の個々のバージョンに対応しない文書を指します。

クラス名と表名の対応については「付録 A.1 システムクラスの名称定義の規則」を参照してください。

(3) ユーザ用 RD エリアの容量の見積もり方法

ここでは、ユーザ用 RD エリアの容量を見積もるための方法について説明します。ここで算出した値を基にして、実際に使用するディスク占有量を算出してください。

ディスク占有量の算出方法については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

「(2) ユーザ用 RD エリアが使用するデータベースリソースの所要量」の表 2-9 の内容を参考にして、次の値を算出してください。

(a) 各表に格納するレコードの総数

表 2-9 のデータサイズおよびレコード数の見積もり方法を参考にして、文書数やフォルダの数などから、各表に格納するレコードの総数を算出してください。

(b) 各表に定義する列の総数

表 2-9 には、DocumentBroker のシステムプロパティが使用する列数が示してあります。サブクラスを定義してユーザプロパティを追加した場合は、この表に示すスーパークラスの列数の値に、ユーザプロパティ数を加算して算出してください。

(c) 各列のデータサイズ

表 2-9 には、DocumentBroker のシステムプロパティが使用するデータサイズが示してあります。サブクラスを定義してユーザプロパティを追加した場合は、この表に示すスーパークラスのデータサイズの値に、ユーザプロパティのデータサイズを加算してください。ユーザプロパティのデータサイズについては、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。対応する HiRDB のデータ型を基に、データサイズを算出してください。

(d) インデクス

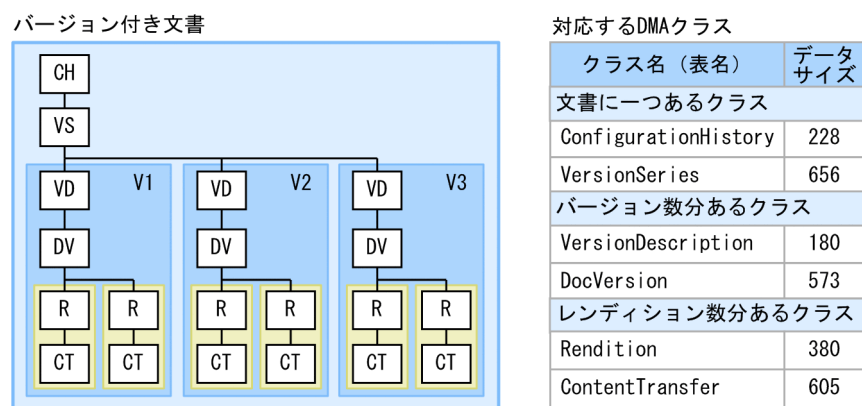
オブジェクトのプロパティに定義されるインデクスの算出については、データベース定義文出力コマンド (dbjertsq1) によって出力されるデータベース定義文を参考にしてください。

メタ情報の表に定義されるインデクスの算出については、createmetatable_hirdb.sql のデータベース定義文を参考にしてください。これらの定義文を基に、マニュアル「HiRDB システム導入・設計ガイド」を参照してインデクスを算出してください。

(4) 見積もりの例

ここでは、文書を管理する表の見積もりの例として、バージョン付き文書を管理する表の見積もりの例を示します。バージョン付き文書を管理する表の見積もりの例を次の図に示します。

図 2-2 バージョン付き文書を管理する表の見積もりの例



(凡例)

CH : ConfigurationHistoryオブジェクト

VS : VersionSeriesオブジェクト

VD : VersionDescriptionオブジェクト

DV : DocVersionオブジェクト

R : Renditionオブジェクト

CT : ContentTransferオブジェクト

この例では、バージョン付き文書で、3つのバージョンと2つのレンディションを管理しています。ユーザプロパティの追加はありません。例えば、このバージョン付き文書の文書数が1,000の場合、バージョン付き文書を管理する表の容量は次のように算出できます。

$$\begin{aligned} & \text{文書数} \times (\text{文書情報} + (\text{バージョン数} \times (\text{バージョン情報} + (\text{レンディション数} - 1) \times \\ & \text{Renditionオブジェクト} + (\text{レンディション数} \times \text{ContentTransferオブジェクト}))) \\ & = 1,000 \times (884 + (3 \times (753 + (2 - 1) \times 380 + (2 \times 605)))) \\ & = \text{約}8.0\text{メガバイト} \end{aligned}$$

文書情報

$$= \text{ConfigurationHistoryのデータサイズ} + \text{VersionSeriesのデータサイズ}$$

$$= 228 \text{ バイト} + 656 \text{ バイト}$$

$$= 884 \text{ バイト}$$

バージョン情報

$$= \text{VersionDescriptionのデータサイズ} + \text{DocVersionのデータサイズ}$$

$$= 180 \text{ バイト} + 573 \text{ バイト}$$

$$= 753 \text{ バイト}$$

2.5.3 ユーザ LOB 用 RD エリアの容量の見積もり

ここでは、DocumentBroker で使用するユーザ LOB 用 RD エリアの容量の見積もりについて説明します。

(1) DocumentBroker で使用するユーザ LOB 用 RD エリア

DocumentBroker で使用するユーザ LOB 用 RD エリアを次の表に示します。

表 2-10 DocumentBroker で使用するユーザ LOB 用 RD エリア

ユーザ LOB 用 RD エリアの種類	説明
コンテンツ格納用 RD エリア	文書の实体であるコンテンツを格納します。 この RD エリアは必ず作成してください。

ユーザ LOB 用 RD エリアの種類	説明
SGMLTEXT データ格納用 RD エリア	コンテンツのテキストデータ (SGMLTEXT) を格納します。 HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した全文検索機能を使用する場合には, この RD エリアを作成してください。
n-gram インデクス情報格納用 RD エリア	コンテンツのテキストデータのインデクス (n-gram インデクス) を格納します。 HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した全文検索機能を使用する場合には, この RD エリアを作成してください。
全文検索機能付き文字列型プロパティのインデクス格納用 RD エリア	全文検索機能付き文字列型プロパティのデータのインデクスを格納します。 HiRDB Text Search Plug-in, または HiRDB XML Extension を利用した文字列型プロパティに対する全文検索機能を使用する場合には, この RD エリアを作成してください。

ユーザ LOB 用 RD エリアに格納するデータについて説明します。

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バイナリ形式のファイルです。コンテンツは, dmaClass_ContentTransfer クラスに対応する表に格納されます。

テキストデータ (SGMLTEXT)

コンテンツから抽出されたテキスト形式のデータです。テキストデータ (SGMLTEXT) は, dmaClass_DocVersion クラスに対応する表に格納されます。

なお, SGML TEXT は, HiRDB Text Search Plug-in, または HiRDB XML Extension でプレーンテキストを扱う抽象データ型 (SGMLTEXT 型) のデータのことです。詳細については, マニュアル「HiRDB Text Search Plug-in」, またはマニュアル「HiRDB XML Extension」を参照してください。

n-gram インデクス

テキストデータ (SGMLTEXT) の列に対するインデクスデータです。n-gram インデクスは, dmaClass_DocVersion クラスに対応する表に格納されます。

なお, n-gram インデクスは, 文書中の n 文字 (n-gram) の出現位置を登録するインデクスのことで, HiRDB Text Search Plug-in, または HiRDB XML Extension で全文検索を実行するときに使用されます。詳細については, マニュアル「HiRDB Text Search Plug-in」, またはマニュアル「HiRDB XML Extension」を参照してください。

全文検索機能付き文字列型プロパティのデータのインデクス

全文検索機能付き文字列型プロパティのデータの列に対するインデクスデータです。全文検索機能付き文字列型プロパティのインデクスデータは, 全文検索機能付き文字列型プロパティを追加するサブクラスに対応する表に格納されます。

(2) ユーザ LOB 用 RD エリアが使用するデータベースリソースの所要量

ここでは, ユーザ LOB 用 RD エリアの容量を見積もるための目安として, DocumentBroker のクラス (表) ごとのデータサイズの見積もり方法を説明した一覧を示します。この一覧を参照して, どのように RD エリアの容量を見積もりするかについては, 「(3) ユーザ LOB 用 RD エリアの容量の見積もり方法」で説明します。

ユーザ LOB 用 RD エリア分のデータベースリソースの所要量を次の表に示します。

表 2-11 ユーザ LOB 用 RD エリア分のデータベースリソースの所要量

分類	クラス	データサイズ(バイト)	備考
コンテンツ格納用	ContentTransfer	バージョンごとの Rendition 数 × (バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数 ¹⁾) × 1 文書に格納するファイルのファイルサイズの平均値	文書の実体であるコンテンツデータを管理するクラス。
全文検索データ格納用	全文検索機能付き文書クラス	登録する文書のコンテンツのサイズに従って算出 ²⁾	-
全文検索機能付き文字列型プロパティのインデクス格納用	全文検索機能付き文字列型プロパティを追加するサブクラス	オブジェクトに設定する全文検索機能付き文字列型プロパティのデータのサイズに従って算出 ²⁾	-

注 1

バージョンなし文書は、バージョン付き文書の個々のバージョンに対応しない文書を指します。

注 2

登録する文書のコンテンツのサイズを求めて、マニュアル「HiRDB Text Search Plug-in」、またはマニュアル「HiRDB XML Extension」に従って算出する必要があります。

(3) ユーザ LOB 用 RD エリアの容量の見積もり方法

ここでは、ユーザ LOB 用 RD エリアの容量を見積もるための方法について説明します。ここで算出した値を基にして、実際に使用するディスク占有量を算出してください。ディスク占有量の算出方法については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

ユーザ LOB 用 RD エリアに格納するデータの容量は、登録するコンテンツのデータサイズや登録する文書の数などを考慮して、的確に見積もってください。

(a) コンテンツの容量

「(2) ユーザ LOB 用 RD エリアが使用するデータベースリソースの所要量」の表 2-11 の内容を参考にし、コンテンツ格納用 RD エリアに格納するコンテンツの容量を算出してください。レンディションの数、ファイルの数、文書の論理構造など、文書の構成を考慮して、登録する文書の総数と文書のコンテンツの平均サイズから容量を算出してください。

(b) 全文検索機能で使用する容量

全文検索機能を使用するためには、全文検索機能付き文書クラスごとに、SGMLTEXT データ格納用 RD エリアと n-gram インデクス情報格納用 RD エリアをそれぞれ作成する必要があります。全文検索機能を使用するための RD エリアの容量の見積もりについては、マニュアル「HiRDB Text Search Plug-in」、またはマニュアル「HiRDB XML Extension」を参照してください。

なお、HiRDB Text Search Plug-in、または HiRDB XML Extension に登録する文書サイズは、登録する文書のコンテンツのサイズ(単位: バイト)に従って算出してください。

(c) 文字列型プロパティに対する全文検索機能で使用する容量

文字列型プロパティに対する全文検索機能を使用するためには、オブジェクトに設定する全文検索機能付き文字列型プロパティごとに、プロパティのデータのインデクスを格納するための RD エリアを作成します。

文字列型プロパティに対する全文検索機能を使用するための RD エリアの容量の見積もりについては、マ

マニュアル「HiRDB Text Search Plug-in」、またはマニュアル「HiRDB XML Extension」を参照してください。

なお、HiRDB Text Search Plug-in、または HiRDB XML Extension に登録するプロパティのデータのサイズは、オブジェクトに設定する全文検索機能付き文字列型プロパティのデータのサイズと同じです。

2.5.4 アクセス制御機能に関連するデータベース容量の見積もり

アクセス制御機能のデータベース容量の見積もり方法を説明します。

ここでは、アクセス制御機能の RD エリアの容量を見積もるための目安として、DocumentBroker のクラス(表)ごとに、各表に定義する列数、各列のデータサイズ、および各表に格納するレコード数の見積もり方法を説明した一覧を示します。アクセス制御機能を使用する場合は、ここで算出した値を、「2.5.2 ユーザ用 RD エリアの容量の見積もり」で算出した値に加算して、データベースの容量を見積もってください。

アクセス制御機能が使用するデータベースリソースの所要量を次の表に示します。

表 2-12 アクセス制御機能が使用するデータベースリソースの所要量

分類	クラス(表名)	列数	データサイズ(バイト)	レコード数見積もり方法	備考
アクセス制御対象オブジェクト	DocumentBroker オブジェクトのトップオブジェクトに該当する表	7	$532 + 17 \times n$	すべての DocumentBroker オブジェクトのトップオブジェクト数	n : DocumentBroker オブジェクトにバインドしたパブリック ACL の平均の個数
ACL	edmClass_ACL	6	$84 + 263 \times (m + n)$	すべての DocumentBroker オブジェクトのトップオブジェクト数	<ul style="list-style-type: none"> m : DocumentBroker オブジェクトのローカル ACL に設定した ACE の平均の個数 n : DocumentBroker オブジェクトのセキュリティ ACL に設定した ACE の平均の個数
パブリック ACL	edmClass_PublicACL	8^1	$343 + 263 \times (m + n)^2$	パブリック ACL の個数	<ul style="list-style-type: none"> m : パブリック ACL のローカル ACL に設定した ACE の平均の個数 n : パブリック ACL のセキュリティ ACL に設定した ACE の平均の個数
バインド	edmClass_BindRelationship	2	68	すべての DocumentBroker オブジェクトのトップオブジェクト数 × DocumentBroker オブジェクトごとのバインドするパブリック ACL の平均の個数	なし

2. システム導入前の検討

注 1

パブリック ACL にユーザプロパティを追加した場合、追加したプロパティ数を加えて算出してください。

注 2

パブリック ACL にユーザプロパティを追加した場合、追加したプロパティのデータサイズを加えて算出してください。

なお、表 2-12 の「レコード数見積もり方法」に記述されている「すべての DocumentBroker オブジェクトのトップオブジェクト数」とは、作成した文書やフォルダなどに含まれるオブジェクトの個数を表します。このオブジェクトを次に示します。

- ComponentDocVersion オブジェクト
- ConfigurationHistory オブジェクト
- Container オブジェクト
- ContainerVersion オブジェクト
- DocVersion オブジェクト
- IndependentPersistence オブジェクト
- VersionTraceableContainer オブジェクト
- VersionTracedComponentDocVersion オブジェクト
- VersionTracedDocVersion オブジェクト

2.6 リファレンスファイル管理機能を使用する場合の ファイルシステムのディスク容量の見積もり

ここでは、リファレンスファイル管理機能を使用する場合の、ファイルシステムのディスク容量の見積もりについて説明します。なお、リファレンスファイル管理機能を使用しない場合、この見積もりは不要です。

ファイルシステムのディスク容量は、コンテンツ格納先ベースパスごとに見積もってください。コンテンツ格納先ベースパスごとに格納する文書の最大サイズを見積もり、見積もりした値より大きいディスク容量を確保する必要があります。コンテンツ格納先ベースパスに対して格納する文書の最大サイズの見積もり式を次に示します。

バージョンごとの Rendition 数 × (バージョン付き文書ごとの平均バージョン数 × バージョン付き文書数 + バージョンなし文書数) × 1 文書に格納するファイルのファイルサイズの平均値

注

バージョンなし文書は、バージョン付き文書の個々のバージョンに対応しない文書を指します。

3

環境設定

この章では、DocumentBroker の環境設定について説明します。

-
- 3.1 環境設定の流れ
 - 3.2 環境設定の準備
 - 3.3 インストールとアンインストール (UNIX の場合)
 - 3.4 インストールとアンインストール (Windows の場合)
 - 3.5 データベースシステムでの環境設定
 - 3.6 ユーザ管理機能の設定
 - 3.7 DocumentBroker の実行環境の作成
 - 3.8 DocumentBroker でデータベースシステムを使用するための設定
 - 3.9 アプリケーションサーバの設定
 - 3.10 Java アプリケーションの設定
 - 3.11 複数の実行環境を構築する場合の設定
 - 3.12 リファレンスファイル管理機能を使用する場合の設定
-

3.1 環境設定の流れ

この節では、DocumentBroker の環境設定の流れについて説明します。DocumentBroker の環境を設定する順序を、次に示します。

1. 各プログラムの環境設定をする前の準備
各プログラムの環境設定をする前の準備として、前提プログラムのインストールやオペレーティングシステムにシステム管理者の登録などをします。
「3.2 環境設定の準備」、「3.3 インストールとアンインストール (UNIX の場合)」または「3.4 インストールとアンインストール (Windows の場合)」、および「3.5 データベースシステムでの環境設定」を参照してください。
2. ユーザ管理の設定
DocumentBroker には、幾つかのユーザ管理方法が用意されています。ここでは、利用環境に合わせたユーザ管理機能を設定します。
「3.6 ユーザ管理機能の設定」を参照してください。
3. DocumentBroker の実行環境の作成
DocumentBroker の実行環境を作成します。
「3.7 DocumentBroker の実行環境の作成」を参照してください。
4. データベースシステムでの設定
データベースシステムの環境設定をします。
「3.8 DocumentBroker でデータベースシステムを使用するための設定」を参照してください。
5. アプリケーションサーバの設定
アプリケーションサーバ (Cosminexus) 上で DocumentBroker を使用するために必要な設定をします。
「3.9 アプリケーションサーバの設定」を参照してください。
6. Java アプリケーションの設定
Java アプリケーションで DocumentBroker を使用するために必要な設定をします。
「3.10 Java アプリケーションの設定」を参照してください。
7. 複数の実行環境を構築する設定
複数の実行環境から、同一の文書空間にアクセスする場合に必要な設定をします。
「3.11 複数の実行環境を構築する場合の設定」を参照してください。
8. リファレンスファイル管理機能の設定
リファレンスファイル管理機能を使用する場合に必要な設定をします。
「3.12 リファレンスファイル管理機能を使用する場合の設定」を参照してください。

3.2 環境設定の準備

この節では、DocumentBroker の環境を設定する場合の準備について説明します。

3.2.1 システム管理者の決定

DocumentBroker を運用および管理するシステム管理者には、Windows の Administrator 権限を持つユーザを割り当てます。UNIX の場合、システム管理者はスーパーユーザ権限、一般ユーザ権限のどちらでも構いません。運用コマンドは、システム管理者だけが実行できます。

3.2.2 利用ユーザおよびグループの決定

DocumentBroker を利用するユーザとユーザが所属するグループを決定してください。また、ここで決定した利用ユーザおよびグループは、ユーザ管理システムに登録してください。ユーザ管理システムについては、「3.6 ユーザ管理機能の設定」を参照してください。

DocumentBroker のアクセス制御機能では、DocumentBroker に登録されたオブジェクトや、そのアクセス権を保守するために、セキュリティ管理者を必ず指定してください。

セキュリティ管理者は、文書空間のすべてのアクセス制御対象オブジェクトに対してフルコントロールのアクセス権を持ちます。セキュリティ管理者は、連携するユーザ管理システムに登録されているユーザから選択して文書空間構成定義ファイルに定義してください。文書空間構成定義ファイルについては、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

3.2.3 DocumentBroker の実行環境を作成するディレクトリの決定

DocumentBroker の実行環境を作成するディレクトリを決定してください。

3.3 インストールとアンインストール (UNIX の場合)

この節では、ご使用の OS が UNIX の場合の、次に示すプログラムのインストール方法およびアンインストール方法について説明します。

- DocBroker Platform または DocumentBroker Runtime Library

また、前提プログラムについてもインストールしてください。インストール方法については、各製品のマニュアルを参照してください。

3.3.1 インストール

DocumentBroker Platform または DocumentBroker Runtime Library のインストールディレクトリは次のとおりです。

- /opt/DocBroker/Platform
- /opt/DocBroker/Library

デバイススペシャルファイル名や CD-ROM のマウントディレクトリは、使用する環境によって異なります。詳細については、オペレーティングシステムのマニュアルを参照してください。また、インストールの手順は、プログラムの提供媒体によって異なります。なお、インストール先のフォルダには、ASCII コードで表せるパスを指定してください。

DocumentBroker Platform または DocumentBroker Runtime Library をインストールする手順を次に示します。

1. スーパーユーザ (id:root) としてオペレーティングシステムにログインする。
2. 「mkdir mnt/cdrom」を実行して CD-ROM をマウントする「mnt/cdrom」ディレクトリを作成する。
「mnt/cdrom」の部分には、CD-ROM をファイルシステムとしてマウントするディレクトリ名を指定してください。
3. DocumentBroker Platform または DocumentBroker Runtime Library の CD-ROM を CD-ROM ドライブにセットする。
4. 「mount -r -o mode=0544 /dev/cdrom /mnt/cdrom」を実行して CD-ROM をマウントする。
「/dev/cdrom」および「/mnt/cdrom」の部分は使用する環境によって異なります。使用するデバイススペシャルファイル名および CD-ROM ファイルシステムを指定してください。
5. CD-ROM のセットアッププログラム「/mnt/cdrom/linux/setup /mnt/cdrom」を実行して日立 PP インストーラを起動する。
初期画面が表示されます。なお、CD-ROM のディレクトリ名や、ファイル名は、ls コマンドで確認して、表示されたファイル名を入力してください。
6. 「i」または「I」を入力して「I) Install Software」を選択する。
インストールできるプログラムの一覧が表示されます。
7. インストールするプログラムにカーソルを移動させて、スペースキーで選択する。
選択したプログラムの左側に「<@>」が付きます。なお、複数のプログラムを選択できます。
8. 「i」または「I」を入力して「I) Install」を選択する。
プログラムをインストールするかどうかについてのメッセージが最下行に表示されます。
9. 最下行に表示されるメッセージに対して「y」または「Y」を入力する。
インストールが始まります。ただし、「n」または「N」を入力すると、インストールが中止されて、手

順 7. で表示されたインストールできるプログラムの一覧が表示されます。

10. プログラムのインストールが終了したら、「q」または「Q」を入力して「Q)Quit」を選択して初期画面に戻る。

インストールを終了する場合、初期画面で「q」または「Q」を入力して「Q)Quit」を選択して終了します。

3.3.2 アンインストール

プログラムを再セットアップする場合、すでにインストールされているプログラムをアンインストールする必要があります。

(1) アンインストール

アンインストールする手順を次に示します。

1. 起動しているプログラムを停止する。
DocumentBroker Platform または DocumentBroker Runtime Library のライブラリを使用しているプログラムを停止してください。
2. スーパーユーザ (id:root) としてオペレーティングシステムにログインする。
3. 「/etc/hitachi_setup」を実行して日立 PP インストーラを起動する。
4. 「d」または「D」を入力して「D) Delete Software」を選択する。
アンインストールできるプログラムの一覧が表示されます。
5. アンインストールするプログラムにカーソルを移動させて、スペースキーで選択する。
選択したプログラムの左側に「<@>」が付きます。なお、複数のプログラムを選択できます。
6. 「d」または「D」を入力して「D) Delete」を選択する。
プログラムをアンインストールするかどうかについてのメッセージが最下行に表示されます。
7. 最下行に表示されるメッセージに対して「y」または「Y」を入力する。
アンインストールが始まります。ただし、「n」または「N」を入力すると、アンインストールが中止されて、手順 5. で表示されたアンインストールできるプログラムの一覧が表示されます。
8. プログラムのアンインストールが終了したら、「q」または「Q」を入力して「Q)Quit」を選択して初期画面に戻る。
アンインストールを終了する場合、初期画面で「q」または「Q」を入力して「Q)Quit」を選択して終了します。

3.3.3 インストール済みプログラムの一覧表示

UNIX の場合、日立 PP インストーラで、すでにマシンにインストールされているプログラムの一覧を表示できます。プログラムの一覧を表示する手順を次に示します。

1. 日立 PP インストーラを起動する。
初期画面が表示されます。なお、日立 PP インストーラの起動については、「3.3.1 インストール」の手順 5.、または「3.3.2 アンインストール」の手順 3. を参照してください。
2. 「l」または「L」を入力して「L) List Installed Software」を選択する。
PP 一覧表示画面が表示されます。この画面には、すでにマシンにインストールされているプログラムの一覧が表示されます。
3. PP 一覧表示画面で「q」または「Q」を入力して「Q)Quit」を選択する。

3. 環境設定

初期画面に戻ります。なお、PP一覧表示画面で「p」または「P」を入力して「P) Print to」を選択すると、すでにインストールされているプログラムの一覧が「/tmp/hitachi_PPLIST」に出力されます。

3.4 インストールとアンインストール (Windows の場合)

この節では、ご使用の OS が Windows の場合の、次に示すプログラムのインストール方法およびアンインストール方法について説明します。

- DocumentBroker Platform
- DocumentBroker Developer
- DocumentBroker Runtime Library

また、前提プログラムについてもインストールしてください。インストール方法については、各製品のマニュアルを参照してください。

上書きインストールを実行する場合、次に示すパスのディレクトリ、およびファイルは上書きされます。必要に応じて退避してください。

DocumentBroker Platform の場合

- <インストールディレクトリ>¥DocBroker¥Platform¥conf 以下のディレクトリ、およびファイル
- <インストールディレクトリ>¥DocBroker¥Platform¥sql 以下のディレクトリ、およびファイル

DocumentBroker Developer の場合

- <インストールディレクトリ>¥DocBroker¥Developer¥conf 以下のディレクトリ、およびファイル
- <インストールディレクトリ>¥DocBroker¥Developer¥sql 以下のディレクトリ、およびファイル

DocumentBroker Runtime Library の場合

- <インストールディレクトリ>¥DocBroker¥Library¥conf 以下のディレクトリ、およびファイル

3.4.1 統合 CD-ROM からのインストール

ここでは、統合 CD-ROM からプログラムをインストールする方法について説明します。インストール実行中にエラーが発生した場合、警告ダイアログまたはエラーダイアログが表示されます。警告ダイアログが表示された場合、メッセージに従って対策してから、インストールを続行してください。エラーダイアログが表示された場合、インストールは中止されます。エラーの要因を取り除いたあと、再度インストールを実行してください。

統合 CD-ROM からインストールする手順を次に示します。なお、インストール先のフォルダには、ASCII コードで表せるパスを指定してください。

1. システム管理者 (Administrator) の権限を持つユーザでログインします。
2. 統合 CD-ROM の指示に従いインストールする製品名を選択して、インストールを開始します。
セットアップ開始画面の表示後、[ようこそ] ダイアログが表示されます。
3. インストールする製品 (DocumentBroker Platform, DocumentBroker Developer または DocumentBroker Runtime Library) を確認して、「次へ」をクリックします。
[ユーザ情報] ダイアログが表示されます。
4. 「ユーザ名」および「所属」を入力します。
5. 「次へ」をクリックします。
[インストール先のフォルダ] ダイアログが表示されます。デフォルトのインストール先フォルダを次に示します。

DocumentBroker Platform の場合

<インストールディレクトリ> %DocBroker%\Platform

DocumentBroker Developer の場合

<インストールディレクトリ> %DocBroker%\Developer

DocumentBroker Runtime Library の場合

<インストールディレクトリ> %DocBroker%\Library

6. インストール先のフォルダを変更する場合は「変更」をクリックし、インストール先を選択します。
7. 「次へ」をクリックします。
[インストール準備]ダイアログが表示されます。
8. 「インストール」をクリックすると、ファイルのコピーが開始されます。ファイルのコピー後,[完了]ダイアログが表示されます。
9. 「完了」をクリックします。
ダイアログ表示が終了して、インストールが完了します。

3.4.2 リモートインストール

DocumentBroker Platform, DocumentBroker Developer または DocumentBroker Runtime Library をリモートインストールする場合、以下にインストールされます。

DocumentBroker Platform の場合

OS がインストールされたドライブ :%Program Files%\Hitachi%\DocBroker%\Platform

- ドライブだけを指定した場合
指定したドライブ :%Program Files%\Hitachi%\DocBroker%\Platform
- パスを指定した場合
指定したパス :%DocBroker%\Platform

DocumentBroker Developer の場合

OS がインストールされたドライブ :%Program Files %Hitachi%\DocBroker%\Developer

- ドライブだけを指定した場合
指定したドライブ :%Program Files %Hitachi%\DocBroker%\Developer
- パスを指定した場合
指定したパス :%DocBroker%\Developer

DocumentBroker Runtime Library の場合

OS がインストールされたドライブ :%Program Files %Hitachi%\DocBroker%\Library

- ドライブだけを指定した場合
指定したドライブ :%Program Files %Hitachi%\DocBroker%\Library
- パスを指定した場合
指定したパス :%DocBroker%\Library

注 OS が 64bit の場合は、Program Files(x86) と表示されます。

上書きインストールの場合は、インストールされている DocumentBroker Platform, DocumentBroker Developer または DocumentBroker Runtime Library と同じフォルダにインストールされます。

3.4.3 アンインストール

プログラムを再セットアップする場合、すでにインストールされているプログラムをアンインストールする必要があります。アンインストールするときの手順を、次に説明します。

アンインストールを実行すると、インストール時に配置されたファイルがすべて消去されます。必要に応じてバックアップを取得してください。

削除されないディレクトリおよびファイル

次に示す情報は、アンインストール時に削除されません。必要に応じて削除してください。

DocumentBroker Platform の場合

- <インストールディレクトリ>¥DocBroker¥Platform¥conf 以下のディレクトリ，およびファイル
- <インストールディレクトリ>¥DocBroker¥Platform¥sql 以下のディレクトリ，およびファイル

DocumentBroker Developer の場合

- <インストールディレクトリ>¥DocBroker¥Developer¥conf 以下のディレクトリ，およびファイル
- <インストールディレクトリ>¥DocBroker¥Developer¥sql 以下のディレクトリ，およびファイル

DocumentBroker Runtime Library の場合

- <インストールディレクトリ>¥DocBroker¥Library¥conf 以下のディレクトリ，およびファイル

アンインストールする手順を次に示します。

1. 起動しているプログラムを停止します。
2. Administrators グループのユーザでログインします。
3. [コントロールパネル] - [すべてのコントロールパネル項目] の [プログラムと機能] からアンインストールする製品名を選択してください。

3.5 データベースシステムでの環境設定

この節では、DocumentBroker の前提プログラムであるデータベースシステムに必要な環境設定について説明します。

3.5.1 HiRDB の環境設定の準備

(1) データベース容量の算出

HiRDB の環境設定を始める前に、DocumentBroker で管理する文書数、クラス数、プロパティ数などを基に、データベースの容量を見積もってください。データベースの容量を決定するための計算式については、「2.5 データベース容量の見積もり」を参照してください。

(2) データベースへの接続情報の決定

データベースに接続するための情報を決定しておく必要があります。

必要な情報を次に示します。

データベースに接続するためのユーザ名

データベースに接続するためのパスワード

データベースで使用するテーブルを格納するための RD エリア名

データベースで使用するインデクスを格納するための RD エリア名

データベースで使用する BLOB データを格納するための RD エリア名

データベースのポート番号

データベースのホスト名または IP アドレス

3.5.2 HiRDB の環境設定

DocumentBroker を運用するために必要な RD エリアなどを作成します。HiRDB の環境設定については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。このとき、DocumentBroker の文書空間で使用する文字コード種別に合わせて、HiRDB で使用する文字コード種別を設定してください。

3.5.3 HiRDB のユーザ権限の設定

DocumentBroker で使用するテーブルやインデクスを定義する場合、HiRDB に対して接続する必要があります。このため、データベースを作成するユーザに対して、CONNECT 権限を付与してください。また、スキーマを定義する必要もあるので、スキーマ定義権限も付与してください。

HiRDB のユーザ権限は、HiRDB のデータベース定義ユティリティ (pddef) を実行して、定義系 SQL の GRANT 文を使用して与えます。CONNECT 権限およびスキーマ定義権限を与える GRANT 文の例を次に示します。

(例)

ユーザ (ユーザ名: USER01 パスワード: PASS01) に CONNECT 権限およびスキーマ定義権限を与え、スキーマを作成します。

<記述例>

```
GRANT CONNECT TO USER01 IDENTIFIED BY PASS01;  
GRANT SCHEMA TO USER01;  
CREATE SCHEMA AUTHORIZATION USER01;
```

HiRDB のデータベース定義ユティリティについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。また、GRANT 文の文法および詳細な注意事項などについては、マニュアル「HiRDB SQL リファレンス」を参照してください。

3.5.4 HiRDB Text Search Plug-in , HiRDB XML Extension での環境設定

全文検索および概念検索といった全文検索機能を利用する場合には、HiRDB のプラグインプログラムである HiRDB Text Search Plug-in , または HiRDB XML Extension が必要です。これらを使用する場合、HiRDB へのセットアップや RD エリアの追加などが必要です。HiRDB Text Search Plug-in , および HiRDB XML Extension の環境設定については、マニュアル「HiRDB Text Search Plug-in」, またはマニュアル「HiRDB XML Extension」を参照してください。また、プラグインプログラムの環境設定については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

3.6 ユーザ管理機能の設定

この節では、DocumentBroker のユーザ管理機能について説明します。

DocumentBroker は、LDAP 対応のディレクトリサービス、ユーザが作成したアクセスルーチンと連携したユーザ管理機能を提供します。

3.6.1 LDAP 対応のディレクトリサービスによるユーザ管理機能を使用する場合の設定

LDAP 対応のディレクトリサービスによるユーザ管理機能を使用する場合、文書空間構成定義ファイルでの設定が必要です。文書空間構成定義ファイルについては、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

(1) ディレクトリサーバの設定

LDAP ディレクトリサーバをインストールして初期設定します。インストールと初期設定については、使用している LDAP ディレクトリサーバのマニュアルを参照してください。すでに LDAP ディレクトリサーバを使用している場合には、設置する必要はありません。

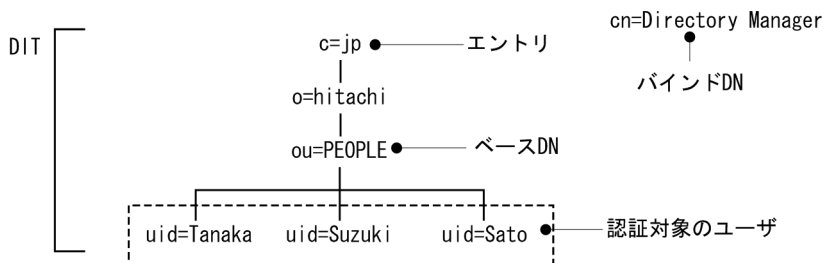
(2) ディレクトリサービスの情報管理の概念

連携する LDAP 対応のディレクトリサービスのデータ構造について次に示します。

フラット型

ベース DN 直下にすべてのユーザエントリが登録されているデータ構造です。

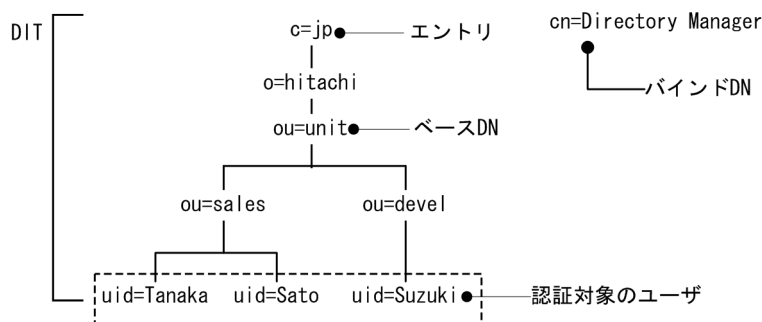
図 3-1 ディレクトリサービスで管理する情報の構造の例



階層型

ベース DN 以下が分岐しており、各階層にユーザエントリが登録されているデータ構造です。定義によっては、階層構造の場合に関連付いた情報を上位にあるプロパティから取得したり、再帰的に取得することも可能です。定義例については、「3.7.3(1) ユーザ情報の取得方法」を参照してください。

図 3-2 ディレクトリサービスで管理する情報の構造の例 (階層型)



(a) DIT

ディレクトリサービスが管理する情報は、図 3-1 および図 3-2 の形式の構造図によって示されます。この構造を DIT と呼びます

(b) エントリ

エントリまたはディレクトリエントリは、DIT を構成する情報で、図 3-1 および図 3-2 の DIT の各節に該当します (以降、エントリと呼びます)。

図 3-1 および図 3-2 の「c=jp」「o=hitachi」、「ou=PEOPLE」、「uid=Tanaka」などがエントリです。エントリは、左辺と右辺を「=」(等号) でつないでいます。左辺は属性で右辺は属性値を表します。また、エントリは一つ以上の属性によって構成されています。ディレクトリサービスでは、データをエントリおよび属性によって管理します。

(c) DN

DN は、DIT の各エントリを一意に識別するための情報名です。ファイルシステム内のファイルパスのように扱われます。DN は、「,」(コンマ) で区切られたエントリの集まりで、各エントリはディレクトリ内の位置を表す相対識別名 (RDN) で構成されます。相対識別名は、「属性 = 属性値」の形式で表現されます。例えば、uid=Tanaka や ou=People1 などが相対識別名に当たります。DN は、ファイルシステム内のファイルパスのように扱われますが、相対識別名の並びがファイルシステムとは逆の順番になります。つまり、ファイルシステムでは、ファイルの名称をいちばん右に指定して、左から右へパスを指定していきますが、DN では、いちばん左の位置に DIT の最下層のエントリを指定し、いちばん右側の位置に DIT の最上位のエントリを指定していきます。DN の記述例を次に示します。

```
uid=Tanaka,ou=People1,o=hitachi,c=jp
```

この記述例は、図 3-1 に示した「jp」というディレクトリ内の、「hitachi」というサブディレクトリ内にある「PEOPLE」というサブディレクトリの「Tanaka」というエントリの DN を示します。

(3) LDAP 対応のディレクトリサービスへのバインド

DocumentBroker で LDAP ディレクトリに接続 (バインド) するために、情報検索用のユーザを登録してアクセス権を設定します。情報検索用ユーザの DN のことを情報検索用のバインド DN といいます。情報検索用のバインド DN には、DocumentBroker で参照するベース DN 以下すべてのエントリ、およびエントリに設定されたすべての属性に対して、読み取り権、比較権、検索権を付与してください。情報検索用のバインド DN は、LDAP ディレクトリサーバからユーザ情報などを取得するときに使用します。

情報検索用のバインド DN は、文書空間構成定義ファイル (docspace.properties) の次に示すプロパティに設定します。

3. 環境設定

- LdapBindUserDN エントリ
- LdapBindPassword エントリ

文書空間構成定義ファイル (docspace.properties) の設定の詳細は、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

(4) LDAP 対応のディレクトリサービスのユーザ認証

ユーザ管理システムとして LDAP 対応のディレクトリサービスを使用する場合は、セッションを確立するときにユーザが指定したユーザ名を基に、文書空間構成定義ファイルで指定した DIT を検索してユーザエントリを特定し、その DN によりバインドを試みます。

このため、ディレクトリ構成を意識しないでユーザを特定できます。ただし、ディレクトリ内に存在するユーザが、一意に識別できる必要があります。

(5) LDAP 対応のディレクトリサービスでのユーザ情報取得

DocumentBroker のアクセス制御機能を使用して、アクセス制御されている文書に接続する場合、DocumentBroker は LDAP 対応のディレクトリサービスに登録されているユーザに関する情報を使用して、目的の文書に対するアクセスが許可されているかどうか判断します。DocumentBroker がアクセス制御機能で使用するユーザに関する情報は、ユーザ識別子およびグループ識別子です。ユーザ識別子およびグループ識別子について、次に説明します。

(a) ユーザ識別子

ユーザを一意に識別するための情報です。ユーザ識別子は、1 ~ 254 バイト (ただし、「¥0」を含まない) の ASCII 半角英数字、「-」、「.」、「@」、「_」で構成されます。LDAP 対応のディレクトリサービスを使用する場合、ユーザ識別子として使用する属性は、文書空間構成定義ファイルに指定する必要があります。文書空間構成定義ファイルについては、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

(b) グループ識別子

グループを一意に識別する情報です。ユーザが複数のグループに所属する場合があるため、グループ識別子は、一人のユーザに対して複数存在できます。また、プライマリグループも設定できます。

グループ識別子は、1 ~ 254 バイト (ただし、「¥0」を含まない) の ASCII 半角英数字、「-」、「.」、「@」、「_」で構成されます。

LDAP 対応のディレクトリサービスでは、「組織」と「グループ」をグループとして扱えます。組織とグループについて次に説明します。

組織

オブジェクトクラス「organizationalUnit」またはそのクラスから派生したクラスのオブジェクトです。DocumentBroker では、ユーザの属性として格納される組織情報 (例えば、ou 属性や departmentnumber 属性など) およびユーザを識別する DN に記述される組織情報をそれぞれ抽出して利用できます。

グループ

オブジェクトクラス「groupOfUniqueNames」またはそのクラスから派生したクラスのオブジェクトです。

(6) LDAP 対応のディレクトリサービスとの連携時の注意

組織構成の変更などによって、ディレクトリ (LDAP) 上のグループ識別子を変更されることがあります。

す。このような場合、オブジェクトに付与しているアクセス制御情報で表されるアクセス権を保持するには、ACLに記載されているグループ識別子を変更する必要があります。グループ識別子の変更に伴うオブジェクトのアクセス権の設定の変更については、マニュアル「DocumentBroker Version 5 概説」を参照してください。

DocumentBroker で利用する LDAP のユーザ識別子およびグループ識別子は、1 ~ 254 バイト（ただし、「¥0」を含まない）の ASCII 半角英数字、「-」、「.」、「@」、「_」で構成される必要があります。

