
uCosminexus DocumentBroker
Version 3
Standard GUI Development Kit

文法書

3020-3-U75

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参
照ください。

対象製品

R-1595C-43 uCosminexus DocumentBroker Version 3 Standard GUI Development Kit 03-60 (適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 R2 (x64) , Windows Server 2008 x86 , Windows XP Professional)

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

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

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

発行

2009 年 4 月 (第 1 版) 3020-3-U75

著作権

All Rights Reserved. Copyright (C) 2009, Hitachi Systems & Services, Ltd.

はじめに

このマニュアルは、次に示すプログラムプロダクトで提供する Java クラスライブラリの詳細、インターフェースの詳細、メソッドの文法、およびメッセージの詳細についてリファレンス形式で説明したものです。

- R-1595C-43 uCosminexus DocumentBroker Version 3 Standard GUI Development Kit

対象読者

このマニュアルは、uCosminexus DocumentBroker Version 3 Standard GUI Development Kit が提供する Java クラスライブラリを使用して、ユーザアプリケーションプログラムを開発し、システムを構築、運用および管理する方を対象にしています。なお、次の内容を理解されていることを前提としています。

- Windows に関する知識
- Java™ 言語に関する知識
- XML に関する知識
- SQL 言語に関する知識

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 DocumentBroker Standard GUI Development Kit の概要

uCosminexus DocumentBroker Standard GUI Development Kit で実現できる機能の概要、および前提プログラムについて説明しています。

第 2 章 DocumentBroker Standard GUI Development Kit 提供のクラス一覧

uCosminexus DocumentBroker Standard GUI Development Kit で提供するクラスの一覧について説明しています。

第 3 章 共通操作パッケージ提供クラスの詳細

共通操作パッケージ (jp.co.hitachi_system.sContents.API.Common) 提供のクラス、およびメソッドについて説明しています。

第 4 章 文書操作パッケージ提供クラスの詳細

文書操作パッケージ (jp.co.hitachi_system.sContents.API.Document) 提供のクラス、およびメソッドについて説明しています。

第 5 章 フォルダ操作パッケージ提供クラスの詳細

フォルダ操作パッケージ (jp.co.hitachi_system.sContents.API.Folder) 提供のクラス、およびメソッドについて説明しています。

第 6 章 検索操作パッケージ提供クラスの詳細

検索操作パッケージ (jp.co.hitachi_system.sContents.API.Select) 提供のクラス、およびメソッドについて説明しています。

はじめに

第 7 章 文書種別マスタパッケージ提供クラスの詳細

文書種別マスタパッケージ (jp.co.hitachi_system.sContents.API.DocMaster) 提供のクラス、およびメソッドについて説明しています。

第 8 章 DocumentBroker で提供されるクラスライブラリの詳細

Java クラスライブラリのクラス、およびメソッドについて説明しています。

第 9 章 コーディング例

クラスライブラリを使用したコーディング例について説明しています。

第 10 章 DocumentBroker Standard GUI Development Kit の環境構築

uCosminexus DocumentBroker Standard GUI Development Kit のインストール、および環境構築手順について説明しています。

付録 A DocumentBroker Standard GUI Development Kit で使用するプロパティ一覧

uCosminexus DocumentBroker Standard GUI Development Kit で扱うことができるプロパティについて説明しています。

付録 B 用語解説

uCosminexus DocumentBroker Standard GUI Development Kit の用語について説明しています。

関連マニュアル

このマニュアルは次のマニュアルと関連がありますので、必要に応じてお読みください。

uCosminexus DocumentBroker Standard GUI のマニュアル

uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド
(3020-3-U73)

uCosminexus DocumentBroker Standard GUI の環境設定、および運用方法について知りたい場合に参照してください。

uCosminexus DocumentBroker Version 3 Standard GUI 操作ガイド (3020-3-U74)

uCosminexus DocumentBroker Standard GUI の GUI の操作方法などについて知りたい場合に参照してください。

関連するプログラムのマニュアル

uCosminexus DocumentBroker Version 3 クラスライブラリ Java 解説 (3000-3-F16)

Java クラスライブラリの機能および環境設定のしかたについて知りたい場合に参照してください。

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

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

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

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

Cosminexus システム構築ガイド (3020-3-M06)

Cosminexus を使用して構築するアプリケーションサーバの構築方法について知りたい場合に参照してください。

読書手順

このマニュアルは、利用目的に合わせて章を選択してお読みいただけます。次に示す表を参考にして、お読みになる章を選択してください。

このマニュアルを読む目的	記述箇所
DocumentBroker Standard GUI Development Kit 概要、および前提条件について知りたい	1 章
どのような Java クラスライブラリのクラス、インターフェースおよびメソッドがあるかを知りたい、また、クラス、インターフェースおよびメソッドの説明形式について知りたい	2 章
共通パッケージクラスのインターフェースおよびメソッドについて知りたい	3 章
文書操作クラスのインターフェースおよびメソッドについて知りたい	4 章
フォルダ操作クラスのインターフェースおよびメソッドについて知りたい	5 章
検索操作クラスのインターフェースおよびメソッドについて知りたい	6 章
文書種別マスタ操作クラスについて知りたい	7 章
DocumentBroker Standard GUI Development Kit で使用する DocumentBroker のクラス、およびメソッドについて知りたい	8 章
コーディング例について知りたい	9 章
DocumentBroker Standard GUI Development Kit のインストール、および環境構築について知りたい	10 章
DocumentBroker Standard GUI Development Kit で使用するプロパティについて知りたい	付録 A
DocumentBroker Standard GUI Development Kit の用語について知りたい	付録 B

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ように表記しています。

正式名称	略称
HiRDB/Single Server Version 6	HiRDB
HiRDB/Single Server Version 7	
HiRDB/Single Server Version 8	
HiRDB/Parallel Server Version 6	
HiRDB/Parallel Server Version 7	
HiRDB/Parallel Server Version 8	
Java(TM)	Java

はじめに

正式名称	略称	
Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版	Windows Server 2003 Enterprise Edition	Windows Server 2003
Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版	Windows Server 2003 Standard Edition	
Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition 日本語版	Windows Server 2003 R2, Enterprise Edition	Windows Server 2003 R2
Microsoft(R) Windows Server(R) 2003 R2, Standard Edition 日本語版	Windows Server 2003 R2, Standard Edition	
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition 日本語版	Windows Server 2003 R2, Enterprise Edition (x64)	Windows Server 2003 R2 (x64)
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition 日本語版	Windows Server 2003 R2, Standard x64 Edition (x64)	
Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版	Windows Server 2008 Enterprise Edition x86	Windows Server 2008 , または Windows Server 2008 x86
Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版	Windows Server 2008 Standard Edition x86	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP Professional	Windows XP
uCosminexus Application Server Enterprise Version 7	uCosminexus Application Server または Cosminexus	
uCosminexus Application Server Enterprise Version 8		
uCosminexus Application Server Standard Version 7		
uCosminexus Application Server Standard Version 8		
uCosminexus Developer Professional Version 7		
uCosminexus Developer Professional Version 8		
uCosminexus Developer Standard Version 7		
uCosminexus Developer Standard Version 8		
uCosminexus DocumentBroker Development Kit Version 3	DocumentBroker Development Kit	DocumentBroker
uCosminexus DocumentBroker Runtime Version 3	DocumentBroker Runtime	
uCosminexus DocumentBroker Server Version 3	DocumentBroker Server , DocumentBroker サーバまたは , 文書管理サーバ	

正式名称	略称
uCosminexus DocumentBroker Text Search Index Loader Version 2	DocumentBroker Text Search Index Loader
uCosminexus DocumentBroker Version 3 Standard GUI	DocumentBroker Standard GUI
uCosminexus DocumentBroker Version 3 Standard GUI Development Kit	DocumentBroker Standard GUI Development Kit
uCosminexus DocumentBroker Version 3 オブジェクト操作ツール	オブジェクト操作ツール
DocumentBroker Rendering Option Version 3	DocumentBroker Rendering Option
DocumentBroker イメージ登録 for DocuCentre(R)	DocumentBroker イメージ登録 for DocuCentre