3.6.2 ユーザ作成のアクセスルーチンを使用する場合の設定

DocumentBroker は、文書空間構成定義ファイルの定義によって、使用するユーザ管理システムのアクセスルーチンに、ユーザが作成したアクセスルーチンを組み込む機能を提供しています。これは、ユーザ管理システムに LDAP 対応のディレクトリサービス以外を使用する場合や、すでに使用している LDAP 対応のディレクトリサービスの運用形態が DocumentBroker のサポートしていない形態である場合に利用する機能です。なお、ユーザが作成したアクセスルーチン（コーディング部分）を UOC と呼びます。

(1) アクセスルーチンの作成方法

DocumentBroker に組み込むユーザ管理システムのアクセスルーチンは、決められた形式および仕様に基づいて作成する必要があります。DocumentBroker で提供するインターフェースとクラスを使用して、運用中のユーザ管理システムにアクセスできるように、アクセスルーチンを作成してください。

DocumentBroker で提供するインターフェースについては、「付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス」を参照してください。

また、作成したアクセスルーチンを組み込む場合は、文書空間構成定義ファイル（docspace.properties）での設定が必要です。文書空間構成定義ファイルの UserAuthentication プロパティに UOC を指定し、UOCClass プロパティに必要な値を指定してください。なお、文書空間構成定義ファイルについては、「4.3 文書空間構成定義ファイル（docspace.properties）」を参照してください。

(2) ユーザ作成のアクセスルーチンを使用する場合のユーザ情報の取得

DocumentBroker のアクセス制御機能を使用して、アクセス制御されている文書に接続する場合、DocumentBroker はユーザの属性についてのユーザ情報を取得します。そして、取得したユーザ情報を基に、ユーザの接続したい文書に対してアクセスが許可されているかどうかを判断します。UOC によるユーザ管理機能を使用してユーザ情報を取得する場合に、取得できるユーザ情報を次に示します。

(a) ユーザ識別子

関数 DbjAuthUserInfo インターフェースに格納されたユーザ識別子文字列を使用します。インターフェースの詳細については、「付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス」を参照してください。

(b) グループ識別子

関数 DbjAuthUserInfo インターフェースに格納されたプライマリグループ識別子およびグループ識別子文字列を使用します。インターフェースの詳細については、「付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス」を参照してください。

3.7 DocumentBroker の実行環境の作成

この節では、DocumentBroker の実行環境の作成について説明します。

3.7.1 DocumentBroker 実行環境ディレクトリの作成

DocumentBroker の実行環境を格納するためのディレクトリを作成してください。実行環境ディレクトリは、インストールディレクトリとは、別のディレクトリとしてください。ここで作成したディレクトリの下に DocumentBroker のインストールディレクトリ下の conf ディレクトリをコピーします。ディレクトリの名称は任意ですが、UNIX の場合、ディレクトリへのアクセスパーミッションに「0777」を設定する必要があります。また、Windows の場合は、ディレクトリに読み込み、書き込み権限を設定してください。

3.7.2 DocumentBroker の環境変数の設定

DocumentBroker の動作に必要な環境変数の設定について説明します。UNIX の場合、環境変数は各サーバマシンのログインシェル環境に合わせて次のファイルに設定してください。なお、\$HOME は、ログインユーザのホームディレクトリを示します。

Bourne シェル環境の場合

\$HOME/.profile

C シェル環境の場合

\$HOME/.cshrc または \$HOME/.login

(1) DOCB_CONF_DIR の設定

環境変数「DOCB_CONF_DIR」に「3.7.1 DocumentBroker 実行環境ディレクトリの作成」で作成した DocumentBroker 実行環境ディレクトリを設定します。

注 Windows の場合は、システム環境変数、またはユーザ環境変数となります。

(2) PATH の設定

データベース運用のコマンドを実行する環境の場合は、環境変数「PATH」に、java のパスを追加します。

3.7.3 文書空間の定義

この節では、DocumentBroker の文書空間の構成を定義する文書空間構成定義ファイル (docspace.properties) およびユーザ権限定義ファイル (userperm.properties) について説明します。設定内容については、「4. 環境設定に必要なファイル」を参照してください。

(1) ユーザ情報の取得方法

ここでは、ログインするときに生成されて値が設定されるユーザ情報の取得方法について説明します。

(a) ログインするときに生成されるユーザ情報 (LDAP ディレクトリサーバを使用する場合)

ユーザ情報は、DocumentBroker オブジェクトに対するアクセス権を判定する場合に使用されます。ユーザ情報には、「ユーザ識別子」および「グループ識別子」があります。これらの情報は、ログインしたときに生成されます。

ユーザ識別子

ユーザを識別する情報です。ユーザ識別子については、「3.6.1(5)(a) ユーザ識別子」を参照してください。

ユーザ識別子として取得される情報は、文書空間構成定義ファイルで任意に指定できます。ユーザ識別子を取得する場合に文書空間構成定義ファイルで指定するプロパティを次に示します。

LdapUserId プロパティ

LdapUserCase プロパティ

これらのプロパティの値は、ユーザ識別子として取得する情報および取得方法に従って指定してください。各プロパティの指定方法については、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

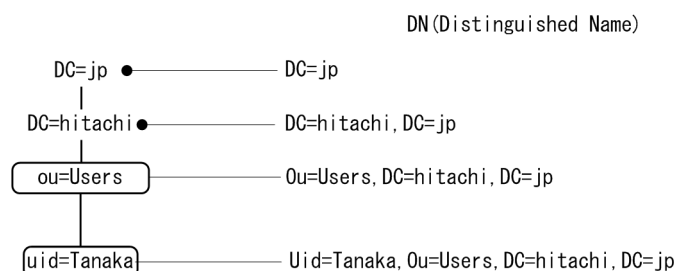
グループ識別子

ユーザが所属するグループを識別する情報です。グループ識別子については、「3.6.1(5)(b) グループ識別子」を参照してください。

グループ識別子の取得方法は、文書空間構成定義ファイルで定義します。グループ情報の取得方法について、次に説明します。なお、DocumentBroker にログインする場合のログイン名を、属性として登録してあるユーザエントリのことを、ログインユーザエントリと呼ぶことにします。

グループ識別子を取得する場合に、文書空間構成定義ファイルで指定するプロパティを次の図に示します。

図 3-3 LDAP ディレクトリサーバのツリー構成例



グループエントリの属性から取得する方法

グループ管理用のエントリの属性に、ユーザが所属するグループ名称が登録されている場合に使用する方法です。定義例を次の図に示します。

図 3-5 ログインユーザエントリの属性から取得する場合の定義例

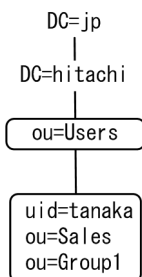


表 3-4 ログインユーザエントリの属性から取得する場合の定義例で指定するプロパティ

指定するプロパティ	指定するユーザ識別子	取得するグループ識別子
LdapGroupFromUserAttr=Yes LdapGroupIdFromUserAttr=ou	tanaka	Sales Group1

組織エントリの属性から取得する方法

ログインユーザエントリの属性に、そのユーザが所属する組織エントリの DN が登録されている場合に使用する方法です。定義例を次の図に示します。

図 3-6 組織エントリの属性から取得する場合の定義例

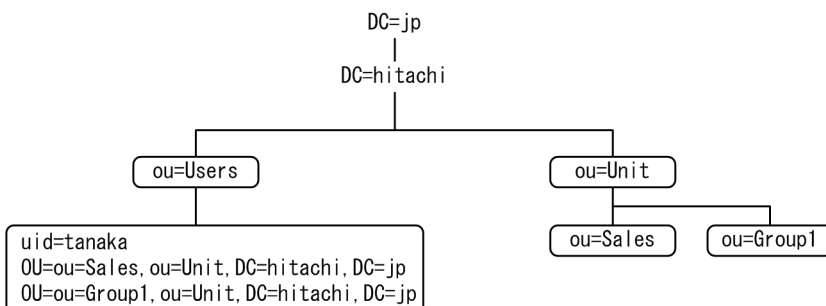


表 3-5 組織エントリの属性から取得する場合の定義例で指定するプロパティ

指定するプロパティ	指定するユーザ識別子	取得するグループ識別子
LdapGroupFromUserAttr=Yes LdapGroupIsDnFromUserAttr=Yes LdapGroupIdFromUserAttr=ou LdapGroupIdAttrFromUserAttr=dn	tanaka	ou=Sales,ou=Unit,DC=hitachi,DC=jp ou=Group1,ou=Unit,DC=hitachi,DC=jp

ログインユーザエントリの DN から取得する方法

ログインユーザエントリの DN の構成要素 (RDN) の属性値を、グループ識別子として使用する方法です。定義例を次の図に示します。

図 3-7 ログインユーザエントリの DN から取得する場合の定義例

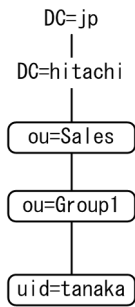


表 3-6 ログインユーザエントリの DN から取得する場合の定義例で指定するプロパティ

指定するプロパティ	指定するユーザ識別子	取得するグループ識別子
LdapGroupFromUserDn=Yes LdapGroupIdFromUserDn=ou	tanaka	Group1 Sales

これらのグループ識別子を取得する方法は、選択して使用したり、組み合わせて使用したりできます。ただし、「ログインユーザエントリの属性から取得する方法」と「組織エントリの属性から取得する方法」を組み合わせることはできません。また、取得するグループ識別子は LDAP ディレクトリサーバ上でユニークとなる値を取得する必要があります。

オブジェクトの識別子名 (DN) に次の文字が含まれる場合、グループエントリの属性から取得する方法を使用するとグループ情報が取得できません。

- * 0x2a
- (0x28
-) 0x29
- ¥ 0x5c
- NUL 0x00
- / 0x2f

ログインユーザエントリの属性から取得する方法、または組織エントリの属性から取得する方法、またはログインユーザエントリの DN から取得する方法を使用して取得してください。

文書空間構成定義ファイルで指定した方法に従って、グループ識別子の取得方法ごとに、指定する必要があるプロパティを次の表に示します。

表 3-7 グループ識別子の取得方法ごとに指定する必要があるプロパティの一覧

グループ情報の取得方法	指定するプロパティ
グループエントリの属性から取得する方法	<ul style="list-style-type: none"> • LdapGroup=Yes (必須) • LdapGroupRoot=<DN> (必須) • LdapGroupScope プロパティ • LdapGroupTimeout プロパティ • LdapGroupClass プロパティ • LdapGroupId プロパティ • LdapGroupFilterLeft プロパティ • LdapGroupFilterRight プロパティ • LdapGroupMember プロパティ • LdapGroupCase プロパティ • LdapGroupFromGroupDn プロパティ • LdapGroupFromGroupMember プロパティ

グループ情報の取得方法	指定するプロパティ
ログインユーザエントリの属性から取得する方法	<ul style="list-style-type: none"> • LdapGroupFromUserAttr=Yes (必須) • LdapGroupIdFromUserAttr プロパティ • LdapGroupCase プロパティ
組織エントリの属性から取得する方法	<ul style="list-style-type: none"> • LdapGroupFromUserAttr=Yes (必須) • LdapGroupIsDnFromUserAttr=Yes (必須) • LdapGroupIdFromUserAttr プロパティ • LdapGroupIdAttrFromUserAttr プロパティ • LdapGroupCase プロパティ
ログインユーザエントリの DN から取得する方法	<ul style="list-style-type: none"> • LdapGroupFromUserDn=Yes (必須) • LdapGroupIdFromUserDn プロパティ • LdapGroupCase プロパティ

各プロパティの指定方法については、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

(b) ログインするときに生成されるユーザ情報 (ユーザ作成のアクセスルーチンを使用する場合)

ユーザ管理機能として LDAP 対応のディレクトリサービスに連携していない場合や運用中の LDAP 対応のディレクトリサービスが DocumentBroker の運用形態に合わない場合、DocumentBroker は、ユーザが作成したアクセスルーチンを組み込む機能を提供しています。

ユーザ管理システムへのアクセスルーチンとして、ユーザが作成したアクセスルーチンを使用する場合は、文書空間構成定義ファイルの UserAuthentication プロパティで「UOC」を指定する必要があります。この場合、UOCClass プロパティの値として、ユーザが作成したアクセスルーチンのクラス名で指定します。UserAuthentication プロパティで「UOC」を指定した場合に生成されるユーザ情報には、ユーザ識別子とグループ識別子があります。ユーザ識別子については、「3.6.2(2)(a) ユーザ識別子」をグループ識別子については、「3.6.2(2)(b) グループ識別子」を参照してください。

(2) ユーザ権限定義ファイルの設定

文書空間にオブジェクトを作成する権限や文書空間内のオブジェクトに対する操作の範囲を定義するためにユーザ権限定義ファイルを作成します。

セキュリティ管理者は、このユーザ権限定義ファイルを更新することでユーザ権限を変更できます。ユーザ権限定義プロパティファイルの設定内容については、「4.4 ユーザ権限定義ファイル (userperm.properties)」を参照してください。

3.8 DocumentBroker でデータベースシステムを使用するための設定

この節では、DocumentBroker でデータベースシステムを使用するために必要な設定について説明します。

3.8.1 データベースシステムの設定に必要なファイル

この節では、DocumentBroker でデータベースシステムを使用するための設定に必要なファイルについて説明します。データベースの設定を始める前に、次のファイルを準備、作成してください。なお、各ファイルの詳細は、「4. 環境設定で必要なファイル」を参照してください。

- データベース情報ファイル
- メタ情報初期化ファイル
- 定義情報ファイル
- RD エリア定義情報ファイル
- インデクス情報ファイル

3.8.2 データベースシステムの設定手順

ここでは、DocumentBroker でデータベースシステムを使用するための設定手順について説明します。なお、データベースシステムの設定は、システム管理者が実行してください。また、データベースを設定するユーザに対しては、あらかじめ CONNECT 権限およびスキーマ定義権限を付与しておいてください。

DocumentBroker で使用できるデータベースは HiRDB です。したがって、ここで説明するデータベースシステムの設定手順は、HiRDB を使用する場合の設定手順です。また、データベースシステムを設定する前に、HiRDB の環境設定とデータベースシステムを設定するために必要なファイルを作成しておいてください。HiRDB の環境設定については「3.5.2 HiRDB の環境設定」を、必要なファイルについては「4. 環境設定で必要なファイル」を参照してください。

DocumentBroker で使用するデータベースシステムの設定手順を次に示します。設定中は、HiRDB を起動させておいてください。HiRDB の起動方法については、マニュアル「HiRDB システム運用ガイド」を参照してください。

(1) コマンド用環境設定ファイルを編集します

DocumentBroker のコマンド用環境設定ファイルを環境に合わせて編集します。コマンドは java で作成し、java コマンドから呼び出すバッチファイル形式で提供しています。

次のファイルで各コマンド共通で設定する環境変数を指定できます。

Windows の場合
dbjcmdconf.bat

UNIX の場合
dbjcmdconf

指定できる環境変数と設定値を次の表に示します。

表 3-8 指定できる環境変数と設定値

環境変数	設定値
DOCB_JAVA_CMD	実行する Java コマンドのパスを指定します。
DOCB_JVM_ARGS	JavaVM の起動オプションを指定します。特に値を指定しない場合は、java コマンドのデフォルト値となります。
DOCB_CLASSPATH	設定されている値をクラスパスに追加します。複数の値を指定したい場合は、Windows の場合「;」(セミコロン)で、UNIX の場合「:」(コロン)で区切りで指定してください。

注意

- パッチファイル内でエラーが発生する、または環境変数に不正な値を指定した場合の動作は保証しません。
- 半角スペースを含むパスなどを指定する場合、「」（半角ダブルクォーテーション）で囲まないでください。

Windows の場合の指定例を次に示します。

```
set DOCB_JAVA_CMD=c:\program files\Hitachi\cosminexus\jdk\bin\java.exe
set DOCB_JVM_ARGS=
set DOCB_CLASSPATH=C:\win32app\hitachi\hirdb_s\CLIENT\UTL\pdjdbc2.jar
```

UNIX の場合の指定例を次に示します。

```
export DOCB_JAVA_CMD=/opt/Cosminexus/jdk/bin/java
export DOCB_JVM_ARGS=
export DOCB_CLASSPATH=/opt/HiRDB_S/client/lib/pdjdbc2.jar
```

(2) 文書空間識別子を生成する

文書空間識別子生成コマンド (dbjenguid) を実行して、文書空間識別子を生成します。コマンドの詳細については、「dbjenguid (GUID の生成)」を参照してください。

生成した文書空間識別子を、文書空間構成定義ファイルの DocSpaceId プロパティに設定してください。

(3) メタ情報用のテーブルを定義します

DocumentBroker のメタ情報用のテーブルやインデクスを作成します。テーブルやインデクスは、テーブル定義用の SQL ファイルを使用して作成します。

(a) SQL ファイルの編集

メタ情報用のテーブルやインデクスは、次に示す SQL ファイルを編集して作成します。

UNIX の場合

```
/opt/DocBroker/Platform/sql/createmetatable_hirdb.sql
```

Windows の場合

```
<インストールディレクトリ>\DocBroker\Platform\sql\createmetatable_hirdb.sql, または <インストールディレクトリ>\DocBroker\Developer\sql\createmetatable_hirdb.sql
```

SQL ファイル中の文字列を、テキストエディタを使用して書き換えます。

テーブルやインデクスを作成するために SQL ファイルの書き換えが必要な文字列、および書き換える内容を次の表に示します。

表 3-9 SQL ファイルの書き換えが必要な内容

書き換えが必要な文字列	書き換える内容
(META)	メタ情報用のテーブルを格納する「ユーザ表用 RD エリア」の名称
(METAINDEX)	メタ情報用のインデクスを格納する「ユーザインデクス用 RD エリア」の名称

(b) SQL ファイルの実行

編集した SQL ファイルを入力ファイルとして、HiRDB でデータベース定義ユティリティ (pddef) を実行して定義します。HiRDB のデータベース定義ユティリティについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

(4) メタ情報の初期設定コマンド (dbjinitmeta) を実行します

メタ情報をデータベースに登録するために、メタ情報の初期設定コマンド (dbjinitmeta) を実行します。メタ情報の初期設定コマンドの使用方法については、「7.3 コマンドの文法」を参照してください。動作環境メタ情報ファイルは、この時点で作成されます。

(5) メタ情報の追加コマンド (dbjaddmeta) を実行します

作成した定義情報ファイルを入力ファイルとして、メタ情報の追加コマンド (dbjaddmeta) を実行することで、サブクラスおよびプロパティを追加します。定義情報ファイルについては、「4.9 定義情報ファイル」を参照してください。

メタ情報の追加コマンドの使用方法については、「7.3 コマンドの文法」を参照してください。この時点で、動作環境メタ情報ファイルも更新されます。

(6) DocumentBroker 用データベース定義文の作成コマンド (dbjrtssql) を実行します

RD エリア定義情報ファイルを入力ファイルとして、DocumentBroker 用データベース定義文の作成コマンド (dbjrtssql) を実行し、データベース定義文を出力します。さらに、ユーザが追加するプロパティにインデクスを定義する場合は、インデクス情報ファイルも入力ファイルとして指定してください。RD エリア定義情報ファイルについては「4.10 RD エリア定義情報ファイル」を、インデクス情報ファイルについては「4.11 インデクス情報ファイル」を参照してください。DocumentBroker 用データベース定義文の作成コマンドの使用方法については、「7.3 コマンドの文法」を参照してください。また、全文検索機能を利用する場合のデータベース定義文の内容については、「3.8.3 全文検索機能を使用する場合に出力されるデータベース定義文」を参照してください。

(7) データベース定義ユティリティを実行します

データベース定義文を入力ファイルとして、HiRDB でデータベース定義ユティリティ (pddef) を実行して定義を追加します。HiRDB のデータベース定義ユティリティについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

3.8.3 全文検索機能を使用する場合に出力されるデータベース定義文

ここでは、全文検索機能を使用する場合に出力されるデータベース定義文について説明します。

(1) 出力されるデータベース定義文

全文検索機能を使用する場合、DocumentBroker 用データベース定義文の作成コマンド (dbjrtssql) を実行すると、次に示すデータベース定義文が出力されます。

- 全文検索機能を使用する文書クラスを定義したデータベース定義文
 - 概念検索機能を使用する文書クラスを定義したデータベース定義文
 - 文字列型プロパティに対する全文検索機能を使用するためのプロパティを定義したデータベース定義文
- 以降、それぞれのデータベース定義文について説明します。

(2) 全文検索機能を使用する文書クラスを定義したデータベース定義文

全文検索機能を使用する文書クラスに定義される全文検索インデクス用プロパティ（検索タームを指定する全文検索機能用）ごとに出力されるデータベース定義文について説明します。ここでは、edmProp_TextIndex プロパティを定義したクラスのデータベース定義文について説明します。

(a) edmProp_TextIndex プロパティの定義

プレーンテキストを対象とした全文検索機能を使用する文書クラスに、edmProp_TextIndex プロパティを定義します。これによって、次の図に示すデータベース定義文が出力されます。

図 3-8 edmProp_TextIndex プロパティの定義で出力されるデータベース定義文

```
CREATE TABLE "usrClass_DocTextSearch"
(
    "dmaProp_01ID" CHAR(16),
    "dmaProp_This" CHAR(52),
    "dmaProp_Parent" CHAR(52),
    "dmaProp_ParentContainer" CHAR(52),
    "dmaProp_CurrentOfSeriesCnt" INT,
    "edmProp_TextIndex" SGMLTEXT RECOVERY ALL
        ALLOCATE (SGMLTEXT IN (SEARCHDB1))
        PLUGIN '<TEXTTYPE>PLAIN</TEXTTYPE>',
    "edmProp_DocLength" INT,
    "edmProp_ContentIndexStatus" INT,
    :
) IN (USER01);

CREATE UNIQUE INDEX "usrClass_DocTextSearch01"
ON "usrClass_DocTextSearch" ("dmaProp_01ID") IN (INDEX01);
:

CREATE INDEX "usrClass_DocTextSearch06"
USING TYPE NGRAM ON "usrClass_DocTextSearch" ("edmProp_TextIndex")
IN (SEARCHINDEX1);
```

(b) データベース定義文の定義内容の解説

図 3-8 のデータベース定義文中の 1. , 2. , および 3. の意味を次に示します。

1. 全文検索機能付き文書クラスのための列作成定義
全文検索用の文書の内容を格納するためのユーザ LOB 用 RD エリアを使用します。例では、SEARCHDB1 というユーザ LOB 用 RD エリアを指定しています。
2. 全文検索機能付きクラスのための列作成定義
3. 全文検索機能付きクラスのためのインデクス作成定義
全文検索用のユーザ LOB 用 RD エリアを使用します。例では、SEARCHINDEX1 または SGMLINDEX1 というユーザ LOB 用 RD エリアを指定しています。

(3) 概念検索機能を使用する文書クラスを定義したデータベース定義文

概念検索機能を使用する文書クラスに定義される全文検索インデクス用プロパティ（概念検索機能用）ご

3. 環境設定

とに出力されるデータベース定義文について説明します。ここでは、次に示す全文検索インデクス用プロパティを定義したクラスのデータベース定義文について説明します。

- edmProp_ConceptTextIndex プロパティ

(a) edmProp_ConceptTextIndex プロパティを定義した場合

プレーンテキストを対象とする概念検索機能を使用する文書クラスに、edmProp_ConceptTextIndex プロパティを定義します。これによって、次の図に示すデータベース定義文が出力されます。

図 3-9 edmProp_ConceptTextIndex プロパティを定義した場合に出力されるデータベース定義文

```
CREATE TABLE "usrClass_DocTextSearch"
(
    "dmaProp_01ID" CHAR(16),
    :
    "edmProp_ConceptTextIndex" SGMLTEXT RECOVERY ALL 1.
        ALLOCATE (SGMLTEXT IN (SEARCHDB1)) 1.
        PLUGIN '<TEXTTYPE>PLAIN</TEXTTYPE>', 1.
    "edmProp_DocLength" INT, 2.
    "edmProp_ContentIndexStatus" INT, 2.
    :
) IN (AREA01);

CREATE UNIQUE INDEX "usrClass_DocTextSearch01"
ON "usrClass_DocTextSearch"("dmaProp_01ID") IN (INDEX01);
:

CREATE INDEX "usrClass_DocTextSearch06" 3.
USING TYPE NGRAM ON "usrClass_DocTextSearch"("edmProp_ConceptTextIndex") 3.
IN (SEARCHINDEX1) 3.
PLUGIN 'CONCEPT_ON'; 3.
```

(b) データベース定義文の定義内容の解説

図 3-9 のデータベース定義文中の 1. , 2. , および 3. の意味を次に示します。

1. 概念検索機能を使用する文書クラスのための列作成定義
概念検索用の文書の内容を格納するためのユーザ LOB 用 RD エリアを使用します。例では、SEARCHDB1 というユーザ LOB 用 RD エリアを指定しています。
2. 概念検索機能を使用する文書クラスのための列作成定義
3. 概念検索機能を使用する文書クラスのためのインデクス作成定義
概念検索用のユーザ LOB 用 RD エリアを使用します。例では、SEARCHINDEX1 というユーザ LOB 用 RD エリアを指定しています。

(4) 文字列型プロパティに対する全文検索機能を使用するためのプロパティを定義したデータベース定義文

ここでは、文字列型プロパティに対する全文検索機能を使用するためのプロパティを定義した場合に出力されるデータベース定義文について説明します。

usrClass_PropTextSearch クラスに対して、全文検索機能付き文字列型プロパティである usrProp_DocSummary プロパティを定義した場合に出力されるデータベース定義文を次の図に示します。

図 3-10 全文検索機能付き文字列型プロパティを定義した場合に出力されるデータベース定義文

```

CREATE TABLE "usrClass_PropTextSearch"
(
    "dmaProp_01ID" CHAR(16),
    ⋮
    "usrProp_DocSummary" FREEWORD ●----- 1.
) IN (USERAREA);

CREATE UNIQUE INDEX "usrClass_PropTextSearch01"
    ON "usrClass_PropTextSearch"("dmaProp_01ID") IN (INDEXAREA);
    ⋮
CREATE INDEX "usrClass_PropTextSearch02" ●----- 2.
    USING TYPE IXFREEWORD ON "usrClass_PropTextSearch"("usrProp_DocSummary") ●-----
    IN (SEARCHINDEX1); ●-----

```

図 3-10 のデータベース定義文中の 1. および 2. の意味を次に示します。

1. 全文検索機能付き文字列型プロパティのための列作成定義
2. 全文検索機能付き文字列型プロパティのためのインデクス作成定義
全文検索機能付き文字列型プロパティ用のユーザ LOB 用 RD エリアを使用します。例では、SEARCHINDEX1 というユーザ LOB 用 RD エリアを指定しています。

(5) データベース定義文中の RD エリア名についての注意事項

全文検索機能を使用する場合に出力されるデータベース定義文に定義されている RD エリア名について説明します。

これらの RD エリア名は、「3.5.2 HiRDB の環境設定」で作成した RD エリア名に変更する必要があります。ただし、DocumentBroker 用データベース定義文の作成コマンド (dbjertsq1) の -r オプションで RD エリア定義情報ファイル名を指定している場合、この作業は不要です。RD エリア定義情報ファイルについては、「4.10 RD エリア定義情報ファイル」を参照してください。

3.9 アプリケーションサーバの設定

この節では、アプリケーションサーバ (Cosminexus) 上で DocumentBroker を使用するための設定方法について説明します。

3.9.1 アプリケーションサーバの構築

J2EE アプリケーションを実行するシステムを構築します。構築方法については、マニュアル「Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド」を参照してください。

3.9.2 セキュリティポリシーの設定

アプリケーションサーバ (Cosminexus) 上で DocumentBroker を使用するために、セキュリティポリシーを設定する必要があります。J2EE サーバのセキュリティポリシーファイル (server.policy) に、以下の内容を追加してください。

```
//
// Grant permissions to JSP/Servlet
//
grant codeBase "file:${ejbserver.http.root}/web/${ejbserver.serverName}/-" {
    permission java.util.PropertyPermission "*", "read, write";
    permission java.util.logging.LoggingPermission "control";
    permission java.lang.RuntimePermission "getenv.DOCB_CONF_DIR";
};
//
// Grant permissions to uCosminexus DocumentBroker
//
grant codeBase "file:${ejbserver.http.root}/ejb/${ejbserver.serverName}/apps/-" {
    permission java.io.FilePermission "<<ALL FILES>>", "read, write, delete";
    permission java.util.logging.LoggingPermission "control";
    permission java.lang.RuntimePermission "getenv.DOCB_CONF_DIR";
};
```

セキュリティポリシーファイルについては、マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の「server.policy (J2EE サーバ用セキュリティポリシーファイル)」の説明を参照してください。

3.9.3 保護区リストファイルの設定

保護区リストファイルは製品単位の定義で、Cosminexus 上で DocumentBroker を使用するために必要な設定です。

「保護区リストファイル」の要素として、次に示すプリフィックス名を設定してください。

- jp.co.Hitachi.soft.docbroker.*

設定した保護区リストファイルは、J2EE サーバの起動時に読み込まれ、Cosminexus をインストールしたマシン上で動作するすべての J2EE サーバで有効となります。

保護区リストファイルについては、マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の「criticalList.cfg (保護区リストファイル)」の説明を参照してください。

3.9.4 コンテナ拡張ライブラリの取り込み

J2EE サーバ用オプション定義ファイルへの設定をします。Cosminexus Component Container の `usrconf.cfg` ファイルに、次のパスを追加してください。

- `add.class.path=JDBC` ドライバのファイルパス

`add.class.path` については、マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の「`usrconf.cfg` (J2EE サーバ用オプション定義ファイル)」の説明を参照してください。

3.9.5 J2EE サーバの起動

J2EE サーバを起動します。J2EE サーバの起動方法は、マニュアル「Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド」を参照してください。

3.9.6 リソースアダプタの設定

データベースにアクセスするためのリソースアダプタを設定します。設定方法については、マニュアル「Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド」、および「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (アプリケーション/リソース定義)」を参照してください。リソースアダプタは DB Connector を使用します。

UNIX の場合に、コマンドでリソースアダプタを設定するときの例について説明します。

(1) リソースアダプタのインポート

次のコマンド (`cjimportres`) を実行して、HiRDB Type4 Driver に対応するリソースアダプタをインポートします。

```
$COSMINEXUS_HOME/CC/admin/bin/cjimportres J2EEサーバ名 -type rar -f $COSMINEXUS_HOME/CC/DBConnector/DBConnector_HiRDB_Type4_CP.rar
```

(2) Connector 属性ファイルの取得

次のコマンド (`cjgetresprop`) を実行して、Connector 属性ファイルをカレントディレクトリに取得します。

```
$COSMINEXUS_HOME/CC/admin/bin/cjgetresprop J2EEサーバ名 -type rar -resname DB_Connector_for_HiRDB_Type4 -c DBConnector_docb.xml
```

(3) Connector 属性ファイルの編集

取得した Connector 属性ファイルを次のとおり編集してください。

表 3-10 Connector 属性ファイルの設定項目

設定項目	設定内容
<code>display-name</code>	リソースアダプタの表示名
<code>description</code>	HiRDB のポート番号
<code>DBHostName</code>	HiRDB のホスト名
<code>LONGVARBINARY_Access</code>	LOCATOR

3. 環境設定

設定項目	設定内容
LONGVARIABLE_ACCESS_SIZE	1024 ¹
User	ユーザ名 ²
Password	パスワード ²
MinPoolSize	コネクションプールの最小値 ³
MaxPoolSize	コネクションプールの最大値 ³

注 1

docspace.properties の BinaryAccessSize と一致させることを推奨します。

注 2

データベースにアクセスするための DocumentBroker 用のユーザ名とパスワードです。

注 3

同時実行数以上の値を設定してください。

なお、コネクション枯渇時の動作は、リソースアダプタの Connector 属性ファイルの property タグの設定値に依存します。

コネクション枯渇時のコネクション取得待ち行列を有効にするかどうか

RequestQueueEnable (デフォルト: True)

コネクション枯渇時のコネクション取得待ち行列のとどまることのできる時間の最大値

RequestQueueTimeout (デフォルト: 30 秒)

デフォルトの設定では、データベースアクセス処理でコネクションが枯渇した場合、コネクションが取得できるまで最大 30 秒の待ちが発生します。30 秒以内にコネクションが取得できない場合は SQL エラーが発生します。

設定値の詳細については、マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の <property> タグに指定できるプロパティについて記載されている個所を参照してください。また、

Cosminexus の同時実行スレッド数についても適切に設定してください。同時実行スレッド数の詳細については、

マニュアル「Cosminexus V9 アプリケーションサーバ リファレンス 定義編 (サーバ定義)」の

usrconf.properties (J2EE サーバ用ユーザプロパティファイル) の webserver.connector.ajp13.max_threads について記載されている個所を参照してください。

(4) Connector 属性ファイルの反映

次のコマンド (cjsetresprop) を実行して、編集した Connector 属性ファイルの内容をリソースアダプタに反映します。

```
$COSMINEXUS_HOME/CC/admin/bin/cjsetresprop J2EEサーバ名 -type rar -resname DB_Connector_for_HiRDB_Type4 -c DBConnector_docb.xml
```

(5) リソースアダプタのデプロイ

次のコマンド (cjdeployrar) を実行して、リソースアダプタをデプロイします。

```
$COSMINEXUS_HOME/CC/admin/bin/cjdeployrar J2EEサーバ名 -resnameリソースアダプタの表示名
```

(6) リソースアダプタの接続テスト

次のコマンド (cjtestres) を実行して、リソースアダプタが正常に接続できるかテストします。

```
$COSMINEXUS_HOME/CC/admin/bin/cjtestres J2EEサーバ名 -type rar -resnameリソースアダプタの表示名
```

(7) リソースアダプタの開始

次のコマンド (cjstartrar) を実行して、リソースアダプタを開始します。

```
$COSMINEXUS_HOME/CC/admin/bin/cjstartrar J2EEサーバ名 -resnameリソースアダプタの表示名
```

3.10 Java アプリケーションの設定

Java アプリケーションで DocumentBroker を使用できます。この場合、環境変数と Java 実行時の `-classpath` オプション、データベース情報ファイル (`databaseinfo.properties`) の設定が必要です。

3.10.1 環境変数の設定

「3.7.2 DocumentBroker の環境変数の設定」に従い `DOCB_CONF_DIR` 環境変数を設定します。

3.10.2 JAR ファイルの追加

Java アプリケーションを実行する際の `-classpath` オプションに、次に示すファイルを追加します。

UNIX の場合

- `/opt/DocBroker/Platform/lib/djlib5.jar`
または
- `/opt/DocBroker/Library/lib/djlib5.jar`

Windows の場合

- `<インストールディレクトリ>%DocBroker%Platform%lib%djlib5.jar`
または
- `<インストールディレクトリ>%DocBroker%Library%lib%djlib5.jar`
または
- `<インストールディレクトリ>%DocBroker%Developer%lib%djlib5.jar`

また、使用する JDBC ドライバ (HiRDB の場合は「HiRDB Type4 JDBC Driver」) を `-classpath` オプションに追加してください。

3.10.3 データベース情報ファイルの設定

「4.7 データベース情報ファイル (`databaseinfo.properties`)」を参照して、データベースに接続するための情報を設定します。

3.11 複数の実行環境を構築する場合の設定

この節では、複数の実行環境から、同一の文書空間にアクセスする場合の実行環境の構築方法について説明します。DocumentBroker では、使用ユーザ数や単位時間あたりのトランザクション数の増加などのシステム負荷を軽減するために、一つのデータベースに複数の DocumentBroker の実行環境を配置したシステム構成を構築できます。これによって、システム全体の負荷分散が実現できます。

3.11.1 新規に実行環境を構築する場合

複数の実行環境から同一文書空間にアクセスするために、新規に実行環境を構築する手順を次に示します。なお、次の手順では、接続する文書空間にアクセスする実行環境がすでに構築されていることが前提です。

1. DocumentBroker の実行環境を作成する。

なお、DocumentBroker の実行環境の作成については、「3.7 DocumentBroker の実行環境の作成」を参照してください。

2. 文書空間構成定義ファイルを設定する。

このとき、同一文書空間として使用したい実行環境の文書空間構成定義ファイルの DocSpaceId と同じ値を指定します。

文書空間構成定義ファイルについては、「4.3 文書空間構成定義ファイル (docspace.properties)」を参照してください。

3. DocumentBroker 実行環境の情報の登録コマンド (dbjregenvi -r) を実行する。

これによって、新規に構築した実行環境から同一文書空間にアクセスできるようになります。

3.11.2 実行環境を変更する場合

別マシンへの実行環境の移動や、実行環境ディレクトリのパスの変更によって実行環境を変更する場合の手順を次に示します。

1. 変更後の実行環境を作成する。

DocumentBroker の実行環境の作成については、「3.7 DocumentBroker の実行環境の作成」を参照してください。

2. 変更前の実行環境ディレクトリ /conf に格納されているファイルを、変更後の実行環境ディレクトリ /conf に移動する。

3. 変更後の実行環境で、DocumentBroker 実行環境の情報の登録コマンド (dbjregenvi -u) を実行する。

3.11.3 実行環境を削除する場合

実行環境を削除する手順を次に示します。

1. DocumentBroker 実行環境の情報の登録コマンド (dbjregenvi -d) を実行する。

3.12 リファレンスファイル管理機能を使用する場合の設定

この節では、リファレンスファイル管理機能を使用するための設定について説明します。

3.12.1 コンテンツを管理するための設定

ここでは、リファレンスファイル管理機能を使用して、ファイルシステム上の任意のディレクトリにコンテンツを格納するための設定について説明します。

(1) コンテンツを格納する領域の確保

コンテンツを格納するために、ファイルシステム上に領域を確保する必要があります。確保する領域のディスク容量の見積もり方法については「2.5 リファレンスファイル管理機能を使用する場合のファイルシステムのディスク容量の見積もり」を参照してください。見積もりした値よりも大きいディスク容量を確保してください。

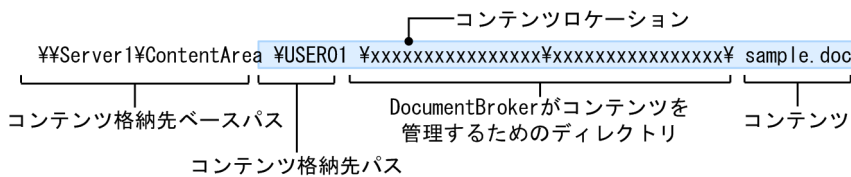
(2) コンテンツを格納するディレクトリの作成

コンテンツの格納先とするディレクトリを基点としたパスをコンテンツ格納先ベースパスといいます。コンテンツ格納先ベースパスに設定するディレクトリは、あらかじめファイルシステム上の任意の領域に作成しておく必要があります。

作成したディレクトリの下に、DocumentBroker によってコンテンツが格納されます。なお、リファレンスファイル文書の格納方式は、文書空間構成定義ファイルの ReferenceStorageMode プロパティで設定します。ReferenceStorageMode プロパティについては「4.3.2 文書空間構成定義ファイルの記述形式」を参照してください。

コンテンツ格納先ベースパス下のコンテンツ格納先の例を、次の図に示します。なお、ここでは Windows の場合を例としています。

図 3-11 文書空間構成定義ファイルの ReferenceStorageMode プロパティに Origin を指定した場合

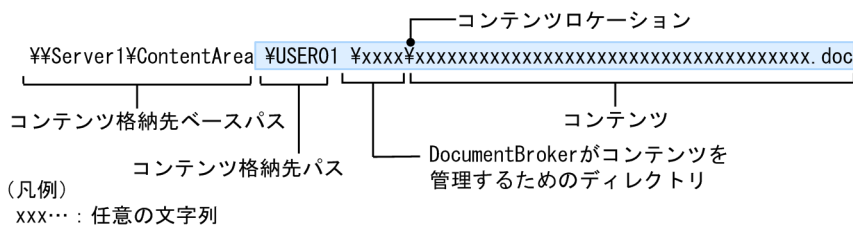


(凡例)
xxx…: 任意の文字列

注 UNC形式のパスでコンテンツ格納先ベースパスを指定した場合です。

Origin を指定した場合の例では、コンテンツ格納先ベースパスにはディレクトリ「¥¥Server1¥ContentArea」を、コンテンツ格納先パスにはディレクトリ「USER01」を設定して、コンテンツ「sample.doc」が格納された場合を示しています。

図 3-12 文書空間構成定義ファイルの ReferenceStorageMode プロパティに Divide を指定した場合



注 UNC形式のパスでコンテンツ格納先ベースパスを指定した場合です。

コンテンツ格納先ベースパス

ファイルシステム上の任意の領域に、コンテンツの格納先としてシステム管理者が作成するディレクトリです。このディレクトリは、あらかじめ作成しておく必要があります。作成したディレクトリをコンテンツ格納先ベースパスとして設定します。

文書空間で使用する文字コード種別が UTF-8 の場合、コンテンツ格納先ベースパスには、ASCII コードで表せるパスを指定してください。なお、コンテンツ格納先ベースパスは、絶対パスで指定してください。相対パスでの指定はできません。

コンテンツ格納先パス

コンテンツ格納先ベースパスからの相対パスで指定するディレクトリです。このディレクトリは、コンテンツ格納時に DocumentBroker によって作成されます。

文書空間で使用する文字コード種別が UTF-8 の場合、コンテンツ格納先パスには、ASCII コードで表せるパスを指定してください。

DocumentBroker がコンテンツを管理するためのディレクトリ

DocumentBroker によって作成される、コンテンツ管理するためのディレクトリです。このディレクトリの下にコンテンツが格納されます。

コンテンツ

DocumentBroker によってコンテンツが登録、管理されます。

コンテンツ格納先パス、DocumentBroker がコンテンツを管理するためのディレクトリ、およびコンテンツを合わせて、コンテンツロケーションとしてデータベースで管理されます。

なお、コンテンツ格納先ベースパスの長さ、コンテンツ格納先パスの長さおよびファイル名の長さの合計が、次の値を超えないようにしてください。

- 文書空間構成定義ファイルの ReferenceStorageMode プロパティに Origin (省略値) を指定した場合
コンテンツ格納先ベースパスの長さ (バイト) + コンテンツ格納先パスの長さ (バイト) + ファイル名の長さ (バイト) < OS の最大ファイルパス長 (バイト) - 35 (バイト)
- 文書空間構成定義ファイルの ReferenceStorageMode プロパティに Divide を指定した場合
コンテンツ格納先ベースパスの長さ (バイト) + コンテンツ格納先パスの長さ (バイト) + ファイル名の長さ (59 バイト) < OS の最大ファイルパス長 (バイト) - 6 (バイト)

(3) コンテンツを格納するディレクトリおよびコンテンツのアクセス権限の設定

UNIX の場合、コンテンツ格納先ベースパスに指定するディレクトリには、システム管理者およびシステム管理者グループに対してだけ読み取り権、書き込み権、実行権 (770) のアクセス権限を設定してください。

Windows の場合、コンテンツ格納先ベースパスに指定するディレクトリには、プロセスを起動するユーザに対してだけフルコントロールのアクセス権限を設定してください。

また、どちらの OS の場合でも、そのほかのグループまたはユーザに対しては、操作可能な権限を設定しないでください。

(4) システム構成に応じたコンテンツの格納先の設定

次に示すシステム構成の場合、必要に応じてコンテンツの格納先を設定してください。

複数の実行環境を構成する場合

DocumentBroker の実行環境ごとに、コンテンツの格納先ディレクトリを確保してください。このとき、コンテンツの格納先とするファイルシステムは、DocumentBroker が存在するマシンから接続可能なネットワーク上のマシンの共有ディスクを使用してください。また、Windows の場合、コンテンツ格納先ベースパスは、UNC 形式のパスで指定してください。

対象のネットワーク上のマシンには、プロセスを起動するユーザのユーザ名とパスワードを使ってログインします。必要なアクセス権限を設定してください。

複数の異なる文書空間で、コンテンツの格納先を同一のファイルシステムとする場合

各文書空間で、コンテンツ格納先ベースパスが重複しないようにしてください。複数の異なる文書空間で同一のコンテンツ格納先ベースパスを使用した場合、ファイル破壊の原因となります。

3.12.2 ネットワーク上のマシンの共有ディスクでコンテンツを管理する場合の設定

ここでは、複数の実行環境を構成するシステムでリファレンスファイル管理機能を使用する場合に、ネットワーク上のマシンの共有ディスクにコンテンツを格納するための設定について説明します。

(1) Windows の場合の設定

ネットワーク上のマシンの共有ディスクを使用する場合は、コンテンツ格納先ベースパスを UNC 形式のパスで指定してください。

対象のネットワーク上のマシンには、プロセスを起動するユーザのユーザ名とパスワードを使用してログインします。必要なアクセス権限を設定してください。なお、対象のネット上のマシンへの接続ユーザ数が上限に達した場合は、エラーが発生します。

(2) UNIX の場合の設定

NFS を使用するためには、DocumentBroker を起動する前に次の設定を実施してください。

NFS サーバの構成

コンテンツの格納先となるサーバマシンで、NFS サーバを構成しておきます。

NFS ファイルシステムのエクスポート

コンテンツの格納先となるサーバマシンでコンテンツの格納先ディレクトリを作成し、NFS ファイルシステムのエクスポートを行います。

NFS クライアントの構成

DocumentBroker が存在するマシンで、NFS クライアントの構成を定義します。

NFS ファイルシステムのマウント

DocumentBroker が存在するマシンで、NFS ファイルシステムをマウントします。

NFS の設定の詳細については、オペレーティングシステムのマニュアルを参照してください。

なお、コンテンツの格納先となるサーバマシンと、DocumentBroker が存在するマシンは同一のオペレーティングシステムを使用してください。また、パスワードファイルに設定するシステム管理者ユーザの

ユーザ ID，およびシステム管理者ユーザが所属するシステム管理者グループのグループ ID は，マシン間で同一の内容にしておく必要があります（同じパスワードファイルを使用することを推奨します）。

システム管理者ユーザのユーザ ID，およびシステム管理者ユーザが所属するシステム管理者グループのグループ ID が異なる場合，ファイルアクセスでのエラーの原因となります。

4

環境設定で必要なファイル

この章では、DocumentBroker の環境設定で必要なファイルについて説明します。

-
- 4.1 ファイルの種類

 - 4.2 動作環境定義ファイル (docbconf.properties)

 - 4.3 文書空間構成定義ファイル (docspace.properties)

 - 4.4 ユーザ権限定義ファイル (userperm.properties)

 - 4.5 レン디션定義ファイル (mime.properties)

 - 4.6 メタ情報初期化ファイル (docbmetainit.properties)

 - 4.7 データベース情報ファイル (databaseinfo.properties)

 - 4.8 メタ情報ファイル

 - 4.9 定義情報ファイル

 - 4.10 RD エリア定義情報ファイル

 - 4.11 インデクス情報ファイル

 - 4.12 変更情報ファイル
-

4.1 ファイルの種類

DocumentBroker の環境設定で必要なファイルには、プロパティファイルとコマンドで使用するファイルがあります。

4.1.1 プロパティファイル

DocumentBroker の環境設定で必要なプロパティファイルの一覧を次に示します。これらのファイルは、DocumentBroker の実行環境を作成するときに作成してください。

表 4-1 環境設定で必要なプロパティファイル

名称	内容	参照先
動作環境定義ファイル (docbconf.properties)	DocumentBroker の動作環境を定義する。	4.2
文書空間構成定義ファイル (docspace.properties)	DocumentBroker の文書空間の構成を定義する。	4.3
ユーザ権限定義ファイル (userperm.properties)	文書空間のすべてのオブジェクトに対するの権限をユーザ、またはグループ単位で定義する。	4.4
レンディション定義ファイル (mime.properties)	登録するファイルの拡張子からレンディションタイプ (MIME 形式) を自動的に設定する。	4.5

注 次の場所に提供されています。コピー先にコピーしてから使用してください。

UNIX の場合

格納先:

- /opt/DocBroker/Platform/conf

または

- /opt/DocBroker/Library/conf

コピー先: 実行環境ディレクトリ /conf 以下

Windows の場合

格納先:

- <インストールディレクトリ>%DocBroker%\Platform\conf

または

- <インストールディレクトリ>%DocBroker%\Developer\conf

または

- <インストールディレクトリ>%DocBroker%\Library\conf

コピー先: 実行環境ディレクトリ %conf 以下

(1) プロパティファイルの記述規則

ここでは、プロパティファイルの記述規則について説明します。

プロパティファイルは、Java 2 Platform, Standard Edition のプロパティファイル形式です。

プロパティは、「プロパティ名 = 値」の形式で指定します。

プロパティ名、「=」、値の間に空白文字を指定すると無視されます。

プロパティ名の次の文字 (空白文字を除く) が「=」の場合、その次の文字 (空白文字を除く) から行

末文字までを値として解釈されます。

値の後ろに空白文字を付加した場合、空白文字は無視されます。

値の中に「=」、「#」、または「!」が使用されている場合、値の一部と解釈されます。

プロパティファイルの文字は、ISO 8859-1 (Latin1) でエンコードします。ファイルの中に ISO 8859-1 以外の文字コードの文字が含まれる場合は、Java ツールの native2ascii などを使ってエンコードする必要があります。

行頭文字に「#」を指定した場合、行全体がコメントと見なされます。

行頭に続く空白文字（半角スペース、タブまたは改行）は無視されます。

ディレクトリを指定する場合、区切り文字（記号）には、「¥¥」または「/」を使用してください。

指定した値の大文字と小文字は区別されます。

4.1.2 コマンドで使用するファイル

DocumentBroker のコマンドに必要なファイルの一覧を次に示します。これらのファイルは、DocumentBroker でデータベースシステムを使用するための設定を始める前に、作成してください。

表 4-2 コマンドに必要なファイル

名称	内容	参照先
メタ情報初期化ファイル (docbmetainit.properties) ¹	DocumentBroker のシステム設定を記述する。	4.6
データベース情報ファイル (databaseinfo.properties) ¹	コマンドが接続するデータベースの情報を記述する。	4.7
メタ情報ファイル (edmsys.csv, edmclass.csv, edmclassprop.csv, edmprop.csv)	システムクラスを定義したメタ情報を提供する。	4.8
定義情報ファイル ²	サブクラスおよびプロパティを追加するときに、追加するオブジェクトの定義情報を定義する。	4.9
RD エリア定義情報ファイル	DocumentBroker が使用するユーザ表およびインデクスを、どの RD エリアに格納するか定義する。	4.10
インデクス情報ファイル	追加するプロパティのインデクス情報を定義する。	4.11
変更情報ファイル	ユーザ識別子、グループ識別子、ユーザプロパティの変更内容を定義します。 変更情報ファイルには、次の 3 種類があります。 <ul style="list-style-type: none"> ユーザ識別子変更情報ファイル グループ識別子変更情報ファイル プロパティ変更情報ファイル 	4.12

注 1

コピー先にコピーしてから使用してください。提供場所については「4.1.1 プロパティファイル」を参照してください。

注 2

次の場所にサンプルファイルが提供されています。

4. 環境設定に必要なファイル

- UNIX の場合 : /opt/DocBroker/Platform/sample
- Windows の場合 : <インストールディレクトリ>\DocBroker\Platform\sample , または <インストールディレクトリ>\DocBroker\Developer\sample

4.2 動作環境定義ファイル (docbconf.properties)

この節では、動作環境定義ファイルについて説明します。

4.2.1 動作環境定義ファイルの概要

動作環境定義ファイルは、DocumentBroker の動作環境を定義するファイルです。

システム管理者はコピーした「docbconf.properties」を、テキストエディタなどを使用して編集してください。

動作環境定義ファイルで指定するプロパティを次の表に示します。各プロパティの詳細については、「4.2.2 動作環境定義ファイルの記述形式」を参照してください。

表 4-3 動作環境定義ファイルのプロパティ一覧

種類	設定項目	プロパティ名	指定内容	省略の可否
メッセージログ・トレースファイル	出力先ディレクトリ	LogPath	メッセージログ・トレースファイルの出力先ディレクトリの絶対パス	
	メッセージログファイルの設定	MsgLogSize	メッセージログファイルのファイルサイズの上限值	
		MsgLogNumber	メッセージログファイルの最大面数	
	トレースファイルの設定	TraceLevel	トレースファイルのトレースレベル	
		TraceSize	トレースファイルのファイルサイズの上限值	
		TraceNumber	トレースファイルの最大面数	
アプリケーションエラーログ・アプリケーショントレースファイル	出力先ディレクトリ	APTracePath	アプリケーションエラーログ・アプリケーショントレースファイルの出力先ディレクトリの絶対パス	
	アプリケーションエラーログファイルの設定	APErrorLogSize	アプリケーションエラーログファイルのファイルサイズの上限值	
		APErrorLogNumber	アプリケーションエラーログファイルの最大面数	
	アプリケーショントレースファイルの設定	APTraceLevel	アプリケーショントレースファイルのトレースレベル	
		APTraceSize	アプリケーショントレースファイルのファイルサイズの上限值	
		APTraceNumber	アプリケーショントレースファイルの最大面数	

4. 環境設定に必要なファイル

種類	設定項目	プロパティ名	指定内容	省略の可否
		PromptOutput	アプリケーショントレース情報を標準出力に出力するか、標準エラー出力に出力するか	

(凡例)

: 指定を省略できます。

× : 指定を省略できません。

4.2.2 動作環境定義ファイルの記述形式

動作環境定義ファイルの記述形式を次に示します。なお、動作環境定義ファイルの記述規則については、「4.1.1(1) プロパティファイルの記述規則」を参照してください。

< プロパティ名 >=< 値 >

以降、動作環境定義ファイルに記述するプロパティについて説明します。

LogPath プロパティ

メッセージログ・トレースの出力先ディレクトリを絶対パスで指定します。パスは印刷可能な ASCII コードで指定してください。

指定を省略した場合、または次に示す値を指定した場合、UNIX の場合は「\$DOCB_CONF_DIR/log」が、Windows の場合は「%DOCB_CONF_DIR%\log」が仮定されます。

- 存在しないディレクトリ
- ディレクトリでない
- 書き込み権限のないディレクトリ

MsgLogSize プロパティ

メッセージログファイルの 1 面当たりの最大サイズを指定します。指定できる値の範囲は、4,096 ~ 2,147,483,647 (バイト) です。指定を省略した場合、「16,777,216」が仮定されます。

MsgLogNumber プロパティ

ファイルサイズの上限を超えた場合に出力する、メッセージログファイルの最大面数を指定します。最新の情報は、常に最新の面 (ファイル名 0) に記載されます。

指定できる値の範囲は、1 ~ 16 の整数です。指定を省略した場合は、「10」が仮定されます。

TraceLevel プロパティ

トレースファイルのトレースレベルを指定します。指定できる値は「-1」「0」「10」「20」「30」です。指定する値ごとのトレースレベルと出力情報を次の表に示します。指定を省略した場合、「0」が仮定されます。

表 4-4 トレースレベルと出力情報

トレースレベル	設定基準	出力内容
-1	ログを出力しない。	何も出力されません。
0	通常運用でのトレースレベル。	<ul style="list-style-type: none"> • 例外発生 • 例外発生時のスタックトレース • 製品公開メソッドの開始・終了 • SQL 実行時の SQL 文

トレースレベル	設定基準	出力内容
10	<ul style="list-style-type: none"> • 通常運用よりも詳細な情報を取得するために利用されることを想定したレベル。 • 発生頻度の低い障害の再発監視や本番前の現地テストへの適用を想定。 	<ul style="list-style-type: none"> • トレースレベルが 0 の場合に出力される情報 • 他プログラムの呼出 / 戻り
20	<ul style="list-style-type: none"> • 障害の個所を詳細に特定するために使用するログレベル。 • プログラムの動作シーケンスを把握できるレベル。 	<ul style="list-style-type: none"> • トレースレベルが 10 の場合に出力される情報 • 主要メソッドの開始 / 終了
30	<ul style="list-style-type: none"> • 障害の個所を詳細に特定するために使用するログレベル。 	<ul style="list-style-type: none"> • トレースレベルが 20 の場合に出力される情報 • すべてのメソッドの開始 / 終了

注

パラメタクラスのメソッドを除きます。

TraceSize プロパティ

トレースファイルの 1 面当たりの最大サイズを指定します。指定できる値の範囲は、4,096 ~ 2,147,483,647 (バイト) です。指定を省略した場合、「16,777,216」が仮定されます。

TraceNumber プロパティ

ファイルサイズの上限を超えた場合に出力する、トレースファイルの最大面数を指定します。最新の情報は、常に最新の面 (ファイル名 0) に記載されます。

指定できる値の範囲は、1 ~ 16 の整数です。指定を省略した場合、「10」が仮定されます。

APTracePath プロパティ

アプリケーショントレースおよびアプリケーションエラーログの出力先ディレクトリを絶対パスで指定します。パスは印刷可能な ASCII コードで指定してください。

指定を省略した場合、または次に示す値を指定した場合、UNIX の場合は「\$DOCB_CONF_DIR/log」が、Windows の場合は「%DOCB_CONF_DIR%\log」が仮定されます。

- 存在しないディレクトリ
- ディレクトリでない
- 書き込み権限のないディレクトリ

APTraceLevel プロパティ

アプリケーショントレースファイルのトレースレベルを指定します。この設定は、次に示す出力先にトレース情報を出力する場合に有効になります。

- コマンドプロンプト
- アプリケーショントレースファイル

指定できる値は「-1」「0」「10」「20」「30」です。指定する値ごとのトレースレベルと出力情報については、表 4-4、およびマニュアル「DocumentBroker Version 5 リファレンス API」を参照してください。指定を省略した場合、「0」が仮定されます。

APTraceSize プロパティ

アプリケーショントレースファイルの 1 面当たりの最大サイズを指定します。指定できる値の範囲は、4,096 ~ 2,147,483,647 (バイト) です。指定を省略した場合、「1,048,576」が仮定されます。

APTraceNumber プロパティ

ファイルサイズの上限を超えた場合に出力する、アプリケーショントレースファイルの最大面数を指定します。最新の情報は、常に最新の面 (ファイル名 0) に記載されます。

指定できる値の範囲は、1 ~ 16 です。指定を省略した場合、「2」が仮定されます。

4. 環境設定で必要なファイル

APErrLogSize プロパティ

アプリケーションエラーログファイルの 1 面当たりの最大サイズを指定します。指定できる値の範囲は、4,096 ~ 2,147,483,647 (バイト) です。指定を省略した場合、「1,048,576」が仮定されます。

APErrLogNumber プロパティ

ファイルサイズの上限を超えた場合に出力する、アプリケーションエラーログファイルの最大面数を指定します。最新の情報は、常に最新の面 (ファイル名 0) に記載されます。

指定できる値の範囲は、1 ~ 16 です。指定を省略した場合、「2」が仮定されます。

PromptOutput プロパティ

アプリケーショントレース出力時の出力先にコマンドプロンプトを指定した場合、アプリケーショントレース情報を標準出力に出力するか、標準エラー出力に出力するかを指定します。

- STDOUT

アプリケーショントレース情報を標準出力に出力します。

- STDERR

アプリケーショントレース情報を標準エラー出力に出力します。

指定を省略した場合、「STDOUT」が仮定されます。

4.3 文書空間構成定義ファイル (docspace.properties)

この節では、文書空間構成定義ファイルについて説明します。

4.3.1 文書空間構成定義ファイルの概要

文書空間構成定義ファイルは、DocumentBroker の文書空間の構成を定義するファイルです。システム管理者はコピーした「docspace.properties」を、テキストエディタなどを使用して編集してください。

文書空間構成定義ファイルで指定するプロパティを次の表に示します。各プロパティの詳細については、「4.3.2 文書空間構成定義ファイルの記述形式」を参照してください。

表 4-5 文書空間構成定義ファイルのプロパティ一覧

項番	種類	設定項目	プロパティ名	指定内容	省略の可否
1	サービス	文書空間	DocSpaceId	文書空間の識別子	×
2			ConnectorName	DB Connector の表示名	1
3	データベース	接続するデータベースの設定	DBType	データベース種別	
4	ユーザ認証	ユーザ認証	UserAuthentication	ユーザ認証方式	
5		LDAP 設定	LdapHost	ディレクトリサーバのホスト名または IP アドレス	
6			LdapPort	ディレクトリサーバのポート番号	
7		UOC ライブラリの設定 (UOC 指定時)	UOCClass	UOC で使用するクラスのクラス名	
8	ユーザ情報の検索	検索条件	LdapUserRoot	ユーザ検索対象のベースになる DN	
9			LdapUserScope	ユーザ検索時の検索範囲	
10			LdapUserClass	ユーザとして定義しているクラスの名称	
11			LdapUserId	ユーザを検索するときのキーとなる属性	
12			LdapUserFilterLeft	検索フィルタの「(」(左括弧) および検索フィルタの内容	
13			LdapUserFilterRight	検索フィルタの「)」(右括弧) および検索フィルタの内容	
14		検索オプション	LdapUserTimeout	ユーザ検索時の最大待ち時間	
15		LdapUserCase	ユーザ識別子の文字種の制限		
16	グループ情報の検索	ディレクトリエントリからの検索	LdapGroup	グループ識別子のオブジェクトからの検索方法	
17			LdapGroupRoot	グループ検索対象のベースになる DN	
18			LdapGroupScope	グループ検索時の検索範囲	

4. 環境設定で必要なファイル

項番	種類	設定項目	プロパティ名	指定内容	省略の可否
19			LdapGroupClass	グループ識別子にするオブジェクトのクラス	
20			LdapGroupId	グループを検索するときのキーとなる属性	
21			LdapGroupFilterLeft	検索フィルタの「(」(左括弧)および検索フィルタの内容	
22			LdapGroupFilterRight	検索フィルタの「)」(右括弧)および検索フィルタの内容	
23			LdapGroupMember	グループに所属しているユーザを記述する属性	
24		ユーザの属性値からの検索	LdapGroupFromUserAttr	ユーザの属性値からのディレクトリのノード情報の検索方法	
25			LdapGroupIdFromUserAttr	グループ情報が格納されているユーザの属性	
26			LdapGroupIsDnFromUserAttr	ユーザの属性に格納されているグループ情報	
27			LdapGroupIdAttrFromUserAttr	グループ識別子として利用する属性	
28		ユーザの DN からの検索	LdapGroupFromUserDn	グループ識別子の検索方法	
29			LdapGroupIdFromUserDn	グループ識別子として使用する属性名	
30		検索オプション	LdapGroupTimeout	グループ検索時の最大待ち時間	
31			LdapGroupCase	グループ識別子の文字種の制限	
32		グループ識別子の取得	LdapGroupFromGroupDn	グループ識別子をグループプロパティの上位にあるプロパティから取得	
33			LdapGroupFromGroupMember	グループ識別子をグループプロパティに関連付けているグループプロパティから取得	
34	ディレクトリサービスに対するバインド方法	ディレクトリサービスに対するバインド方法	LdapBindUserDN	ディレクトリサーバにアクセスする認証者の DN	
35			LdapBindPassword	ディレクトリサーバにアクセスする際の認証者に対応するパスワード	
36	アクセス制御機能	セキュリティ管理者	SecurityAdmin	セキュリティ管理者	2
37		ユーザ権限の定義	UserPermDef	ユーザ権限の定義	
38		パーミッション	DefaultACFlagOwner	オブジェクトの所有者に対するパーミッション	
39			DefaultACFlagGroup	グループに対するパーミッション	
40		DefaultACFlagEveryone	すべてのユーザに対するパーミッション		

項番	種類	設定項目	プロパティ名	指定内容	省略の可否
41	サイズ設定	コンテンツサイズの設定	MaxContentSize	コンテンツサイズの最大バイト数	
42		全文検索テキストのサイズ設定	MaxFullTextSize	全文検索テキストの最大バイト数	
43		データサイズ設定	BinaryAccessSize	バイナリデータのアクセスサイズ	
44	リファレンスファイル文書化	格納方式設定	ReferenceStorageMode	リファレンスファイル文書の格納方式	
45		ディレクトリ数設定	ReferenceDirDivideNum	リファレンスファイル文書を格納するディレクトリ数	
46		ベースパス設定	ReferenceBasePath	リファレンスファイル文書を格納するベースパス	
47	ストリーム形式での操作	出力先ディレクトリの設定	TempDirectory	一時ディレクトリ名	

(凡例)

: 指定を省略できます。

x: 指定を省略できません。

注 1

DocumentBroker クラスライブラリを使用した J2EE アプリケーションを実行する場合、指定は省略できません。

注 2

DocumentBroker クラスライブラリを使用したアプリケーションを実行する場合、指定は省略できません。

4.3.2 文書空間構成定義ファイルの記述形式

文書空間構成定義ファイルの記述形式を次に示します。なお、文書空間構成定義ファイルの記述規則については、「4.1.1(1) プロパティファイルの記述規則」を参照してください。

<プロパティ名>=<値>

以降、文書空間構成定義ファイルに記述するプロパティについて説明します。

DocSpaceId プロパティ

文書空間の識別子として、GUID を指定します。GUID は dbjgenguid コマンドで生成できます。

GUID は「X」を 0 ~ 9、および a ~ f (小文字) で表される 16 進数とした

「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8-4-4-4-12)」の形式で表されます。デフォルト値は存在しません。また、このプロパティの指定は省略できません。

ConnectorName プロパティ

データベースに対応するリソースアダプタの設定で指定する DB Connector の表示名を指定します。

DBType プロパティ

文書空間が接続するデータベース種別指定します。「HiRDB」を指定します。指定を省略した場合は「HiRDB」が仮定されます。

UserAuthentication プロパティ

文書空間で使用するユーザ認証方式を指定します。

- LDAP

LDAP 対応のディレクトリサービスと連携してユーザ認証を実行します。

4. 環境設定に必要なファイル

DocumentBroker では、LDAP 対応のディレクトリサービスとして、次の製品との連携をサポートしています。

- Windows Server 2008 R2 および Windows Server 2012 の Active Directory
- Oracle Directory Server Enterprise Edition
- UOC
ユーザ管理システムへのアクセスルーチンとして、ユーザが作成したアクセスルーチンを使用する場合に指定します。

指定を省略した場合、「LDAP」が仮定されます。

LdapHost プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。

LDAP 対応のディレクトリサービスのホスト名または IP アドレスを指定します。指定を省略した場合、何も仮定されません。

UOCClass プロパティ

UserAuthentication プロパティで「UOC」を指定した場合には、必ず UOC で使用する Class のクラス名を指定します。指定を省略した場合、何も仮定されません。

LdapPort プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。LDAP 対応のディレクトリサービスのポート番号を指定します。指定を省略した場合、標準のポート番号（389）が仮定されます。

LdapUserId プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザ識別子として利用するユーザを検索するときのキーとなる属性を指定します。ただし、指定する属性は、ユーザの一意性を保証している必要があります。指定を省略した場合、「uid」が仮定されます。

LDAP 対応のディレクトリサービスとして Active Directory を使用する場合、「sAMAccountName」または「userPrincipalName」を指定してください。ただし、「userPrincipalName」を指定した場合、ログイン名として「ユーザ名@ドメイン名」の形式で指定する必要があります。

LdapUserTimeout プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリプロパティからユーザを検索するときの最大待ち時間を指定します。0 ~ 180（秒）の範囲で指定します。0 を指定した場合、待ち時間が無制限になります。指定を省略した場合、「60（秒）」が仮定されます。

LdapUserCase プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。DocumentBroker で扱うユーザ識別子の文字種についての制限を指定します。

- Upper
ユーザ識別子の文字列を、すべて大文字に変換する場合に指定します。LDAP 対応のディレクトリサービスから取得したユーザ識別子を、DocumentBroker 内で大文字に変換します。
- Lower
ユーザ識別子の文字列を、すべて小文字に変換する場合に指定します。LDAP 対応のディレクトリサービスから取得したユーザ識別子を、DocumentBroker 内で小文字に変換します。
- Default
ユーザ識別子の文字列を変換しません。

指定を省略した場合、「Default」が仮定されます。

LdapUserRoot プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。DIT 上でユーザを検索するベースとなるノードの DN を指定します。指定を省略した場合、何も仮定されません。

LdapUserScope プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリプロパティからユーザを検索するときの検索範囲を指定します。

- Onelevel
検索開始点の一つ下のレベルにあるすべてのプロパティを検索します。
- Subtree
検索開始点と、その下のすべてのレベルにあるすべてのプロパティを検索します。

指定を省略した場合、「Subtree」が仮定されます。

LdapUserClass プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザとして定義しているクラスの名称を指定します。

LDAP 対応のディレクトリサービスとして Active Directory を使用する場合、「user」または user のサブクラスを指定してください。指定を省略した場合、「inetOrgPerson」が仮定されます。

LdapUserFilterLeft プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザを検索する場合に指定する検索フィルタの「(」(左括弧) および追加指定する検索フィルタの内容を指定します。

指定を省略した場合、「(」が仮定されます。

LdapUserFilterRight プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザを検索する場合に指定する検索フィルタの「)」(右括弧) および追加指定する検索フィルタの内容を指定します。

指定を省略した場合、「)」が仮定されます。

LdapGroup プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループ識別子を、メンバ(ユーザ)をリストとしたオブジェクトから検索することを指定します。

- Yes
メンバ(ユーザ)をリストとしたオブジェクトから検索します。
- No
メンバ(ユーザ)をリストとしたオブジェクトから検索しません。

指定を省略した場合、「Yes」が仮定されます。

LdapGroupRoot プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。DIT 上でグループを検索する基となるノードの DN を指定します。指定を省略した場合、何も仮定されません。

LdapGroupScope プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリプロパティでグループを検索する場合の検索範囲を指定します。

- Onelevel
検索開始点の一つ下のレベルにあるすべてのプロパティを検索します。
- Subtree
検索開始点とその下のすべてのレベルにあるすべてのプロパティを検索します。

4. 環境設定で必要なファイル

指定を省略した場合、「Subtree」が仮定されます。

LdapGroupClass プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループ識別子としてメンバ（ユーザ）をリストとしたオブジェクトのクラスを指定します。指定を省略した場合、「groupOfUniqueNames」が仮定されます。

なお、LDAP 対応のディレクトリサービスとして Active Directory を使用する場合、「group」または group のサブクラスを指定してください。

LdapGroupId プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループ識別子として利用する属性を指定します。指定を省略した場合、「cn」が仮定されます。

なお、LDAP 対応のディレクトリサービスとして Active Directory を使用する場合、「sAMAccountName」を指定してください。

LdapGroupFilterLeft プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループを検索するときに指定する検索フィルタの「(」(左括弧)および追加指定する検索フィルタの内容を指定します。指定を省略した場合、「(」が仮定されます。

LdapGroupFilterRight プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループを検索する場合に LDAP に指定する検索フィルタの「)」(右括弧)および追加指定する検索フィルタの内容を指定します。指定を省略した場合、「)」が仮定されます。

LdapGroupMember プロパティ

UserAuthentication プロパティで「LDAP」を指定し、LdapGroup プロパティに「Yes」を指定したときに有効になります。グループに所属しているユーザを記述する属性を指定します。指定を省略した場合、「uniqueMember」が仮定されます。

なお、LDAP 対応のディレクトリサービスとして Active Directory を使用する場合、「member」を指定してください。

LdapGroupFromUserAttr プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザの属性値からのグループ識別子を取得するか、取得しないかを指定します。

- Yes
ユーザの属性値からグループ識別子を取得します。
- No
ユーザの属性値からグループ識別子を取得しません。

指定を省略した場合、「No」が仮定されます。

LdapGroupIdFromUserAttr プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループ情報が格納されているユーザの属性を指定します。指定を省略した場合、「ou」が仮定されます。

LdapGroupsDnFromUserAttr プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザの属性に格納されているグループの情報が DN かどうかを指定します。

- Yes

ユーザの属性に格納されているグループの情報が DN の場合、指定します。この場合、DocumentBroker は、LdapGroupIdAttrFromUserAttr プロパティで指定されたグループの属性の情報を取得します。

- No

ユーザの属性に格納されているグループの情報が DN でない場合、指定します。この場合、格納された情報を、そのままグループ識別子とします。

指定を省略した場合、「No」が仮定されます。

LdapGroupIdAttrFromUserAttr プロパティ

LdapGroupFromUserAttr プロパティおよび LdapGroupIsDnFromUserAttr プロパティに「Yes」を指定した場合に有効になります。

ユーザの属性から取得した情報で、グループのグループ識別子として利用する属性を指定します。指定を省略した場合、「dn」が仮定されます。

LdapGroupFromUserDn プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。グループ識別子をユーザプロパティの DN の構成要素となる属性値から検索するか、検索しないかを指定します。

- Yes

グループ識別子をユーザプロパティの DN の構成要素となる属性から取得します。

- No

グループ識別子をユーザプロパティの DN の構成要素となる属性から取得しません。

指定を省略した場合、「No」が仮定されます。

LdapGroupIdFromUserDn プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ユーザプロパティの DN の構成要素の中で、グループ識別子として使用する属性名を指定します。指定を省略した場合、「ou」が仮定されます。

LdapGroupTimeout プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリプロパティからグループを検索するときの最大待ち時間を指定します。0 ~ 180 (秒) の範囲で指定します。0 を指定した場合、待ち時間が無制限になります。指定を省略した場合、「60 (秒)」が仮定されます。

LdapGroupCase プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。DocumentBroker で扱うグループ識別子の文字種についての制限を指定します。

- Upper

グループ識別子の文字列を、すべて大文字に変換する場合に指定します。LDAP 対応のディレクトリサービスから取得したグループ識別子を、DocumentBroker 内で大文字に変換します。

- Lower

グループ識別子の文字列を、すべて小文字に変換する場合に指定します。LDAP 対応のディレクトリサービスから取得したグループ識別子を、DocumentBroker 内で小文字に変換します。

- Default

グループ識別子の文字列を変換しません。

指定を省略した場合、「Default」が仮定されます。

LdapGroupFromGroupDn プロパティ

UserAuthentication プロパティで「LDAP」を指定し、LdapGroup プロパティに「Yes」を指定した

4. 環境設定に必要なファイル

場合に有効になります。グループ識別子をグループプロパティの上位にあるプロパティから取得するかどうかを指定します。

なお、最上位プロパティは LdapGroupRoot プロパティ、グループを検索するときのキーとなる属性は LdapGroupId プロパティの指定値となります。

- Yes
グループ識別子をグループプロパティの上位にあるプロパティから取得します。
- No
グループ識別子をグループプロパティの上位にあるプロパティから取得しません。デフォルトの値です。

指定を省略した場合、「No」が仮定されます。

LdapGroupFromGroupMember プロパティ

UserAuthentication プロパティで「LDAP」を指定し、LdapGroup プロパティに「Yes」を指定した場合に有効になります。グループ識別子をグループプロパティに関連付いているグループプロパティから、再帰的に取得する場合に指定します。なお、関連付いているグループを記述する属性は LdapGroupMember プロパティ、グループを検索するときのキーとなる属性は LdapGroupId プロパティで指定した値となります。

- Yes
グループ識別子をグループプロパティに関連付いているグループプロパティから、再帰的に取得します。
- No
グループ識別子をグループプロパティに関連付いているグループプロパティから、再帰的に取得しません。デフォルトの値です。

指定を省略した場合、「No」が仮定されます。

LdapBindUserDN プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリサービスからのデータ読み込み時に、ディレクトリサービスに対して認証を行ってバインドする場合は、認証に使用する DN を指定します。

指定を省略した場合、ディレクトリサービスからのデータ読み込み時に、ディレクトリサービスに対して匿名バインドが使用されます。

なお、ディレクトリサービスに対して認証を行ってバインドする場合は、必ず LdapBindUserDN プロパティと LdapBindPassword プロパティの両方を指定してください。どちらか一方でも指定を省略した場合は、ディレクトリサービスに対して匿名バインドが使用されます。

LdapBindPassword プロパティ

UserAuthentication プロパティで「LDAP」を指定した場合に有効になります。ディレクトリサービスからのデータ読み込み時に、ディレクトリサービスに対して認証を行ってバインドする場合は、認証に使用するパスワードを指定します。

なお、ディレクトリサービスに対して認証を行ってバインドする場合は、必ず LdapBindUserDN プロパティと LdapBindPassword プロパティの両方を指定してください。どちらか一方でも指定を省略した場合は、ディレクトリサービスに対して匿名バインドが使用されます。

SecurityAdmin プロパティ

DocumentBroker に登録されたオブジェクトやオブジェクトのアクセス権を保守するユーザ（セキュリティ管理者）を指定します。このプロパティは、アクセス制御機能に対応するデータベーススキーマが構築されている場合、有効になります。セキュリティ管理者名には、ユーザ識別子を指定してください。

セキュリティ管理者を複数指定する場合は、ユーザ識別子を「,」（コンマ）で区切って指定します。セキュリティ管理者の人数に制限はありません。セキュリティ管理者の指定例を次に示します。

セキュリティ管理者の指定例

SecurityAdmin= ユーザ識別子 1, ユーザ識別子 2

ユーザ識別子については、「3.7.3(1) ユーザ情報の取得方法」を参照してください。なお、このプロパティの指定は省略できません。

UserPermDef プロパティ

文書空間のすべてのオブジェクトに対しての権限をユーザ、またはグループ単位で定義するか、定義しないかを指定します。定義する場合、ユーザ権限定義ファイルを作成します。ユーザ権限定義ファイルについては、「4.4 ユーザ権限定義ファイル (userperm.properties)」を参照してください。このプロパティは、アクセス制御機能に対応するデータベーススキーマが構築されている場合、有効になります。

- Yes

文書空間のすべてのオブジェクトに対しての権限をユーザ、またはグループ単位で定義します。

- No

文書空間のすべてのオブジェクトに対しての権限をユーザ、またはグループ単位で定義しません。

指定を省略した場合、「No」が仮定されます。

DefaultACFlagOwner プロパティ

オブジェクトの作成時にデフォルトで付加される ACFlag を指定します。オブジェクトの所有者に対するパーミッションを指定します。ACFlag は、表 4-6 および表 4-7 に示されるパーミッション文字列を「|」（ストローク）で結合して指定します。指定を省略した場合、「FULL_CONTROL」が仮定されます。

DefaultACFlagGroup プロパティ

オブジェクトの作成時にデフォルトで付加される ACFlag を指定します。グループに対するパーミッションを指定します。ACFlag は、表 4-6 および表 4-7 に示されるパーミッション文字列を「|」（ストローク）で結合して指定します。指定を省略した場合、「FULL_CONTROL」が仮定されます。

DefaultACFlagEveryone プロパティ

オブジェクトの作成時にデフォルトで付加される ACFlag を指定します。すべてのユーザに対するパーミッションを指定します。ACFlag は、表 4-6 および表 4-7 に示されるパーミッション文字列を「|」（ストローク）で結合して指定します。指定を省略した場合、「FULL_CONTROL」が仮定されます。

DefaultACFlagOwner プロパティ、DefaultACFlagGroup プロパティ、および DefaultACFlagEveryone プロパティに設定できるパーミッションには、基本パーミッションと組み合わせパーミッションがあります。どちらのパーミッションも指定できます。また、この 2 種類の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。したがって、各プロパティの値には、次に示す表のパーミッション文字列を指定します。基本パーミッションと対応する文字列を表 4-6 に示します。また、組み合わせパーミッションと対応する文字列を表 4-7 に示します。

表 4-6 基本パーミッションと対応する文字列

パーミッション文字列	パーミッション	意味
PRIM_READ_PROPS	基本プロパティ参照権	プロパティを参照する権限
PRIM_WRITE_PROPS	基本プロパティ更新権	プロパティを更新する権限
PRIM_READ_CONTENTS	基本コンテンツ参照権	文書の内容を参照する権限

4. 環境設定で必要なファイル

パーミッション文字列	パーミッション	意味
PRIM_WRITE_CONTENTS	基本コンテンツ更新権	文書の内容を更新する権限
PRIM_LINK	基本リンク権	オブジェクト同士を関連づける権限
PRIM_VERSION	基本バージョン管理権	オブジェクトのバージョンに対する操作をする権限
PRIM_DELETE	基本削除権	オブジェクトを削除する権限

基本プロパティ参照権は、そのほかの基本パーミッションにも含まれます。したがって、基本プロパティ参照権以外の基本パーミッションを指定すると、各基本パーミッションで許可される操作に加えて、オブジェクトのプロパティの参照が許可されます。なお、基本パーミッションは、アクセス制御の対象となるオブジェクトごとに許可される操作の内容が異なります。詳細については、マニュアル「DocumentBroker Version 5 概説」を参照してください。

表 4-7 組み合わせパーミッションと対応する文字列

パーミッション文字列	パーミッション	意味
READ_PROPS	プロパティ参照権	基本プロパティ参照権と同じ権限
READ	参照権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 基本プロパティ参照権 • 基本コンテンツ参照権
WRITE_PROPS	プロパティ更新権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 基本プロパティ参照権 • 基本プロパティ更新権
READ_WRITE	参照更新権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 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
DELETE	削除権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 基本プロパティ参照権 • 基本削除権
LINK	リンク権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 基本プロパティ参照権 • 基本リンク権
VERSION	バージョン権	次の基本パーミッションを組み合わせた権限 <ul style="list-style-type: none"> • 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 • 基本バージョン管理権
FULL_CONTROL	フルコントロール	すべての基本パーミッションを組み合わせた権限

なお、表 4-6 および表 4-7 に示したパーミッションのほかに、「NONE」が指定できます。NONE を指定すると、新規にオブジェクトを作成した場合に権限は設定されません。また「NONE」は、ほかのパーミッション文字列と一緒に指定できません。例えば、「DefaultACFlagEveryone=NONE | LINK」といった指定はできません。

MaxContentSize プロパティ

登録する文書のファイルの最大サイズを、0 ~ 2147483647 の範囲で指定します。単位はバイトです。0 を指定した場合、登録する文書のサイズをチェックしません。指定を省略した場合は「0」が仮定されます。

MaxFullTextSize プロパティ

登録する文書の全文検索インデクスを作成するファイルの最大サイズを、0 ~ 2147483647 の範囲で指定します。単位はバイトです。0 を指定した場合、ファイルのサイズをチェックしません。指定を省略した場合は「0」が仮定されます。

BinaryAccessSize プロパティ

コンテンツ、および全文検索テキストを操作する場合に一度に要求するバイト数を、1 ~ 2097151 の範囲で指定します。単位はキロバイトです。実データのサイズが指定したサイズよりも小さい場合、実データのサイズでアクセスします。指定を省略した場合は「1024」が仮定されます。