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。

- JavaTM を Java と表記します。
- uCosminexus DocumentBroker Development Kit、および uCosminexus DocumentBroker Runtime が提供する Java 言語対応のクラスライブラリを Java クラスライブラリと表記します。
- Windows Server 2003、Windows Server 2003 R2、Windows Server 2003 R2 (x64)、Windows Server 2008 x86、および Windows XP を合わせて Windows と表記することがあります。

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
ACL	<u>A</u> ccess <u>C</u> ontrol <u>L</u> ist
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
DB	<u>D</u> atabase
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
GUID	<u>G</u> lobally <u>U</u> nique <u>I</u> dentifier
ID	<u>I</u> dentifier
J2EE	<u>J</u> ava <u>2</u> <u>E</u> nterprise <u>E</u> dition
OIID	<u>O</u> bject <u>I</u> nstance <u>I</u> dentifier
OS	<u>O</u> perating <u>S</u> ystem
PC	<u>P</u> ersonal <u>C</u> omputer
PDF	<u>P</u> ortable <u>D</u> ocument <u>F</u> ormat
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage

英略語	英字での表記
UCS-2	U <u>n</u> iversal multi-o <u>c</u> tet C <u>h</u> aracter S <u>e</u> t 2
UCS-4	U <u>n</u> iversal multi-o <u>c</u> tet C <u>h</u> aracter S <u>e</u> t 4
UTF-8	8-bit U <u>C</u> S T <u>r</u> ansformation F <u>o</u> rmat
XML	E <u>x</u> tensible M <u>a</u> rku <u>p</u> L <u>a</u> ngu <u>a</u> ge

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。
...	記述が省略されていることを示します。 (例) ABC... ABC の後ろに記述があり、その記述が省略されていることを示します。
...	この記号の直前の項目を繰り返し、複数個指定できることを示します。 (例) A ... A を複数個指定できることを示します。

このマニュアルで使用する構文要素記号

このマニュアルで使用する構文要素の種類を次に示します。

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

注 すべての半角文字を使用してください。

常用漢字以外の漢字の使用について

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

個所（かしよ）

KB（キロバイト）などの単位表記について

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

目次

1	DocumentBroker Standard GUI Development Kit の概要	1
1.1	DocumentBroker Standard GUI Development Kit とは	2
1.2	DocumentBroker Standard GUI Development Kit で実現する機能	3
1.2.1	共通処理機能	3
1.2.2	文書操作機能	3
1.2.3	フォルダ操作機能	3
1.2.4	検索操作機能	4
1.2.5	文書種別マスタの操作機能	4
1.3	DocumentBroker Standard GUI Development Kit のシステム構成	5
1.3.1	DocumentBroker Standard GUI Development Kit のシステム構成例	5
1.4	DocumentBroker Standard GUI Development Kit の前提プログラム	7
1.4.1	DocumentBroker Standard GUI Development Kit の前提 OS	7
1.4.2	DocumentBroker Standard GUI Development Kit の同一装置内の前提ソフトウェア	7
1.4.3	DocumentBroker Standard GUI Development Kit のシステム内の前提ソフトウェア	8
1.5	DocumentBroker Standard GUI Development Kit のパッケージ一覧	10
1.6	インターフェースの処理の流れ	11
1.7	コーディング時の注意事項	14
1.7.1	マルチスレッド環境での注意事項	14
1.7.2	使用できる文字コード種別	14
1.8	DocumentBroker Standard GUI Development Kit で使用するユーザ種別	15
1.9	トランザクションの開始と終了	16
2	DocumentBroker Standard GUI Development Kit 提供のクラス一覧	17
2.1	DocumentBroker Standard GUI Development Kit パッケージが提供するクラス	18
2.2	共通操作パッケージのクラス	20
2.2.1	共通操作パッケージのメソッド一覧	20
2.3	共通操作パッケージの例外クラス	24
2.3.1	共通操作パッケージの例外クラス	24
2.4	文書操作パッケージ提供クラス	26
2.4.1	文書操作パッケージのクラス	26
2.4.2	文書操作パッケージのメソッド一覧	26
2.5	フォルダ操作パッケージ提供クラス	29
2.5.1	フォルダ操作パッケージのクラス一覧	29

2.5.2	フォルダ操作パッケージのメソッド一覧	29
2.6	検索操作パッケージ提供クラス	32
2.6.1	検索操作パッケージのクラス一覧	32
2.6.2	検索操作パッケージのメソッド一覧	32
2.7	文書種別マスタ操作パッケージ提供クラス	35
2.7.1	文書種別マスタ操作パッケージのクラス一覧	35
2.7.2	文書種別マスタ操作パッケージのメソッド一覧	35
2.8	パッケージ提供クラス，およびメソッドの説明形式	37
2.8.1	メソッドで説明する項目	37
2.9	メソッドの関係	39
2.9.1	文書空間オブジェクトの操作時に影響する setter メソッド	39
2.9.2	文書空間オブジェクト操作時に必要なアクセス権限	41
2.9.3	文書の状態によるメソッドの実行可否	43

3

	共通パッケージ提供クラスの詳細	45
3.1	DecmFactory クラス	46
3.1.1	DecmFactory	46
3.1.2	createDecmACLPerm	47
3.1.3	createDecmDoc	47
3.1.4	createDecmSubFile	48
3.1.5	createDecmDocProp	48
3.1.6	createDecmDocPerm	49
3.1.7	createDecmFolderTree	49
3.1.8	createDecmFolder	50
3.1.9	createDecmFolderProp	50
3.1.10	createDecmFolderPerm	51
3.1.11	createDecmSelect	51
3.1.12	createDecmCondition	52
3.1.13	createDecmCondGroup	52
3.1.14	createDecmDocMasterList	53
3.1.15	createDecmDocMaster	53
3.1.16	createDecmDocMasterProp	54
3.1.17	createDecmDocPropMaster	54
3.1.18	createDecmDocPropMasterProp	55
3.1.19	initUserProp (文字列型)	55
3.1.20	initUserProp (文字列型以外)	56

3.1.21	isCapacityUse	57
3.2	DecmObject クラス	58
3.3	DecmCommonProp クラス	59
3.3.1	getOiid	59
3.3.2	getStrProp	59
3.3.3	setStrProp	60
3.3.4	getIntProp	61
3.3.5	setIntProp	61
3.3.6	getLongProp	62
3.3.7	setLongProp	62
3.4	DecmCommonPerm クラス	64
3.4.1	getOwnerId	64
3.4.2	setOwnerId	64
3.4.3	getOwnerName	65
3.4.4	setOwnerName	65
3.4.5	getEveryonePerm	66
3.4.6	setEveryonePerm	66
3.4.7	getACLList	67
3.4.8	setACLList	67
3.5	DecmACLPerm クラス	69
3.5.1	getSubject	69
3.5.2	setSubject	69
3.5.3	getSubjectType	70
3.5.4	setSubjectType	70
3.5.5	getPermission	71
3.5.6	setPermission	71
3.6	DecmDBControl クラス	73

4

	文書操作パッケージ提供クラスの詳細	75
4.1	DecmDoc クラス	76
4.1.1	getOiid	78
4.1.2	setOiid	78
4.1.3	setFolderOiid	79
4.1.4	setDocProp	79
4.1.5	setDocPerm	80
4.1.6	setFilePath	81

4.1.7	setTextPath	81
4.1.8	setDocMaster	82
4.1.9	setRenditionType	83
4.1.10	setSubFileList	83
4.1.11	isSubTraceMode	84
4.1.12	setSubTraceMode	84
4.1.13	createDoc	85
4.1.14	deleteDoc	87
4.1.15	trashDoc	88
4.1.16	trashBackDoc	89
4.1.17	fetchDocProp	90
4.1.18	updateDocProp	90
4.1.19	fetchDocPerm	92
4.1.20	updateDocPerm	93
4.1.21	downloadContent	94
4.1.22	fetchDocHistory	96
4.1.23	deleteDocHistory	97
4.1.24	checkoutDoc	98
4.1.25	cancelCheckoutDoc	99
4.1.26	checkinDoc	100
4.1.27	moveDoc	102
4.1.28	fetchSubFile	103
4.1.29	lockRead	104
4.1.30	lockWrite	105
4.2	DecmDocProp クラス	106
4.3	DecmDocPerm クラス	107
4.4	DecmSubFile クラス	108
4.4.1	setFilePath	108
4.4.2	setRenditionType	109
4.4.3	fetchFileProp	110
4.4.4	downloadContent	110