ReferenceStorageMode プロパティ

リファレンスファイル文書の格納方式を設定します。

「Divide」を指定すると、DocumentBroker がコンテンツを管理するためのディレクトリ階層が少なくなります。

リファレンスファイル文書のコンテンツ数が、「ノードあたりのディレクトリの最大数 /2」を超える可能性がある場合は、「Divide」を指定してください。それ以外の場合は、「Origin」を指定してください。

運用開始後にこのプロパティの指定値を変更してもシステムは問題なく動作しますが、異なる格納形式の文書が混在するようになります。そのため、リファレンスファイル文書のコンテンツ数を見積もり、運用を開始する前に適切な値を設定してください。運用開始後にこのプロパティの指定値を変更しないことをお勧めします。

- Origin

リファレンスファイル文書のコンテンツを管理するためのディレクトリが、コンテンツ単位に異なります。デフォルトの値です。

- Divide

リファレンスファイル文書のコンテンツを管理するためのディレクトリが、複数のコンテンツで共有されます。

指定を省略した場合、「Origin」が仮定されます。

ReferenceDirDivideNum プロパティ

ReferenceStorageMode プロパティで「Divide」を指定した場合に有効となります。リファレンスファイル文書のコンテンツを格納するための、ディレクトリの数を指定します。1 ~ 10,000 の範囲で指定してください。デフォルトの値は 100 です。指定を省略した場合、「100」が仮定されます。

運用開始後にこのプロパティの指定値を変更してもシステムは問題なく動作しますが、DocumentBroker がコンテンツを管理するためのディレクトリ内のコンテンツ数に偏りが発生するため、運用開始後は指定値を変更しないことを推奨します。

ReferenceBasePath プロパティ

リファレンスファイル管理機能で、コンテンツの格納先となるベースパスのデフォルトを指定します。コンテンツの格納先となるベースパスは DbjSession.setReferencePath() でセッションごとに変更できます。相対パスや存在しないパスを指定した場合、リファレンスファイル文書の操作時に例外が発生します。指定を省略した場合、「null」が仮定されます。

TempDirectory プロパティ

文書のコンテンツをストリーム形式でダウンロードする時に生成される一時ファイルの出力先ディレクトリを指定します。複数の実行環境を構築する場合、実行環境ごとに異なるディレクトリを指定してください。また、このディレクトリ下にはユーザデータを格納しないでください。

指定するディレクトリ下のディスク容量は、次の計算式で算出して必要量を割り当ててください。

最大文書サイズ × 同時ダウンロード実行数

指定を省略した場合、「%DOCB_CONF_DIR%\tmp (Windows の場合)」または「\$DOCB_CONF_DIR/tmp (UNIX の場合)」が仮定されます。

4.3.3 文書空間構成定義ファイルの記述例

文書空間構成定義ファイルの記述例を、次に示します。

(1) 前提条件

定義例は次の前提条件に従っていることとします。

LDAP 対応のディレクトリサービスを使用してユーザ管理をしている。

LDAP 対応のディレクトリサービスで構築されている DIT は次の図に示す。

各エントリに対応するディレクトリ情報は次の表に示す。

図 4-1 DIT の例

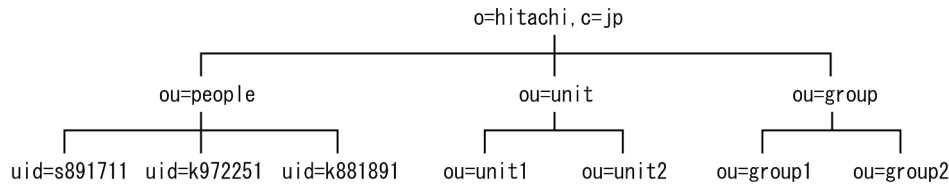


表 4-8 エントリに対応するディレクトリ情報の例

エントリ	各エントリに対応するディレクトリ情報
o: hitachi,c=jp	dn: o=hitachi,c=jp objectclass: top objectclass: organization o: hitachi,c=jp
ou=people	dn: ou=people,o=hitachi,c=jp objectclass: top objectclass: person objectclass: organizationalperson objectclass: inetOrgPerson sn: person cn: personal
uid: s891711	dn: uid=s891711,ou=people,o=hitachi,c=jp objectclass: top objectclass: person objectclass: organizationalperson objectclass: inetOrgPerson cn: Taro Hitachi cn:lang-ja: 日立太郎 uid: s891711 sn: hitachi ou: unit1 ou: unit userpassword: s891711

エン트리	各エン 트리に対応するディレクトリ情報
uid: k972251	dn: uid=k972251,ou=people,o=hitachi,c=jp objectclass: top objectclass: person objectclass: organizationalperson objectclass: inetOrgPerson cn: Ichiro Yamada cn:lang-ja: 山田 一郎 uid: k972251 sn: yamada ou: unit2 ou: unit userpassword: k972251
uid:k881891	dn: uid=k881891,ou=people,o=hitachi,c=jp objectclass: top objectclass: person objectclass: organizationalperson objectclass: inetOrgPerson cn: Jiro Suzuki cn:lang-ja: 鈴木 二郎 uid: k881891 sn: suzuki ou: unit2 ou: unit userpassword: k881891
ou: unit	dn: ou=unit,o=hitachi,c=jp objectclass: top objectclass: organizationalUnit ou: unit ou:lang-ja: ユニット
ou: unit1	dn: ou=unit1,ou=unit,o=hitachi,c=jp objectclass: top objectclass: organizationalUnit ou: unit1 ou:lang-ja: ユニット 1
ou: unit2	dn: ou=unit2,ou=unit,o=hitachi,c=jp objectclass: top objectclass: organizationalUnit ou: unit2 ou:lang-ja: ユニット 2
ou: group	dn: ou=group,o=hitachi,c=jp objectclass: top objectclass: groupOfUniqueNames objectclass: hdsgroupofuniquenames ou: group ou:lang-ja: グループ cn:grp0
ou: group1	dn: ou=group1,ou=group,o=hitachi,c=jp objectclass: top objectclass: groupOfUniqueNames objectclass: hdsgroupofuniquenames cn: grp1 ou: group1 ou:lang-ja: グループ 1 uniqueMember: uid=s891711,ou=people,o=hitachi,c=jp

4. 環境設定に必要なファイル

エントリ	各エントリに対応するディレクトリ情報
ou: group2	dn: ou=group2,ou=group,o=hitachi,c=jp objectclass: top objectclass: groupOfUniqueNames objectclass: hdsgroupofuniquenames cn: grp2 ou: group2 ou:lang-ja: グループ 2 uniqueMember:uid=k972251,ou=people,o=hitachi,c=jp uniqueMember:uid=k881891,ou=people,o=hitachi,c=jp

(2) 定義例

前提条件に基づいた文書空間構成定義ファイルの定義例を次に示します。

```
DocSpaceId = 673d2be0-d1fd-11d0-ab59-08002be29e1d
SecurityAdmin = admin01
DefaultACFlagOwner = FULL_CONTROL
DefaultACFlagGroup = READ_WRITE | LINK | VERSION
DefaultACFlagEveryone = NONE

UserAuthentication = LDAP
LdapHost = 123.45.67.78
LdapPort = 389
LdapUserRoot = o=hitachi,c=jp
LdapGroup = Yes
LdapGroupFromUserAttr = Yes
LdapUserScope = Subtree
LdapUserTimeout = 60
LdapGroupTimeout = 60
LdapUserClass = inetOrgPerson
LdapUserId = uid
LdapUserFilterLeft = (
LdapUserFilterRight = )
LdapUserCase = Default
LdapGroupFromUserDn = Yes
LdapGroupClass = hdsgroupOfUniqueNames
LdapGroupId = cn
LdapGroupRoot = o=hitachi,c=jp
LdapGroupFilterLeft = (
LdapGroupFilterRight = )
LdapGroupIdFromUserAttr = ou
LdapGroupIsDnFromUserAttr = No
LdapGroupIdAttrFromUserAttr = ou
LdapGroupIdFromUserDn = ou
LdapGroupCase = Default
LdapBindUserDN = uid=Tanaka,ou=People,o=hitachi,c=jp
LdapBindPassword = password
```

(3) 認証, ユーザ情報の取得例

ここでは, 例で示した文書空間構成定義ファイルを使用した場合のユーザ認証とユーザ情報の取得について説明します。なお, 説明ではログイン名は「s891711」であると仮定します。

認証

UserAuthentication プロパティが LDAP であるため, DocumentBroker は次のような検索条件に適合するユーザのエントリを検索します。

検索基点

LdapUserRoot プロパティで指定されている「o=hitachi,c=jp」です。

検索範囲

LdapUserScope プロパティに「Subtree」が指定されているので検索基点からのすべてのサブツリーが検索範囲となります。

検索フィルタによる検索条件

LdapUserFilterLeft プロパティ, LdapUserClass プロパティ, LdapUserId プロパティおよび

LdapUserFilterRight プロパティの指定によって、「(&(objectclass=inetOrgPerson)(uid=s891711))」となります。

この検索によって見つかったエントリの userpassword 属性の値と入力されたパスワードを比較して、一致すれば認証成功になります。

ユーザ情報取得

LdapGroup プロパティが Yes であるため、グループのエントリを検索します。この場合、次のような検索条件を用いて検索します。

検索基点

LdapGroupRoot プロパティで指定されている「o=hitachi,c=jp」です。

検索範囲

LdapGroupScope プロパティに「Subtree」が指定されているので検索基点からのすべてのサブツリーが検索範囲となります。

検索条件

LdapGroupFilterLeft プロパティ、LdapGroupClass プロパティおよび LdapGroupFilterRight プロパティの指定によって、
「(&(objectclass=hdsgroupOfUniqueNames)(uniqueMember=uid=s891711,ou=people,o=hitachi,c=jp))」となります。

この検索によって見つかったエントリの cn 属性 (LdapGroupId プロパティで指定されている属性) の値「grp1」をグループ識別子として取得します。

LdapGroupFromUserAttr プロパティが Yes であるため、ユーザの属性値からグループ情報を取得します。この場合、ユーザの ou 属性 (LdapGroupIdFromUserAttr プロパティで指定されている属性) の値は「unit1」および「unit」ですが、LdapGroupIsDnFromUserAttr プロパティが No であるため、その値をそのままグループ識別子として取得します。

LdapGroupFromUserDn プロパティが Yes であるため、ユーザの dn からグループ情報を取得します。この場合、ユーザの dn:uid=s891711,ou=people,o=hitachi,c=jp の ou (LdapGroupIdFromUserDn プロパティで指定した DN のキー) の値「people」をグループ識別子として取得します。

このような検索結果から「grp1」、「unit1」、「unit」および「people」をこのユーザのグループ識別子として取得できます。

4.3.4 文書空間構成定義ファイルの注意事項

文書空間構成定義ファイルで文書空間を定義する場合の注意事項について説明します。

(1) 検索フィルタへ検索条件を追加する場合の注意事項

アクセス制御されている文書に接続する場合、DocumentBroker は、ユーザの属性からのユーザ情報を取得して、アクセスしたい文書にアクセスできるかどうか判断します。ユーザ情報を取得するとき、検索フィルタに検索条件を指定してユーザを検索します。この検索フィルタには、デフォルトの検索フィルタとデフォルトの検索フィルタに検索条件を追加した検索フィルタがあります。

文書空間構成定義ファイルでデフォルトの検索フィルタに検索条件を追加するためには、LdapUserFilterLeft プロパティおよび LdapUserFilterRight プロパティを指定しておく必要があります。

LdapUserFilterLeft プロパティおよび LdapUserFilterRight プロパティの概要

LdapUserFilterLeft プロパティおよび LdapUserFilterRight プロパティは、ユーザ識別子を検索す

4. 環境設定で必要なファイル

るときにデフォルトの検索フィルタに対して、検索条件を追加して指定します。したがって、デフォルトフィルタに情報を付与して、特定の情報を参照できるユーザを検索できます。

ユーザ識別子を検索する場合に DocumentBroker が指定するフィルタ

DocumentBroker はユーザ識別子を検索する場合、次に示すデフォルトフィルタを指定します。

`&(objectclass=$LdapUserClass$)($LdapUserId$= ログインユーザ名)`

ただし、`x` は、文書空間構成定義ファイルでプロパティ `x` に指定された値です。

このデフォルトフィルタの意味は、`(objectclass=$LdapUserClass$)` および `($LdapUserId$= ログインユーザ名)` が真となるユーザ識別子を検索条件として指定するということです。

このデフォルトフィルタに対して `LdapUserFilterLeft` プロパティおよび `LdapUserFilterRight` プロパティにユーザ識別子の検索情報を指定して、検索フィルタを設定します。

デフォルトフィルタへの検索条件の追加例

次の例を基に検索フィルタの設定方法を説明します。

(例)

ユーザが使用するアプリケーションプログラムを示す「`ApplicationType=DocumentBroker`」という属性および属性値を、デフォルトフィルタに追加する。

「`ApplicationType` という属性の属性値が `DocumentBroker` であるユーザ」をデフォルトフィルタに追加条件として付与する場合、`LdapUserFilterLeft` プロパティおよび `LdapUserFilterRight` プロパティには次のように指定します。

- `LdapUserFilterLeft=(`

- `LdapUserFilterRight=(ApplicationType=DocumentBroker))`

ただし、指定する際、「`(`」(左括弧)と「`)`」(右括弧)の個数をそろえる必要があります。これによって作成される検索フィルタを次に示します。

`&(objectclass=$LdapUserClass$)($LdapUserId$= ログインユーザ名)`

`(ApplicationType=DocumentBroker))`

この検索フィルタは、`(objectclass=$LdapUserClass$)`、`($LdapUserId$= ログインユーザ名)`、および `(ApplicationType=DocumentBroker)` が真となるユーザ識別子を検索することを意味します。

なお、グループ識別子の検索フィルタに検索情報を追加指定するためのプロパティとして、`LdapGroupFilterLeft` プロパティおよび `LdapGroupFilterRight` プロパティがあります。これらのプロパティの指定方法は、`LdapUserFilterLeft` プロパティおよび `LdapUserFilterRight` プロパティの指定方法を参考にしてください。

(2) 文書空間構成定義ファイルの内容を変更する場合の注意事項

業務アプリケーションの運用を開始してから、文書空間の構成を変更するような場合が考えられます。このような場合は、次の手順で文書空間の構成を変更してください。

1. 業務アプリケーションを終了する。
2. 文書空間構成定義ファイルを変更する。
3. 業務アプリケーションを再起動する。

なお、業務アプリケーションの起動と終了については、「5. DocumentBroker の起動と終了」を参照してください。

(3) アクセス制御機能の注意事項

アクセス制御機能を利用する環境では、UserAuthentication プロパティの値として、LDAP、または UOC を必ず指定してください。これらの値以外が指定されている場合や指定が省略されている場合は、業務アプリケーションが起動しません。

4.4 ユーザ権限定義ファイル (userperm.properties)

この節では、ユーザ権限定義ファイルについて説明します。

DocumentBroker の運用形態に合わせて、コピーしたサンプルファイルを編集してください。

4.4.1 ユーザ権限定義ファイルの概要

ユーザ権限定義ファイルはユーザ権限を定義するためのファイルです。ユーザ権限とは、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。

システム管理者は、このユーザ権限定義ファイルを更新することでユーザ権限を変更できます。

4.4.2 ユーザ権限定義ファイルの記述形式

ユーザ権限定義ファイルの記述形式を次に示します。なお、ユーザ権限定義ファイルの記述規則については、「4.1.1(1) プロパティファイルの記述規則」を参照してください。

Subject=SubjectType: パーミッション文字列

Subject

権限を与える対象となるユーザ、グループ、またはシステムサブジェクトを指定します。

SubjectType

アクセス権を与えるサブジェクトが、ユーザなのかグループなのかまたはシステムサブジェクトなのかを識別するための情報です。次に示す識別子のどれかを指定します。

- U

Subject に指定した権限を与える対象が、ユーザであることを示します。

- G

Subject に指定した権限を与える対象が、グループであることを示します。

- S

Subject に指定した権限を与える対象が、システムサブジェクトであることを示します。なお、この値を指定する場合、Subject には「everyone」だけ指定できます。

パーミッション文字列

Subject に設定するパーミッションに対応する文字列を指定します。オブジェクト作成権限を表すパーミッション文字列「CREATE」および基本パーミッションを表すパーミッション文字列を指定します。基本パーミッションを表すパーミッション文字列については、「4.3.2 文書空間構成定義ファイルの記述形式」の表 4-6 基本パーミッションと対応する文字列を参照してください。パーミッション文字列は、「|」（ストローク）で結合して複数指定できます。

なお、プロパティの指定で、次に示す場合については、複数のプロパティに指定されたパーミッション文字列の論理和のユーザ権限が与えられます。

- 同一ユーザや同一グループを複数指定して、それぞれにパーミッション文字列を指定した場合
- すべてのユーザ（everyone）と特定のユーザや、特定のユーザとそのユーザが所属するグループなど、同一ユーザが該当するサブジェクトを複数指定して、それぞれにパーミッション文字列を指定した場合

ユーザ権限は、ユーザ権限定義ファイルに、ユーザおよびグループ単位でパーミッションを指定します。なお、ユーザ権限定義ファイルにプロパティが指定されなかった場合、すべてのユーザにユーザ権限が与えられません。

4.4.3 ユーザ権限定義ファイルの記述例

ユーザ権限定義ファイルの記述例を例題に沿って説明します。

(1) ユーザ権限定義ファイルで使用する例題

例題として使用するユーザ権限定義ファイルの定義内容を次の表に示します。

表 4-9 ユーザ権限定義ファイルの定義内容

指定項目	指定する値	説明
DocumentBroker を利用 するすべてのユーザ (everyone) の権限	CREATE PRIM_READ _PROPS	DocumentBroker を利用するすべてのユーザ (everyone) に対し て、次の操作を許可します。 オブジェクトの作成 すべてのオブジェクトのプロパティの参照
ユーザ「User1」の権限	PRIM_READ_CONTEN TS	「User1」というユーザに対して、次の操作を許可します。 文書のプロパティの参照 文書の内容の参照
グループ「Group1」の権 限	PRIM_DELETE	「Group1」というグループに所属するユーザに対して、すべての オブジェクトの削除を許可します。

(2) ユーザ権限定義ファイルの記述例

ユーザ権限定義ファイルの記述例を次に示します。

```
everyone=S:CREATE|PRIM_READ_PROPS
User1=U:PRIM_READ_CONTENTS
Group1=G:PRIM_DELETE
```

4.5 レンディション定義ファイル (mime.properties)

この節では、レンディション定義ファイルについて説明します。

レンディション定義ファイルは、文書のコンテンツを登録するときに、登録するファイルの拡張子からレンディションタイプ (MIME 形式) を自動的に設定するために使用するファイルです。

(1) レンディション定義ファイルの記述形式

プロパティ名には、ファイルの拡張子を指定します。値には、レンディションタイプを指定します。レンディションタイプは、MIME 形式で指定します。ただし、先頭の「MIME::」は省略して指定してください。

形式は、次のとおりです。

拡張子 = レンディションタイプ

複数の拡張子に同じレンディションタイプを割り当てる場合も、拡張子ごとに 1 列ずつ、同じレンディションタイプを指定してください。

記述例を示します。

```
txt = text/plain
htm = text/html
html = text/html
```

(2) ファイル内容の有効期間

レンディション定義プロパティファイルの内容は、DocumentBroker を初期化するときに読み込まれます。一度読み込まれたあとは、プロセスが終了するまで再読み込みはされません。

(3) 注意事項

複数の実行環境でシステムを構築する場合、すべての実行環境でレンディション定義プロパティファイルの内容を一致させておく必要があります。レンディション定義プロパティファイルの内容が不一致の場合、DocumentBroker に登録されるファイルの形式に関する情報が、使用する実行環境ごとに異なります。

4.6 メタ情報初期化ファイル (docbmetainit.properties)

この節では、メタ情報初期化ファイルについて説明します。

メタ情報初期化ファイルは、DocumentBroker のシステム設定を記述するファイルです。メタ情報の登録コマンド (dbjinitmeta) の実行時に入力ファイルとして指定することで、システム情報が登録されます。

4.6.1 メタ情報初期化ファイルの記述形式

ここでは、メタ情報初期化ファイルの記述形式について説明します。なお、メタ情報初期化の記述規則については、「4.1.1(1) プロパティファイルの記述規則」を参照してください。

メタ情報初期化ファイルは、Java 2 Platform, Standard Edition のプロパティファイル形式です。

<プロパティ名>=<値>

以降、メタ情報初期化ファイルに記述するプロパティについて説明します。

DocSpaceCharacterSet プロパティ

文書空間で使用する文字コード種別を指定します。

- SJIS
文書空間で使用する文字コード種別に「Shift-JIS」を使用します。
- UTF-8
文書空間で使用する文字コード種別に「UTF-8」を使用します。

指定を省略した場合、「UTF-8」が仮定されます。なお、不正な値を指定した場合、エラーとなります。

4.7 データベース情報ファイル (databaseinfo.properties)

この節では、データベース情報ファイルについて説明します。

データベース情報ファイルは、コマンドが接続するデータベースの情報を記述するファイルです。次のコマンドの実行時に、データベースへの接続に使用します。

- メタ情報の初期設定コマンド (dbjinitmeta)
- メタ情報の追加コマンド (dbjaddmeta)
- DocumentBroker 用データベース定義文の作成コマンド (dbjcrtsql)
- メタ情報の削除コマンド (dbjdelmeta)
- DocumentBroker 実行環境の情報の登録コマンド (dbjregenvid)
- メタ情報の内容変更コマンド (dbjchgmeta)
- メタ情報の CSV 出力コマンド (dbjprintmeta)

4.7.1 データベース情報ファイルの記述形式

ここでは、データベース情報ファイルの記述形式について説明します。なお、メタ情報初期化の記述規則については、「4.1.1(1) プロパティファイルの記述規則」を参照してください。

データベース情報ファイルは、Java 2 Platform, Standard Edition のプロパティファイル形式です。

< プロパティ名 >=< 値 >

以降、データベース情報ファイルに記述するプロパティについて説明します。

DBType プロパティ

使用するデータベースの種類として、「HiRDB」を指定します。このプロパティの指定は省略できません。

DBHost プロパティ

接続先データベースのホスト名を指定します。

DBPort プロパティ

接続先データベースのポート番号を指定します。

DBUser プロパティ

データベースに接続するユーザ名を指定します。

DBPass プロパティ

データベースに接続するユーザのパスワードを指定します。

4.8 メタ情報ファイル

この節では、メタ情報ファイルについて説明します。

DocumentBroker で使用するクラスの定義情報をメタ情報といいます。このメタ情報は、メタ情報ファイルという初期設定用のファイルに記述されています。メタ情報ファイルは、DocumentBroker をインストールしたときに以下に格納されます。

UNIX の場合

- /opt/DocBroker/Platform/meta 以下

Windows の場合

- <インストールディレクトリ>\DocBroker\Platform\meta 以下
または
- <インストールディレクトリ>\DocBroker\Developer\meta 以下

メタ情報ファイルは、直接編集しないでください。メタ情報を追加する場合は、定義情報ファイルに必要なオブジェクトの定義を追加します。定義情報ファイルについては、「4.9 定義情報ファイル」を参照してください。

メタ情報ファイルの一覧を次の表に示します。

表 4-10 メタ情報ファイルの一覧

ファイル名	内容
edmclass.csv	DocumentBroker が提供するシステムクラスを定義したファイル
edmclassprop.csv	DocumentBroker が提供するシステムクラスが持つプロパティを定義したファイル
edmprop.csv	DocumentBroker が提供するシステムプロパティを定義したファイル
edmsys.csv	システム情報のデフォルト値を定義したファイル

4.9 定義情報ファイル

この節では、定義情報ファイルについて説明します。

定義情報ファイルは、サンプルファイルが提供されています。任意のディレクトリ下に、すべての定義情報ファイルを格納して使用してください。

4.9.1 定義情報ファイルの概要

定義情報ファイルは、ユーザの環境に合わせてサブクラスやプロパティを追加する場合に作成する、オブジェクトを定義するファイルです。このファイルに定義した情報は、メタ情報として追加されます。

作成する定義情報ファイルを次の表に示します。

表 4-11 定義情報ファイルの一覧

ファイル名	内容
addclass.csv	ユーザ定義のサブクラスを追加する場合に作成します。
addprop.csv	ユーザ定義のプロパティを追加する場合に作成します。ユーザ定義のプロパティを追加した場合、addclassprop.csv でプロパティをクラスに追加する必要があります。
addclassprop.csv	プロパティをクラスに追加する場合に作成します。

4.9.2 サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則

定義情報ファイルに記述するサブクラス名やプロパティ名の規則を説明します。ここでは、表識別子（表の名称として一意に識別できる値）、列名を定義する場合の規則について説明します。

なお、システムクラスおよびシステムプロパティのデータベース定義の名称一覧は、「付録 A システムクラス、システムプロパティおよびデータベース定義の名称規則」を参照してください。

(1) サブクラス名の規則

- 英小文字、英大文字、数字、空白、「_」（下線文字）、および「-」（マイナス記号）で指定します。
- サブクラス名には、「dmaClass」および「edmClass」で始まる文字列は使用できません。
- サブクラスの名称は重複しないようにしてください。
- 先頭 1 バイトは、英小文字、または英大文字を指定します。
- 末尾に、空白は設定できません。
- 1 ~ 28 バイトで指定します。

(2) プロパティ名の規則

- 英小文字、英大文字、数字、空白、「_」（下線文字）、および「-」（マイナス記号）で指定します。
- 「dmaProp」、「edmProp」および「dbrProp」で始まる文字列は使用できません。
- 先頭 1 バイトは、英小文字か英大文字で指定します。
- 末尾に、空白は設定できません。
- VariableArray 型のプロパティ名は、1 ~ 14 バイトで指定します。
VariableArray 型のプロパティは、データ型が VariableArray であるプロパティである必要があります。
- VariableArray 型の要素のプロパティ名は、1 ~ 15 バイトで指定します。
VariableArray 型の要素のプロパティとは、edmClass_Struct クラスのサブクラスに追加したプロパティです。

- 重複しないように付ける必要があります。
列名は、プリフィクスが「VariableArray 型プロパティの列名 + '_'」と重複しないように付ける必要があります。例えば列名「varray_elem」と VariableArray 型のプロパティの列名「varray」がある場合、両方ともプリフィクス「varray_」が重複するため、片方に別の名称を付ける必要があります。
- VariableArray 型のプロパティおよび VariableArray 型の要素のプロパティ以外は 1 ~ 28 バイトで指定します。

4.9.3 定義情報ファイルの記述形式

ここでは、定義情報ファイルの記述形式について説明します。

(1) 記述規則

定義情報ファイルを記述する場合の規則を次に示します。

ファイルの文字コードは UTF-8 とします。

コメント以外は、ASCII の範囲で記述します。

区切り文字は「,」とします。

先頭(1 バイト目)が「#」で始まる行はコメント行として扱われます。(空白のトリミングは行われません。)

空白行は無視されます。

(2) 記述形式

定義情報ファイルの記述形式について説明します。

(a) addclass.csv ファイル

addclass.csv ファイルは次の形式に従って記述します。

クラス ID, クラス名, スーパークラス名

各項目について説明します。

クラス ID

追加するクラス ID を GUID 形式で指定します。GUID は、「X」を 0 ~ 9 および a ~ f (小文字) で表される 16 進数とした「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8-4-4-4-12)」の形式で表されます。GUID 値はシステム内で重複できません。なお、大文字・小文字は区別されません。指定を省略した場合、システムで自動採番されます。

クラス名

追加するクラス名を指定します。

クラス名の規則は、「4.9.2(1) サブクラス名の規則」を参照してください。この項目の指定は省略できません。

スーパークラス名

追加するクラスのスーパークラス名を指定します。なお、この項目の指定は省略できません。

サブクラスを生成できるスーパークラスを次に示します。

- dmaClass_ConfigurationHistory クラス
- dmaClass_Container クラス
- dmaClass_DocVersion クラス

4. 環境設定に必要なファイル

- edmClass_ContainerVersion クラス
- edmClass_IndependentPersistence クラス
- edmClass_Struct クラス
- edmClass_VersionTracedDocVersion クラス
- edmClass_VersionTracedComponentDocVersion クラス

(b) addprop.csv ファイル

addprop.csv ファイルは次の形式に従って記述します。

プロパティ ID, プロパティ名, データ型, プロパティのデフォルト値, 文字列の最大長, VariableArray クラス名, 繰り返し列の最大要素数

各項目について説明します。

プロパティ ID

追加するプロパティ ID を GUID 形式で指定します。GUID は、「X」を 0 ~ 9 および a ~ f (小文字) で表される 16 進数とした「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8-4-4-4-12)」の形式で表されます。GUID 値はシステム内で重複できません。なお、大文字・小文字は区別されません。指定を省略した場合、システムで自動採番されます。

プロパティ名

追加するプロパティ名を指定します。

プロパティ名の規則は、「4.9.2(2) プロパティ名の規則」を参照してください。この項目の指定は省略できません。

データ型

追加するプロパティのデータ型を指定します。なお、この項目の指定は省略できません。

- String
- Integer
- Boolean
- VariableArray
- FreeWordIndex

プロパティのデフォルト値

プロパティにデフォルト値として設定する値を指定します。指定するデータ型によって指定できる値が異なります。

- データ型が「String」の場合
0 ~ 255 バイトの文字列を指定します。NULL 値を指定したい場合は「NULL」を指定します。「文字列の最大長」に指定する最大長より大きい文字列を指定した場合、エラーとなります。指定を省略した場合、0 バイトの文字列が仮定されます。
- データ型が「Integer」の場合
-2147483648 ~ 2147483647 の範囲で指定します。指定を省略した場合、「0」が仮定されます。
- データ型が「Boolean」の場合
「0」または「1」を指定します。指定を省略した場合、「0」が仮定されます。
- データ型が「FreeWordIndex」の場合
0 ~ 255 バイトの文字列を指定します。NULL 値を指定したい場合は「NULL」を指定します。指定を省略した場合、0 バイトの文字列が仮定されます。

上記以外のデータ型の指定値は無視されます。

文字列の最大長

文字列の最大長（バイト）を指定します。データ型に「String」、または「FreeWordIndex」を指定した場合に指定が必要です。

指定できる範囲は、データベースのデータ型に依存します。DocumentBroker では指定値の妥当性はチェックしません。データ型が「String」の場合はデータベースの「MVARCHAR」に対応します。「FreeWordIndex」の場合は「FREEWORD」に対応します。

VariableArray クラス名

edmClass_Struct クラスのサブクラスの名前を指定します。データ型に「VariableArray」を指定した場合に指定が必要です。

繰り返し列の最大要素数

VariableArray 型のプロパティをデータベースの繰り返し列に格納する場合に指定します。データ型に「VariableArray」を指定した場合に指定が必要です。指定できる値の範囲は、2 ~ 30000 です。別表に格納する場合は「-1」を指定します。

(c) addclassprop.csv ファイル

addclassprop.csv ファイルは次の形式に従って記述します。

クラス名, プロパティ名, デフォルト値

各項目について説明します。

クラス名

プロパティを追加するクラス名を指定します。なお、この項目の指定は省略できません。プロパティを追加できるクラスを次に示します。

- dmaClass_DirectContainmentRelationship クラス
- dmaClass_ReferentialContainmentRelationship クラス
- edmClass_PublicACL クラス
- edmClass_Relationship クラス
- 追加したサブクラス

VariableArray 型のプロパティを追加できるクラスを次に示します。

- edmClass_IndependentPersistence クラスのサブクラス
- dmaClass_ConfigurationHistory クラスのサブクラス
- dmaClass_Container クラスのサブクラス
- edmClass_ContainerVersion クラスのサブクラス
- dmaClass_DocVersion クラスのサブクラス
- edmClass_VersionTracedDocVersion クラスのサブクラス
- edmClass_VersionTracedComponentDocVersion クラスのサブクラス

FreeWordIndex 型のプロパティを追加できるクラスを次に示します。

- edmClass_IndependentPersistence クラスのサブクラス
- dmaClass_ConfigurationHistory クラスのサブクラス
- dmaClass_Container クラスのサブクラス
- edmClass_ContainerVersion クラスのサブクラス
- dmaClass_DocVersion クラスのサブクラス
- edmClass_VersionTracedDocVersion クラスのサブクラス
- edmClass_VersionTracedComponentDocVersion クラスのサブクラス

全文検索インデクス用プロパティを追加できるクラスを次に示します。

4. 環境設定に必要なファイル

- dmaClass_DocVersion クラスのサブクラス
- edmClass_VersionTracedDocVersion クラスのサブクラス
- dmaClass_ConfigurationHistory クラスのサブクラス

なお、クラスに全文検索インデクス用プロパティを追加した場合、次のシステムプロパティもそのクラスに追加する必要があります。追加しない場合は、エラーとなります。

- edmProp_DocLength プロパティ
- edmProp_ContentIndexStatus プロパティ

プロパティ名

追加するプロパティ名を指定します。この項目の指定は省略できません。

デフォルト値

クラス固有のプロパティのデフォルト値を指定します。指定を省略した場合、addprop.csv ファイルの内容が仮定されます。また、"" だけが指定された場合は、addprop.csv の内容を参照しないで、空値が指定されます。

4.9.4 定義情報ファイルの記述例

サブクラスおよびプロパティを追加する場合の定義情報ファイルについて、例題に沿って説明します。

(1) 使用する例題

サブクラスおよびプロパティを追加する場合の定義情報ファイルを作成するための例題を次に示します。

(例)

「問題点管理ドキュメント」という文書を管理するためにサブクラスを追加します。追加するサブクラスには、それぞれ文書の管理に必要なプロパティを追加します。管理する文書の詳細情報を次の表に示します。

表 4-12 管理する文書の詳細情報

文書の種類	概要	付加情報	管理方法
問題点管理ドキュメント	製品の開発時の問題点を管理するための文書	<ul style="list-style-type: none">• 問題点分類• 問題点発生日時• 検討結果• 解決日時• 対策日	直接型のリンクを使用して管理する。

この文書を管理するためのサブクラスを次のように設計します。追加するサブクラスとそのスーパークラスを次の表に示します。

表 4-13 追加するサブクラスとそのスーパークラス

管理する文書	追加するサブクラス	スーパークラス
問題点管理ドキュメント	usrClass_ProbInfoDoc	dmaClass_DocVersion

問題点管理ドキュメントは、dmaClass_DocVersion クラスのサブクラスとして usrClass_ProbInfoDoc クラスを追加して管理します。

次に、文書に対して付加する情報を各クラスのプロパティとして追加します。文書の付加情報とクラスに追加するプロパティの対応を、次の表に示します。

表 4-14 文書の付加情報とサブクラスに追加するプロパティの対応

サブクラス名	付加情報	プロパティ名	データ型
usrClass_ProbInfoDoc	問題点分類	usrProp_Category	String 型
	問題点発生日時	usrProp_DateOfProb	Integer32 型
	検討結果	usrProp_Status	Boolean 型
	解決日時	usrProp_DateOfResolve	Integer32 型

なお、追加するサブクラスは、全文検索機能付き文書クラスとして追加するため、表に示したプロパティのほかに、全文検索用のプロパティを追加する必要があります。全文検索機能付き文書クラスに追加するプロパティについては、「4.9.6 定義情報ファイルの記述例（全文検索機能を使用する文書クラスを追加する場合）」を参照してください。

このように、管理する文書の種類や付加情報などから必要なクラスとプロパティを決定して、その設計情報を基に必要なクラスおよびプロパティを追加します。

(2) 記述例

addclass.csv

```
,usrClass_ProbInfoDoc,dmaClass_DocVersion
```

addclassprop.csv

```
usrClass_ProbInfoDoc,usrProp_Category,
usrClass_ProbInfoDoc,usrProp_DateOfProb,
usrClass_ProbInfoDoc,usrProp_Status,
usrClass_ProbInfoDoc,usrProp_DateOfResolve,
```

addprop.csv

```
,usrProp_Category,String,,64,,
,usrProp_DateOfProb,Integer,,,,
,usrProp_Status,Boolean,,,,
,usrProp_DateOfResolve,Integer,,,,
```

4.9.5 定義情報ファイルの記述例（VariableArray 型のプロパティを追加する場合）

プロパティの基本単位が VariableArray 型であるプロパティを追加する場合の定義情報ファイルについて、例題に沿って説明します。

(1) 使用する例題

プロパティは、データ型に従った値を 1 個持つか複数個持つかが決められています。これを基本単位といいます。プロパティの基本単位が VariableArray 型のプロパティとは、そのプロパティが配列の要素を保持するプロパティです。

VariableArray 型のプロパティとして定義するプロパティは、edmClass_Struct クラスのインスタンスをリファレンスとして保持します。VariableArray 型のプロパティを定義するための例題を次に示します。

(例)

管理する文書として「文書リスト」を定義します。この文書は、書籍の著者に関する情報を管理します。この「文書リスト」について、サブクラスを追加して必要な付加情報を追加します。追加するサブクラスとそのスーパークラスを次の表に示します。

表 4-15 追加するサブクラスとそのスーパークラス

管理する情報	追加するサブクラス	スーパークラス
文書リスト	usrClass_DocList	edmClass_IndependentPersistence
著者の情報	usrClass_WriterInfo	edmClass_Struct

「文書リスト」は、書籍の著者に関する情報を管理する文書で、edmClass_IndependentPersistence クラスのサブクラスとして、「usrClass_DocList クラス」を定義します。「著者の情報」は、著者の情報を管理する文書であり、edmClass_Struct クラスのサブクラスとして「usrClass_WriterInfo クラス」を定義します。

「文書リスト」には付加情報としてプロパティを追加します。付加情報に対応するプロパティを次の表に示します。

表 4-16 usrClass_DocList クラスの構造

クラス名	付加情報	プロパティ名 (VariableArray 型のプロパティ)	データ型
usrClass_DocList	文書名	uP_DocName	String 型
	著者の情報	uP_WriterInfo	VariableArray 型

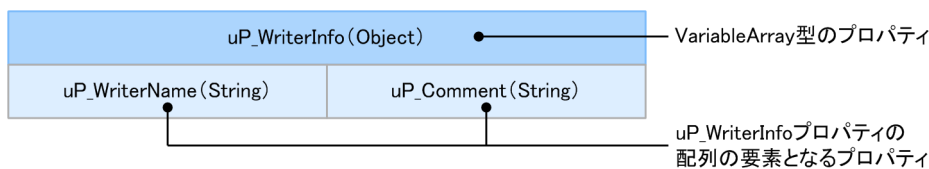
「uP_WriterInfo プロパティ」は、VariableArray 型のプロパティです。表 4-17 に、「uP_WriterInfo プロパティ」に定義するプロパティを示します。「uP_WriterInfo プロパティ」は、配列の要素としてプロパティを管理します。VariableArray 型の「uP_WriterInfo プロパティ」の要素を管理する「usrClass_WriterInfo クラス」に格納されているプロパティを説明します。

表 4-17 VariableArray 型の「uP_WriterInfo プロパティ」の要素となるクラスの構造

クラス名 (VariableArray 型のプロパティの要素)	付加情報	プロパティ名 (VariableArray 型のプロパティ要素の構成)	データ型
usrClass_WriterInfo	著者名	uP_WriterName	String 型
	著者の紹介	uP_Comment	String 型

VariableArray 型の「uP_WriterInfo プロパティ」の構造を次の図に示します。

図 4-2 VariableArray 型の「uP_WriterInfo プロパティ」の構造



配列の要素として管理されるこれらのプロパティは、edmClass_Struct クラスのプロパティとして追加します。

(2) 記述例

addclass.csv


```
,usrClass_DocList,edmClass_IndependentPersistence
,usrClass_WriterInfo,edmClass_Struct
```

addclassprop.csv

```
usrClass_DocList,uP_DocName,
usrClass_DocList,uP_WriterInfo,
usrClass_WriterInfo,uP_WriterName,
usrClass_WriterInfo,uP_Comment,
```

addprop.csv

```
,uP_DocName,String,,64,,
,uP_WriterInfo,VariableArray,,,usrClass_WriterInfo,100
,uP_WriterName,String,,64,,
,uP_Comment,String,,64,,
```

4.9.6 定義情報ファイルの記述例（全文検索機能を使用する文書クラスを追加する場合）

ここでは、全文検索機能を使用する文書クラスを追加するための定義情報ファイルの記述例を次に示します。なお、文書に対する全文検索機能を使用する場合は、全文検索用のプロパティを追加した dmaClass_DocVersion クラスのサブクラスを作成するために必要な情報を、定義情報ファイルに記述します。

（1）記述例

dmaClass_DocVersion クラスのサブクラス usrClass_DocTextSearch クラスを生成する場合の記述例を次に示します。

addclass.csv

```
,usrClass_DocTextSearch,dmaClass_DocVersion
```

addclassprop.csv

```
usrClass_DocTextSearch,edmProp_TextIndex,
usrClass_DocTextSearch,edmProp_DocLength,
usrClass_DocTextSearch,edmProp_ContentIndexStatus,
```

4.9.7 定義情報ファイルの記述例（文字列型プロパティに対する全文検索機能を使用する場合）

文字列型プロパティに対する全文検索機能を使用する場合の定義情報ファイルについて、例題に沿って説明します。

全文検索機能を使用できる文字列型プロパティのことを <INDEXWORD PRONOUNCE=" せんぶんけんさくきのうつきもしれつかたふるはてい " INDEXITEM=" 全文検索機能付き文字列型プロパティ "> 全文検索機能付き文字列型プロパティ <INDEXWORD> といいます。文字列型プロパティに対して全文検索機能を使用する場合、全文検索機能付き文字列型プロパティを追加します。

なお、全文検索機能付き文字列型プロパティの追加方法は、基本的に String 型のプロパティの追加方法と同じです。そのため、ここでは String 型のプロパティの追加方法と異なる個所について説明します。

String 型のプロパティの追加方法については、「4.9.4 定義情報ファイルの記述例」を参照してください。

(1) 使用する例題

全文検索機能付き文字列型プロパティを文書クラスに定義するための例題を次に示します。

(例)

管理する文書として「設計ドキュメント」を定義します。この文書は、製品の開発に関する情報を管理します。

「設計ドキュメント」を管理するためにサブクラスを追加します。追加するサブクラスには、全文検索機能付き文字列型プロパティを追加します。

追加するサブクラスとそのスーパークラスを次に示します。

表 4-18 追加するサブクラスとそのスーパークラス (文字列型プロパティに対する全文検索機能を使用する場合)

管理する文書	追加するサブクラス	スーパークラス
設計ドキュメント	usrClass_PropTextSearch	dmaClass_DocVersion

dmaClass_DocVersion クラスのサブクラスとして usrClass_PropTextSearch クラスを追加して、設計ドキュメントを管理します。

また、文書に付加する情報としてプロパティを追加します。設計ドキュメントの付加情報と、付加情報に対応するプロパティを次の表に示します。

表 4-19 usrClass_PropTextSearch クラスの構造

クラス名	付加情報	プロパティ名	データ型
usrClass_PropTextSearch	文書名	usrProp_DocName	String 型
	執筆者	usrProp_DocWriter	String 型
	文書の概要	usrProp_DocSummary (全文検索機能付き文字列型プロパティ)	FreeWordIndex 型

usrProp_DocSummary プロパティは、全文検索機能付き文字列型プロパティです。

(2) 記述例

全文検索機能付き文字列型プロパティを追加するには、プロパティを追加するために必要な情報を定義情報ファイルに記述します。

全文検索機能付き文字列型プロパティを追加する場合の定義情報ファイルの記述方法について、「(1) 使用する例題」で設定した例題に沿って説明します。

addclass.csv

```
,usrClass_PropTextSearch,dmaClass_DocVersion
```

addclassprop.csv

```
usrClass_PropTextSearch,usrProp_DocName,
usrClass_PropTextSearch,usrProp_DocWriter,
usrClass_PropTextSearch,usrProp_DocSummary,
```

addprop.csv

```
,usrProp_DocName,String,,64,,  
,usrProp_DocWriter,String,,64,,  
,usrProp_DocSummary,FreeWordIndex,,255,,
```

4.10 RD エリア定義情報ファイル

この節では、RD エリア定義情報ファイルについて説明します。

4.10.1 RD エリア定義情報ファイルの概要

RD エリア定義情報ファイルは、データベース定義文中の RD エリア名を、ユーザが指定した RD エリア名で出力させるために作成するファイルです。作成したファイルは、DocumentBroker 用データベース定義文の作成コマンド (dbjrtsql)、およびメタ情報の追加コマンド (dbjaddmeta) の `-r` オプションで指定します。指定したファイルの内容が、`-o` オプションで指定したデータベース定義文格納ファイル中の RD エリア名に反映されます。これによって、DocumentBroker 用データベース定義文の作成コマンド (dbjrtsql)、およびメタ情報の追加コマンド (dbjaddmeta) で出力されるデータベース定義文中の RD エリアの手動による変更が不要になります。

4.10.2 RD エリア定義情報ファイルの記述形式

ここでは、RD エリア定義情報ファイルの記述形式について説明します。

(1) 記述規則

RD エリア定義情報ファイルを記述する場合の規則を次に示します。

ファイルの文字コードは UTF-8 とします。

区切り文字は「,」とします。

先頭 (1 バイト目) が「#」で始まる行はコメント行として扱われます。(空白のトリミングは行われません。)

空白行は無視されます。

(2) 記述形式

RD エリア定義情報ファイルの記述形式を次に示します。

種別 ,RD エリア名 [, クラス名 [, プロパティ名 ,...]]