5

フォルダ操作パッケージ提供クラスの詳細 113

5.1	DecmFolderTree クラス	114
5.1.1	setParentOiid	115
5.1.2	getFolderList	115

5.2	DecmFolder クラス	117
5.2.1	getOiid	118
5.2.2	setOiid	119
5.2.3	setParentOiid	119
5.2.4	setFolderProp	120
5.2.5	setFolderPerm	120
5.2.6	setCapacityMax	121
5.2.7	resetCapacityMax	122
5.2.8	createNodeFolder	122
5.2.9	createDocFolder	123
5.2.10	createPubFolder	125
5.2.11	deleteFolder	127
5.2.12	fetchFolderProp	127
5.2.13	updateNodeFolderProp	128
5.2.14	updateDocFolderProp	129
5.2.15	updatePubFolderProp	130
5.2.16	fetchFolderPerm	131
5.2.17	updateFolderPerm	132
5.2.18	changeCapacityMode	133
5.2.19	lockRead	134
5.2.20	lockWrite	134
5.3	DecmFolderProp クラス	136
5.4	DecmFolderPerm クラス	137

6

	検索操作パッケージ提供クラスの詳細	139
6.1	DecmSelect クラス	140
6.1.1	setFolderOiid	141
6.1.2	setFetchCount	142
6.1.3	setSortKey	143
6.1.4	setSortMode	143
6.1.5	setTrashMode	144
6.1.6	setFolderMode	145
6.1.7	isSubFileSearchMode	145
6.1.8	setSubFileSearchMode	146
6.1.9	addCondition	146
6.1.10	setSelectList	147

6.1.11	setCondGroup	147
6.1.12	setKeyword	148
6.1.13	setSearchExpanse	149
6.1.14	setSearchDictionary	149
6.1.15	searchDoc	150
6.1.16	resetCondition	151
6.2	DecmCondition クラス	152
6.2.1	createStrCond	152
6.2.2	createIntCond	153
6.3	DecmCondGroup クラス	154
6.3.1	addCond	154
6.3.2	addCondGroup	155

7

文書種別マスタ操作パッケージ提供クラスの詳細 157

7.1	DecmDocMasterList クラス	158
7.1.1	getDocMasterList	158
7.2	DecmDocMaster クラス	160
7.2.1	getOiid	160
7.2.2	setOiid	161
7.2.3	fetchDocMasterProp	161
7.2.4	getNewDocMaster	162
7.2.5	getSearchDocMaster	162
7.2.6	getListDocMaster	163
7.3	DecmDocMasterProp クラス	164
7.4	DecmDocPropMaster クラス	165
7.4.1	setParentOiid	165
7.4.2	getDocPropMasterList	166
7.5	DecmDocPropMasterProp クラス	167

8

DocumentBroker で提供されるクラスライブラリ 169

8.1	DocumentBroker で提供されるクラスライブラリ	170
8.1.1	使用するクラス, およびインターフェースの一覧	170
8.1.2	使用するメソッド一覧	170
8.2	DbjFactory0200 クラス	172
8.2.1	getFactory	172

8.3	DbjFactory インターフェース	173
8.3.1	createSession	173
8.4	DbjSession インターフェース	175
8.4.1	login	175
8.4.2	logout	176
8.4.3	begin	177
8.4.4	rollback	177
8.4.5	commit	178
8.5	DbjDocSpace インターフェース	180

9

9	コーディング例	181
9.1	作成するコーディング例	182
9.1.1	前提条件	182
9.2	共通操作パッケージのコーディング例	183
9.2.1	DecmFactory インスタンスの作成	183
9.3	文書操作パッケージのコーディング例	184
9.3.1	文書の作成・取得・削除	184
9.3.2	文書のプロパティの更新・取得	186
9.3.3	文書のアクセス権限の更新・取得	187
9.3.4	文書の履歴の取得・削除・更新	188
9.3.5	文書をごみ箱へ移動したあと元に戻す	189
9.3.6	文書の移動	190
9.4	フォルダ操作パッケージのコーディング例	191
9.4.1	フォルダの作成・削除	191
9.4.2	フォルダのプロパティの作成・更新	192
9.4.3	フォルダのアクセス権限の更新・取得	194
9.5	検索操作パッケージのコーディング例	195
9.5.1	文書の検索	195
9.5.2	組み合わせ検索条件を指定した文書の検索	196
9.5.3	全文検索を利用した文書の検索	197
9.6	文書種別マスタ操作パッケージのコーディング例	199
9.6.1	文書種別マスター一覧の取得	199
9.6.2	特定の文書種別マスタの OIID の取得	199
9.6.3	属性値マスター一覧の取得	199

10	DocumentBroker Standard GUI Development Kit の環境構築	201
10.1	DocumentBroker Standard GUI Development Kit のインストール，および環境構築手順	202
10.1.1	DocumentBroker Standard GUI Development Kit のインストール	202
10.1.2	Java 実行環境の設定	202
10.1.3	uCosminexus で使用する場合の設定	203
付録		205
付録 A	DocumentBroker Standard GUI Development Kit で使用するプロパティ一覧	206
付録 A.1	DocumentBroker Standard GUI Development Kit で使用するプロパティ	206
付録 B	用語解説	215
索引		221

図目次

図 1-1	DocumentBroker Standard GUI Development Kit のシステム構成例（推奨）	6
図 1-2	インターフェースの処理の流れ	12
図 2-1	DocumentBroker Standard GUI Development Kit で提供するクラスの関連図	18
図 2-2	DocumentBroker Standard GUI Development Kit で提供している例外クラスの関連図	19

表目次

表 1-1	DocumentBroker Standard GUI Development Kit のパッケージ一覧	10
表 2-1	共通操作パッケージのクラス一覧	20
表 2-2	DecmFactory クラスのメソッド一覧	20
表 2-3	DecmFactory クラスのフィールド一覧	21
表 2-4	DecmCommonProp クラスのメソッド一覧	21
表 2-5	DecmCommonPerm クラスのメソッド一覧	22
表 2-6	DecmACLPerm クラスのメソッド一覧	22
表 2-7	DecmACLPerm クラスのフィールド一覧	23
表 2-8	共通操作パッケージの例外クラス	24
表 2-9	文書操作パッケージのクラス一覧	26
表 2-10	DecmDoc クラスのメソッド一覧	26
表 2-11	DecmSubFile クラスのメソッド一覧	28
表 2-12	フォルダ操作パッケージのクラス一覧	29
表 2-13	DecmFolderTree クラスのメソッド一覧	29
表 2-14	DecmFolder クラスのメソッド一覧	29
表 2-15	DecmFolder クラスのフィールド一覧	30
表 2-16	検索操作パッケージのクラス一覧	32
表 2-17	DecmSelect クラスのメソッド一覧	32
表 2-18	DecmSelect クラスのフィールド一覧	33
表 2-19	DecmCondition クラスのメソッド一覧	33
表 2-20	DecmCondition クラスのフィールド一覧	33
表 2-21	DecmCondGroup クラスのメソッド一覧	34
表 2-22	DecmCondGroup クラスのフィールド一覧	34
表 2-23	文書種別マスタ操作パッケージのクラス一覧	35
表 2-24	DecmDocMasterList クラスのメソッド一覧	35
表 2-25	DecmDocMaster クラスのメソッド一覧	35
表 2-26	DecmDocPropMaster クラスのメソッド一覧	36
表 2-27	DecmDoc クラスのメソッド実行時に影響する setter メソッド (1/2)	39
表 2-28	DecmDoc クラスのメソッド実行時に影響する setter メソッド (2/2)	40
表 2-29	DecmFolder クラスのメソッド実行時に影響する setter メソッド	41
表 2-30	文書空間オブジェクト操作時に必要なアクセス権限	42
表 2-31	文書の状態によるメソッドの実行可否	43
表 4-1	DecmDoc クラスで扱うプロパティ	76