各項目について説明します。

種別

RD エリアの種別を指定します。指定できる値を次の表に示します。なお、この項目の指定は省略できません。

表 4-20 RD エリアの種別

種別	説明	出力するデータベース定義文に種別に対応する指定がない場合の指定値	クラス名の指定	プロパティ名の指定
TableArea	表として定義されるクラスと RD エリア名を対応付ける。	USERAREA	省略可	指定不要

種別	説明	出力するデータベース定義文に種別に対応する指定がない場合の指定値	クラス名の指定	プロパティ名の指定
IndexArea	データベースのインデクスとして定義されるプロパティと RD エリアを対応付ける。	INDEXAREA	省略可	1 から最大 16 個指定
LobArea	文書実体を管理するクラスと LOB 列格納用 RD エリア名を対応付ける。	BLOBAREA	省略不可 dmaClass_ContentTransfer クラスを指定する。	指定不要
SGMLTEXTLobArea	全文機能付き文書クラスと全文検索用の文書内容格納用 RD エリア名を対応付ける。	BLOBAREA	省略不可	指定不要
NgramIndexArea	全文検索機能付き文書クラスに定義される全文検索インデクスと全文検索インデクス格納用 RD エリア名を対応付ける。	BLOBAREA	省略不可	指定不要
FreeWordIndexArea	全文検索機能付き文字列型プロパティに定義される全文検索インデクスと全文検索インデクス格納用 RD エリア名を対応付ける。	BLOBAREA	省略不可	1 個指定する

RD エリア名

格納先の RD エリア名を指定します。この項目の指定は省略できません。

なお、種別が LobArea、SGMLTEXTLobArea、NgramIndexArea、および FreeWordIndexArea の場合に指定する RD エリア名は、ファイル内で一意になっている必要があります。

クラス名

クラス名（定義情報ファイルに設定した値）を指定します。クラスごとに RD エリアを設定する場合に指定する必要があります。指定を省略すると、種別は 1 つの RD エリアに対応付けられます。

プロパティ名

インデクスのキーに選択するプロパティ名（定義情報ファイルに設定した値）を指定します。インデクスごとに RD エリアを設定する場合に指定する必要があります。

4.10.3 RD エリア定義情報ファイルの記述例

RD エリア定義情報ファイルの記述例を次に示します。

```
# 下記の場合、ClassA は TBLArea1、ClassB は TBLArea2、それ以外のクラスは TBLArea3 に格納される。
TableArea, TBLArea1, ClassA
TableArea, TBLArea2, ClassB
TableArea, TBLArea3
```

4. 環境設定に必要なファイル

```
# 下記の場合、ClassA の PropA をキーとするインデクスは IDXArea1, ClassA のそれ以外インデクスは IDXArea2, それ以外のクラスのインデクスは IDXArea3 に格納される。
```

```
IndexArea, IDXArea1, ClassA, PropA  
IndexArea, IDXArea2, ClassA  
IndexArea, IDXArea3
```

```
# 以下の LOB 格納用 RD エリアは格納先 RD エリアが重複しないように指定する。
```

```
LobArea, LOBArea, dmaClass_ContentTransfer  
SGMLTEXTLobArea, SGMLTEXTArea  
NgramIndexArea, NGRMArea1, ClassA  
NgramIndexArea, NGRMArea2, ClassB  
FreeWordIndexArea, FREEArea, ClassC, PropB
```

4.10.4 RD エリア定義情報ファイルの注意事項

RD エリア定義情報ファイルの注意事項について説明します。

指定された RD エリア名がデータベースに定義されているかどうかについてはチェックしません。

指定されたクラス名、プロパティ名の妥当性についてはチェックしません。

種別が LobArea, SGMLTEXTLobArea, NgramIndexArea, および FreeWordIndexArea の場合に指定する RD エリア名は、ユニークな名称を指定してください。重複している場合、該当するエリア名には BLOBAREA という文字列を出力します。

ファイルの内容に誤りがあった場合、警告メッセージを表示してその行の解析を中止し、処理を続行します。

RD エリア定義情報ファイルには、空白を含むクラス名、プロパティ名、および RD エリア名は指定できません。空白を含むクラス名、プロパティ名、および RD エリア名を指定した場合、データベース定義文出力時に警告メッセージが出力されるか、またはデータベース定義ユティリティ実行時にデータベースのエラーでエラーとなります。

4.11 インデクス情報ファイル

この節では、インデクス情報ファイルについて説明します。

インデクス情報ファイルは、ユーザが追加するプロパティにインデクスを定義し、インデクスの作成をデータベース定義に追加する場合に作成してください。

4.11.1 インデクス情報ファイルの概要

サブクラスに追加するプロパティをキーにして検索したい場合、プロパティにインデクスを定義します。インデクス情報ファイルは、ユーザが追加するプロパティにインデクスを定義する場合に、インデクスの情報を記述するファイルです。

作成したインデクス情報ファイルをメタ情報の追加コマンド (dbjaddmeta)、および DocumentBroker 用データベース定義文の作成コマンド (dbjertsq1) 実行時に入力ファイルとして指定することで、データベース定義文にインデクスの作成を定義できます。インデクス情報ファイルでは、ユーザが定義するプロパティに対するインデクスの作成を定義できます。

4.11.2 インデクス名をデータベース定義の名称に使用する場合の規則

定義情報ファイルに記述したサブクラス名やプロパティ名を、データベース定義の名称定義として使用する場合の規則を説明します。ここでは、インデクス名 (表を検索するためのキーとして列に付けた索引の名称) を定義する場合の規則について説明します。

インデクス名を定義する場合、英小文字、英大文字、数字、空白、「_」(下線文字)、および「-」(マイナス記号) で指定します。インデクス名の指定では、DocumentBroker で付けるインデクス名を含めてデータベースのスキーマで一意に付ける必要があります。また、インデクス名を指定する場合は、次に示す規則およびデータベース定義の制約に従ってください。

- ユーザがインデクスを追加する場合、インデクス情報ファイルで指定したインデクス名が、データベース定義文のインデクス名になります。
- インデクス名には、「dmaClass」および「edmClass」で始まる名称を指定しないでください。
- インデクス名の先頭 1 バイトは、英小文字または英大文字で指定します。
- インデクス名の末尾は、空白にできません。
- インデクス名は、1 ~ 30 バイトで指定します。

4.11.3 インデクス情報ファイルの記述形式

ここでは、インデクス情報ファイルの記述形式について説明します。

(1) 記述規則

インデクス情報ファイルを記述する場合の規則を次に示します。

ファイルの文字コードは UTF-8 とします。

区切り文字は「,」とします。

先頭 (1 バイト目) が「#」で始まる行はコメント行として扱われます。(空白のトリミングは行われません。)

空白行は無視されます。

(2) 記述形式

インデクス情報ファイルの記述形式を次に示します。

[インデクス名], クラス名, [UNIQUE], [EXCEPT], プロパティ名, [{ASC | DESC}] [, プロパティ名, [{ASC | DESC}]]...

各項目について説明します。

インデクス名

インデクス名を指定します。インデクス名の規則は、「4.11.2 インデクス名をデータベース定義の名称に使用する場合の規則」を参照してください。

指定が省略された場合、インデクス名はシステムインデクス名に続けて「クラス名 + VV (32進数)」の形で自動採番されます。VV を超える場合はエラーとなります。インデクス名を指定し、クラス名に VariableArray クラス名を指定した場合、VariableArray プロパティを複数の箇所で使用すると同名のインデクス名が複数出力されます。

クラス名

インデクスを定義するクラス名を指定します。この項目は省略できません。

UNIQUE

インデクスのキーに選択するプロパティを一意に制約する場合に指定します。指定を省略した場合、インデクスのキーに選択するプロパティを一意に制約しません。

EXCEPT

NULL 値だけで構成されるキーを除外してインデクスを定義する場合に指定します。指定を省略した場合、NULL 値だけで構成されるキー値を除外しません。

プロパティ名

インデクスのキーに選択するプロパティ名を指定します。最大 16 個まで指定できます。複数指定すると、マルチカラムインデクス定義となります。システム定義のプロパティ（「dma」および「edm」で始まるプロパティ）は指定できません。なお、この項目は省略できません。

ASC|DESC

インデクスを昇順に定義する場合は ASC、降順に定義する場合は DESC を指定します。直前のカラムにはプロパティ名を指定する必要があります。なお、指定を省略した場合、キーに選択したプロパティに ORDER BY 句が設定されません。

4.11.4 インデクス情報ファイルの記述例

インデクス情報ファイルの記述例を次に示します。

```
INDEXA,usrClassCH,UNIQUE,EXCEPT,usrProp_Name,ASC  
INDEXA,usrClassCH,,usrProp_Author,,usrProp_Count
```


4.12 変更情報ファイル

この節では、変更情報ファイルについて説明します。

4.12.1 変更情報ファイルの概要

変更情報ファイルは、オブジェクトに設定されている、ユーザ識別子、グループ識別子、ユーザプロパティを変更する場合に作成してください。dbjchgusrid コマンドの実行時に入力ファイルとして指定することで、オブジェクトに設定されているユーザ識別子、グループ識別子、ユーザプロパティの内容を一括で変更することができます。変更情報ファイルには、次の種類があります。変更内容に合わせて必要なファイルを作成してください。なお、コマンドを実行すると、変更情報ファイルで空文字となっている各行の1フィールド目に実行結果が出力されます。コマンドの実行結果でエラーが発生した場合、出力された変更情報ファイルの内容を修正し再実行すると、エラーになったオブジェクトだけ、処理されます。

- ユーザ識別子変更情報ファイル
- グループ識別子変更情報ファイル

ユーザプロパティを変更する場合は、変更するユーザプロパティを記述した下記のファイルが追加が必要です。

- プロパティ変更情報ファイル

それぞれのファイルの記述形式については、「4.12.3 ユーザ識別子変更情報ファイル」「4.12.4 グループ識別子変更情報ファイル」「4.12.5 プロパティ変更情報ファイル」を参照してください。

4.12.2 変更情報ファイルの記述規則

変更情報ファイルを記述する場合の規則を次に示します。

- 行の終わりは、<EOF>または行末文字とします。
- 先頭(1バイト目)が「#」および「;」で始まる行はコメント行として扱われます。(空白のトリミングは行われません。)
- 値に「,」が含まれている場合、「"」(引用符)で囲んでください。
- ファイルの文字コードは、文書空間で使用する文字コード (UTF-8 または Shift-JIS), または印刷可能な ASCII コードとします。文書空間で使用する文字コードでファイルの内容を読み込むため、文書空間で使用する文字コード種別に応じた文字コードで記述してください。
- 次の場合は、エラーとなります。
 - 変更前の識別子と変更後の識別子に同じ値を指定している場合

```
A,A
```

- 変更前の識別子を重複して指定している場合

```
A,B
A,C
```

- 変更後の識別子を以降の行で変更前の識別子に指定している場合

```
A,B
B,C
```

4.12.3 ユーザ識別子変更情報ファイル

ユーザ識別子を変更する場合に作成します。

(1) 形式

ユーザ識別子変更情報ファイルは、次の形式で記述します。

登録結果フラグ, ユーザ識別子変更エントリ [, ユーザプロパティ変更エントリ]

各項目について説明します。

登録結果フラグ

変更処理を実行すると、登録結果フラグに実行結果が出力されます。

<登録結果フラグ> ::= { - | 1 | 2 }

表 4-21 登録結果フラグの意味

登録結果フラグ	意味	コマンド実行時に指定した場合の動作
-	一括変更処理は未実行です。	一括変更処理を実行します。
1	一括変更処理が正常終了しました。	一括変更処理を実行しません。
2	一括変更処理が異常終了しました。	一括変更処理を実行します。

凡例

- : 値を設定しません。

ユーザ識別子変更エントリ

<ユーザ識別子変更エントリ> ::= <変更前ユーザ識別子>, <変更後ユーザ識別子>

<変更前ユーザ識別子>

変更前のユーザ識別子を 1 ~ 254 バイトの範囲で指定します。

<変更後ユーザ識別子>

変更後のユーザ識別子を 1 ~ 254 バイトの範囲で指定します。

ユーザプロパティ変更エントリ

<ユーザプロパティ変更エントリ> ::= <変更前ユーザプロパティ値>, <変更後ユーザプロパティ値>

<変更前ユーザプロパティ値>

変更前のユーザプロパティ値を指定します。

<変更後ユーザプロパティ値>

変更後のユーザプロパティ値を指定します。

(2) 記述例

ユーザ識別子変更情報ファイルの記述例を次に示します。

```

:result,before,after,user prop before,user prop after
,user1,user2,日立 太郎 1,日立 太郎 2
,user3,user4
,user5,user6,日立 太郎 5,日立 太郎 6
    
```

参考

userA を userB へ、userB を userA に変更する場合、コマンドを分割して実行する必要があります。1 回のコマンドで実行すると、記述規則に反するためエラーとなります。1 回目のコマンド実行で、ユーザ A を work に、ユーザ B をユーザ A に変更します。2 回目のコマンド実行で work をユーザ B に変更します。

1 回目

ユーザ A を work に、ユーザ B をユーザ A に変更します。

```
!result,before,after,user prop before,user prop after
,userA,work, ユーザ A,work
,userB,userA, ユーザ B, ユーザ A
```

2 回目

work をユーザ B に変更します。

```
!result,before,after,user prop before,user prop after
,work,userB,work, ユーザ B
```

4.12.4 グループ識別子変更情報ファイル

グループ識別子を変更する場合に作成します。

(1) 形式

グループ識別子変更情報ファイルは、次の形式で記述します。

登録結果フラグ、グループ識別子変更エントリ [ユーザプロパティ変更エントリ]

各項目について説明します。

グループ識別子変更エントリ

<グループ識別子変更エントリ> : : = <変更前グループ識別子> , <変更後グループ識別子>

<変更前グループ識別子>

変更前のグループ識別子を 1 ~ 254 バイトの範囲で指定します。

<変更後グループ識別子>

変更後のグループ識別子を 1 ~ 254 バイトの範囲で指定します。

登録結果フラグ、ユーザプロパティ変更エントリ

ユーザ識別子変更情報ファイルの場合と同じです。「4.12.3(1) 形式」を参照してください。

(2) 記述例

グループ識別子変更情報ファイルの記述例を次に示します。

```
!result,before,after,user prop before,user prop after
,group1,group2, グループ 1, グループ 2
,group3,group4
,group5,group6, グループ 5, グループ 6
```

参考

groupA を groupB へ , groupB を groupC にシフトする場合は , groupB を groupC に変更する指定の後で , groupA を groupB に変更する指定を行います。

```
;result,before,after,user prop before,user prop after  
;groupA,groupB の指定は groupB,groupC の指定行以降に記述します  
,groupB,groupC, グループ B, グループ C  
,groupA,groupB, グループ A, グループ B
```

4.12.5 プロパティ変更情報ファイル

ユーザプロパティの変更を行う場合に作成します。

(1) 形式

プロパティ変更情報ファイルは、次の形式で記述します。

クラス名 , プロパティ名 ,[VariableArray 型プロパティの要素となるプロパティ名],[ユーザプロパティ値に対応するプロパティ名],[ユーザプロパティ値に対応する VariableArray 型プロパティの要素となるプロパティ名], プロパティ種別

各項目について説明します。

クラス名

変更対象のプロパティを持つクラス名を指定します。

プロパティ名

変更対象のプロパティ名を指定します。

VariableArray 型プロパティの要素となるプロパティ名

変更対象のプロパティが VariableArray 型の要素のプロパティ名を指定します。変更対象のプロパティが VariableArray 型でない場合は、指定は無効となります。

ユーザプロパティ値に対応するプロパティ名

<プロパティ名>を一意に識別できるプロパティ名を指定します。

例えば、「ユーザ識別子」(一意な値)と「対応する氏名」(一意でない値)が定義されたクラスで、同姓同名のユーザが存在する場合、<プロパティ名>に「氏名を格納するプロパティ名」を指定しただけではユーザ識別子の異なる同姓同名の氏名についても値が一律に変更されてしまいます。このような場合は、<プロパティ名>には、「氏名を格納するプロパティ名」を指定し、<ユーザプロパティ値に対応するプロパティ名>に一意に識別できる「ユーザ識別子を格納するプロパティ名」を指定することで、ユーザ識別子変更情報ファイルで指定したユーザ識別子を一意に識別して「対応する氏名」を変更できます。

ユーザプロパティ値に対応する VariableArray 型プロパティの要素となるプロパティ名

< VariableArray 型プロパティの要素となるプロパティ名 >を一意に識別できる要素となるプロパティ名を指定します。<ユーザプロパティ値に対応するプロパティ名>が VariableArray 型のプロパティでない場合は、指定は無効となります。

プロパティ種別

変更対象のユーザプロパティがユーザ識別子、グループ識別子、ユーザプロパティ値のどれかを指定

します。

<プロパティ種別> ::= {U|G|UU|UG}

表 4-22 指定できるプロパティ種別

プロパティ種別	指定時の動作	
	<プロパティ名>または < VariableArray 型プロパティの要素となるプロパティ名 > で指定されたプロパティ値	<ユーザプロパティ値に対応するプロパティ名>または は <ユーザプロパティ値に対応する VariableArray 型プロパティの要素となるプロパティ名 > で指定されたプロパティ値
U	ユーザ識別子変更情報ファイルの変更前ユーザ識別子から、変更後ユーザ識別子に変更します。	ユーザ識別子変更情報ファイルの変更前ユーザ識別子から、変更後ユーザ識別子に変更します。
G	グループ識別子変更情報ファイルの変更前グループ識別子から、変更後グループ識別子に変更します。	グループ識別子変更情報ファイルの変更前グループ識別子から、変更後グループ識別子に変更します。
UU	ユーザ識別子変更情報ファイルの変更前ユーザプロパティ値から、変更後ユーザプロパティ値に変更します。	ユーザ識別子変更情報ファイルの変更前ユーザ識別子から、変更後ユーザ識別子に変更します。
UG	グループ識別子変更情報ファイルの変更前ユーザプロパティ値から、変更後ユーザプロパティ値に変更します。	グループ識別子変更情報ファイルの変更前グループ識別子から、変更後グループ識別子に変更します。

(2) 記述例

プロパティ変更情報ファイルの記述例を次に示します。

```

;class name,property name,VA element property name,ex property name,ex VA element property name,type
.edmClass_PublicACL,esysProp_Owner,,U
.edmClass_PublicACL,dbuUserProp_OwnerName,,UU
.ecmClass_DocContainer,esysProp_Creator,,U
.ecmClass_DocContainer,dbuUserProp_OwnerName,,edmProp_OwnerId,,UU
.ecmClass_DocContainer,dbuUserProp_ModifierName,,esysProp_Modifier,,UU
.ecmClass_DocContainer,dbuUserProp_PrimaryGroupName,,edmProp_PrimaryGroupId,,UG
.ecmClass_Flow,ecmProp_Req,ecmProp_Name,ecmProp_Req,ecmProp_Id,UU

```

(3) 注意事項

プロパティ変更情報ファイルには、次のような指定はできません。

- 一括変更の対象でないクラスのクラス名
- 全文検索機能付き文字列型プロパティのプロパティ名
- <プロパティ名>および< VariableArray 型プロパティの要素となるプロパティ名 >において、
「dmaProp」、「edmProp」で始まるプロパティ名
ただし、次のプロパティ名は除きます。
edmProp_Name, edmProp_Owner, edmProp_CreateTime, edmProp_Creator, edmProp_ModifyTime, edmProp_Modifier, edmProp_Comment
- データ型が異なるプロパティの一つのプロパティ変更情報ファイル中への指定

なお、次に示すプロパティ値はデータ型により検索、および登録時に扱われる値が異なります。データ型による扱われるプロパティ値の違いを次の表に示します。

4. 環境設定に必要なファイル

表 4-23 データ型による扱われるプロパティ値の違い

プロパティ値 (指定例)	String 型	Integer 型	Boolean 型
1	1	1	1
0	0	0	0
001	001	1	エラー
10.5	10.5	エラー	エラー
0.5	0.5	エラー	エラー
-1	-1	-1	エラー
2147483648	2147483648	エラー	エラー

注

メッセージ KMBJ20708-E を出力します。

5

DocumentBroker の起動と終了

この章では , DocumentBroker の起動と終了の方法について説明します。

5.1 DocumentBroker の起動方法

5.2 DocumentBroker の終了方法

5.1 DocumentBroker の起動方法

この節では、DocumentBroker の起動方法について説明します。

起動手順

次の手順に従って、DocumentBroker を起動してください。なお、手順 1.、手順 2. および手順 3. の順序は問いません。

1. データベースシステムを起動する。
2. LDAP 対応のディレクトリサービスを起動する。
ユーザ管理機能に LDAP 対応のディレクトリサービスを使用する場合、実行してください。ユーザ管理機能については、「3.6 ユーザ管理機能の設定」を参照してください。
3. アプリケーションサーバを起動する。
4. 業務アプリケーションを起動する。
次のどちらかの方法で起動します。
 - ・ cjstartapp コマンドを使用する
 - ・ Cosminexus Server Plug-in の GUI を使用する開始方法については、マニュアル「Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド」の「J2EE アプリケーションの開始」を参照してください。

5.2 DocumentBroker の終了方法

この節では、DocumentBroker の終了方法について説明します。

終了手順

次の手順に従って、DocumentBroker を終了してください。なお、手順 2.、手順 3. および手順 4. の順序は問いません。

1. 業務アプリケーションを停止する。
次のどちらかの方法で停止します。
 - ・ `cjstopapp` コマンドを使用する
 - ・ Cosminexus Server Plug-in の GUI を使用する

停止方法については、マニュアル「Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド」の「J2EE アプリケーションの停止」を参照してください。

1. アプリケーションサーバを停止する。
2. LDAP 対応のディレクトリサービスを停止する。
ユーザ管理機能に、LDAP 対応のディレクトリサービスを使用する場合、実行してください。
ユーザ管理機能については、「3.6 ユーザ管理機能の設定」を参照してください。
3. データベースシステムを停止する。

6

運用と障害対策

この章では、DocumentBroker で使用するデータベースの運用方法、およびリファレンスファイル管理機能を使用する場合の運用方法について説明します。また、障害対策、および障害情報の取得についても説明します。

6.1 データベースの運用

6.2 クラス定義の変更

6.3 リファレンスファイル管理機能を使用する場合の運用

6.4 障害対策

6.1 データベースの運用

この節では、データベースの運用方法について説明します。

6.1.1 バックアップとリストア

DocumentBroker で使用する表のバックアップと回復は、データベースシステムの機能を利用してください。なお、使用できるデータベースシステムは HiRDB のため、データのバックアップと回復は HiRDB の機能を利用してください。HiRDB でのバックアップの取得方法と回復方法の詳細については、マニュアル「HiRDB システム運用ガイド」を参照してください。

また、ハードウェア障害の発生に備えて、DocumentBroker がインストールされているシステムのレジストリ情報などの環境情報のバックアップも取得するようにしてください。

(1) バックアップとリストアについて

DocumentBroker は、データベース上のレコードを組み合わせることで、アプリケーションで扱う論理的なオブジェクトや、オブジェクト間の関連を実現しています。したがって、障害時の回復には、これらの関連の整合性を保持するようにバックアップおよびリストアを行う必要があります。

(2) バックアップとリストアの方法

DocumentBroker では、オブジェクト間の関連を、文書空間単位に管理しています。また、文書空間はデータベースのスキーマに対応しており、文書空間単位で運用できます。

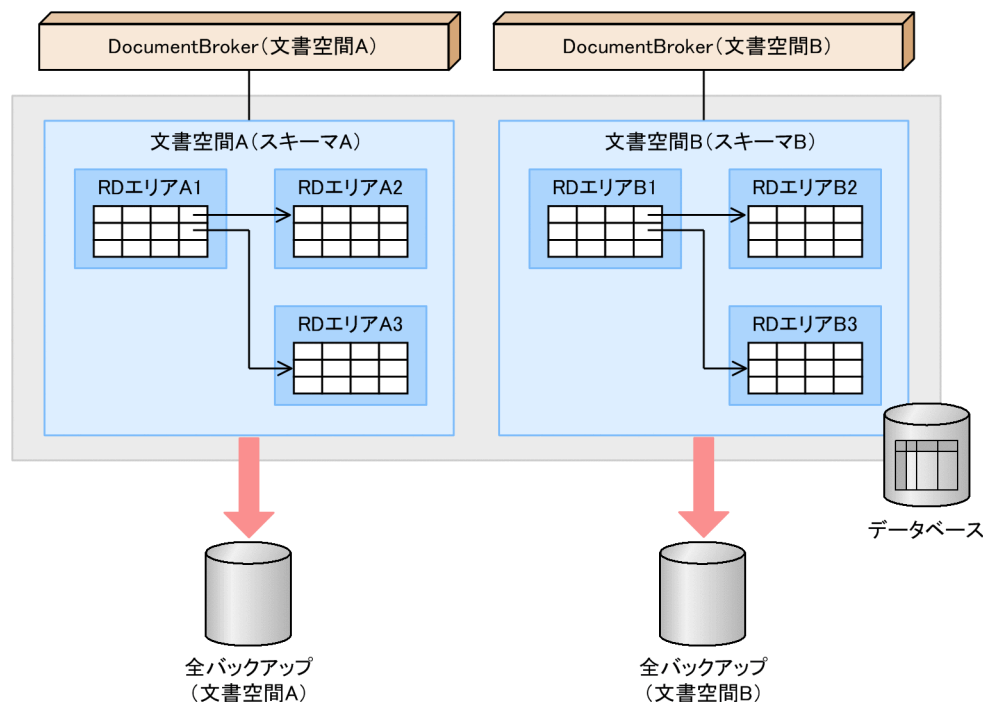
DocumentBroker では、アプリケーションで扱うすべてのデータをデータベース上に格納しています。格納しているデータの障害回復には、データベースの機能を利用してください。

オブジェクト間の関連の整合性を保持したバックアップと回復の方法としては、文書空間単位でバックアップと回復を行う方法と RD エリア単位のバックアップと回復を行う方法があります。次に、これらの方法を説明します。

(a) 文書空間単位のバックアップと回復

文書空間単位でのバックアップの取得の概念を次の図に示します。

図 6-1 文書空間単位でのバックアップの取得



文書空間（スキーマ）に属する全データのバックアップを、同期を取って取得します。障害が発生した場合は、全データをバックアップしたデータを基に回復します。ただし、同期を取って全データのバックアップを取得する運用では、文書空間に属するデータ量が多い場合、バックアップおよび回復に必要なとする時間の増大や、障害回復までに、停止する業務範囲が大きいなどの問題が発生します。

この運用の場合、更新ログの取得方法として次に示す方法のどちらかを選択してください。

ロールバックおよびロールフォワードに必要なログを取得する。

この方法の場合、バックアップから取得したデータに対して、更新ログでの任意の同期点までの論理的な整合性を保って回復します。

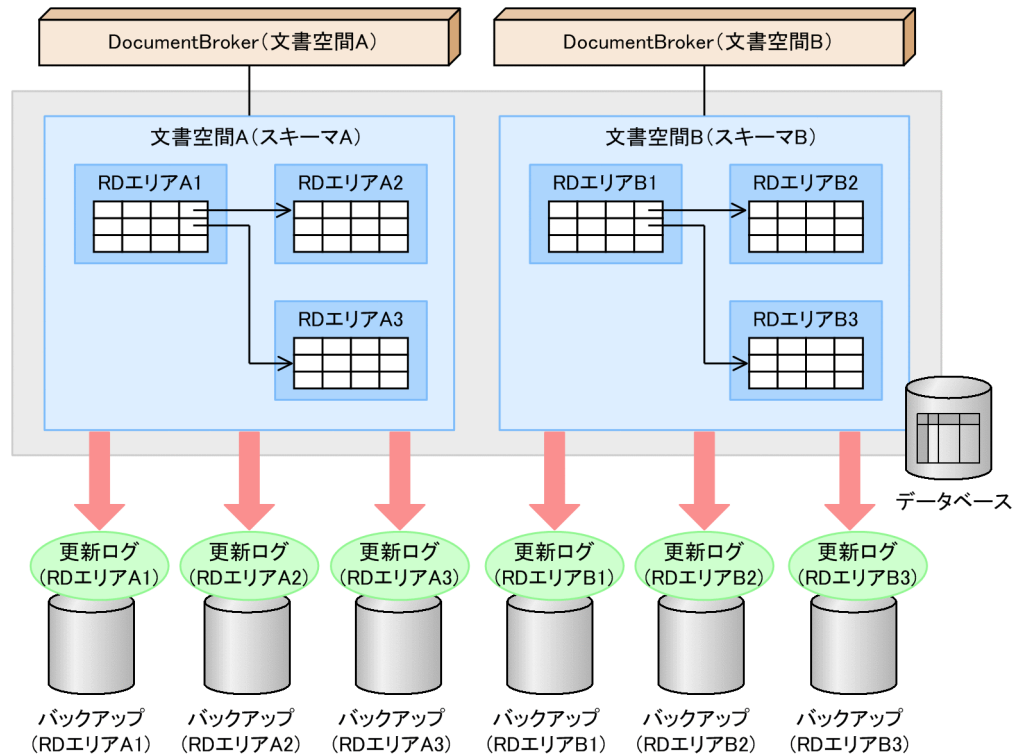
ロールバックに必要なログだけを取得する。

この方法の場合、最後に全データのバックアップを取得した時点まで回復します。

(b) RD エリア単位のバックアップと最新同期点までの回復

RD エリア単位でのバックアップの取得の概念図を次の図に示します。

図 6-2 RD エリア単位でのバックアップ取得



文書空間を構成する個々の RD エリア，またはディスクのパーティション単位にバックアップします。障害が発生した場合は，バックアップしたデータを基に，障害の発生した RD エリアのデータを回復します。このとき，ほかの RD エリアのデータとの関連の整合性を保証するため，データベースの更新ログから最新の同期点まで回復します。このため，更新ログの取得方法は，ロールバックおよびロールフォワードに必要なログを取得する設定にしてください。

この方式では，バックアップした時に RD エリア相互の同期を取る必要がないため，対象データが多い場合にも，日々のバックアップ対象を全体の一部ずつとするなど，柔軟なバックアップの運用ができます。

(3) データベース運用上の注意事項

LOB 列格納用 RD エリアの更新ログの取得方法は，ログ取得モードとなります。次の注意が必要です。

(a) ログファイル容量

当該 LOB 列に対する追加・更新のデータ量の分だけログが増加するため，ログ容量（特にアンロードログファイルのディスク容量）が不足しないようにファイルサイズ（バイト）を見積もってください。1 文書当たりの見積もりの概算を次に示します。

$$(\text{文書ファイル数} + 4) \times \text{平均ファイルサイズ (バイト)}$$

システムファイルの容量見積もりの詳細については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(b) ログ取得性能

ログの出力量の増加による性能劣化を抑えるため，次の 2 点を検討してください。

LOB 列格納用 RD エリアへのグローバルバッファの割り当て

グローバルバッファが指定されていない場合は，データの追加および更新ごとにログがファイルに出

力されるため入出力が増加します。しかし、グローバルバッファが指定されている場合は、コミット時にログのファイルをバッファリングして出力するため、前者と比べて入出力の増加を抑えた運用ができます。グローバルバッファのオペランドの設定の詳細については、マニュアル「HiRDB システム定義」を参照してください。

HiRDB のログ入出力バッファ長の拡大

ログ量の増加に伴ってログ用バッファの使用量が増え、ログバッファの空き待ちになる場合が想定されます。このため、ログ入出力バッファ長の拡大を検討してください。サーバ共通定義の詳細については、マニュアル「HiRDB システム定義」を参照してください。

(4) バックアップの取得

データベースシステムの機能を利用して、同期を取って、DocumentBroker が使用している表一式のバックアップを取得してください。バックアップ対象になる表やエリアなどの情報は、データベースの初期設定で使用したデータベース定義文を基に事前に用意してください。

また、ハードウェア障害の発生に備えて、DocumentBroker がインストールされているシステムのレジストリ情報などの環境情報のバックアップも取得するようにしてください。

(5) リストアの方法

取得したバックアップのリストアは、データベースシステムの機能を利用して実行してください。リストアを実行する前には、必ず DocumentBroker を停止させてください。

6.1.2 ジャーナルファイルの運用

障害が発生した場合は、データベースシステムが取得するジャーナルファイルを利用してデータを回復してください。

なお、使用できるデータベースシステムは HiRDB です。したがって、HiRDB が取得するジャーナルファイルを利用して、データを回復してください。HiRDB のジャーナルファイルの運用方法については、マニュアル「HiRDB システム運用ガイド」を参照してください。

6.1.3 レプリケーション

データベースの内容をほかのデータベースに反映する機能のことを、レプリケーション機能といいます。レプリケーション機能を利用することで、DocumentBroker が使用するデータベースのデータを抽出して、ほかのデータベースへ反映できます。

なお、使用できるデータベースシステムは HiRDB です。したがって、HiRDB のレプリケーション機能を利用します。HiRDB のレプリケーション機能を利用するには、HiRDB に HiRDB Datareplicator を組み込む必要があります。HiRDB のレプリケーション機能については、マニュアル「HiRDB Datareplicator」を参照してください。

(1) レプリケーションの方法

データベースシステムのレプリケーション機能を利用して、ほかのデータベースへデータを反映できます。レプリケーションの対象となる表やエリアなどの情報は、データベースの初期設定で使用したデータベース定義文を基に事前に用意してください。

(2) クラス、プロパティとデータベースの表、列との対応関係

データベースシステムの定義系 SQL を利用して、クラス、プロパティとデータベースの表、列との対応

が取れます。使用できるデータベースシステムは、HiRDB です。したがって、ここでは HiRDB を利用する場合の対応関係について説明します。

DocumentBroker 用データベース定義文の作成コマンド (dbjertsq1) では、メタ情報ファイルに記述されているクラス識別子を、オブジェクトを格納する RDB 表名として設定します。また、メタ情報ファイルに記述されているプロパティ識別子を表の各列名として設定します。

6.2 クラス定義の変更

この節では、DocumentBroker の運用中に、サブクラスやプロパティの追加、削除など、データベースの表を定義・削除する場合の操作手順、および注意事項について説明します。

「クラスの定義」および「プロパティの定義」とは、メタ情報として格納している DocumentBroker のクラスおよびプロパティの定義を表します。また、「表」および「列」は、データベースの表および列を表します。

6.2.1 クラスの定義を変更する場合について

DocumentBroker の運用中にクラスを追加したり、プロパティを追加したりする場合は、クラスの定義を変更する必要があります。DocumentBroker では、次のような場合にクラス定義を変更します。

クラスの追加

クラスの削除

クラスの定義の変更（プロパティの追加、プロパティの定義の変更、プロパティの削除）

次に、これらのクラスの定義を変更する手順を説明します。

6.2.2 クラスの追加

クラスを追加する場合、追加するクラスの定義を記述した定義情報ファイルを作成します。そして、作成した定義情報ファイルを指定してメタ情報の追加コマンド（dbjaddmeta）で、メタ情報へのクラスの定義の追加、および追加するクラスに対応する表を作成するデータベース定義文の生成を実行します。

（1）クラスの追加手順

クラスを追加する手順を次に示します。

1. 定義情報ファイルを作成します。
追加するクラスの定義を記述した定義情報ファイルを作成します。作成した定義情報ファイルは、メタ情報の追加コマンド（dbjaddmeta）の `-f` オプションに指定します。定義情報ファイルの作成については、「4.9 定義情報ファイル」を参照してください。
2. インデクス情報ファイルを作成します。
追加するクラスのプロパティにインデクスを定義する場合、インデクス情報ファイルを作成します。メタ情報の追加コマンド（dbjaddmeta）の `-i` オプションに、作成したインデクス情報ファイル名を指定して実行することで、インデクス情報ファイルの内容を基に、データベース定義文に定義系 SQL 「CREATE INDEX」が出力されます。なお、データベース定義文は、`-o` オプションで指定するデータベース定義文格納ファイルに出力されます。
インデクス情報ファイルについては「4.11 インデクス情報ファイル」を参照してください。
3. RD エリア定義情報ファイルを作成します。
データベース定義文中の RD エリア名称を、ユーザが指定した RD エリア名で出力させるためには、RD エリア定義情報ファイルを作成します。
メタ情報の追加コマンド（dbjaddmeta）の `-r` オプションに、作成した RD エリア定義情報ファイル名を指定することで、RD エリア定義情報ファイルの内容がデータベース定義文の RD エリア名に反映されます。なお、データベース定義文は、`-o` オプションで指定するデータベース定義文格納ファイルに出力されます。
RD エリア定義情報ファイルの作成については、「4.10 RD エリア定義情報ファイル」を参照してください。