表 4-2	DecmDoc クラスで扱うプロパティ	108
表 5-1	DecmFolderTree クラスで扱うプロパティ	114
表 5-2	DecmFolder クラスで扱うプロパティ	117
表 6-1	DecmSelect クラスで扱うプロパティ	140
表 7-1	DecmDocMasterList クラスで扱うプロパティ	158
表 7-2	DecmDocMaster クラスで扱うプロパティ	160
表 7-3	DecmDocPropMaster クラスで扱うプロパティ	165
表 8-1	DocumentBroker Development Kit , Runtime Java クラスライブラリのクラス, および インターフェース一覧	170
表 8-2	DbjFactory0200 クラスのメソッド一覧	171
表 8-3	DbjFactory インターフェースのメソッド一覧	171
表 8-4	DbjSession インターフェースのメソッド一覧	171
表 A-1	DocumentBroker Standard GUI Development Kit で使用するプロパティ	206

1

DocumentBroker Standard GUI Development Kit の概要

この章では、DocumentBroker Standard GUI Development Kit で実現できる機能の概要について説明します。

-
- 1.1 DocumentBroker Standard GUI Development Kit とは
 - 1.2 DocumentBroker Standard GUI Development Kit で実現する機能
 - 1.3 DocumentBroker Standard GUI Development Kit のシステム構成
 - 1.4 DocumentBroker Standard GUI Development Kit の前提プログラム
 - 1.5 DocumentBroker Standard GUI Development Kit のパッケージ一覧
 - 1.6 インターフェースの処理の流れ
 - 1.7 コーディング時の注意事項
 - 1.8 DocumentBroker Standard GUI Development Kit で使用するユーザ種別
 - 1.9 トランザクションの開始と終了
-

1.1 DocumentBroker Standard GUI Development Kit とは

DocumentBroker Standard GUI Development Kit では、DocumentBroker Standard GUI で提供している文書モデルに対してアクセスを行うために必要な、Java クラスライブラリとメソッドを提供しています。

1.2 DocumentBroker Standard GUI Development Kit で実現する機能

DocumentBroker Standard GUI Development Kit で提供している Java クラスライブラリで、実現できる機能について説明します。

1.2.1 共通処理機能

DocumentBroker Standard GUI の文書モデルに対して、アクセスを行うための共通処理を提供します。

1.2.2 文書操作機能

DocumentBroker Standard GUI Development Kit の文書モデルで使用する文書に対する操作機能を提供します。

- 文書の作成
- 文書の削除
- ごみ箱への移動
- ごみ箱から元に戻す処理
- 文書のプロパティ取得
- 文書のプロパティ更新
- 文書のアクセス権限取得
- 文書のアクセス権限更新
- 文書の实体（コンテンツ）のダウンロード
- 文書の履歴取得
- 文書の履歴削除
- チェックアウト処理
- キャンセルチェックアウト処理
- チェックイン処理
- 文書の移動
- 関連ファイル一覧の取得

1.2.3 フォルダ操作機能

DocumentBroker Standard GUI の文書モデルで使用するフォルダに対する操作機能

1. DocumentBroker Standard GUI Development Kit の概要

提供します。

フォルダの作成

フォルダの削除

フォルダ一覧の取得

フォルダのプロパティ取得

フォルダのプロパティ更新

フォルダのアクセス権限取得

フォルダのアクセス権限更新

1.2.4 検索操作機能

DocumentBroker Standard GUI Development Kit の文書モデルで使用する文書に対する検索機能を提供します。

フォルダ下の文書一覧の取得

全文検索結果一覧の取得

プロパティ検索結果一覧の取得

ごみ箱の文書一覧の取得

1.2.5 文書種別マスタの操作機能

DocumentBroker Standard GUI Development Kit の文書モデルで使用する文書に対する、文書種別マスタの操作機能を提供します。

文書種別マスタ一覧の取得

文書種別マスタ定義項目の取得

属性値マスタデータの取得

1.3 DocumentBroker Standard GUI Development Kit のシステム構成

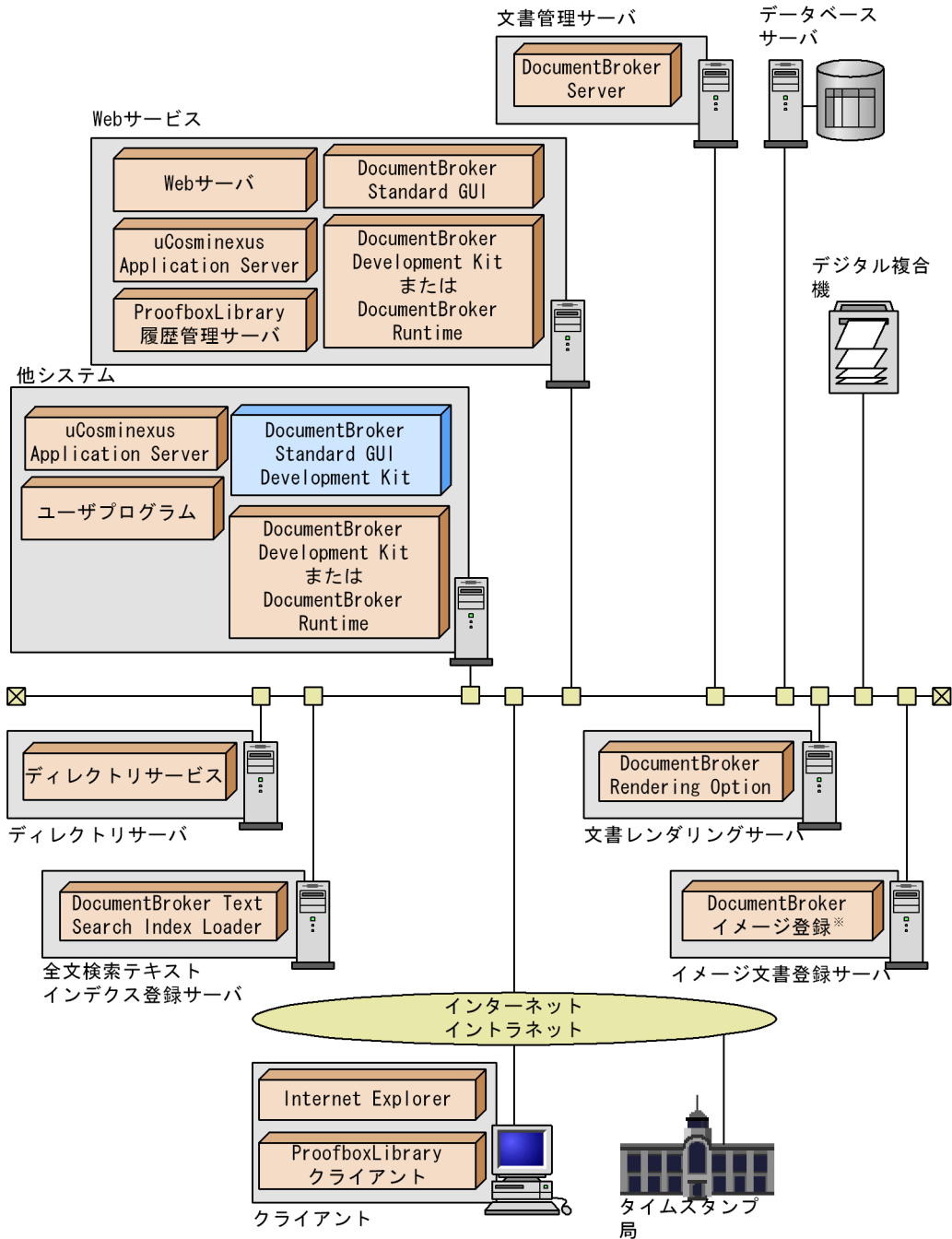
この節では、DocumentBroker Standard GUI Development Kit を使用する場合のシステム構成について説明します。

1.3.1 DocumentBroker Standard GUI Development Kit のシステム構成例

DocumentBroker Standard GUI Development Kit の推奨するシステムの構成例を次の図に示します。

1. DocumentBroker Standard GUI Development Kit の概要

図 1-1 DocumentBroker Standard GUI Development Kit のシステム構成例（推奨）



注※ 次の中から、ご使用のデジタル複合機に対応したプログラムを使用します。

- ・ DocumentBroker イメージ登録 for DocuCentre
- ・ DocumentBroker イメージ登録 for iR
- ・ DocumentBroker イメージ登録 for imagio and IPSiO
- ・ DocumentBroker イメージ登録 for Konica

1.4 DocumentBroker Standard GUI Development Kit の前提プログラム

ここでは、DocumentBroker Standard GUI Development Kit の前提プログラムについて説明します。

1.4.1 DocumentBroker Standard GUI Development Kit の前提 OS

次のどれかが必要です。

Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版 (Service Pack2 以降)

Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版(Service Pack2 以降)

Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition 日本語版 (Service Pack2 以降)

Microsoft(R) Windows Server(R) 2003 R2, Standard Edition 日本語版 (Service Pack2 以降)

Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition 日本語版(Service Pack2 以降)

Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition 日本語版 (Service Pack2 以降)

Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版

Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版

Microsoft(R) Windows(R) XP Professional Operating System (Service Pack2 以降)

1.4.2 DocumentBroker Standard GUI Development Kit の同一装置内の前提ソフトウェア

(1) DocumentBroker Standard GUI Development Kit を使用してユーザアプリケーションを開発する場合

次のどちらかのプログラムが必要です。

uCosminexus DocumentBroker Development Kit Version 3 03-60

uCosminexus DocumentBroker Runtime Version 3 03-60

1. DocumentBroker Standard GUI Development Kit の概要

次のどれかのプログラムが必要です。

- uCosminexus Application Server Standard 07-00 以降
- uCosminexus Application Server Enterprise 07-00 以降
- uCosminexus Developer Standard 07-00 以降
- uCosminexus Developer Professional 07-00 以降
- uCosminexus Application Server Standard 08-00 以降
- uCosminexus Application Server Enterprise 08-00 以降
- uCosminexus Developer Standard 08-00 以降
- uCosminexus Developer Professional 08-00 以降

(2) DocumentBroker Standard GUI Development Kit を使用してユーザアプリケーションを実行する場合

次のどちらかのプログラムが必要です。

- uCosminexus DocumentBroker Development Kit Version 3 03-60
- uCosminexus DocumentBroker Runtime Version 3 03-60

次のどれかのプログラムが必要です。

- uCosminexus Application Server Standard 07-00 以降
- uCosminexus Application Server Enterprise 07-00 以降
- uCosminexus Application Server Standard 08-00 以降
- uCosminexus Application Server Enterprise 08-00 以降

1.4.3 DocumentBroker Standard GUI Development Kit のシステム内の前提ソフトウェア

次のプログラムが必要です。

- DocumentBroker Version 3 Standard GUI 03-60 以降

! 注意事項

DocumentBroker Standard GUI Development Kit では、DocumentBroker Standard GUI に対する DocumentBroker Server の環境を前提とします。

ただし、DocumentBroker Standard GUI の配置ディスクリプタ (web.xml) の定義、各種プロパティファイルによる定義内容、ログインユーザの種別等については参照しません。

DocumentBroker Standard GUI での定義情報との同期は、DocumentBroker Standard GUI Development Kit を使用するユーザアプリケーション側で行う必要があります。

詳細については、マニュアル「uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド」を参照してください。

1.5 DocumentBroker Standard GUI Development Kit のパッケージ一覧

DocumentBroker Standard GUI Development Kit のパッケージ一覧を次の表に示します。

表 1-1 DocumentBroker Standard GUI Development Kit のパッケージ一覧

パッケージ名	パッケージ概要
jp.co.hitachi_system.sContents.API.Common	共通操作パッケージ
jp.co.hitachi_system.sContents.API.Document	文書操作パッケージ
jp.co.hitachi_system.sContents.API.Folder	フォルダ操作パッケージ
jp.co.hitachi_system.sContents.API.Select	検索操作パッケージ
jp.co.hitachi_system.sContents.API.DocMaster	文書種別マスタ操作パッケージ

1.6 インターフェースの処理の流れ

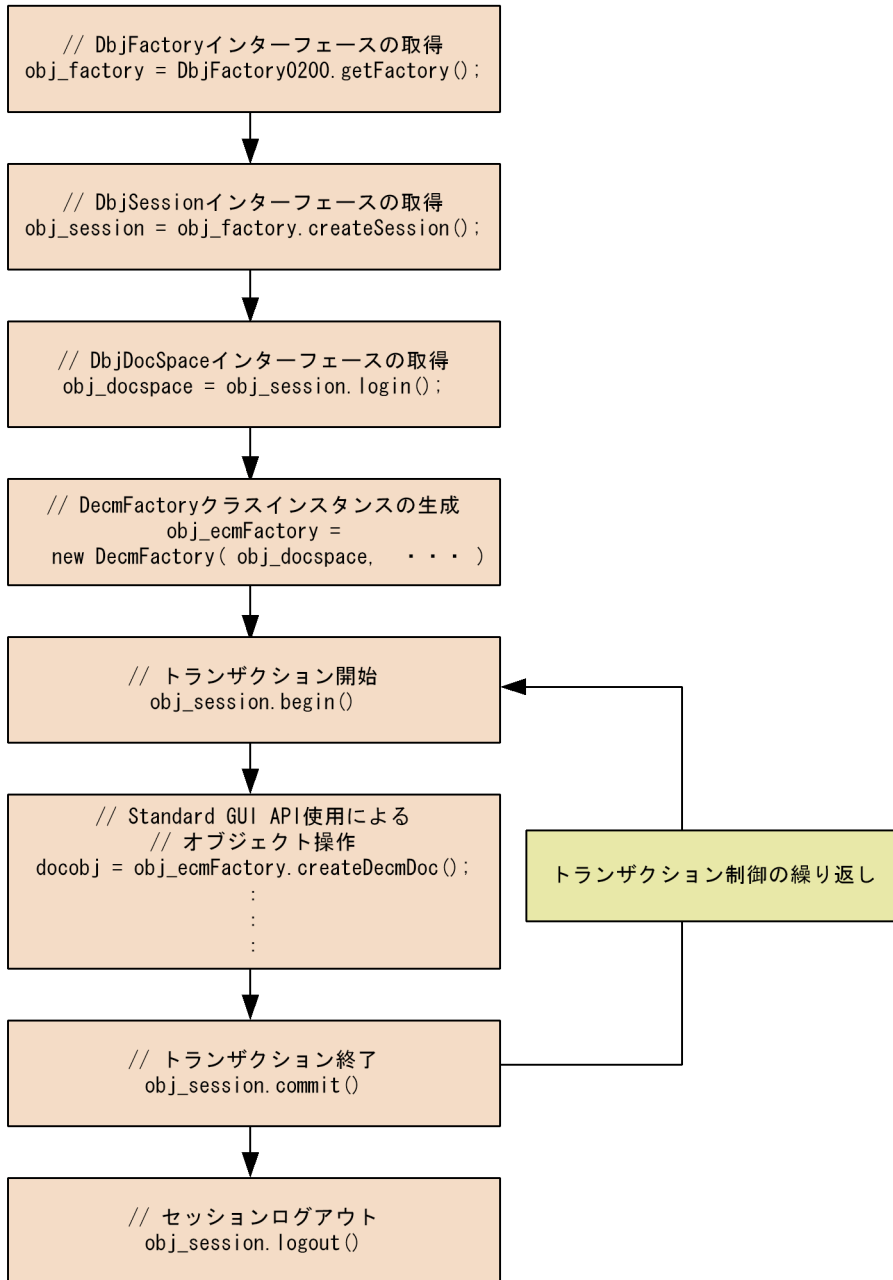
DocumentBroker Standard GUI Development Kit は、DocumentBroker Development Kit または DocumentBroker Runtime で提供される Java クラスライブラリの使用が前提です。

DocumentBroker Standard GUI Development Kit におけるファクトリクラス (DecmFactory) に、DbjDocSpace インタフェースを渡す必要があります。

クラスの詳細については、「2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧」を参照してください。

インターフェースの処理の流れを次の図に示します。

図 1-2 インターフェースの処理の流れ



(凡例)

→ : 処理の流れを示します。

注

- DocumentBroker Standard GUI Development Kit のメソッド発行で、エラー例外が発生した場合は、DbjSession インタフェースの rollback メソッドを必ず発行してください。
- DocumentBroker Standard GUI Development Kit のメソッド発行で、エラー例外が発生した場合は、DbjSession インタフェースの commit メソッドは発行しないでください。正しく動作しないおそれがあります。

1.7 コーディング時の注意事項

この節では、コーディング時の注意事項について説明します。

1.7.1 マルチスレッド環境での注意事項

DocumentBroker Standard GUI Development Kit で提供するクラスのインスタンスに対して、複数スレッドから同時にアクセスはしないで下さい。