さい。

4. メタ情報のバックアップを取得します。
メタ情報を更新する前に、メタ情報ファイルの出力コマンド (dbjprintmeta) でメタ情報のバックアップを取得します。
メタ情報ファイルの出力コマンド (dbjprintmeta) には、-l オプションを指定します。-l オプションには、メタ情報のバックアップを出力するディレクトリ名を指定して、データベースからメタ情報を出力します。
5. データベースのバックアップを取得します。
クラスに対応する表を作成するので、「3.8.2(3) メタ情報用のテーブルを定義します」で定義したメタ情報用の表を作成した RD エリアのバックアップを取得します。バックアップの取得方法については、マニュアル「HiRDB システム運用ガイド」のバックアップの取得方法に関する説明で、定義系 SQL「CREATE TABLE」の記述を参照してください。
6. メタ情報の追加コマンド (dbjaddmeta) を実行します。
手順 1. ~ 手順 3. で作成した定義情報ファイル、インデクス情報ファイル、および RD エリア定義情報ファイルを指定して、メタ情報の追加コマンド (dbjaddmeta) を実行して、クラスの定義をメタ情報へ追加します。メタ情報の追加コマンド (dbjaddmeta) の -o オプションで、追加するクラスに対応する表を作成する定義系 SQL「CREATE TABLE」を格納するデータベース定義文格納ファイルを指定します。
7. データベース定義ユティリティを実行します。
手順 6. で作成したデータベース定義文を入力ファイルとして指定し、HiRDB のデータベース定義ユティリティ (pddef) を実行して定義するクラスに対応する表を追加します。データベース定義ユティリティ (pddef) は、文書空間構成定義ファイルの PdUser エントリに指定するユーザが実行してください。
HiRDB のデータベース定義ユティリティ (pddef) については、マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.2.3 クラスの削除

クラスを削除する場合、メタ情報の削除コマンド (dbjdelmeta) でクラスの定義の削除を実行します。ただし、削除するのはメタ情報の削除コマンド (dbjdelmeta) の -r オプションで指定したサブクラスに対応した表のデータだけです (同じ RD エリアに含まれる別の表のデータは削除されません)。RD エリア内のデータを削除する場合、ここでの手順を実行する前にアプリケーションなどで削除対象のクラスのすべてのオブジェクトを削除しておいてください。

また、VariableArray 型のプロパティの要素を格納するために定義する edmClass_Struct クラスのサブクラスの削除は、次に示す条件を満たしている必要があります。

edmClass_Struct クラスのサブクラスの削除の条件

削除する edmClass_Struct クラスのサブクラスを要素とする、基本単位が VariableArray 型のプロパティを定義しているクラスが存在しない。

クラスを削除する手順を次に示します。

1. メタ情報のバックアップを取得します。
「6.2.2 クラスの追加」のメタ情報のバックアップの取得に関する記述を参照してください。
2. データベースのバックアップを取得します。
クラスに対応する表を削除するため、「3.8.2(3) メタ情報用のテーブルを定義します」で定義したメタ

情報用の表を作成した RD エリア, および dbjdelmeta (メタ情報の削除) コマンドの -r オプションで指定したサブクラス名に対応した表を格納している RD エリアのバックアップを取得します。バックアップの取得方法については, マニュアル「HiRDB システム運用ガイド」のバックアップの取得方法に関する説明で, 定義系 SQL「DROP TABLE」の記述を参照してください。

3. メタ情報の削除コマンド (dbjdelmeta) を実行します。
メタ情報の削除コマンド (dbjdelmeta) でクラスの定義を削除します。メタ情報の削除コマンド (dbjdelmeta) には, -r オプションを指定します。
4. データベース定義ユーティリティを実行します。
HiRDB のデータベース定義ユーティリティ (pddef) を実行して定義するクラスに対応する表を削除します。削除する SQL 文は,「DROP TABLE "-r オプションに指定するサブクラス名"」です。データベース定義ユーティリティ (pddef) は, 文書空間構成定義ファイルの PdUser エントリに指定するユーザが実行します。HiRDB のデータベース定義ユーティリティ (pddef) については, マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.2.4 クラスの定義の変更

クラスの定義の変更では, プロパティの追加, プロパティの定義の変更, またはプロパティの削除のどれかがあります。

(1) プロパティの追加

プロパティを追加するには, 追加するプロパティの定義を記述した定義情報ファイルを作成します。そして, 作成した定義情報ファイルをメタ情報の追加コマンド (dbjaddmeta) で指定して, プロパティの定義の追加, およびプロパティに対応する列を追加するデータベース定義文の生成を実行します。

プロパティを追加する手順を次に示します。

1. 定義情報ファイルを作成します。
追加するプロパティの定義を記述する定義情報ファイルを作成します。定義情報ファイルの作成については,「4.9 定義情報ファイル」を参照してください。
2. インデクス情報ファイルを作成します。
追加するプロパティに対応する列の値は, 手順 5. で示すように NULL 値になります。これによって, 追加するプロパティにインデクスを定義する定義系 SQL「CREATE INDEX」で「除外値指定」はできません。インデクス情報ファイルの作成については,「6.2.2 クラスの追加」のインデクス情報ファイルの作成に関する記述を参照してください。
3. メタ情報のバックアップを取得します。
「6.2.2 クラスの追加」のメタ情報のバックアップの取得に関する記述を参照してください。
4. データベースのバックアップを取得します。
表に列を追加するので,「3.8.2(3) メタ情報用のテーブルを定義します」で定義したメタ情報用の表を作成した RD エリアのバックアップを取得します。バックアップの取得方法については, マニュアル「HiRDB システム運用ガイド」のバックアップの取得方法に関する説明で, 定義系 SQL「ALTER TABLE」の記述を参照してください。
5. メタ情報の追加コマンド (dbjaddmeta) を実行します。
手順 1., および手順 2. で作成した定義情報ファイル, およびインデクス情報ファイルを指定して, メタ情報の追加コマンド (dbjaddmeta) を実行して, プロパティを追加します。
メタ情報の追加コマンド (dbjaddmeta) の -o オプションで, 列を追加する定義系 SQL「ALTER TABLE」を格納するデータベース定義文格納ファイルを指定します。定義系 SQL「ALTER TABLE」

の「ADD」句で追加する列の値には NULL 値が指定されます。

メタ情報の追加コマンド (dbjaddmeta) で追加するプロパティ定義に指定する初期値は、UAP でオブジェクトを新規に作成する場合の初期値であり、追加するプロパティには値を設定しません。

6. データベース定義ユーティリティを実行します。
手順 5. で作成したデータベース定義文を入力ファイルとして指定し、HiRDB のデータベース定義ユーティリティ (pddef) を実行して追加するプロパティに対応する列を追加します。詳細については、「6.2.2 クラスの追加」のデータベース定義ユーティリティの実行に関する記述を参照してください。

(2) プロパティ定義の変更

プロパティの定義を変更する場合、メタ情報の変更コマンド (dbjchgmeta) で、プロパティの定義の変更を実行します。

プロパティの定義を変更する手順を次に示します。

1. メタ情報のバックアップを取得します。
「6.2.2 クラスの追加」のメタ情報のバックアップの取得に関する記述を参照してください。
2. データベースのバックアップを取得します。
表の定義を変更するので、全 RD エリアのバックアップを取得します。バックアップの取得方法については、マニュアル「HiRDB システム運用ガイド」のバックアップの取得方法に関する説明で、定義系 SQL「ALTER TABLE」の記述を参照してください。
3. メタ情報の変更コマンド (dbjchgmeta) を実行します。
メタ情報の変更コマンド (dbjchgmeta) でプロパティの定義を変更します。メタ情報の変更コマンド (dbjchgmeta) の -o オプションで、変更するプロパティに対応する列を変更する定義系 SQL「ALTER TABLE」を格納する、データベース定義文格納ファイルを指定します。
4. データベース定義ユーティリティを実行します。
手順 3. で作成したデータベース定義文を入力ファイルとして指定し、HiRDB のデータベース定義ユーティリティ (pddef) を実行して変更するプロパティに対応する列を変更します。詳細については、「6.2.2 クラスの追加」のデータベース定義ユーティリティの実行に関する記述を参照してください。

(3) プロパティの削除

プロパティを削除する場合、メタ情報の削除コマンド (dbjdelmeta) で、プロパティの定義の削除を実行します。ただし、メタ情報の削除コマンド (dbjdelmeta) では、データベース定義は変更されないため、登録済みのデータは削除されません。また、該当プロパティを持つクラスにデータを追加した場合は、該当プロパティには NULL 値が設定されます。

プロパティの定義を削除する手順を次に示します。

1. メタ情報のバックアップを取得します。
「6.2.2 クラスの追加」のメタ情報のバックアップの取得に関する記述を参照してください。
2. メタ情報の削除コマンド (dbjdelmeta) を実行します。
メタ情報の削除コマンド (dbjdelmeta) でプロパティの定義を削除します。メタ情報の削除コマンド (dbjdelmeta) の -p オプション、および -q オプションを指定して実行します。

コンテンツ格納先ベースパスは、ユーザが管理するディレクトリです。そのため、コンテンツ格納先ベースパスを変更しても、データベースで管理されるコンテンツロケーションと不一致にならないので、ユーザがコンテンツ格納先ベースパスを変更できます。ただし、ファイルシステムのディスクの移行などでコンテンツ格納先ベースパスを変更する場合は、移行先のディスクでも、コンテンツロケーションを移行前と同様のディレクトリ構成にする必要があります。このため、移行前のコンテンツロケーションのディレクトリ構成を移行後のコンテンツ格納先にコピーしてください。

なお、ファイルシステムのディスク移行時などに、コンテンツの格納先ベースパスを誤って変更した場合、またはコンテンツ格納先ベースパスを誤って指定した場合、ファイルシステム上のコンテンツ格納先とデータベースで管理するコンテンツロケーションが不一致となり、リファレンスファイル文書进行操作したときエラーになります。

6.3.2 ファイルシステムのバックアップとリストア

コンテンツを格納するファイルシステムのバックアップおよびリストアは、OS で提供されるシステムツールなどを利用して実施します。

ファイルシステムのバックアップおよびリストアは、必ず DocumentBroker の停止中に実施してください。また、必ずデータベースのバックアップおよびリストアと同期して実施する必要があります。データベースのバックアップとリストアについては、「6.1.1 バックアップとリストア」を参照してください。

6.4 障害対策

この節では、障害が発生した時に DocumentBroker が出力する情報、障害が発生した場合に考えられる要因とその対処方法について説明します。

なお、トランザクションの処理など、データベースシステムの機能に依存している対処方法については、マニュアル「HiRDB システム運用ガイド」、またはマニュアル「HiRDB UAP 開発ガイド」を参照してください。

6.4.1 障害が発生した時に DocumentBroker が出力する情報

ここでは、障害が発生した場合に、DocumentBroker が出力する情報について説明します。

(1) ログの種類

DocumentBroker が運用中に出力するログの種類を次の表に示します。

表 6-1 ログの種類と用途

ログの種類	用途
メッセージログ	稼働中のインフォメーション情報や障害時のエラー情報を出力します。アプリケーションおよびコマンドの稼働監視の情報として使用します。メッセージの詳細については、マニュアル「DocumentBroker Version 5 メッセージ」を参照してください。
トレース	システムでトラブルが発生した場合の障害保守情報を出力します。保守員が障害解析用に使用します。
アプリケーショントレース	DocumentBroker クラスライブラリを使用したアプリケーションを開発・運用する上で、必要な情報を出力します。
アプリケーションエラーログ	DocumentBroker クラスライブラリを使用したアプリケーションで、エラーが発生した場合にエラーの内容を出力します。

トラブルシューティングに必要な資料は、運用を開始する前に資料取得のための設定をしておく必要があります。各ログの出力方法や出力内容については、動作環境定義ファイルで設定できます。動作環境定義ファイルについては、「4.2 動作環境定義ファイル (dochconf.properties)」を参照してください。

(a) 出力先ディレクトリ

ログは、動作環境定義ファイルで指定したディレクトリに出力されます。デフォルトの出力先ディレクトリは、次のとおりです。

- 実行環境ディレクトリ /log

(b) 出力ファイル名

トレースファイルの出力ファイル名は、環境変数に指定できません。次の表に示すファイル名で出力されます。

表 6-2 ログの出力先とファイル名

種別	コンポーネント	ログ出力先およびファイル名
メッセージログ	ライブラリ	UNIX の場合：<LogPath>/dbjmessage_x.log Windows の場合：<LogPath>%dbjmessage_x.log

種別	コンポーネント	ログ出力先およびファイル名
	コマンド	UNIX の場合：<LogPath>/< コマンド名 >_message_x.log Windows の場合：<LogPath>¥< コマンド名 >_message_x.log
トレース	ライブラリ	UNIX の場合：<LogPath>/dbjtrace_x.log Windows の場合：<LogPath>¥dbjtrace_x.log
	コマンド	UNIX の場合：/<LogPath>/< コマンド名 >_trace_x.log Windows の場合：<LogPath>¥< コマンド名 >_trace_x.log
アプリケーショントレース	ライブラリ	UNIX の場合：次のどちらかとなります。 <ul style="list-style-type: none"> • <APTracePath>/dbjaptrace_x.log • <APTracePath>/<tracePrefix>_x.log Windows の場合：次のどちらかとなります。 <ul style="list-style-type: none"> • <APTracePath>¥dbjaptrace_x.log • <APTracePath>¥<tracePrefix>_x.log
アプリケーションエラーログ	ライブラリ	UNIX の場合：次のどちらかとなります。 <ul style="list-style-type: none"> • <APTracePath>/dbjaptraceerr_x.log • <APTracePath>/<tracePrefix>err_x.log Windows の場合：次のどちらかとなります。 <ul style="list-style-type: none"> • <APTracePath>¥dbjaptraceerr_x.log • <APTracePath>¥<tracePrefix>err_x.log

注意：「x」は面の番号（0 から面数（最大 15）まで）です。

6.4.2 障害要因の特定

障害要因の特定方法について説明します。

障害が発生した場合、その対処方法を決めるためには、その障害の要因を特定する必要があります。障害の要因を特定するには、次の項目を確認してください。

(1) エラーメッセージの確認

障害の要因を特定するために、まず、DocumentBroker が作成するメッセージログファイルにエラーメッセージが出力されているかどうかを確認します。

メッセージログファイルの出力先については、「6.4.1(1) ログの種類」を参照してください。

エラーメッセージが出力されている場合、エラーメッセージのメッセージ ID からエラーが発生しているプログラム、障害の要因が特定できます。

(2) ユーザの操作内容の確認

障害が発生したときに実行していた操作の詳細な手順を確認します。正しい操作手順をしていないことが要因で、障害が発生している場合が考えられます。操作手順を確認することで、障害の要因がある程度特定できます。

6.4.3 障害への対処

障害への対処方法について説明します。

障害要因が判明したら、適切な対処方法で障害の要因を取り除きます。エラーメッセージが出力されている場合、メッセージの対処に従って対処してください。サポートサービスへの連絡が必要になる障害の場

合、または対処方法が不明の障害の場合、必要な障害情報を取得して、サポートサービスに連絡し、その指示に従って対処してください。

6.4.4 IT Report Utility を利用した保守情報の採取

IT Report Utility を利用した保守情報の採取方法について説明します。

(1) 使用方法

IT Report Utility の次のコマンド (systoru) を使用して保守情報を採取します。

```
systoru -t failure -g "DocumentBroker" -g "os"
```

(2) 採取できる資料一覧

IT Report Utility を利用して採取する保守情報は、次のとおりです。

- OS 資料 (syslog ファイル (UNIX の場合) またはイベントログ (Windows の場合) など)
- DocumentBroker の実行環境ディレクトリ (環境変数「DOCB_CONF_DIR」) の内容

(3) 注意事項

- このコマンドを実行する前に、IT Report Utility をインストールしておいてください。IT Report Utility のインストール方法、詳細な使用方法については、IT Report Utility が提供しているドキュメント「IT Report Utility (Windows 版) の機能」または「システム情報採取ツール (Linux 版) の機能」を参照してください。
- このコマンドを実行する前に、DocumentBroker の実行環境ディレクトリを環境変数「DOCB_CONF_DIR」に指定してください。
- このコマンドは、DocumentBroker のシステム管理者が実行してください。
- 動作環境定義ファイルで各種ログおよびトレースの出力先ディレクトリに実行環境ディレクトリの配下以外を指定している場合、各種ログおよびトレースの採取はできません。
- DocumentBroker の実行環境ディレクトリは 300 メガバイトを超える場合があります。このため、IT Report Utility の資料採取で消費するディスク容量の上限値「SYSTORU_MAX_OUTPUT_DATA_SIZE」には、716800 (700 メガバイト) 以上の指定を推奨します。設定方法の詳細については IT Report Utility の提供するドキュメントを参照してください。

7

コマンドリファレンス

この章では、DocumentBroker で使用するコマンドの種類と文法について説明します。

7.1 コマンドの種類

7.2 コマンドの形式

7.3 コマンドの文法

7.4 コマンドが出力するファイル

7.1 コマンドの種類

この節では、DocumentBroker で使用するコマンドの種類について説明します。

注意事項

コマンドを実行する場合、次の点に注意してください。

- コマンドを実行できるユーザは、システム管理者だけです。
- コマンドを実行する前に、データベース情報ファイルおよびコマンド用環境設定ファイルを設定しておいてください。
- コマンドを実行する前に、DocumentBroker の実行環境ディレクトリを環境変数「DOCB_CONF_DIR」に指定してください。
- コマンドを実行する前に、データベースを起動しておいてください。

格納先

これらのコマンドを実行する場合、以下に格納されているコマンドを使用してください。

UNIX の場合

- /opt/DocBroker/Platform/bin
- /opt/DocBroker/Library/bin

Windows の場合

- <インストールディレクトリ>%DocBroker%Platform%bin
- <インストールディレクトリ>%DocBroker%Developer%bin
- <インストールディレクトリ>%DocBroker%Library%bin

注 DocumentBroker Runtime Library の場合に使用できるのは、dbjregenvid コマンドだけです。

7.1.1 コマンド一覧

コマンドの一覧を次の表に示します。

表 7-1 コマンド一覧

種別	コマンド	機能
データベース運用	dbjinitmeta	メタ情報の初期設定
	dbjaddmeta	メタ情報の追加
	dbjrtssql	DocumentBroker 用データベース定義文の作成
	dbjdelmeta	メタ情報の削除
	dbjregenvid	DocumentBroker 実行環境の情報の登録
	dbjchgmeta	メタ情報の変更
	dbjchgusrid	ユーザ / グループの一括変更
一覧出力	dbjlsfldr	フォルダ一覧の出力
	dbjlsdoc	文書一覧の出力
その他	dbjprintmeta	登録済みメタ情報の出力
	dbjgenguid	GUID の生成

7.2 コマンドの形式

この節では、コマンドの入力形式、使用方法および注意事項について説明します。

(1) 入力形式

コマンドの入力形式を次に示します。

コマンド名称 [オプション...]

(a) コマンド名

コマンド名は、実行するコマンドのファイル名です。

(b) オプション

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

オプションの形式

オプションはマイナス記号で始まる文字列で、次に示すように、引数を取らないか、または1個の引数を取ります。

形式 1 : - オプションフラグ

形式 2 : - オプションフラグ < 空白またはタブ > フラグ引数

(凡例)

オプションフラグ : 1 文字の英数字で、英大文字・小文字は区別される

フラグ引数 : オプションフラグに対する引数

オプションの指定規則

フラグ引数を取らないオプションフラグは、一つのマイナス記号のあとにまとめて指定できません。

誤った指定例 : \$ cmd -abc

正しい指定例 : \$ cmd -a -b -c

フラグ引数を必要とするオプションフラグのフラグ引数は省略できません。

例えば、オプションフラグ -a がフラグ引数を取る場合、次のように入力すると -b はフラグ引数とみなされます。

\$ cmd -a -b

オプションフラグとフラグ引数の間には空白またはタブが必要です。

誤った指定例 : \$ cmd -afile

正しい指定例 : \$ cmd -a file

同じオプションフラグを 2 回以上指定できません。例えば、「\$ cmd -a 1 -a 2」とは入力できません。

マイナス記号だけのオプションは入力できません。例えば、「\$ cmd -」と入力すると「-」はコマンド引数とみなされます。

文書空間で使用する文字コード種別が UTF-8 の場合に、オプションのフラグ引数としてファイルのパスを指定するとき、パスは印刷可能な ASCII コードで記述します。

(2) 入出力

(a) 入力

入力は、すべてコマンドのオプション、引数の並びです。

7. コマンドリファレンス

(b) 出力

コマンド処理が正常に終了した場合の出力は、すべて標準出力に対して実行します。また、終了コードの一覧を次の表に示します。

表 7-2 終了コード一覧

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

コマンド処理がエラーになった場合（終了コードが「0」以外）は、メッセージをすべて標準エラー出力に出力します。引数エラーになった場合（終了コードが「2」）は、標準エラー出力にコマンドの使用方法（USAGE）を出力します。出力形式は次のとおりです。

出力形式

Usage:xxxx yyyy

- xxxx：コマンド名称が出力されます。
- yyyy：コマンドの指定形式が表示されます。

出力例

```
Usage:command -a -b option_arg_1 [-c][-d option_arg_2] ...
       {-e | -f option_arg_3}
```

(3) コマンドの実行可能ファイルに対するアクセス権限

コマンドの実行可能ファイルにアクセスしてコマンドを実行できるユーザはシステム管理者だけです。

7.3 コマンドの文法

この節では、コマンドの文法について説明します。なお、各コマンドは、アルファベット順に説明します。

コマンドは、DocumentBroker の停止中に実行してください。

使用できるデータベースシステムは HiRDB です。したがって、コマンドの文法は HiRDB を使用する場合の記述形式になっています。

dbjaddmeta (メタ情報の追加)

機能

引数に指定した定義情報ファイルの内容を基に、データベース中のメタ情報にユーザが定義したサブクラスとプロパティを追加します。サブクラスまたはプロパティを追加した場合は、自動的に動作環境メタ情報ファイルも更新されます。

データベースに対するデータベース定義後に、このコマンドでサブクラスやプロパティを追加した場合、データベース定義文を出力し、データベース定義をする必要があります。

形式

```
dbjaddmeta
    -f 格納先ディレクトリ名
    [-o データベース定義文格納ファイル
    [-i インデクス情報ファイル名]
    [-r RDエリア定義情報ファイル名]
    ]
```

オプション

- f 格納先ディレクトリ名
定義情報ファイルを格納したディレクトリ名を絶対パスで指定します。定義情報ファイルについては、「4.9 定義情報ファイル」を参照してください。
- o データベース定義文格納ファイル
作成したデータベース定義文を格納するファイルを絶対パスで指定します。このオプションを指定した場合、定義情報ファイルの内容によって、データベース定義文に出力される SQL 文が異なります。なお、出力先にファイルが存在する場合は、上書きされます。
 - サブクラスを追加する場合
定義系 SQL「CREATE TABLE」文がデータベース定義文に出力されます。
 - 追加済みのクラスにプロパティを追加する場合
定義系 SQL「ALTER TABLE」文がデータベース定義文に出力されます。
- i インデクス情報ファイル名
-o オプションを指定した場合だけ指定できるオプションです。
追加するプロパティに対して定義系 SQL「CREATE INDEX」文（インデクス定義文）を出力する場合に、インデクス情報ファイル名を絶対パスで指定します。インデクス情報ファイルについては、「4.11 インデクス情報ファイル」を参照してください。
- r RD エリア定義情報ファイル名
-o オプションを指定した場合だけ指定できるオプションです。
RD エリア定義情報ファイル名を絶対パスで指定します。

-o オプションで指定したデータベース定義文格納ファイル中の RD エリア名を、指定した RD エリア定義情報ファイルの情報に従って出力する場合に指定します。RD エリア定義情報ファイルについては、「4.10 RD エリア定義情報ファイル」を参照してください。

注意事項

このコマンドを実行する前には、dbjinitmeta コマンドでメタ情報をデータベースに登録しておく必要があります。

データベースの初期設定をする場合は、dbjcrtsql コマンドを実行する前に、このコマンドを実行してください。

このコマンドはデータベース自身の定義であるエリアの定義は実行しません。必要なエリアの定義は、出力されるデータベース定義文に追加してください。

このコマンドで追加した定義を削除するには、コマンド実行前に取得したデータベースのバックアップをリストアするか、メタ情報の削除コマンド (dbjdelmeta) を実行してください。コマンドを実行する前にはデータベースのバックアップを取得してください。

次に示す場合は、インデクス格納 RD エリアの名称として「USR_INDEX_AREA」が出力されます。

- -i オプションにインデクス情報ファイル名を指定し、-r オプションを省略してこのコマンドを実行した場合
- RD エリア定義情報ファイルで種別「IndexArea」に、インデクス情報ファイルに記述した項目に対応する RD エリアの定義を記述していない状態で、-i オプションにインデクス情報ファイル名を指定し、-r オプションに RD エリア定義情報ファイル名を指定してこのコマンドを実行した場合

実行結果であるデータベース定義文に対してエリア定義などの必要な項目を追加、修正してからデータベースを作成してください。ただし、-r オプションを指定した場合は、RD エリア定義情報ファイルの指定に従って、RD エリア名が出力されます。

一度このコマンドを実行した環境で、再度メタ情報の追加を行う場合、新たに追加するメタ情報だけを記述した定義情報ファイルを使用する必要があります。

dbjchgmeta (メタ情報の変更)

機能

メタ情報の内容を変更します。変更できる内容を次に示します。

- 文字列型プロパティの最大長を大きくする。
- 繰り返し列の最大要素数を大きくする。

形式

```
dbjchgmeta  -p プロパティ名  
             {-l 最大長 | -e 最大要素数}  
             -o データベース定義文格納ファイル名
```

オプション

- p プロパティ名
変更するプロパティ名を指定します。
- l 最大長
変更後の文字列の最大長を指定します。指定できる範囲はデータベースに依存します。

- e 最大要素数
変更後の繰り返し列の最大要素数を指定します。2 ~ 30,000 の範囲の 10 進数の値で指定してください。
- o データベース定義文格納ファイル名
データベース定義文を格納するファイルを、絶対パスで指定します。なお、出力先にファイルが存在する場合は、上書きされます。

注意事項

文字列プロパティの最大長を変更できるのは、ユーザプロパティだけです。

このコマンドの実行で変更した値を、このコマンドの再実行により小さい値に変更することはできません。コマンドを実行する前に、データベースのバックアップを取得しておいてください。

dbjcrtsql (DocumentBroker 用データベース定義文の作成)

機能

データベースに登録されたメタ情報を基に、DocumentBroker のデータベースを初期設定するために必要な以下のデータベース定義文を作成します。

- システムクラスのテーブル定義文
- ユーザクラスのテーブル定義文
- システムプロパティに対するインデクス定義文
- ユーザプロパティに対するインデクス定義文 (-i オプション指定時)

形式

```
dbjcrtsql  -o データベース定義文格納ファイル名
           [-i インデクス情報ファイル名]
           [-r RDエリア定義情報ファイル名]
```

オプション

- o データベース定義文格納ファイル名
作成したデータベース定義文を格納するファイルを絶対パスで指定します。なお、出力先にファイルが存在する場合は、上書きされます。
- i インデクス情報ファイル名
インデクス定義をするための情報を記述したファイル名を絶対パスで指定します。ユーザが追加したプロパティに対して定義系 SQL 「CREATE INDEX」文 (インデクス定義文) を出力する場合に指定します。インデクス情報ファイルについては、「4.11 インデクス情報ファイル」を参照してください。
- r RD エリア定義情報ファイル名
RD エリア定義情報ファイル名を絶対パスで指定します。
 - o オプションで指定したデータベース定義文格納ファイル中の RD エリア名を、指定した RD エリア定義情報ファイルの情報に従って出力する場合に指定します。RD エリア定義情報ファイルについては、「4.10 RD エリア定義情報ファイル」を参照してください。

注意事項

このコマンドを実行する前には、dbjinitmeta コマンドでメタ情報をデータベースに登録しておく必要

があります。

このコマンドは、データベース自身の定義であるエリア定義は実行しません。必要なエリアの定義は、出力されるデータベース定義文に追加してください。

次に示す場合は、インデクス格納 RD エリアの名称として「USR_INDEX_AREA」が出力されます。

- `-i` オプションにインデクス情報ファイル名を指定し、`-r` オプションを省略してこのコマンドを実行した場合
- RD エリア定義情報ファイルで種別「IndexArea」に、インデクス情報ファイルに記述した項目に対応する RD エリアの定義を記述していない状態で、`-i` オプションにインデクス情報ファイル名を指定し、`-r` オプションに RD エリア定義情報ファイル名を指定してこのコマンドを実行した場合

実行結果であるデータベース定義文に対してエリア定義などの必要な項目を追加、修正してからデータベースを作成してください。ただし、`-r` オプションを指定した場合は、RD エリア定義情報ファイルの指定に従って、RD エリア名が出力されます。

dbjdelmeta (メタ情報の削除)

機能

dbjaddmeta コマンドで追加したサブクラスやプロパティを削除します。

ユーザが追加したサブクラスを削除する場合は、`-r` オプションを指定します。ユーザが追加したプロパティを削除する場合は、`-p` オプションと `-q` オプションを指定します。ただし、全文検索用のプロパティは削除できません。

形式

```
dbjdelmeta { -r サブクラス名 | -p プロパティ名 -q クラス名 }
```

オプション

- r サブクラス名
サブクラスを削除する場合に、サブクラス名を指定します。
- p プロパティ名
プロパティを削除する場合に、プロパティ名を指定します。
- q クラス名
`-p` オプションで指定したプロパティを持つクラスの、クラス名を指定します。

注意事項

指定されたサブクラスが `edmClass_Struct` クラスのサブクラスの場合、`VariableArray` 型プロパティから参照されていないときだけ削除できます。

このコマンドを実行すると、データベースに登録されているメタ情報は更新されますが、データベース定義は変更されません。

dbjgenguid (GUID の生成)

機能

文書空間識別子、クラス ID、プロパティ ID に使用する GUID を生成します。

形式

```
dbjgenguid [-n 生成するGUIDの数] [-o 出力ファイル名]
```

オプション

- n 生成する GUID の数
生成する GUID の数を指定します。1 ~ 256 の範囲の 10 進数の値で指定してください。このオプションの指定を省略した場合、1 つの GUID を生成します。
- o 出力ファイル名
生成した GUID を格納するファイル名を、絶対パスで指定します。このオプションの指定を省略した場合、標準出力に出力します。なお、出力先にファイルが存在する場合は、上書きされます。

実行例

- n オプションを指定しないで実行した場合の、出力例を次に示します。

```
KMBJ20001-I dbjgenguid start.
aa2e7f2f-d4e9-4a5a-9c20-1d87a882556c
KMBJ20002-I dbjgenguid ended.
```

- n オプションに 2 を指定して実行した場合の、出力例を次に示します。

```
KMBJ20001-I dbjgenguid start.
f3ba558e-bd73-4e6b-9aad-1da72f833039
7ea15529-899d-4372-88b0-bb49afc47170
KMBJ20002-I dbjgenguid ended.
```

dbjinitmeta (メタ情報の初期設定)

機能

DocumentBroker のメタ情報を表にして、文書空間構成定義ファイルで指定されているデータベースに登録します。

形式

```
dbjinitmeta [-d [-e]] [-f 格納先ディレクトリ名]
```

オプション

- d
メタ情報をデータベースに再登録する場合に指定します。このオプションを指定すると、既存のメタ情報をデータベースから削除して、新規にメタ情報を登録します。
- e
複数の実行環境から同一の文書空間にアクセスしている場合、それらの実行環境の情報をすべて削除するときに指定します。このオプションを指定できるのは、実行環境識別子が 0 の実行環境だけです。
このオプションを指定すると、登録されている実行環境の情報が削除されるため、複数の実行環境から同一の文書空間にアクセスする実行環境を新規に登録し直す必要があります。
- f 格納先ディレクトリ名
メタ情報ファイルを格納したディレクトリ名を、絶対パスで指定します。
このオプションは、dbjprintmeta コマンドで出力したメタ情報ファイルを登録する場合に使用しま

す。

注意事項

-d オプションを指定してこのコマンドを実行した場合は、必ず dbjrtssql コマンドを実行して、出力されたデータベース定義文を基に表を再定義してください。

文字コード種別を変更する場合、事前にデータベースの文字コード種別を変更する必要があります。また、データベースで文字コード種別を変更する場合は、データベースの初期化（RD エリアの再作成）を行う必要があります。

dbjprintmeta (登録済みメタ情報の出力)

機能

データベースに登録されているメタ情報を csv 形式で出力します。出力するファイルを次に示します。

- edmsys.csv
- edmclass.csv
- edmclassprop.csv
- edmprop.csv

形式

dbjprintmeta [-v] [-l 出力先ディレクトリ名]

オプション

-v

DocumentBroker Version 3 以前のメタ情報を基にメタ情報を出力します。このオプションは、DocumentBroker Version 3 から DocumentBroker Version 5 に移行する場合に使用します。

-l 出力先ディレクトリ名

メタ情報ファイルの出力先を絶対パスで指定します。このオプションの指定を省略した場合は、実行環境ディレクトリ /env_info 下に出力します。なお、出力先に同名のファイルが存在する場合は、上書きされます。

dbjregenvid (DocumentBroker 実行環境の情報の登録)

機能

DocumentBroker の実行環境についての情報（実行環境識別子、環境変数「DOCB_CONF_DIR」に指定している実行環境ディレクトリのパス、およびホスト名）をデータベース中の表に登録します。また、登録した実行環境の更新、削除、および登録されている実行環境のファイル出力をします。

なお、このコマンドで登録した実行環境識別子は、該当する実行環境下で作成されるすべてのオブジェクトの OIID に付加されます。

形式

dbjregenvid -r [-i 実行環境識別子] |
-u 実行環境識別子 |
-d |
-l

オプション

-r

DocumentBroker の実行環境の情報を登録する場合に指定します。

このオプションを指定すると、データベース中の表に実行環境識別子のレコードが挿入 (INSERT) されます。

-i 実行環境識別子

登録する実行環境識別子を指定します。1 ~ 254 の範囲で、10 進数の値で指定してください。

このオプションを省略した場合は、1 ~ 254 の範囲で未使用の識別子のうち、最小値の識別子が自動的に採番されます。

-u 実行環境識別子

実行環境の情報を更新する場合に指定します。実行環境を別のマシンに移動したり、実行環境ディレクトリのパスを変更したりした場合は、-u オプションを指定してこのコマンドを実行してください。このオプションを指定すると、引数に指定した実行環境識別子に対応するレコードのカラム (環境変数「DOCB_CONF_DIR」に指定している実行環境ディレクトリのパス、およびホスト名) が更新されます。

-d

実行環境を削除する場合に指定します。ただし、このコマンドを実行した実行環境の OIID 通番レコードが 0 件の場合だけ削除できます。

登録されている実行環境の一覧を標準出力に表示する場合に指定します。

-l

登録されている実行環境の一覧を標準出力 (UNIX)、またはコマンドプロンプト (Windows) に表示する場合に指定します。

実行例

-l オプションを指定して実行した場合の、出力例を次に示します。

ID=0	HOST-NAME=koyuki	ENVDIR=c:\program files\hitachi\docb\server1
ID=20	HOST-NAME=koyuki	ENVDIR=c:\program files\hitachi\docb\server2
ID=100	HOST-NAME=Host1	ENVDIR=c:\program files\hitachi\docb100\server

注意事項

dbjinitmeta コマンドで初期構築した実行環境 (実行環境識別子が 0) については、-d オプションで削除できません。

一度登録された実行環境識別子は削除・変更できません。ただし、実行環境識別子が 0 以外の実行環境については、実行環境でオブジェクトを作成していない場合に (実行環境の OIID 通番レコードが 0 件の場合に)、-d オプションを指定してこのコマンドを実行することで、実行環境識別子を削除できます。また、実行環境識別子の削除後、-r オプションを指定してこのコマンドを実行することで、実行環境識別子を変更できます。

dbjchgusrid (ユーザ / グループの一括変更)

機能

オブジェクトに設定されているユーザ識別子, グループ識別子, および関連したユーザプロパティを変更情報ファイルに記述した内容に一括で変更します。

一括変更の対象となるクラスを次に示します。

- dmaClass_ConfigurationHistory
- dmaClass_Container
- dmaClass_DocVersion
- edmClass_ContainerVersion
- edmClass_IndependentPersistence
- edmClass_VersionTracedDocVersion
- edmClass_VersionTracedComponentDocVersion
- dmaClass_DirectContainmentRelationship ¹
- dmaClass_ReferentialContainmentRelationship ¹
- edmClass_Relationship ¹
- edmClass_PublicACL
- edmClass_ACL
- ユーザが定義したすべてのサブクラス

一括変更の対象となるプロパティを次に示します。

- edmProp_OwnerId
- edmProp_PrimaryGroupId
- edmProp_ACL ²
- edmProp_SACL ²
- ユーザプロパティ

注 1

一括変更の対象は, ユーザプロパティだけです。

注 2

ACE に設定したユーザ識別子, またはグループ識別子だけが変更されます。パーミッションは変更されません。

形式

```
dbjchgusrid {-u ユーザ識別子変更情報ファイル名 |
             -g グループ識別子変更情報ファイル名 |
             -u ユーザ識別子変更情報ファイル名
             -g グループ識別子変更情報ファイル名}
             [-p プロパティ変更情報ファイル名]
             [-t トランザクション件数]
             [-c]
             [-v]
             -o 出力ディレクトリ
```

オプション

-u ユーザ識別子変更情報ファイル名

変更するユーザ識別子，および関連したユーザプロパティを記述した変更情報ファイル名を絶対パスで指定します。変更処理後は，実行結果が書き込まれます。
このオプションを省略した場合，ユーザ識別子は変更されません。

-g グループ識別子変更情報ファイル名

変更するグループ識別子，および関連したユーザプロパティを記述した変更情報ファイル名を絶対パスで指定します。変更処理後は，実行結果が書き込まれます。
このオプションを省略した場合，グループ識別子は変更されません。

-p プロパティ変更情報ファイル名

変更するユーザプロパティを記述した変更情報ファイル名を，絶対パスで指定します。
このオプションを省略した場合，ユーザプロパティは変更されません。

-t トランザクション件数

1 トランザクションで変更するレコード数を 0 ~ 4,096 の範囲で指定します。

• 0 を指定した場合

変更情報ファイルの行ごとに各クラスのプロパティを順番に，1 トランザクションでユーザ，グループ，またはプロパティを対象として変更します。

• 1 以上を指定した場合

変更情報ファイルの行ごとに各クラスのプロパティを順番に，指定された件数ずつユーザ，グループ，またはプロパティを変更します。

指定できる範囲以外の値を指定した場合は，エラーとなります。このオプションを省略した場合は，「200」が仮定されます。

-c

ユーザ識別子変更情報ファイル，グループ識別子変更情報ファイル，およびプロパティ変更情報ファイルが正しく記述されているかをチェックする場合に指定します。次の項目がチェックされます。なお，このオプションを指定した場合，変更処理は実行されません。

- ファイルが正しく記述されているか。
- クラス，およびプロパティが文書空間に存在しているか。
- 変更前の識別子，および変更後の識別子が指定できる値か。

次の内容を確認します。

- 型が一致しているか。
- 最大長を超えていないか。

対象となるクラス，プロパティに対応する表，および列がデータベースに存在するかどうかのチェックは行いません。

-v

変更情報ファイルに指定されたユーザ識別子，グループ識別子の変更を 1 行完了するごとに，メッセージを出力する場合に指定します。次のメッセージが出力されます。

- KMBJ20701-I (ユーザ変更の完了)
- KMBJ20702-I (グループ変更の完了)

このオプションを省略した場合は，すべての処理が完了してからメッセージを出力します。

-o 出力ディレクトリ

指定した変更情報ファイルのバックアップファイルを格納するディレクトリを絶対パスで指定します。次のファイルが格納されます。

- ユーザ識別子変更情報バックアップファイル (-u オプションを指定した場合)
- グループ識別子変更情報バックアップファイル (-g オプションを指定した場合)

- プロパティ変更情報バックアップファイル（-p オプションを指定した場合）
 - チェックアウト一覧出力ファイル（チェックアウト中のオブジェクトが存在した場合）
- 各ファイルのファイル名と出力形式を次に示します。

表 7-3 バックアップファイルのファイル名と出力形式

バックアップファイル名	ファイル名	出力形式
ユーザ識別子変更情報バックアップファイル	dbjchg_u_<YYYYMMDDhhmmssxxx>.log	ユーザ識別子変更情報バックアップファイルと同じ形式です。
グループ識別子変更情報バックアップファイル	dbjchg_g_<YYYYMMDDhhmmssxxx>.log	グループ識別子変更情報バックアップファイルと同じ形式です。
プロパティ変更情報バックアップファイル	dbjchg_p_<YYYYMMDDhhmmssxxx>.log	プロパティ変更情報バックアップファイルと同じ形式です。
チェックアウト一覧出力ファイル	dbjcolist_<YYYYMMDDhhmmssxxx>.log	下記の「チェックアウト一覧出力ファイルの出力形式」を参照してください。

（凡例）

- < YYYY > : 西暦年号
- < MM > : 月
- < DD > : 日
- < hh > : 時
- < mm > : 分
- < ss > : 秒
- < xxx > : ミリ秒

チェックアウト一覧出力ファイルの出力形式

チェックアウト一覧出力ファイルの出力形式を次に示します。バージョン付き文書の OIID が出力されます。

```
dma:///<system id>/<docspace id>/<object id>
dma:///<system id>/<docspace id>/<object id>
.
```