1.7.2 使用できる文字コード種別

Java クラスライブラリを使用したプログラミング時に使用できる文字コード種別を次に示します。

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

使用する文字コード種別は、接続先となる DocumentBroker Standard GUI Development Kit のデータベースで使用する文字コード種別と合わせてください。

1.8 DocumentBroker Standard GUI Development Kit で使用するユーザ種別

DocumentBroker Standard GUI Development Kit では、DocumentBroker Standard GUI のユーザの種別は使用されません。DocumentBroker Standard GUI Development Kit で使用するユーザの種別は、セキュリティ管理者とそれ以外の 2 種類だけです。

DocumentBroker Standard GUI のユーザの種別の詳細については、マニュアル「uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド」を参照してください。

1.9 トランザクションの開始と終了

DocumentBroker Standard GUI Development Kit のトランザクションは、ユーザアプリケーションプログラムで明示的に開始と終了を宣言してください。明示的に宣言していない場合は、データベースのデータの整合性が確保できないことがあります。

詳細については、マニュアル「uCosminexus DocumentBroker Version 3 クラスライブラリ Java 解説」を参照してください。

2

DocumentBroker Standard GUI Development Kit 提供のクラス一覧

この章では、DocumentBroker Standard GUI Development Kit で提供するクラスの一覧を示します。

-
- 2.1 DocumentBroker Standard GUI Development Kit パッケージが提供するクラス

 - 2.2 共通操作パッケージのクラス

 - 2.3 共通操作パッケージの例外クラス

 - 2.4 文書操作パッケージ提供クラス

 - 2.5 フォルダ操作パッケージ提供クラス

 - 2.6 検索操作パッケージ提供クラス

 - 2.7 文書種別マスタ操作パッケージ提供クラス

 - 2.8 パッケージ提供クラス，およびメソッドの説明形式

 - 2.9 メソッドの関係
-

2.1 DocumentBroker Standard GUI Development Kit パッケージが提供するクラス

DocumentBroker Standard GUI Development Kit で提供するクラスの関連図，例外クラスの関連図を次の図に示します。

図 2-1 DocumentBroker Standard GUI Development Kit で提供するクラスの関連図

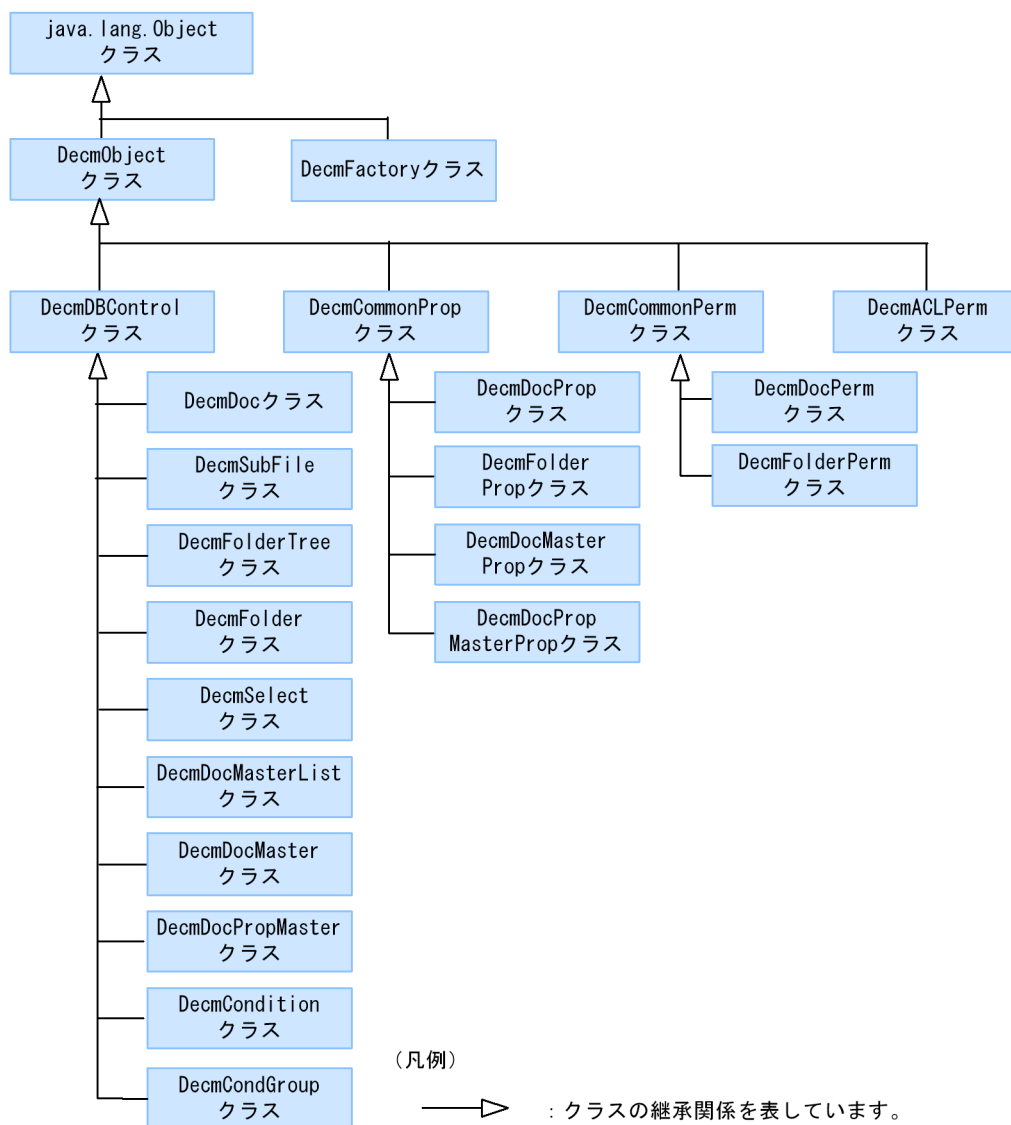
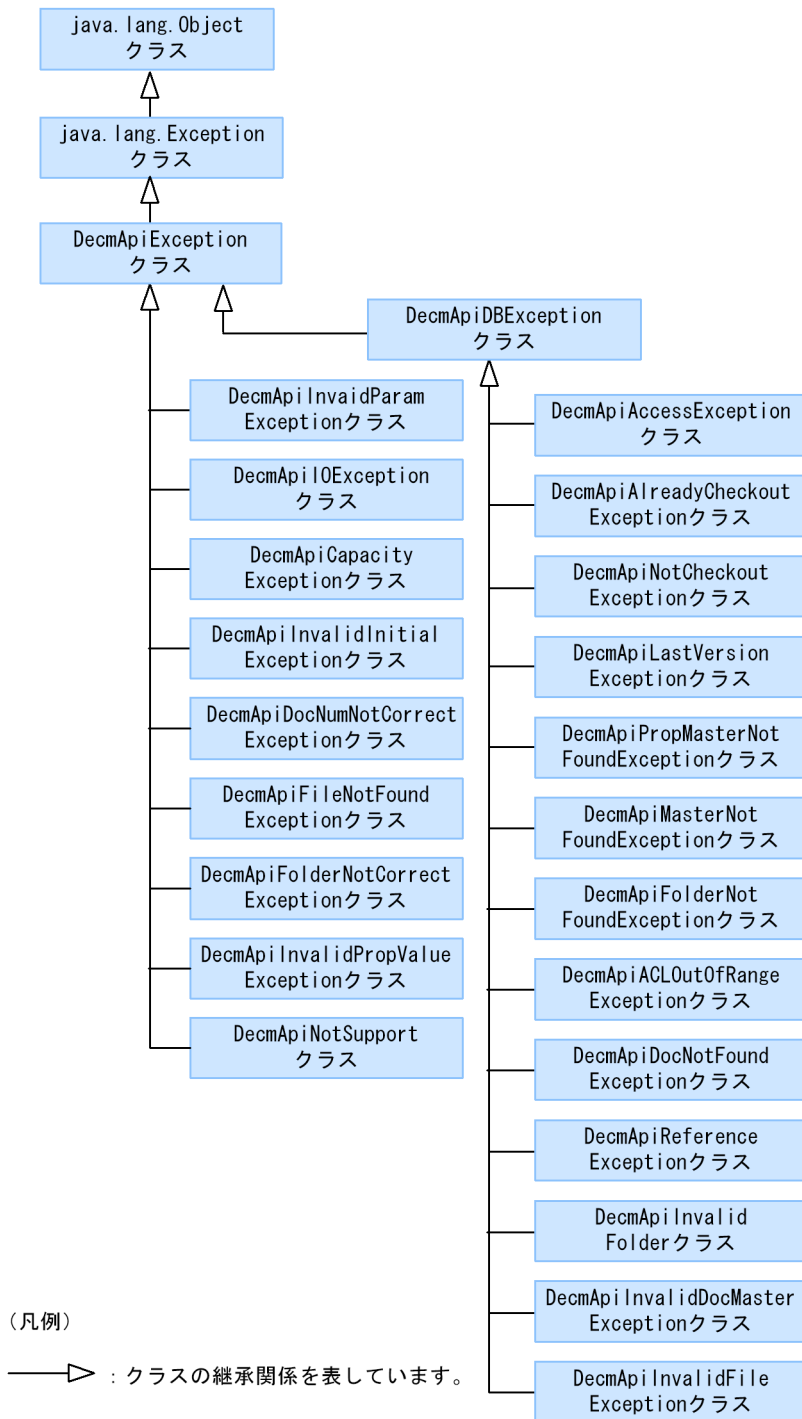


図 2-2 DocumentBroker Standard GUI Development Kit で提供している例外クラスの関連図



2.2 共通操作パッケージのクラス

共通操作パッケージのクラス一覧を次の表に示します。

表 2-1 共通操作パッケージのクラス一覧

クラス	説明
DecmFactory クラス	インスタンスを生成するための Factory クラス
DecmObject クラス	クラスの基底クラス
DecmCommonProp クラス	プロパティの基底クラス
DecmCommonPerm クラス	アクセス権限情報の基底クラス
DecmACLPerm クラス	アクセス権限の基底クラス
DecmDBControl クラス	データベース操作の基底クラス

2.2.1 共通操作パッケージのメソッド一覧

共通操作パッケージのクラスのメソッド一覧を示します。

(1) DecmFactory クラスのメソッド一覧

DecmFactory クラスのメソッド一覧を次の表に示します。スーパークラスは `java.lang.Object` クラスです。

表 2-2 DecmFactory クラスのメソッド一覧

メソッド	機能
DecmFactory	コンストラクタです。 すべてのクラスのインスタンスを生成します。ただし、例外クラスの インスタンスは生成できません。
createDecmACLPerm	アクセス権限表現用クラスのインスタンスを生成します。
createDecmDoc	文書操作クラスのインスタンスを生成します。
createDecmSubFile	関連ファイル操作クラスのインスタンスを生成します。
createDecmDocProp	文書プロパティクラスのインスタンスを生成します。
createDecmDocPerm	文書アクセス権限クラスのインスタンスを生成します。
createDecmFolderTree	フォルダツリー階層情報取得クラスのインスタンスを生成します。
createDecmFolder	フォルダ操作クラスのインスタンスを生成します。
createDecmFolderProp	フォルダプロパティクラスのインスタンスを生成します。
createDecmFolderPerm	フォルダアクセス権限クラスのインスタンスを生成します。
createDecmSelect	文書検索クラスのインスタンスを生成します。
createDecmCondition	検索条件クラスのインスタンスを生成します。
createDecmCondGroup	組み合わせ検索条件クラスのインスタンスを生成します。

メソッド	機能
createDecmDocMasterList	文書種別マスター一覧クラスのインスタンスを生成します。
createDecmDocMaster	文書種別マスタ操作作用クラスのインスタンスを生成します。
createDecmDocMasterProp	文書種別マスタプロパティクラスのインスタンスを生成します。
createDecmDocPropMaster	属性値マスタ取得クラスのインスタンスを生成します。
createDecmDocPropMasterProp	属性値マスタプロパティクラスのインスタンスを生成します。
initUserProp (文字列型)	ユーザプロパティ, および文書種別マスタに使用するプロパティで, 文字列型のプロパティをシステムに登録します。
initUserProp (文字列型以外)	ユーザプロパティ, および文書種別マスタに使用するプロパティで, 文字列型以外のプロパティをシステムに登録します。
isCapacityUse	フォルダ下ファイル容量制限機能の使用有無を取得します。

表 2-3 DecmFactory クラスのフィールド一覧

定義	形式	意味
DB_HIRDB	int 型	データベースの種類: HIRDB
PROP_TYPE_STRING	int 型	ユーザプロパティのタイプ: 文字列型
PROP_TYPE_INTEGER	int 型	ユーザプロパティのタイプ: 整数型
PROP_TYPE_DATE	int 型	ユーザプロパティのタイプ: 日付型
PROP_TYPE_MASTER	int 型	ユーザプロパティのタイプ: マスタ型

注

ユーザプロパティの詳細については、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

(2) DecmObject クラス

DocumentBroker Standard GUI Development Kit で提供するすべてのクラスの基底クラスです。ただし、例外クラスは DecmObject クラスから派生しません。スーパークラスは java.lang.Object クラスです。

(3) DecmCommonProp クラスのメソッド一覧

DecmCommonProp クラスのメソッド一覧を次の表に示します。スーパークラスは DecmObject クラスです。

表 2-4 DecmCommonProp クラスのメソッド一覧

メソッド	機能
getOiid	対象となるオブジェクトの OIID 値を取得します。

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

メソッド	機能
getStrProp	文字列型、またはマスタ型のプロパティに対するプロパティ値を取得します。
setStrProp	文字列型、またはマスタ型のプロパティに対するプロパティ値を設定します。
getIntProp	整数型、または日付型のプロパティに対するプロパティ値を取得します。
setIntProp	整数型、または日付型のプロパティに対するプロパティ値を設定します。
getLongProp	整数型 (Long 型) のプロパティに対するプロパティ値を取得します。
setLongProp	整数型 (Long 型) のプロパティに対するプロパティ値を設定します。

注

プロパティ値の詳細については「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

(4) DecmCommonPerm クラスのメソッド一覧

DecmCommonPerm クラスのメソッド一覧を次の表に示します。スーパークラスは DecmObject クラスです。

表 2-5 DecmCommonPerm クラスのメソッド一覧

メソッド	機能
getOwnerId	オブジェクトの所有者のユーザ ID を取得します。
setOwnerId	オブジェクトの所有者のユーザ ID を設定します。
getOwnerName	オブジェクトの所有者名称を取得します。
setOwnerName	オブジェクトの所有者名称を設定します。
getEveryonePerm	すべてのユーザに対するアクセス権限を取得します。
setEveryonePerm	すべてのユーザに対するアクセス権限を設定します。
getACLList	アクセス権限のリストを取得します。
setACLList	アクセス権限のリストを設定します。

(5) DecmACLPerm クラスのメソッド一覧

DecmACLPerm クラスのメソッド一覧を次の表に示します。スーパークラスは DecmObject クラスです。

表 2-6 DecmACLPerm クラスのメソッド一覧

メソッド	機能
getSubject	アクセス権限対象の ID (ユーザ ID , 組織 ID , グループ ID) を取得します。

メソッド	機能
setSubject	アクセス権限対象の ID (ユーザ ID, 組織 ID, グループ ID) を設定します。
getSubjectType	アクセス権限対象の種別 (ユーザ, 組織, グループ) を取得します。
setSubjectType	アクセス権限対象の種別 (ユーザ, 組織, グループ) を設定します。
getPermission	アクセス権限を取得します。
setPermission	アクセス権限を設定します。

表 2-7 DecmACLPerm クラスのフィールド一覧

定数	形式	意味
PERM_NONE	int 型	アクセス権限の種別: なし
PERM_READ	int 型	アクセス権限の種別: 参照
PERM_LINK	int 型	アクセス権限の種別: リンク
PERM_UPDATE	int 型	アクセス権限の種別: 更新
PERM_FULL	int 型	アクセス権限の種別: 全て
PERM_TYPE_USER	int 型	アクセス権限対象の種別: ユーザ
PERM_TYPE_GROUP	int 型	アクセス権限対象の種別: 組織またはグループ