注意事項

このコマンドを実行する前に、次の項目を実行しておいてください。

- チェックアウト中のオブジェクトが存在しないことを確認する。
- データベースのバックアップを取得する。
- 他のアプリケーションを停止する。

このコマンドの実行中にエラーが発生し処理が終了した場合は、データベースの内容が不整合となります。次の方法で回復してください。

- バックアップからデータベースを回復する。
- エラーの要因を取り除いてから一括変更完了のメッセージが出力されるまで、同じオプションを指定して、コマンドを再実行する。

この場合、引数で指定した変更情報ファイルは変更しないでください。なお、メッセージ「KMBJ20710-E」が出力された場合は、バックアップからデータを復旧してください。

チェックアウト中のオブジェクトが存在する場合、チェックアウト一覧出力ファイルを出力して、警告つき正常終了します。一括変更処理は実行されません。

変更対象のオブジェクト数が多い場合は、処理に時間がかかることがあります。

このコマンドを実行中にキャンセルした場合は、バックアップからデータベースを回復してください。
変更情報ファイルに有効なデータがない場合、バックアップファイルは作成されません。

dbjlsfldr (フォルダ一覧の出力)

機能

指定されたフォルダのリンク先 (直接型リンク) フォルダのパス情報、文書数、プロパティ情報を CSV 形式で出力します。出力内容については、「7.4.1 dbjlsfldr コマンドが出力するファイル」を参照してください。

直接型リンクでフォルダに包含されているフォルダの情報を再帰的に出力します。

形式

```
dbjlsfldr  -f {フォルダのOID | フォルダのクラス名}
           -p プロパティ名
           -o 結果出力ディレクトリ名
```

オプション

-f {フォルダのOID | フォルダのクラス名}

フォルダのOIDまたはフォルダのクラス名を指定します。

フォルダのOIDを指定した場合、指定したフォルダに包含されているフォルダの情報を再帰的に出力します。

フォルダのクラス名を指定した場合、指定したクラスに格納されているトップフォルダを起点に、そのトップフォルダに包含されているフォルダの情報を再帰的に出力します。

-p プロパティ名

フォルダ名を表すユーザプロパティのプロパティ名を指定します。このオプションに指定できるのは、ユーザプロパティだけです。システムプロパティを指定した場合は、エラーとなります。

指定できるプロパティのデータ型は、String型です。

各フォルダクラスで共通のフォルダ名を表すプロパティを指定してください。クラスに指定されたプロパティが存在しない場合、フォルダ名は "(no-data)" と出力されます。指定されたプロパティの値がNULL値の場合、フォルダ名は "(null)" と出力されます。

-o 結果出力ディレクトリ名

結果リストを出力するディレクトリのパスを、絶対パスで指定します。

注意事項

このコマンドを多重実行しないでください。

1つのフォルダ直下に登録 (直接型リンク) されているフォルダおよび文書の合計数が100万件を超えるフォルダが出力対象に含まれる場合は、コマンド実行中にフォルダや文書を削除しないでください。100万件を超えるフォルダ直下のフォルダや文書を削除した場合は、正しく情報が出力されないことがあります。フォルダおよび文書の合計数が100万件を超えるフォルダが出力対象に含まれる場合は、Webサーバを停止するなどして、ユーザがフォルダや文書を削除できない状態で、このコマンドを実行してください。

dbjlsdoc (文書一覧の出力)

機能

指定されたフォルダに登録されている文書の一覧を出力します。バージョン付き文書の場合、最新バージョンの情報を出力します。出力する内容については、「7.4.2 dbjlsdoc コマンドが出力するファイル」を参照してください。

形式

```
dbjlsdoc    -f フォルダのOIID
            -o 結果出力ディレクトリ名
            [-p プロパティ名]
            [-c {CH | DV}]
            [-t プロパティ名, {開始日時, 終了日時 |
                               開始日時 |
                               , 終了日時}]
            [-s]
```

オプション

- f フォルダの OIID
文書一覧を出力する文書が登録されたフォルダの OIID を指定します。
- o 結果出力ディレクトリ名
文書一覧の結果を出力するディレクトリ名を、絶対パスで指定します。
- p プロパティ名
フォルダ名を表すユーザプロパティのプロパティ名を指定します。このオプションに指定できるのは、ユーザプロパティだけです。システムプロパティを指定した場合は、エラーとなります。
指定できるプロパティのデータ型は、String 型です。
各フォルダクラスで共通のフォルダ名を表すプロパティを指定してください。クラスに指定されたプロパティが存在しない場合、フォルダ名は " (no-data) " で出力されます。
- c {CH | DV}
バージョン付き文書の場合、バージョンングオブジェクトとバージョンオブジェクトのどちらのプロパティ情報を出力するかを指定します。
 - CH : バージョニングオブジェクトのプロパティ情報を出力します。
 - DV : バージョンオブジェクトのプロパティ情報を出力します。
 このオプションを省略した場合は CH が仮定されます。
- t プロパティ名, {開始日時, 終了日時 | 開始日時 | , 終了日時}
日時を指定して文書を出力する場合に指定します。
開始日時および終了日時を「YYYYMMDDhhmmss」の形式で指定します。
 - YYYY : 西暦年号 (1970 ~ 2038 の範囲で指定してください。)
 - MM : 月
 - DD : 日
 - hh : 時
 - mm : 分
 - ss : 秒
 開始日時と終了日時の間のカンマ (,) の前後に余分なスペースを指定しないでください。
開始日時と終了日時のどちらかを必ず指定してください。開始日時を省略すると、終了日時以前の文

書だけ出力し、終了日時を省略すると、開始日時以降の文書だけ出力します。

プロパティ名には、文書クラスに定義されている日時を表すユーザプロパティのプロパティ名を指定します。

ユーザプロパティだけが指定でき、システムプロパティを指定した場合、エラーとなります。

指定できるプロパティのデータ型は、String 型、Integer 型です。

String 型プロパティが指定された場合、指定された日時の文字列をそのまま条件とします。Integer 型プロパティが指定された場合、指定された日時を通算秒に変換した値を条件とします。

各文書クラスで共通の日時を表すプロパティを指定してください。該当する文書クラスに指定されたプロパティが存在しない場合、指定した条件は無視されます。

-s

このオプションを指定した場合、直接型リンクされているフォルダに格納されている文書も再帰的に出力します。

このオプションを指定しなかった場合、直接型リンクされているフォルダに格納されている文書は出力しません。

注意事項

このコマンドを多重実行しないでください。

1 つのフォルダ直下に登録（直接型リンク）されているフォルダおよび文書の合計数が 100 万件を超えるフォルダが出力対象に含まれる場合は、コマンド実行中にフォルダや文書を削除しないでください。100 万件を超えるフォルダ直下のフォルダや文書を削除した場合は、正しく情報が出力されないことがあります。フォルダおよび文書の合計数が 100 万件を超えるフォルダが出力対象に含まれる場合は、Web サーバを停止するなどして、ユーザがフォルダや文書を削除できない状態で、このコマンドを実行してください。

7.4 コマンドが出力するファイル

7.4.1 dbjlsfldr コマンドが出力するファイル

dbjlsfldr コマンドを実行すると、次のファイルが出力されます。

- フォルダパス情報ファイル
- フォルダプロパティ情報ファイル

出力形式

フォルダパス情報ファイルの出力形式を次に示します。

- 1フォルダの情報が1行(レコード)で出力されます。CSV形式で出力されます。
- 各項目は、半角のコンマ「,」で区切って出力されます。
- 各項目の値は、ダブルクォーテーション(“)で囲んで出力されます。項目内にダブルクォーテーション(“)を含む場合は、ダブルクォーテーションが二つ重ねて出力されます。
- 文書空間の文字コードで出力されます。

それぞれのファイルの詳細を次に示します。

(1) フォルダパス情報ファイル

(a) ファイル名

ファイル名は「dbjlsfldr_<YYYYMMDDhhmmssxxx>.csv」の形式で出力されます。

- < YYYY > : 西暦年号
- < MM > : 月
- < DD > : 日
- < hh > : 時
- < mm > : 分
- < ss > : 秒
- < xxx > : ミリ秒

(b) 出力内容

フォルダパス情報ファイルの出力内容を次に示します。

表 7-4 フォルダパス情報ファイルの出力内容

項番	出力情報	説明
1	フォルダ OIID	フォルダの OIID が出力されます。
2	フォルダパス	フォルダの階層がフォルダパスで出力されます。フォルダパスは、フォルダを「/」で区切って出力されます。
3	文書数	フォルダにリンク(直接型リンク)されている文書数が出力されます。

(c) 出力例

フォルダパス情報ファイルの出力例を次に示します。

```
#OID, フォルダパス, 文書数
"dma:///07a17522-a626-...0000000101","フォルダ 01","0"
"dma:///07a17522-a626-...0000000102","フォルダ 01/フォルダ 02","100"
"dma:///07a17522-a626-...0000000105","フォルダ 01/フォルダ 02/フォルダ 05","5"
"dma:///07a17522-a626-...0000000103","フォルダ 01/フォルダ 03","200"
"dma:///07a17522-a626-...0000000104","フォルダ 01/フォルダ 04","10"
```

(2) フォルダプロパティ情報ファイル

(a) ファイル名

ファイル名は「dbjlsfldr_<クラス名>_<YYYYMMDDhhmmssxxx>.csv」の形式で出力されます。

- <クラス名> : フォルダが所属するクラスのクラス名
- < YYYY > : 西暦年号
- < MM > : 月
- < DD > : 日
- < hh > : 時
- < mm > : 分
- < ss > : 秒
- < xxx > : ミリ秒

(b) 出力内容

フォルダプロパティ情報ファイルの出力内容を次に示します。

表 7-5 フォルダプロパティ情報ファイルの出力内容

項番	出力情報	説明
1	フォルダ OID	フォルダの OID が出力されます。
2 ~ n	ユーザプロパティの値	フォルダクラスに定義されているユーザプロパティの値が出力されます。ただし、VariableArray 型プロパティおよび FreeWordIndex 型プロパティは出力されません。値が NULL 値の場合は、「」の形式で出力されます。

(c) 出力例

フォルダプロパティ情報ファイルの出力例を次に示します。

```
#OID,usrProp_Name, usrProp_CreateTime
"dma:///07a17522-a626-...0000000101","フォルダ 01","2012/12/12 08:00:00"
"dma:///07a17522-a626-...0000000102","フォルダ 02","2012/09/12 08:00:00"
"dma:///07a17522-a626-...0000000103","フォルダ 03","2012/01/16 10:00:00"
"dma:///07a17522-a626-...0000000104","フォルダ 04","2012/06/12 08:00:00"
```

7.4.2 dbjlsdoc コマンドが出力するファイル

dbjlsdoc コマンドを実行すると、次のファイルが出力されます。

- 文書一覧ファイル

出力形式

文書一覧ファイルの出力形式を次に示します。

- 1 文書の情報が 1 行 (レコード) で出力されます。CSV 形式で出力されます。

7. コマンドリファレンス

- 各項目は、半角のコンマ「,」で区切って出力されます。
- 各項目の値は、ダブルクォーテーション（"）で囲んで出力されます。項目内にダブルクォーテーション（"）を含む場合は、ダブルクォーテーションが二つ重ねて出力されます。
- 文書空間の文字コードで出力されます。

(1) 文書一覧ファイル

ファイル名は「dbjlsdoc_<クラス名>_<YYYYMMDDhhmmssxxx>.csv」の形式で出力されます。

- <クラス名> : 文書が所属するクラスのクラス名
- < YYYY > : 西暦年号
- < MM > : 月
- < DD > : 日
- < hh > : 時
- < mm > : 分
- < ss > : 秒
- < xxx > : ミリ秒

(a) 出力内容

文書一覧ファイルの出力内容を次に示します。

表 7-6 文書一覧ファイルの出力内容

項番	出力情報	説明
1	文書 OIID	文書の OIID が出力されます。
2	フォルダ OIID	文書が格納されているフォルダの OIID が出力されます。
3	フォルダパス	フォルダパスが出力されます。 フォルダパスは、指定されたフォルダのフォルダ名を先頭に「/」で区切って出力されます。 -p オプションを指定した場合だけ出力されます。 -p オプションを指定しない場合は、「""」の形式で出力されます。
4	レンディション数	文書のレンディション数が出力されます。
5	レンディションタイプ	文書のマスタレンディションのレンディションタイプが出力されます。
6	ファイル名	文書のマスタレンディションのファイル名が出力されます。
7	チェックアウトユーザ識別子	文書がチェックアウトされている場合、チェックアウトユーザ識別子が出力されます。 文書がチェックアウトされていない場合は、「""」の形式で出力されます。
8 ~ n	ユーザプロパティの値	文書クラスに定義されているユーザプロパティの値が出力されます。 ただし、VariableArray 型プロパティおよび FreeWordIndex 型プロパティは出力されません。 値が NULL 値の場合は、「""」の形式で出力されます。

(b) 出力例

出力例を次に示します。

```
#OID, フォルダ OID, フォルダパス, レンディション数, レンディションタイプ, ファイル名, チェックアウトユーザ識別子, usrProp_Name,usrProp_CreateTime  
"dma:///07a17522-a6-...0000000101","dma:///07a17522-a6-...0000000201","", "1", "application/  
msword", "a.doc", "", "user01", "2012/12/12 12:00:00"  
"dma:///07a17522-a6-...0000000102","dma:///07a17522-a6-...0000000201","", "1", "text/plain", "b.txt", "", "user02", "2012/12/  
12 12:00:00"  
"dma:///07a17522-a6-...0000000103","dma:///07a17522-a6-...0000000201","", "1", "application/msword", "c.doc"
```


付録

付録 A システムクラス, システムプロパティおよびデータベース定義の名称規則

付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス

付録 C DocumentBroker Version 5 へのバージョンアップ手順

付録 D このマニュアルの参考情報

付録 E 用語解説

付録 A システムクラス, システムプロパティおよびデータベース定義の名称規則

ここでは, システムクラスおよびシステムプロパティの名前, 表識別子, 列名, およびデータベース定義の名称規則を示します。

付録 A.1 システムクラスの名称定義の規則

システムクラスは, データベース定義での名称定義が決まっています。システムクラスに対応する表識別子を次の表に示します。

表 A-1 システムクラスの名称定義の規則

クラス名	表識別子
dmaClass_ConfigurationHistory	dmaClass_ConfigHistory
dmaClass_DirectContainmentRelationship	dmaClass_DCRelationship
dmaClass_ReferentialContainmentRelationship	dmaClass_RCRelationship
dmaClass_Rendition	マスタレンディションの場合 dmaClass_DocVersion の表識別子と同じ サブレンディションの場合 dmaClass_Rendition
dmaClass_Reservation	dmaClass_VersionSeries
dmaClass_VersionDescription	dmaClass_VerDescription
edmClass_ComponentDocVersion	edmClass_CompoDocVersion
edmClass_IndependentPersistence	edmClass_IndPersistence
edmClass_VersionTraceableContainer	edmClass_VTContainer
edmClass_VersionTraceableContainmentRelationship	edmClass_VTCRelationship
edmClass_VersionTracedComponentDocVersion	edmClass_VTCompoDocVersion
edmClass_VersionTracedDocVersion	edmClass_VTDocVersion
<ul style="list-style-type: none"> • dmaClass_Container • dmaClass_ContentTransfer • dmaClass_DocVersion • dmaClass_VersionSeries • edmClass_BindRelationship • edmClass_ContainerVersion • edmClass_ContentReference • edmClass_OIID • edmClass_PublicACL • edmClass_Relationship • edmClass_VTRelationship 	クラス名と同じ

付録 A.2 システムプロパティの名称の規則

システムプロパティは, データベース定義での名称定義が決まっています。システムプロパティに対応する表識別子を次の表に示します。

表 A-2 システムプロパティの名称定義の規則

プロパティ名	列名
dmaProp_ConfigurationHistory	dmaProp_ConfigHistory

プロパティ名	列名
dmaProp_CurrentOfSeriesCount	dmaProp_CurrentOfSeriesCnt
dmaProp_CurrentVersionDescription	dmaProp_CurrntVerDesc
edmProp_ACL	<ul style="list-style-type: none"> • edmProp_ACL_edmProp_Subj • edmProp_ACL_edmProp_Permis
edmProp_HeadVersionDescription	edmProp_HeadVerDescription
edmProp_HeadVTConfigurationHistory	edmProp_HeadVTConfigHistory
dmaProp_PrimaryVersionSeries	dmaProp_PrimVerSeries
edmProp_PrimaryGroupPermission	edmProp_PrimGrpPermission
edmProp_PublicACLIds	edmProp_PACLS_edmProp_ACL Elm
edmProp_SACL	<ul style="list-style-type: none"> • edmProp_SACL_edmProp_Subj • edmProp_SACL_edmProp_Permis
edmProp_TailVersionDescription	edmProp_TailVerDescription
edmProp_TailVTConfigurationHistory	edmProp_TailVTConfigHistory
edmProp_VTConfigurationHistory	edmProp_VTConfigHistory

プロパティ名	列名
<ul style="list-style-type: none"> • dmaProp_ComponentType • dmaProp_Head • dmaProp_IsCurrentVersion • dmaProp_NewVersion • dmaProp_OIID • dmaProp_Parent • dmaProp_ParentContainer • dmaProp_RenditionType • dmaProp_RetrievalName • dmaProp_Reservation • dmaProp_ReservedFor • dmaProp_Tail • dmaProp_This • dmaProp_Version • dmaProp_VersionSeries • edmProp_ConceptTextIndex • edmProp_ConceptStIndex • edmProp_ConfigurationTable • edmProp_Content • edmProp_ContentIndexStatus • edmProp_ContentLocation • edmProp_DocLength • edmProp_EveryonePermission • edmProp_HeadVTVersionSeries • edmProp_HeadVTMode • edmProp_OwnerId • edmProp_OwnerPermission • edmProp_Permission • edmProp_PrimaryGroupId • edmProp_ReferenceType • edmProp_RelationType • edmProp_RenditionsCount • edmProp_RenditionStatus • edmProp_RequiredClassId • edmProp_StIndex • edmProp_Subject • edmProp_TailVTVersionSeries • edmProp_TailVTMode • edmProp_TextIndex • edmProp_VTMode • edmProp_VTVersionSeries 	プロパティ名と同じ

また、システムプロパティは、次に示す規則にも従っています。

規則 1

HiRDB の繰り返し列を利用した VariableArray 型のプロパティの列名は、「4.9.2 サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則」の内容に従って生成します。

規則 2

列名には、_1 という番号が付く場合があります。この番号の意味は、列名に対応するプロパティが属するクラスが、何番目のクラスとして表に格納されているかを表します。番号が付くプロパティを持つクラスを次に示します。

1. マスタ dmaClass_Rendition クラス
"_1" が列名の最後に付きます。
2. dmaClass_Reservation クラス

"_1" が列名の最後に付きます。

詳細については、「4.9.2 サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則」を参照してください。

付録 A.3 データベース定義の名称定義の規則

ここでは、表識別子（表の名称として一意に識別できる値）、列名、インデクス名（表を検索するためのキーとして列に付けた索引の名称）、およびエリア名を定義する場合の規則を説明します。

クラス名やプロパティ名をデータベース定義の名称とする場合の名称定義の規則を次の表に示します。

表 A-3 データベース定義の名称定義の環境ごとの規則

データベース定義の名称定義の環境	クラス名やプロパティ名をデータベース定義の名称とする環境
表識別子	サブクラス名
列名	プロパティ名
列名（二つのクラスを一つの表に格納する場合）	プロパティ名 プロパティ名+'_1'
VariableArray 用のクラス（別表）	VariableArray プロパティ名
VariableArray 型のプロパティ（別表）	要素プロパティ名
VariableArray 型のプロパティ（HiRDB の繰り返し列）	VariableArray プロパティ名+'_'+要素プロパティ名
インデクス名 （キーが一つ、または指定省略時）	サブクラス名
インデクス名 （キーが複数、または指定省略時）	サブクラス名
サブクラス名で表識別子の制約	あり
プロパティ名で列名の制約	あり

サブクラス名に対応する表識別子とプロパティ名に対応する列名の詳細な規則については、「4.9.2 サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則」を参照してください。インデクス名を定義する場合の詳細な規則については、「4.11.2 インデクス名をデータベース定義の名称に使用する場合の規則」を参照してください。エリア名については、データベース定義の制約に従って定義してください。

付録 B ユーザ作成のアクセスルーチンを使用するためのインターフェースとクラス

ここでは、ユーザ管理システムのアクセスルーチンとして DocumentBroker が提供するインターフェースとクラスについて説明します。

ユーザ管理システムのアクセスルーチンは、DocumentBroker が提供する DbjAuthModule インターフェースを実装したクラスをユーザが作成し、そのクラス名を文書空間構成定義ファイル (docspace.properties) に指定します。

DocumentBroker が UOC で提供するインターフェースとクラスの一覧を次の表に示します。

表 B-1 DocumentBroker が UOC で提供するインターフェースとクラス

関数	機能
jp.co.Hitachi.soft.docbroker.auth.DbjAuthModule	ユーザ認証のインターフェースです。DocumentBroker のユーザ認証処理時に発行されます。
jp.co.Hitachi.soft.docbroker.auth.DbjAuthUserInfo	ユーザ情報の取得インターフェースです。
jp.co.Hitachi.soft.docbroker.auth.DbjAuthException	UOC 例外クラスです。UOC でエラーが発生した場合に詳細メッセージを取得します。

各インターフェース、およびクラスに属するメソッドの詳細では、次に示す項目について説明します。

機能

メソッドの概要を記述しています。

形式

メソッドの概要を記述しています。/* */ 内にコメントを記述しています。

機能説明

メソッドの機能の詳細を記述しています。

引数

引数について次のように記述しています。

引数 (入力)

指定する値についての説明

引数 (出力)

指定する領域の説明

指定した領域に設定される値の説明

- 引数は、「形式」で記述した文字列です。
- (入力) は、ユーザが値を指定することを示しています。
- (出力) は、ユーザが、DocumentBroker によって設定される値を格納する領域を指定することを示しています。

戻り値

メソッドを呼び出して、正常な場合に返される値、およびエラーが発生した場合に返される主な値を列挙しています。

注意事項

メソッドを使用する場合の注意事項を記述しています。

付録 B.1 jp.co.Hitachi.soft.docbroker.auth.DbjAuthModule インターフェイス

(1) initialize

機能

DbjAuthModule インターフェイスの初期化

形式

```
public void initialize() throws DbjAuthException
```

機能説明

DbjAuthModule インターフェイスを初期化します。このメソッドは、DbjFactory0200#getFactory メソッド実行時に発行されます。

引数

なし。

戻り値

なし。

注意事項

なし。

(2) login

機能

ユーザとパスワードのチェック

形式

```
public DbjAuthUserInfo login(
    StringuserName,
    StringpassWord,
    ) throws DbjAuthException
```

機能説明

引数で指定されたログインユーザ名とパスワードが正しいかどうかをチェックします。このメソッドは、DbjSession#login メソッド実行時に発行されます。

引数

userName (入力)

DbjSession#login メソッドで指定したユーザ名を指定します。

passWord (入力)

DbjSession#login メソッドで指定したパスワードを指定します。

戻り値

DbjAuthUserInfoインターフェイス

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

(3) commit

機能

ユーザ認証，およびユーザ情報の取得後の後処理

形式

```
public void commit() throws DbjAuthException
```

機能説明

ユーザ認証，およびユーザ情報の取得後の後処理をします。このメソッドは，DbjSession#login メソッド実行時に発行されます。

引数

なし。

戻り値

なし。

注意事項

このメソッドは，マルチスレッド環境で発行されます。そのため，スレッドセーフにしておく必要があります。

(4) abort

機能

ユーザ認証でエラーが発生した場合の後処理

形式

```
public void abort()
```

機能説明

ユーザ認証でエラーが発生した場合の後処理をします。このメソッドは，DbjAuthException がスローされた場合，ユーザ情報が不正な場合に発行されます。

引数

なし。

戻り値

なし。

注意事項

このメソッドは，マルチスレッド環境で発行されます。そのため，スレッドセーフにしておく必要があります。

(5) logout

機能

UOC セッションハンドルの破棄

形式

```
public void abort()
```

機能説明

UOC セッションハンドルを破棄します。このメソッドは，セッション破棄時にセッション内で 1 回だけ発行されます。このメソッドを発行したあとで，UOC セッションハンドルを使用してメソッドを発行しないでください。

引数

userInfo (入力)

DbjAuthModule#login メソッドで指定したユーザ名を指定します。

戻り値

なし。

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

付録 B.2 jp.co.Hitachi.soft.docbroker.auth.DbjAuthUserInfo インターフェイス

(1) getUserId

機能

ユーザ識別子の取得

形式

```
public String getUserId() throws DbjAuthException
```

機能説明

ユーザ識別子を取得します。

引数

なし。

戻り値

なし。

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

(2) getPrimaryGroupId

機能

プライマリグループ識別子の取得

形式

```
public String getPrimaryGroupId() throws DbjAuthException
```

機能説明

プライマリグループ識別子を取得します。

引数

なし。

戻り値

なし。

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

(3) getGroupIds

機能

グループ情報の取得

形式

```
public Collection<String> getGroupIds() throws DbjAuthException
```

機能説明

グループ情報を取得します。

引数

なし。

戻り値

なし。

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

付録 B.3 jp.co.Hitachi.soft.docbroker.auth.DbjAuthException クラス

(1) DbjAuthException

機能

コンストラクタ

形式

```
public DbjAuthException extends Exception(  
    String message
```

引数

message (入力)

詳細メッセージです。

戻り値

なし。

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

(2) getMessage

機能

詳細メッセージの取得

形式

```
public String getMessage()
```

引数

なし。

戻り値

詳細メッセージ

注意事項

このメソッドは、マルチスレッド環境で発行されます。そのため、スレッドセーフにしておく必要があります。

付録 C DocumentBroker Version 5 へのバージョンアップ手順

ここでは、DocumentBrokerVersion 3 以前のバージョンから、DocumentBrokerVersion 5 へバージョンアップする手順について説明します。バージョンアップ時には、HiRDB を起動させておいてください。HiRDB の起動方法については、マニュアル「HiRDB システム運用ガイド」を参照してください。

(1) 手動による定義などの修正

次の表に示す項目について、手動での定義の修正が必要となります。

表 C-1 手動で定義の修正が必要な項目

Version3 以前の定義ファイル	種類	設定項目	エントリ	Version 5 のプロパティファイル	プロパティ	
DocumentSpace 構成定義ファイル (docspace.ini)	サービス	文書空間	SerialId	文書空間構成定義プロパティファイル (docspace.properties)	DocSpaceId	
		ユーザ認証	ユーザ認証		UserAuthentication	UserAuthentication
			LDAP 設定		LdapHost	LdapHost
					LdapPort	LdapPort
	UOC ライブラリの設定	UOCLibrary	UOCClass			
	ユーザ情報の検索	検索条件			LdapPrefixDn	LdapPrefixDn
					LdapUserId	LdapUserId
					LdapUserRoot	LdapUserRoot
					LdapUserScope	LdapUserScope
					LdapUserClass	LdapUserClass
					LdapUserFilterLeft	LdapUserFilterLeft
					LdapUserFilterRight	LdapUserFilterRight
		検索オプション			LdapUserTimeout	LdapUserTimeout
	LdapUserCase				LdapUserCase	
	グループ情報の検索	ディレクトリエントリからの検索			LdapGroup	LdapGroup
					LdapGroupRoot	LdapGroupRoot
					LdapGroupScope	LdapGroupScope
					LdapGroupClass	LdapGroupClass
					LdapGroupId	LdapGroupId
					LdapGroupFilterLeft	LdapGroupFilterLeft
					LdapGroupFilterRight	LdapGroupFilterRight
					LdapGroupMember	LdapGroupMember
		ユーザの属性値からの検索			LdapGroupFromUserAttr	LdapGroupFromUserAttr
					LdapGroupIdFromUserAttr	LdapGroupIdFromUserAttr
					LdapGroupIsDnFromUserAttr	LdapGroupIsDnFromUserAttr

Version3 以前の定義ファイル	種類	設定項目	エントリ	Version 5 のプロパティファイル	プロパティ	
			LdapGroupIdAttrFromUserAttr		LdapGroupIdAttrFromUserAttr	
		ユーザの DN からの検索	LdapGroupFromUserDn		LdapGroupFromUserDn	
			LdapGroupIdFromUserDn		LdapGroupIdFromUserDn	
		検索オプション	LdapGroupTimeout		LdapGroupTimeout	
			LdapGroupCase		LdapGroupCase	
	ディレクトリサービスに対するバインド方法	ディレクトリサービスに対するバインド方法	LdapBindUserDN		LdapBindUserDN	
				LdapBindPassword		LdapBindPassword
	データベース	接続するデータベースの設定	DbType		DbType	
	HiRDB	接続設定	PdHost		ConnectorName	
				PdNamePort		リソースアダプタの設定
				PdUser		リソースアダプタの設定 (User)
			コネクションプールの設定	DBConnectionPoolCount		リソースアダプタの設定 (MaxPoolSize)
				DBConnectionPoolOver		リソースアダプタの設定 (RequestQueueEnable)
				DBConnectionPoolWaitTimeOut		リソースアダプタの設定 (RequestQueueTimeout)
			リファレンスファイル文書の格納	ReferenceStorageMode		ReferenceStorageMode
		ReferenceDirDivideNum			ReferenceDirDivideNum	
	BLOB データ	BLOB データの設定	BlobSubstrMode		未サポート	
		データサイズの設定	BlobSubstrThreshold		未サポート	
		BLOB データの取得先	BlobGettingMethod		未サポート	

Version3 以前の定義ファイル	種類	設定項目	エントリ	Version 5 のプロパティファイル	プロパティ
セキュリティ定義ファイル (docaccess.ini)	-	セキュリティ管理者	SecurityAdmin		SecurityAdmin
		ユーザ権限定義ファイルの名称	UserPermDefFile		UserPermDef
		オブジェクトの所有者に対するパーミッション	DefaultACFlagOwner		DefaultACFlagOwner
		グループに対するパーミッション	DefaultACFlagGroup		DefaultACFlagGroup
		すべてのユーザに対するパーミッション	DefaultACFlagEveryone		DefaultACFlagEveryone
ユーザ権限定義ファイル	-	-	Subject:SubjectType:パーミッション文字列	Subject=SubjectType:パーミッション文字列	
プロパティマッピング定義ファイル	-	-	-	未サポート	-
動作環境定義ファイル (conf.properties)	アプリケーショントレース	アプリケーショントレース情報出力先	APTracePath	動作環境定義ファイル (docbconf.properties)	APTracePath
		トレースレベル	APTraceLevel		APTraceLevel
		アプリケーショントレースファイルのファイルサイズの上限	APTraceSize		APTraceSize
		アプリケーショントレースファイルの数	APTraceNumber		APTraceNumber
		アプリケーションエラーログファイルのファイルサイズの上限	APErrorLogSize		APErrorLogSize
		アプリケーションエラーログファイルの数	APErrorLogNumber		APErrorLogNumber
	クライアント共用トレース	トレース情報出力先	PromptOutput	PromptOutput	

Version3 以前の定義ファイル	種類	設定項目	エントリ	Version 5 のプロパティファイル	プロパティ
		クライアント共用トレースファイル出力先ディレクトリ	TracePath		LogPath
		クライアント共用トレースファイルのファイルサイズの上限	TraceSize		TraceSize
		クライアント共用トレースファイルのファイルサイズの上限	TraceNumber		TraceNumber
環境変数	トレース	出力先ディレクトリ	_HIEDMS_TRACE_DIR		LogPath
		切り替えファイル数	_HIEDMS_TRACE_NUM		TraceNumber
		ファイルサイズ	_HIEDMS_TRACE_SIZE		TraceSize
		トレースレベル	_HIEDMS_TRACE_LEVEL		TraceLevel

次の定義ファイルは、廃止となりました。

- ファイル転送サービス環境定義ファイル (ftpsv.ini)
- サービスプロセス定義ファイル
- ネットワークリソース情報ファイル (netaccess.ini)
- クライアントアプリケーション動作定義ファイル (application.ini)
- クラス定義情報ファイル

また、次の表に示すエントリは廃止となりました。

表 C-2 廃止となったエントリ

定義ファイル	種類	設定項目	エントリ
DocumentSpace 構成定義ファイル (docspace.ini)	文書空間の共通定義	文書空間の数	Count
		ORB および BOA のオプション	DocSpaceOrbBoaOption
		VisiBroker プロパティ	DocSpaceVBProperty
		エラーログ	ErrLogFileCount
	ErrLogFileSize		
	サービス	プロセス設定	Process
			SessionMax
			SessionTimeOut
ProcessOrbBoaOption			

定義ファイル	種類	設定項目	エントリ	
			ProcessVBProperty	
	ユーザ認証	LDAP 設定	LdapClientLib	
	HiRDB	コネクションプールの設定	DBConnectionPoolDynamic	
			DBConnectionPoolTiming	
	DBConnectionScope			
		HiRDB Text Search Plug-in の設定	PdTSPluginOwner	
	アクセスログ	アクセスログの設定	AcLogUse	
			AcLogLevel	
			AcLogFileCount	
			AcLogFileSize	
	メタ情報	共有メモリの設定	XdkShmemManage	
			XdkShmemSize	
	ファイル分割転送機能	データ転送サイズの設定	FtpBufferSize	
	複数の実行環境	サーバの選択	SelectServerInMultiServer	
	オブジェクト操作	VariableArray 型のプロパティでの要素のチェック	VArrayElementCheck	
		オブジェクト操作時のエラーチェックのフラグ	ErrChkFlagOfObjectOperation	
		オブジェクト操作時の動作のフラグ	EnbFncFlagOfObjectOperation	
	DocumentBroker Life Cycle Suite	WorkCoordinator 連携の設定	LifeCycleSuiteConnection	
動作環境定義ファイル (conf.properties)	-	デフォルトの文書空間識別子	DefaultDocSpaceId	
		オブジェクトを操作したときの動作	EnableFunctionFlag	
		文書空間で使用する文字コード種別	DocSpaceCharacterSet	
環境変数	-	-	DOCBROKERDIR	
		トレース	保存日数	_HIEDMS_TRACE_DAYS
		保存日数	_HIEDMS_TRACE_DAYS_C LT	
	ファイル転送	-	-	_HIEDMS_FTPDIR
				_HIEDMS_FTPMODE
				_HIEDMS_FTPORBBOA_OP TION
				_HIEDMS_FTPVB_PROPER TY
				_HIEDMS_CON_TIMEOUT
EDMLckWac her	-	-	_HIEDMS_LCKINF_DIR	

定義ファイル	種類	設定項目	エントリ
			_HIEDMS_LCKWATCH_TIME
	EDMUsrView	-	_HIEDMS_USRVIEW_TIME OUT
	-	-	_HIEDMS_EXCEPTION_FILTER
	-	-	DBJ_CONF_PATH

(2) メタ情報用のテーブルを定義します

DocumentBroker のメタ情報用のテーブルやインデクスを作成します。テーブルやインデクスは、テーブル定義用の SQL ファイルを使用して作成します。

(a) SQL ファイルの編集

メタ情報用のテーブルやインデクスは、次に示す SQL ファイルを編集して作成します。

UNIX の場合

```
/opt/DocBroker/Platform/sql/createmetatable_hirdb.sql
```

Windows の場合

```
<インストールディレクトリ>\DocBroker\Platform\sql\createmetatable_hirdb.sql
```

SQL ファイル中の文字列を、テキストエディタを使用して書き換えます。

テーブルやインデクスを作成するために SQL ファイルの書き換えが必要な文字列、および書き換える内容を次の表に示します。

表 C-3 SQL ファイルの書き換えが必要な内容

書き換えが必要な文字列	書き換える内容
(META)	「2.5.3 ユーザ LOB 用 RD エリアの容量の見積もり」で見積もりした DocumentBroker のメタ情報を格納するユーザ表用 RD エリア名に書き換えします。
(METAINDEX)	「2.5.3 ユーザ LOB 用 RD エリアの容量の見積もり」で見積もりした DocumentBroker のメタ情報を格納するユーザインデクス用 RD エリア名に書き換えします。

(b) SQL ファイルの実行

編集した SQL ファイルを入力ファイルとして、HiRDB でデータベース定義ユティリティ (pddef) を実行して定義します。HiRDB のデータベース定義ユティリティについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

(3) メタ情報の出力コマンド (dbjprintmeta) を実行します

DocumentBroker Version 3 のメタ情報を、DocumentBroker Version 5 のメタ情報として登録するために、メタ情報の出力コマンド (dbjprintmeta -v) を実行します。メタ情報の出力コマンド (dbjprintmeta) の使用方法については、「7.3 コマンドの文法」を参照してください。

(4) メタ情報の初期設定コマンド (dbjinitmeta) を実行します

メタ情報の出力コマンド (dbjprintmeta) で出力したメタ情報をデータベースに登録するために、メタ情

報の初期設定コマンド (dbjinitmeta) を実行します。このとき、-f オプションに「(3) メタ情報の出力コマンド (dbjprintmeta) を実行します」で出力したディレクトリを指定します。メタ情報の初期設定コマンドの使用方法については、「7.3 コマンドの文法」を参照してください。

(5) 実行環境の情報の登録コマンド (dbjregenvid) を実行します

複数の実行環境を登録していた場合、dbjinitmeta を実行した環境以外で実行環境の登録コマンド (dbjregenvid -r) を実行します。実行環境の登録コマンド (dbjregenvid) の使用方法については、「7.3 コマンドの文法」を参照してください。

(6) DocumentBroker Version 3 のメタ情報を削除します

DocumentBroker Version 3 のメタ情報を削除するために、次に示す SQL ファイルを入力ファイルとしたデータベース定義文を入力ファイルとして、HiRDB でデータベース定義ユティリティ (pddef) を実行して定義を追加します。

UNIX の場合

```
/opt/DocBroker/Platform/sql/dropemetatable_v3_hirdb.sql
```

Windows の場合

```
<インストールディレクトリ>%sql%\dropemetatable_v3_hirdb.sql
```

HiRDB のデータベース定義ユティリティについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

付録 D 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 D.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。

なお、本文に記載のマニュアル名称は、「uCosminexus DocumentBroker」を「DocumentBroker」と表記しています。

DocumentBroker のマニュアル

- uCosminexus DocumentBroker Version 5 概説 (3021-3-402)
DocumentBroker の機能について知りたい場合に参照してください。
- uCosminexus DocumentBroker Version 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 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 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 Text Search Index Loader の機能、環境設定、およびコマンドの文法について知りたい場合に参照してください。

関連製品のマニュアル (HiRDB)

- HiRDB データ連動機能 HiRDB Datareplicator Version 8 (3020-6-360)
- ノンストップデータベース HiRDB Version 9 システム導入・設計ガイド (UNIX(R) 用) (3000-6-452)
- ノンストップデータベース HiRDB Version 9 システム導入・設計ガイド (Windows(R) 用) (3020-6-452)
- ノンストップデータベース HiRDB Version 9 システム定義 (UNIX(R) 用) (3000-6-453)
- ノンストップデータベース HiRDB Version 9 システム定義 (Windows(R) 用) (3020-6-453)
- ノンストップデータベース HiRDB Version 9 システム運用ガイド (UNIX(R) 用) (3000-6-454)
- ノンストップデータベース HiRDB Version 9 システム運用ガイド (Windows(R) 用) (3020-6-454)
- ノンストップデータベース HiRDB Version 9 コマンドリファレンス (UNIX(R) 用) (3000-6-455)
- ノンストップデータベース HiRDB Version 9 コマンドリファレンス (Windows(R) 用) (3020-6-455)
- ノンストップデータベース HiRDB Version 9 UAP 開発ガイド (3020-6-456)
- ノンストップデータベース HiRDB Version 9 SQL リファレンス (3020-6-457)
- HiRDB XML 拡張機能 HiRDB XML Extension Version 9 (3020-6-480)
- HiRDB 全文検索プラグイン HiRDB Text Search Plug-in Version 9 (3020-6-481)

関連製品のマニュアル（その他）

- Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド（3020-3-Y02）
- Cosminexus V9 アプリケーションサーバ アプリケーション設定操作ガイド（3020-3-Y13）
- Cosminexus V9 アプリケーションサーバ リファレンス 定義編（サーバ定義）（3020-3-Y16）
- Cosminexus V9 アプリケーションサーバ リファレンス 定義編（アプリケーション/リソース定義）（3020-3-Y17）

関連マニュアルの略称

このマニュアルで使用する関連マニュアルの略称を次に示します。

マニュアル名	略称
HiRDB Datareplicator Version 8	HiRDB Datareplicator
HiRDB Version 9 システム導入・設計ガイド（UNIX(R)用）	HiRDB システム導入・設計ガイド
HiRDB Version 9 システム導入・設計ガイド（Windows(R)用）	
HiRDB Version 9 システム定義（UNIX(R)用）	HiRDB システム定義
HiRDB Version 9 システム定義（Windows(R)用）	
HiRDB Version 9 システム運用ガイド	HiRDB システム運用ガイド
HiRDB Version 9 コマンドリファレンス（UNIX(R)用）	HiRDB コマンドリファレンス
HiRDB Version 9 コマンドリファレンス（Windows(R)用）	
HiRDB Version 9 UAP 開発ガイド	HiRDB UAP 開発ガイド
HiRDB Version 9 SQL リファレンス	HiRDB SQL リファレンス
HiRDB XML Extension Version 9	HiRDB XML Extension
HiRDB 全文検索プラグイン HiRDB Text Search Plug-in Version 9	HiRDB HiRDB Text Search Plug-in

付録 D.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

製品名称	略称
HiRDB Text Search Plug-in Version 9	HiRDB Text Search Plug-in
HiRDB XML Extension Version 9	HiRDB XML Extension
HiRDB/Run Time Version 9	HiRDB/Run Time
Preprocessing Library for Text Search Version 2	Preprocessing Library for Text Search
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。

- HiRDB の製品群を総称して HiRDB と表記します。
- Linux を UNIX と表記することがあります。
- それぞれの製品について個別に説明する場合は、製品名称を使用します。
- Oracle Directory Server Enterprise Edition, および Active Directory を合わせて LDAP 対応のディレクトリサービスと表記することがあります。
- リレーショナルデータベース管理システムをデータベースシステムと表記します。

付録 D.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
ASCII	American Standard Code for Information Interchange
BLOB	Binary Large Object
CD-ROM	Compact Disc Read Only Memory
CIFS	Common Internet File System
CR	Carriage Return
CSV	Comma Separated Value
DAT	Digital Audio Tape
DB	Database
DBMS	Database Management System
DCD	Document Content Description
DIT	Directory Information Tree
DLL	Dynamic Linking Library
DMA	Document Management Alliance
DN	Distinguished Name
GIF	Graphics Interchange Format
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol

英略語	英字での表記
ID	Identifier
ISO	International Organization for Standardization
JIS	Japanese Industrial Standards
LDAP	Lightweight Directory Access Protocol
LF	Line Feed
MIME	Multipurpose Internet Mail Extensions
OIID	Object Instance Identifier
OS	Operating System
PC	Personal Computer
PDF	Portable Document Format
RDB	Relational Database
RDN	Relative Distinguished Name
SGML	Standard Generalized Markup Language
UOC	User Own Coding
UTF-8	8-bit UCS Transformation Format
WWW	World Wide Web
XML	Extensible Markup Language

付録 D.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 E 用語解説

DocumentBroker で使用する用語について説明します。

(英字)

ACE (Access Control Element)

アクセス制御エレメントのことです。

ACFlag (Access Control Flag)

アクセス制御フラグのことです。

ACL (Access Control List)

アクセス制御リストのことです。

AND-NOT 検索

検索オペレータに「AND-NOT」を使用する検索方法です。二つの検索条件を AND-NOT でつないで、左側のオペランドに指定した検索条件は成立するが、右側のオペランドに指定した検索条件は成立しない文書を検索します。例えば、「著者が『日立太郎』であるが、所属は『日立製作所』ではない文書を検索する」というような場合に使用できます。

AND 検索

検索オペレータに「AND」を使用して、検索条件同士の論理積を求める検索方法です。例えば、「著者が『日立太郎』で、文書中に『コンピュータ』という文字列を含む文書を検索する」というような場合に使用できます。

API (Application Programming Interface)

アプリケーションプログラムとのインターフェースを指します。

Child

直接型のリンク (DirectContainment) の場合、オブジェクト (Parent) に包含されるオブジェクトを指します。

ConfigurationHistory オブジェクト

バージョン管理に使用する最上位の DocumentBroker オブジェクトです。文書のバージョンを管理する VersionSeries オブジェクトを管理します。

Containable オブジェクト

フォルダの包含要素になるオブジェクトです。CdbcrContainable クラスのサブクラスを基に作成されたオブジェクトを指します。

Containeer

参照型のリンク (ReferentialContainment) の場合、オブジェクト (Container) に包含されるオブジェクトを指します。

Container

参照型のリンクの場合、オブジェクト (Containee) を包含するオブジェクトを指します。

ContainerVersion オブジェクト

包含しているオブジェクト全体を一つの概念的なオブジェクトとして管理する DocumentBroker オブジェクトです。

Container オブジェクト

リンクを利用してオブジェクトを管理する場合に、オブジェクトを包含する DocumentBroker オブジェクトの一つです。dmaClass_Container クラスおよびユーザが定義した dmaClass_Container クラスのサブクラスを基に作成します。

ContentElement (dmaClass_ContentElement クラス)

文書のコンテンツにアクセスするために使われる DocumentBroker オブジェクトの抽象クラスです。ContentElement の二つのサブクラス (dmaClass_ContentTransfer クラス, dmaClass_ContentReference クラス) は, コンテンツの格納とアクセス機能を提供します。

ContentReference オブジェクト

文書のコンテンツを管理するために使用する DocumentBroker オブジェクトです。コンテンツをデータベースに格納しないで, 位置情報を永続化して管理する場合に使用します。dmaClass_ContentElement クラスのサブクラスである dmaClass_ContentReference クラスを基に作成します。

ContentTransfer オブジェクト

文書のコンテンツを管理するために使用する DocumentBroker オブジェクトです。コンテンツをデータベースに格納 (永続化) して管理する場合に使用します。dmaClass_ContentElement クラスのサブクラスである dmaClass_ContentTransfer クラスを基に作成します。

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。DMA で規定されたクラス (クラス名が dmaClass_ で始まるクラス), DocumentBroker で拡張したクラス (クラス名が edmClass_ で始まるクラス), およびこれらのクラスからユーザが定義したサブクラスを基に作成されます。

DocVersion オブジェクト

DocumentBroker で扱う文書に相当する DocumentBroker オブジェクトです。dmaClass_DocVersion クラスおよびユーザが定義した dmaClass_DocVersion クラスのサブクラスを基に作成します。

edmSQL

DocumentBroker のオブジェクトを検索するための検索条件式を表現するための文法です。SQL の文法に基づいています。

edmSQL 検索

検索条件に, SQL ライクの文法で記述できる edmSQL 文を指定して実行する検索のことです。

GUID (Globally Unique Identifier)

DocumentBroker のクラス, プロパティ, 検索オペレータなどに与えるユニークな識別子です。GUID は, 「X」を 0 ~ 9 および a ~ f (小文字) で表される 16 進数とした 「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8-4-4-4-12)」の形式で表されます。

LDAP (Lightweight Directory Access Protocol)

TCP/IP 上で動作する解放型 DAP を提供し, X.500 のデータモデルを保持しています。

NOT 条件

指定したキーワードとの不一致を求める検索条件です。例えば, 「作成者が『日立』ではない文書を検索する」というような場合に使用できます。

OIID (Object Instance Identifier)

文書空間での永続オブジェクトの存在や格納位置などを明確にするために使用する識別子です。

OR 検索

検索オペレータに「OR」を使用して, 検索条件同士の論理和を求める検索方法です。例えば, 「作成者が『日立太郎』であるオブジェクトか, 作成者の所属が『日立製作所』であるオブジェクトを検索する」というような場合に使用できます。

Parent

直接型のリンクの場合, オブジェクト (Child) を包含するオブジェクトを指します。

RD エリア

データベースの表、インデックスおよびデータディクショナリを格納するデータ領域のことです。データの格納単位の一つで、1 ~ 16 個の HiRDB ファイルから構成されます。

Rendition オブジェクト

文書の表現形式 (HTML, PDF など) を管理する DocumentBroker オブジェクトです。

Reservation オブジェクト

新しいバージョンを作成する権利を VersionSeries オブジェクトに予約する DocumentBroker オブジェクトです。新しいバージョンのチェックインが完了すると、このオブジェクトは削除されて、VersionDescription オブジェクトに置き換えられます。

UOC (User Own Coding)

ユーザによって作成されたプログラムのことです。DocumentBroker では、ユーザ認証や、アクセス制御機能のためのユーザ情報取得に UOC を使用できます。

VariableArray 型

プロパティの基本単位の一つです。基本単位が VariableArray 型であるプロパティは、データ型に従った複数の値を可変長な一次元配列として持ちます。また配列の要素は、構造体で管理できます。

Versionable オブジェクト

バージョン管理の対象になるオブジェクトです。

VersionDescription オブジェクト

文書のバージョン (更新履歴) を管理するために使用する DocumentBroker オブジェクトです。VersionSeries オブジェクトと Versionable オブジェクトを接続するために使用します。

VersionSeries オブジェクト

文書のバージョン (更新履歴) を管理するために使用する DocumentBroker オブジェクトです。連続的な履歴を持つバージョンの構成を保持するオブジェクトです。

VersionTracedComponentDocVersion オブジェクト

直接型、参照型のリンクでフォルダに包含される対象となるためのプロパティを持つ DocumentBroker オブジェクトです。dmaClass_DocVersion クラスのサブクラスの edmClass_VersionTracedComponentDocVersion クラスを基に作成します。

VersionTracedDocVersion オブジェクト

直接型、参照型のリンクで、フォルダに包含される対象となるためのプロパティを持つ DocumentBroker オブジェクトです。dmaClass_DocVersion クラスのサブクラスの edmClass_VersionTracedDocVersion クラスを基に作成します。

VTContainee

オブジェクト (VTContainer) に包含されるオブジェクトを指します。

VTContainer

オブジェクト (VTContainee) を包含するオブジェクトを指します。

W3C (World Wide Web Consortium)

HTML など、WWW に関する技術の標準化を推進する非営利団体です。

(ア行)

アクセス権

オブジェクトを作成したり、すでに作成されているオブジェクトにアクセスしたりする権利です。

アクセス制御エレメント (ACE : Access Control Element)

アクセス制御リスト (ACL) の要素です。一つのサブジェクトと一つのパーミッションの組で構成され、指定されたサブジェクトに対して指定されたパーミッションの範囲のアクセス権を与えることを示す情報です。

アクセス制御機能

DocumentBroker の文書空間でのオブジェクトの作成や、管理されている文書やフォルダなどのオブジェクトに対する操作を、ユーザやグループごとに許可または制限する機能です。

アクセス制御機能付き検索

アクセス制御機能を利用した文書空間で検索を実行した場合に、ユーザにアクセス権がないオブジェクトは検索結果として取得しない検索です。

アクセス制御情報

アクセス制御されている文書空間で、ユーザがメソッドを発行する際に、アクセス権の判定に使用される情報です。

アクセス制御情報変更権

オブジェクトに設定されているアクセス制御情報 (ACFlag および ACL) を変更する権利です。また、パブリック ACL をアクセス制御対象オブジェクトにバインドすることを許可する権利も含まれます。なお、パブリック ACL のアクセス情報変更権には、パブリック ACL のユーザ定義プロパティを変更する権利が含まれます。

アクセス制御フラグ (ACFlag : Access Control Flag)

オブジェクトの所有者、プライマリグループおよび全ユーザという区分でパーミッションを設定できるアクセス制御情報の一つです。

アクセス制御モデル

アクセス制御機能を利用して運用されている文書管理モデルです。

アクセス制御リスト (ACL : Access Control List)

任意のユーザまたはグループにパーミッションを設定できるアクセス制御情報の一つです。アクセス制御エレメント (ACE) のリストで構成されます。

アンバインド

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方法です。例えば、検索タームとして「バイオリン」を指定した場合に、「ヴァイオリン」という検索タームの異表記を含む文書も検索できます。

インデクス情報ファイル

ユーザが追加するプロパティにインデクスを定義する場合に、定義するインデクスの情報を記述するファイルです。

永続オブジェクト

データベースに格納されたオブジェクトを指します。

永続プロパティ

データベースに存在するプロパティを指します。

オブジェクト作成権

オブジェクト作成権限で、文書空間にオブジェクトを作成する権利を与えるパーミッションです。ユーザ権限定義ファイルに指定します。

オブジェクト作成権限

文書空間にオブジェクトを作成する権限で、ユーザ権限の一つです。ユーザ権限定義ファイルで定義します。オブジェクト作成権限を与えられたユーザおよびグループに属するユーザは、オブジェクトを作成するメソッドを実行できます。

オブジェクト操作権限

文書空間内のすべてのオブジェクトを、与えられた権限の範囲で操作する権利で、ユーザ権限の一つです。ユーザ権限定義ファイルで定義します。例えば、オブジェクト操作権限としてプロパティ参照権を与えられたユーザおよびグループに属するユーザは、文書空間内のすべてのオブジェクトのプロパティを参照できます。

オブジェクトリファレンス

DocumentBroker オブジェクトへのリファレンスを示す Object 型プロパティの値です。例えば、dmaProp_ParentContainer プロパティの値がこれに当たります。

(力行)

概念検索

ユーザが任意に指定した文章や文字列を手がかりにして、その条件と似た概念を持つ文書を検索する方法です。全文検索の一種です。概念検索で指定する条件のことを種文章といいます。

仮のバージョン識別子

チェックアウト中の仮のバージョンを識別するための識別子です。チェックアウト中のオブジェクトを参照・更新する時に使用します。この識別子はチェックアウト時に DocumentBroker によって設定される識別子であり、仮のバージョンに該当するオブジェクトの OIID とは異なります。

関連オブジェクト

リンクの型 (直接型、間接型) を定義するための DocumentBroker オブジェクトです。直接型のリンクの場合、オブジェクトを包含するオブジェクト (Parent) とオブジェクトに包含されるオブジェクト (Child) を DirectContainmentRelationship オブジェクトを使用して関連づけます。参照型のリンクの場合、オブジェクトを包含するオブジェクト (Container) とオブジェクトに包含されるオブジェクト (Containee) を ReferentialContainmentRelationship オブジェクトを使用して関連づけます。

基本コンテンツ更新権

基本パーミッションの一つで、オブジェクトのコンテンツを更新する権利を与えるパーミッションです。基本プロパティ参照権を含みます。文書に対して設定する場合は、全文検索インデックスを作成、削除する権利を含みます。

基本コンテンツ参照権

基本パーミッションの一つで、オブジェクトのコンテンツを参照する権利を与えるパーミッションです。基本プロパティ参照権を含みます。文書に対して設定する場合は、全文検索を実行する権利を含みます。

基本削除権

基本パーミッションの一つで、オブジェクトを削除する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本単位

DMA で規定されているプロパティは、データ型に従った値を 1 個持つか複数個持つかが決められています。これを基本単位といいます。

基本バージョン管理権

基本パーミッションの一つで、バージョン管理されているオブジェクトのバージョンを追加、削除する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本パーミッション

ユーザおよびグループがオブジェクトに対して実行できる操作の範囲を定めるパーミッションの基本単位です。オブ

ジェクト操作権限、アクセス制御フラグおよびアクセス制御エレメントで、ユーザおよびグループに許可する操作の範囲を定める場合に使用します。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る）。基本パーミッションには、基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権、基本コンテンツ更新権、基本リンク権、基本バージョン管理権および基本削除権があります。

基本プロパティ更新権

基本パーミッションの一つで、オブジェクトのプロパティを更新する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

基本プロパティ参照権

基本パーミッションの一つで、オブジェクトのプロパティを参照する権利を与えるパーミッションです。そのほかのすべての基本パーミッションに含まれます。フォルダに対して設定する場合は、関連オブジェクトのユーザ定義プロパティの参照と管理されている要素を参照する権利を含みます。

基本リンク権

基本パーミッションの一つで、リンクの設定、変更および関連オブジェクトのユーザ定義プロパティを変更する権利を与えるパーミッションです。基本プロパティ参照権を含みます。

近傍条件検索

検索オペレータに「Prox」を使用する拡張検索（全文検索）です。指定する検索タームが同時に存在する場合に、その検索ターム間の距離を条件として検索します。例えば、「『文書管理』という検索タームと『ドキュメント』という検索タームを含み、これらの検索タームがこのとおりの順序で出現し、かつ検索ターム間に入る文字が5文字以内である文書を検索する」というような場合に使用できます。

組み合わせパーミッション

基本パーミッションを複数組み合わせた権利を与えるパーミッションの単位です。アクセス制御フラグおよびアクセス制御エレメントでユーザおよびグループに許可する操作の範囲を定めるときに使用します。組み合わせパーミッションには、プロパティ参照権、参照権、プロパティ更新権、参照更新権、削除権、リンク権、バージョン管理権およびフルコントロールがあります。例えば、あるユーザに対して、ある文書の参照更新権という組み合わせパーミッションを設定すると、そのユーザは基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権および基本コンテンツ更新権が設定されたのと同じ範囲の操作を、その文書に対して実行できます。

継承 (Inheritance)

既存のクラスを利用して新しいクラスを定義するオブジェクト指向の技術です。

コンテンツ

一般的には、属性に対する文書のデータ部分を指します。DMA では、ある文書に関連づけられていて、DMA で規定されている Content モデルに従ってアクセスされるオブジェクトの集合を指します。また、アクセスされるオブジェクトの実体（例えば、report.doc、document.htm など）をコンテンツデータといいます。

コンテンツ格納先パス

リファレンスファイル管理機能を使用する場合に、コンテンツの格納先として、コンテンツ格納先ベースパスからの相対パスで指定するディレクトリパスです。コンテンツ格納先パスで指定したディレクトリの下に、DocumentBroker がコンテンツを管理するためのディレクトリが作成され、その下にコンテンツが格納されます。コンテンツ格納先パスに指定するディレクトリは、DocumentBroker によって作成されます。

コンテンツロケーション

リファレンスファイル管理機能を使用する場合に、データベースで管理される、コンテンツの格納先の情報です。コンテンツ格納先ベースパスからの相対パスがコンテンツロケーションとしてデータベースに登録されます。

(サ行)

サーババインド

DocumentBroker から LDAP 対応のディレクトリサービスに対して、DN とパスワードを使用して認証を行い、バインドすることです。

削除権

組み合わせパーミッションの一つです。基本削除権と同じ操作を許可するパーミッションです。

サブクラス

あるクラスから派生するクラスのことです。または、それ自身がサブクラスとして参照されているクラスのことです。

サブジェクト (Subject)

アクセス権を与えるユーザまたはグループです。

サブジェクト種別 (SubjectType)

アクセス権を与えるサブジェクトが、ユーザなのか、グループなのかまたはシステムなのかを識別するための情報です。

サブレンディション

マルチレンディション文書に、追加登録されたレンディションのことです。マスタレンディション以外のレンディションを指します。なお、サブレンディションは、登録後にマスタレンディションに変更できます。

参照型のリンク (ReferentialContainment)

フォルダを利用した文書管理の方法です。参照型のリンクで使われるフォルダは、文書にはり付けるインデックスの働きをします。

参照権

組み合わせパーミッションの一つです。基本コンテンツ参照権と同じ操作を許可するパーミッションです。

参照更新権

組み合わせパーミッションの一つです。基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権および基本コンテンツ更新権を組み合わせたパーミッションです。すなわち、参照更新権を設定することで、プロパティを参照、更新する権利とコンテンツを参照、更新する権利を設定できます。

状態フラグ

マルチレンディション文書のマスタレンディションに対するサブレンディションのコンテンツの状態を表すフラグです。マスタレンディションとサブレンディションのコンテンツの状態が一致している、マスタレンディションのコンテンツが更新されたのに対してサブレンディションのコンテンツが更新されていない、またはサブレンディションのコンテンツが存在しない、という 3 種類の状態が表示されます。dbrProp_RenditionStatus プロパティの下位 2 バイトに設定されます。

所有者 (Owner)

オブジェクトの所有者として設定されているユーザです。アクセス制御フラグでパーミッションを与えられます。所有者に設定されているユーザは、そのオブジェクトのアクセス制御フラグで所有者に与えられたパーミッションの範囲の操作をそのオブジェクトに対して実行できます。また、そのオブジェクトの所有者およびセキュリティ ACL の値を変更できます。

スーパークラス

あるクラスのクラス定義に使われたクラスを、派生したクラスのスーパークラスといいます。

セキュリティ ACL

オブジェクトに設定されたアクセス制御情報へのアクセスを制御するためのアクセス制御リストです。任意のユーザまたはグループにアクセス制御情報変更権を設定できます。

セキュリティ管理者

アクセス制御機能を利用した文書空間で、アクセス判定を受けることなく、すべてのオブジェクトに自由にアクセスする特権を持ち、文書空間のすべてのオブジェクトを保守するユーザです。セキュリティ定義ファイルに定義します。

セッション

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。

全文検索

文書に含まれるキーワードを条件（全文検索条件）として、キーワードを含む文書を検索する方法です。

全文検索インデクス

全文検索の対象になるテキストデータに対応するプロパティです。edmProp_TextIndex プロパティ、edmProp_StIndex プロパティ、edmProp_ConceptTextIndex プロパティ、edmProp_ConceptStIndex プロパティおよび edmProp_Content プロパティに相当します。

全文検索機能付き文書クラス

全文検索の対象となる文書を作成するためのクラスです。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、ユーザ定義のクラスです。

(夕行)

チェックアウト (check-out)

文書またはフォルダにバージョンを追加するために、次バージョンの追加を予約して、最新バージョンのコピーを要求することです。

チェックイン (check-in)

文書またはフォルダのバージョンの追加を確定することです。

直接型のリンク (DirectContainment)

フォルダを利用した文書管理の方法です。直接型のリンクで使われるフォルダは、文書を格納するフォルダの働きをします。

定義情報ファイル

サブクラスおよびプロパティを追加するときに、追加するオブジェクトの定義情報を記述するファイルです。

ディレクトリサービス

ネットワーク上にあるユーザや組織の情報などの資源とその属性を記憶し、検索できるようにしたシステムです。DocumentBroker では、SecureWay 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。

同義語展開検索

全文検索条件として指定する検索タームまたは検索タームの同義語を含む文書を検索する方法です。例えば、検索タームとして「パソコン」を指定した場合に、「電子計算機」、「パーソナルコンピュータ」、「PC」など、検索タームと同じ意味を持つ単語を含む文書も検索できます。

動作環境メタ情報ファイル

DocumentBroker を起動するときに参照するメタ情報です。実行環境ディレクトリ /env_info (UNIX の場合)、または実行環境ディレクトリ %env_info (Windows の場合) に格納されます。

匿名バインド

DocumentBroker から LDAP 対応のディレクトリサービスに対して、匿名ユーザで認証を行い、バインドすることです。

独立データ

ほかのオブジェクトに依存しない、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空間オブジェクトです。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空間オブジェクトです。

特権

アクセス制御機能を利用した文書空間で、アクセス判定を受けることなく、すべてのオブジェクトに自由にアクセスする権利です。セキュリティ定義ファイルにセキュリティ管理者として定義されたユーザに与えられます。特権の有無は、ログイン時にセキュリティ定義ファイルが参照され、ログインユーザごとに作成されるユーザ情報に保持されます。

トップオブジェクト

DocumentBroker クラスライブラリのオブジェクトを構成する複数の DocumentBroker オブジェクトのうち、最上位に位置するオブジェクトです。例えば、バージョンなし文書の場合は、DocumentBroker オブジェクトの DocVersion オブジェクトがトップオブジェクトです。

(八行)

バージョン管理権

組み合わせパーミッションの一つです。基本プロパティ参照権、基本プロパティ更新権、基本コンテンツ参照権、基本コンテンツ更新権および基本バージョン管理権を組み合わせたパーミッションです。

バージョン付き文書

複数のバージョンを保持できる文書を表すオブジェクトです。DocumentBroker オブジェクトでは、ConfigurationHistory オブジェクトに相当します。

バージョンなしフォルダ

バージョンを保持しないフォルダを表すオブジェクトです。DocumentBroker オブジェクトでは、Container オブジェクト、および ContainerVersion オブジェクトに相当します。

バージョンなし文書

バージョン管理しない文書を表すオブジェクトです。DocumentBroker オブジェクトでは、DocVersion オブジェクトに相当します。

パーミッション

オブジェクトの作成、オブジェクトのプロパティ参照、オブジェクトのコンテンツ更新などの実行可能な操作の範囲を表す値です。オブジェクト作成権限を与えるパーミッション、オブジェクトの操作の範囲を定めるパーミッションがあります。オブジェクトの操作の範囲を定めるパーミッションには、基本パーミッションと組み合わせパーミッションがあります。

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。

また、LDAP 対応のディレクトリサービスに対して、DN とパスワードを使用して認証を行うことも、バインドといいます。

パブリック ACL

文書空間にオブジェクトとして存在するアクセス制御リスト (ACL) です。複数のオブジェクトが共有できます。

非ナル値制約

非ナル値制約とは、列の値にナル値を許さない制約のことです。

複合データ

複数の異なる型によって表されるデータです。主に、VariableArray 型のプロパティとして設定されている Object 型の要素の値を参照、設定するときに使用します。

フォルダ

リンクでオブジェクトを包含できるオブジェクトの総称です。また、DocumentBroker オブジェクトとしては、Container オブジェクト、VersionTraceableContainer オブジェクトおよび ContainerVersion オブジェクトがあります。

プライマリグループ

アクセス制御フラグ (ACFlag) でパーミッションを与えるグループです。

フルコントロール

組み合わせパーミッションの一つです。すべての基本パーミッションを組み合わせたパーミッションです。オブジェクトに対するすべての操作を許可します。

プロパティ更新権

組み合わせパーミッションの一つです。基本プロパティ更新権と同じ操作を許可するパーミッションです。

プロパティ参照権

組み合わせパーミッションの一つです。基本プロパティ参照権と同じ操作を許可するパーミッションです。

文書

DocumentBroker オブジェクトでは、dmaClass_DocVersion クラスおよびそのサブクラスを基に作成するオブジェクトのことです。コンテンツを保持できます。

文書間リンク (Relationship)

文書と文書の関連づけです。

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。

文書空間構成定義ファイル (docspace.properties)

文書空間の構成を定義するために使用するファイルです。

変換フラグ

マルチレンディション文書のサブレンディションのフォルダを、レンディション変換の対象にするかどうかを表すフラグです。レンディション変換機能を使用してレンディション変換を実行する場合に使用します。また、レンディション変換機能によるレンディション変換でエラーが発生した場合には、エラーを示すフラグとしても使われます。dbrProp_RenditionStatus プロパティの上位 2 バイトに設定されます。

(マ行)

マスタレンディション

マルチレンディション文書に登録されたレンディションのうち、主要なレンディションのことです。マルチレンディション文書を参照・更新する場合には、レンディション形式を指定しますが、レンディション形式を指定しないときは、マスタレンディションが対象になります。なお、マスタレンディションとして扱うレンディションは、登録後に変更できません。

マルチレンディション機能

一つの文書に、同一内容の複数の異なる形式のフォルダを登録する機能です。

マルチレンディション文書

複数のレンディションを登録している文書のことです。一つと同じ内容を表す複数の形式のフォルダを保持する文書です。バージョンなし文書またはバージョン付き文書は、マルチレンディション文書として扱えます。

メタ情報ファイル

DocumentBroker が利用する DocumentBroker のクラス、プロパティ、検索オペレータなどの詳細情報を定義した

ファイルです。クラスおよびプロパティを追加、変更する場合に使用できます。

メタデータ (metadata)

クラス、プロパティおよびオペレータに関する詳細情報を定義するデータです。

(ヤ行)

ユーザ権限

文書空間にオブジェクトを作成する権利 (オブジェクト作成権限) と、文書空間内のすべてのオブジェクトに対する操作の範囲 (オブジェクト操作権限) をユーザまたはグループ単位で定めるアクセス制御情報の一つです。ユーザ権限定義ファイルに定義します。ユーザ権限の内容は、ログイン時にユーザ権限定義ファイルが参照され、ログインユーザごとに作成されるユーザ情報に保持されます。

ユーザ権限定義ファイル

ユーザ権限 (オブジェクト作成権限およびオブジェクト操作権限) を定義するためのファイルです。

ユーザ情報

ログインユーザのユーザ識別子、所属グループ、特権およびユーザ権限を表す情報です。ログイン時にユーザごとに生成され、アクセス権の判定に使用されます。

(ラ行)

リファレンスファイル管理機能

DocumentBroker が存在するマシンから接続可能なファイルシステムの任意のディレクトリで文書のコンテンツを管理し、文書のプロパティおよびコンテンツの格納先の情報をデータベースで管理する機能です。

リファレンスファイル文書

DocumentBroker が存在するマシンから接続可能なファイルシステムの任意のディレクトリに格納されているファイルをコンテンツとして持つ文書のことで、データベースでは、文書のプロパティとコンテンツの格納先の情報を管理しています。バージョンなし文書または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扱えます。

リレーション

DocumentBroker オブジェクトの Relationship オブジェクトに相当します。文書と文書間の参照関係を表すオブジェクトです。

リンク

DocumentBroker オブジェクトの DirectContainmentRelationship オブジェクト、ReferentialContainmentRelationship オブジェクトまたは VersionTraceableContainmentRelationship オブジェクトによって表される、文書とフォルダの関連づけです。

リンク権

組み合わせパーミッションの一つです。基本リンク権と同じ操作を許可するパーミッションです。

リンク識別子

フォルダオブジェクトへのリンク (関連づけ) を識別するための識別子です。リンクを解除したり、リンクのプロパティを参照または更新したりする時に使用します。この識別子は関連づけをした時に DocumentBroker によって設定される識別子です。同じフォルダオブジェクトに、同じ要素を 2 度関連づけした場合は、それぞれ異なるリンク識別子が設定されます。

レンディション

文書のコンテンツの形式およびそのコンテンツをあわせてレンディションと呼びます。DocumentBroker オブジェクトの Rendition オブジェクトおよび ContentTransfer オブジェクトに相当します。

レンディションタイプ

Word などのアプリケーションで編集したファイル，HTML 形式のファイル，GIF などの画像データのファイルのように，登録した文書のコンテンツのファイルの形式を表す文字列です。レンディションごとに設定できます。

DocumentBroker では，レンディションタイプとして，MIME 名を指定することを推奨しています。

レンディション変換機能

マルチレンディション文書の，マスタレンディションのフォルダの文書形式を変換して，サブレンディションのフォルダを作成，登録することです。

ローカル ACL

オブジェクトごとに設定できるアクセス制御リスト (ACL) です。VariableArray 型のプロパティとして設定されます。

ロケール (Locale)

言語や使用する文字コードの種別，特定の国や地域で特別な意味を持つ属性などの定義のことです。ロケールは地域化した形でアプリケーションの拡張性を提供するために使われます。

索引

A

ACE 192
ACFlag 192
ACL 192
AND-NOT 検索 192
AND 検索 12, 192
API 192

B

BinaryAccessSize プロパティ 95

C

Child 192
ConfigurationHistory オブジェクト 192
ConnectorName プロパティ 87
Containable オブジェクト 192
Containeer 192
Container 192
ContainerVersion オブジェクト 192
Container オブジェクト 192
ContentElement 193
ContentReference オブジェクト 193
ContentTransfer オブジェクト 193

D

databaseinfo.properties 106
dbjaddmeta 153
dbjchgmta 154
dbjchgusrid 160
dbjcrtsql 155
dbjdelmeta 156
dbjgenguid 156
dbjinitmeta 157
dbjlsdoc 164
dbjlsfldr 163
dbjprintmeta 158
dbjregenvid 158
DBType プロパティ 87
DefaultACFlagEveryone プロパティ 93
DefaultACFlagGroup プロパティ 93
DefaultACFlagOwner プロパティ 93
DIT 51
DN 51
DOCB_CONF_DIR 54
dochmetainit.properties 105

DocSpaceId プロパティ 87
DocumentBroker オブジェクト 193
DocumentBroker 実行環境の情報の登録 158
DocumentBroker の環境変数の設定 54
DocumentBroker の起動方法 130
DocumentBroker の機能 2
DocumentBroker のシステム構成 14
DocumentBroker の終了方法 131
DocumentBroker 用データベース定義文の作成 155
DocVersion オブジェクト 193

E

edmProp_ConceptTextIndex プロパティ 23
edmProp_ContentIndexStatus プロパティ 23
edmProp_DocLength プロパティ 23
edmProp_TextIndex プロパティ 23
edmSQL 193
edmSQL 検索 193

G

GUID 193
GUID の生成 156

H

HiRDB Text Search Plug-in , HiRDB XML
Extension での環境設定 49

I

IT Report Utility を利用した保守情報の採取 147

L

LDAP 193
LdapBindPassword プロパティ 92
LdapBindUserDN プロパティ 92
LdapGroupCase プロパティ 91
LdapGroupClass プロパティ 90
LdapGroupFilterLeft プロパティ 90
LdapGroupFilterRight プロパティ 90
LdapGroupFromGroupDn プロパティ 91
LdapGroupFromGroupMember プロパティ 92
LdapGroupFromUserAttr プロパティ 90
LdapGroupFromUserDn プロパティ 91
LdapGroupIdAttrFromUserAttr プロパティ 91
LdapGroupIdFromUserAttr プロパティ 90

LdapGroupIdFromUserDn プロパティ 91
 LdapGroupId プロパティ 90
 LdapGroupIsDnFromUserAttr プロパティ 90
 LdapGroupMember プロパティ 90
 LdapGroupRoot プロパティ 89
 LdapGroupScope プロパティ 89
 LdapGroupTimeout プロパティ 91
 LdapGroup プロパティ 89
 LdapHost プロパティ 88
 LdapPort プロパティ 88
 LdapUserCase プロパティ 88
 LdapUserClass プロパティ 89
 LdapUserFilterLeft プロパティ 89
 LdapUserFilterRight プロパティ 89
 LdapUserId プロパティ 88
 LdapUserRoot プロパティ 89
 LdapUserScope プロパティ 89
 LdapUserTimeout プロパティ 88
 LDAP 対応のディレクトリサービスによるユーザ管理
 機能を使用する場合 50
 LDAP 対応のディレクトリサービスのユーザ認証 52

M

MaxContentSize プロパティ 94
 MaxFullTextSize プロパティ 95
 mime.properties 104

N

NOT 検索 12
 NOT 条件 193

O

OIID 193
 OR 検索 12, 193

P

Parent 193
 PATH 54

R

RD エリア 194
 RD エリア定義情報ファイル 118
 ReferenceBasePath プロパティ 95
 ReferenceDirDivideNum プロパティ 95
 ReferenceStorageMode プロパティ 95
 Rendition オブジェクト 194
 Reservation オブジェクト 194

S

SecurityAdmin プロパティ 92

T

TempDirectory プロパティ 95

U

UOC 53, 194
 UOCClass プロパティ 88
 UserAuthentication プロパティ 87
 userperm.properties 102
 UserPermDef プロパティ 93

V

VariableArray 型 194
 Versionable オブジェクト 194
 VersionDescription オブジェクト 194
 VersionSeries オブジェクト 194
 VersionTracedComponentDocVersion オブジェクト
 194
 VersionTracedDocVersion オブジェクト 194
 VTContainee 194
 VTContainer 194

W

W3C 194

あ

アクセス権 195
 アクセス制御エレメント 195
 アクセス制御機能 9, 195
 アクセス制御機能付き検索 195
 アクセス制御機能に関連するデータベース容量の見積
 もり 35
 アクセス制御情報 195
 アクセス制御情報変更権 195
 アクセス制御フラグ 195
 アクセス制御モデル 195
 アクセス制御リスト 195
 アンインストール 47
 アンバインド 195

い

異表記展開検索 195
 異表記展開検索〔文書に対する全文検索〕 11

異表記展開検索〔文字列型プロパティに対する全文検索〕 12
 インスタンス 19
 インストールとアンインストール (UNIX の場合) 42
 インストールとアンインストール (Windows の場合) 45
 インデクス情報ファイル 121, 195
 インデクス情報ファイルの記述例 122
 インデクス名をデータベース定義の名称に使用する場合の規則 121

え

永続オブジェクト 195
 永続プロパティ 195
 エントリ 51

お

オブジェクト作成権 195
 オブジェクト作成権限 102, 196
 オブジェクト操作権限 102, 196
 オブジェクトリファレンス 196

か

概念検索 11, 12, 196
 仮想メモリ所要量の見積もり 26
 仮のバージョン識別子 196
 環境設定 40
 環境設定の準備 41
 環境設定の流れ 40
 関連オブジェクト 196

き

基本コンテンツ更新権 196
 基本コンテンツ参照権 196
 基本削除権 196
 基本単位 113, 196
 基本バージョン管理権 196
 基本パーミッション 93, 196
 基本プロパティ更新権 197
 基本プロパティ参照権 197
 基本リンク権 197
 近傍条件検索 197
 近傍条件検索〔文書に対する全文検索〕 11
 近傍条件検索〔文字列型プロパティに対する全文検索〕 12

く

組み合わせパーミッション 93, 197
 クラスおよびプロパティの追加例 24
 グループ識別子 52, 53, 55

け

継承 197
 検索機能 11
 検索ターム〔文書に対する全文検索〕 11
 検索ターム〔文字列型プロパティに対する全文検索〕 12

こ

コマンド一覧 150
 コマンドの形式 151
 コマンドの文法 153
 コンテンツ 197
 コンテンツ格納先パス 73, 197
 コンテンツ格納先ベースパス 6, 73
 コンテンツロケーション 6, 197

さ

サーババインド 198
 削除権 198
 サブクラス 19, 198
 サブクラスの追加 20
 サブクラス名およびプロパティ名をデータベース定義の名称に使用する場合の規則 108
 サブクラス名の規則 108
 サブジェクト 198
 サブジェクト種別 198
 サブレンディション 5, 198
 参照型のリンク 198
 参照権 198
 参照更新権 198

し

システムクラス, システムプロパティおよびデータベース定義の名称規則 172
 システムクラスの名称定義の規則 172
 システムプロパティの名称の規則 172
 実行環境識別子 13
 ジャーナルファイルの運用 137
 出力されるデータベース定義文 62
 障害対策 145
 状態フラグ 198
 所有者 198

す

スーパークラス 198

せ

セキュリティ ACL 198
セキュリティ管理者 199
セッション 199
全文検索 199
全文検索インデクス 199
全文検索インデクス一括登録機能 13
全文検索機能 11
全文検索機能付き文書クラス 199
全文検索機能付き文書クラスの追加 22
全文検索機能付き文字列型プロパティ 12

そ

相対識別名 51
属性検索機能 11

た

種文章 12

ち

チェックアウト 3, 199
チェックイン 3, 199
直接型のリンク 199

つ

追加するクラスおよびプロパティの検討 19

て

定義情報ファイル 108, 199
ディレクトリエントリ 51
ディレクトリサービス 199
ディレクトリサービスの情報管理の概念 50
データ型の対応 21
データベース運用上の注意事項 136
データベース情報ファイル 106
データベース定義の名称定義の規則 175
データベースの運用 134
データベースの文字コード種別の設定 25
データベース容量の見積もり 27
データベース容量の見積もり方法 27
適用業務の検討 18

と

同義語展開検索 199
同義語展開検索〔文書に対する全文検索〕 11
同義語展開検索〔文字列型プロパティに対する全文検索〕 12
動作環境メタ情報ファイル 199
登録済みメタ情報の出力 158
匿名バインド 199
独立データ 9, 200
独立データの管理機能 9
特権 200
トップオブジェクト 200

は

バージョン 2
バージョン管理権 200
バージョン付き文書 3, 200
バージョンなしフォルダ 200
バージョンなし文書 2, 200
パーミッション 200
バインド 200
バックアップとリストア 134
バックアップとリストアの方法 134
バックアップの取得 137
パブリック ACL 200

ひ

非ナル値制約 200

ふ

ファイルシステムのバックアップ 144
ファイルシステムのリストア 144
フォルダ 3, 201
フォルダ一覧の出力 163
フォルダ管理機能 3
複合データ 200
複数の実行環境機能 13
プライマリグループ 201
フルコントロール 201
プロパティ更新権 201
プロパティ参照権 201
プロパティの追加 21
プロパティ名の規則 108
プロパティを追加する場合の注意事項 22
文書 201
文書一覧の出力 164
文書間リンク 6, 201
文書間リンク機能 6

文書空間 201
 文書空間構成定義ファイル 85, 201
 文書空間構成定義ファイルの内容を変更する 100
 文書空間で使用する文字コード種別の検討 25
 文書空間の文字コード種別の設定 25
 文書の属性情報の管理機能 4
 文書の登録機能とバージョン管理機能 2

へ

変換フラグ 201
 変更情報ファイル 123

ま

マスタレンディション 5, 201
 マルチレンディション管理機能 4
 マルチレンディション機能 201
 マルチレンディション文書 5, 201

め

メタ情報初期化ファイル 105
 メタ情報の削除 156
 メタ情報の初期設定 157
 メタ情報の追加 153
 メタ情報の変更 154
 メタ情報ファイル 107, 201
 メタデータ 202

ゆ

ユーザ/グループの一括変更 160
 ユーザLOB用RDエリアの容量の見積もり 32
 ユーザ管理機能の設定 50
 ユーザ権限 102, 202
 ユーザ権限定義ファイル 102, 202
 ユーザ権限定義ファイルの設定 59
 ユーザ作成のアクセスルーチンを使用する場合 53
 ユーザ識別子 52, 53, 55
 ユーザ情報 52, 54, 202
 ユーザ用RDエリアの容量の見積もり 28

り

リストアの方法 137
 リファレンスファイル文書 202
 リファレンスファイル管理機能 6, 202
 リファレンスファイル管理機能を使用する場合の運用
 143
 リファレンスファイル管理機能を使用する場合の設定
 72

リファレンスファイル管理機能を使用する場合のファ
 イルシステムのディスク容量の見積もり 37
 リファレンスファイル文書 6
 リレーション 202
 リンク先文書 6
 リンク元文書 6
 リンク 202
 リンク権 202
 リンク識別子 202

れ

列名 172
 レプリケーション 137
 レンディション 5, 202
 レンディションタイプ 203
 レンディション定義ファイル 104
 レンディションのコンテンツ種別変換機能 7
 レンディション変換機能 203

ろ

ローカルACL 203
 ロケール 203