(6) DecmDBControl クラス

スーパークラスは DecmObject クラスです。

2.3 共通操作パッケージの例外クラス

共通操作パッケージの例外クラスを一覧形式で説明します。

2.3.1 共通操作パッケージの例外クラス

共通操作パッケージの例外クラス一覧を次の表に示します。

表 2-8 共通操作パッケージの例外クラス

クラス	説明
DecmApiException クラス	例外クラスの基底クラスです。 発行される例外はこのクラスを基底とします。
DecmApiInvalidParamException クラス	引数不正の例外です。 メソッドに指定した引数が不正であることを示します。
DecmApiInvalidInitialException クラス	初期化処理呼び出し不正の例外です。 初期化処理メソッドの呼び出しが、不正であることを示します。
DecmApiIOException クラス	入出力エラーの例外です。 ファイル I/O エラーの発生を示します。
DecmApiDBException クラス	データベースエラーの例外です。 データベースシステムで異常が発生したことを示します。
DecmApiAccessException クラス	アクセス権限エラーの例外です。 対象となるオブジェクト（文書、フォルダなど）に対して、アクセス権限がないことを示します。
DecmApiAlreadyCheckoutException クラス	チェックアウト済みの例外です。 文書に対してチェックアウトを行った際、すでにチェックアウト済みであることを示します。
DecmApiNotCheckoutException クラス	チェックアウト状態不正エラーの例外です。 文書のチェックイン処理で、対象となる文書の状態が不正（未チェックアウト）であることを示します。
DecmApiLastVersionException クラス	履歴の削除処理エラーの例外です。 文書の履歴で最後の 1 バージョンを削除しようとした場合のエラーを示します。
DecmApiDocNotFoundException クラス	指定した文書が存在しないことを示す例外です。 指定した文書が削除されているか、存在しないことを示します。
DecmApiPropMasterNotFoundException クラス	指定した属性値マスタが存在しないことを示す例外です。 指定した属性値マスタが削除されているか、存在しないことを示します。
DecmApiMasterNotFoundException クラス	指定した文書種別マスタが存在しないことを示す例外です。 指定した文書種別マスタが削除されているか、存在しないことを示します。
DecmApiFolderNotFoundException クラス	指定したフォルダが存在しないことを示す例外です。 指定したフォルダが削除されているか、存在しないことを示します。

クラス	説明
DecmApiACLOutOfRangeException クラス	設定したアクセス権限の数が最大値を超えたことを示す例外です。 アクセス権限の設定処理で、権限の設定数が最大数を超えたことを示します。
DecmApiReferenceException クラス	対象の文書ファイルがデータベース中に存在しないことを示す例外です。 対象となる文書のファイルがデータベース中に存在しないため、操作ができないことを示します。
DecmApiCapacityException クラス	フォルダ下に登録できる文書が容量制限を越えたことを示す例外です。
DecmApiInvalidFolder クラス	フォルダ種別が不正であることを示す例外です。
DecmApiFolderNotCorrectException クラス	上位フォルダの種別、またはアクセス権限方式の指定が、不正であることを示す例外です。
DecmApiDocNumNotCorrectException クラス	採番用に指定した文書種別マスタが、不正であることを示す例外です。
DecmApiFileNotFoundException クラス	指定したレンディションにファイルが存在しないことを示す例外です。
DecmApiNotSupport クラス	指定した操作がサポートされていないことを示す例外です。
DecmApiInvalidPropValueException クラス	採番に使用する値が不正であることを示す例外です。
DecmApiInvalidDocMasterException クラス	指定した文書種別マスタが不正であることを示す例外です。 チェックインおよび文書の更新で、前の版（バージョン）と異なる文書種別マスタを指定した場合に、この例外が発行されません。
DecmApiInvalidFileException クラス	登録するファイルが不正であることを示す例外です。

2.4 文書操作パッケージ提供クラス

このパッケージは、DocumentBroker Standard GUI Development Kit 文書操作パッケージです。

文書操作パッケージのクラス、およびメソッドを一覧形式で説明します。

2.4.1 文書操作パッケージのクラス

文書操作パッケージのクラス一覧を次の表に示します。

表 2-9 文書操作パッケージのクラス一覧

クラス	説明
DecmDoc クラス	文書操作クラス
DecmDocProp クラス	文書プロパティクラス
DecmDocPerm クラス	文書アクセス権限クラス
DecmSubFile クラス	関連ファイル操作クラス

2.4.2 文書操作パッケージのメソッド一覧

文書操作パッケージのクラスのメソッド一覧を示します。

(1) DecmDoc クラス

DecmDoc クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-10 DecmDoc クラスのメソッド一覧

メソッド	機能
getOiid	操作対象文書のバージョンングオブジェクトの OIID を取得します。
setOiid	操作対象文書のバージョンングオブジェクトの OIID を設定します。
setFolderOiid	文書の上位フォルダの OIID を設定します。
setDocProp	文書のプロパティを設定します。
setDocPerm	文書のアクセス権限を設定します。
setFilePath	文書の実体（コンテンツ）ファイルのローカルディスク上でのパスを設定します。
setTextPath	全文検索用のテキストファイルのローカルディスク上でのパスを設定します。
setDocMaster	文書種別マスタの OIID を設定します。
setRenditionType	登録する文書ファイルの参照用のファイル（PDF）を設定します。

メソッド	機能
setSubFileList	関連ファイルリストを設定します。
isSubTraceMode	関連ファイルの監査証跡ログの出力有無を取得します。
setSubTraceMode	主ファイルと関連ファイルについて操作するメソッドで、関連ファイルの監査証跡ログを出力するかどうかを設定します。
createDoc	一般文書を作成します。
deleteDoc	文書をデータベースから完全に削除します。
trashDoc	文書をごみ箱へ移動します。
trashBackDoc	文書をごみ箱から戻します。
fetchDocProp	文書の最新バージョンのプロパティを取得します。
updateDocProp	文書の最新バージョンのプロパティを更新します。
fetchDocPerm	文書のアクセス権限情報を取得します。
updateDocPerm	文書のアクセス権限情報を更新します。
downloadContent	文書の実体（コンテンツ）をダウンロードします。
fetchDocHistory	文書の履歴を取得します。
deleteDocHistory	文書の履歴を削除します。
checkoutDoc	文書をチェックアウトします。
cancelCheckoutDoc	文書のチェックアウトをキャンセルします。
checkinDoc	文書をチェックインします。
moveDoc	文書を移動します。
fetchSubFile	主ファイルと関連付けられた関連ファイルリストを取得します。
lockRead	文書に対する参照ロックを予約します。
lockWrite	文書に対する更新ロックを予約します。

(2) DecmDocProp クラス

スーパークラスは DecmCommonProp クラスです。

(3) DecmDocPerm クラス

スーパークラスは DecmCommonPerm クラスです。

(4) DecmSubFile クラス

DecmSubFile クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

表 2-11 DecmSubFile クラスのメソッド一覧

メソッド	機能
setFilePath	文書の実体（コンテンツ）ファイルのローカルディスク上でのパスを設定します。
setRenditionType	登録する文書ファイルに対して、参照用のファイル（PDF）を設定します。
fetchFileProp	文書のプロパティを取得します。
downloadContent	文書の実体（コンテンツ）をダウンロードします。

2.5 フォルダ操作パッケージ提供クラス

このパッケージは、DocumentBroker Standard GUI Development Kit フォルダ操作パッケージです。

フォルダ操作パッケージのクラス、およびメソッドを一覧形式で説明します。

2.5.1 フォルダ操作パッケージのクラス一覧

フォルダ操作パッケージのクラス一覧を次の表に示します。

表 2-12 フォルダ操作パッケージのクラス一覧

クラス	説明
DecmFolderTree クラス	フォルダツリー階層情報取得クラス
DecmFolder クラス	フォルダ操作クラス
DecmFolderProp クラス	フォルダプロパティクラス
DecmFolderPerm クラス	フォルダアクセス権限クラス

2.5.2 フォルダ操作パッケージのメソッド一覧

フォルダ操作パッケージのクラスのメソッド一覧を示します。

(1) DecmFolderTree クラス

DecmFolderTree クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-13 DecmFolderTree クラスのメソッド一覧

メソッド	機能
setParentOiid	上位フォルダの OIID を設定します。
getFolderList	下位フォルダの一覧を取得します。

(2) DecmFolder クラス

DecmFolder クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-14 DecmFolder クラスのメソッド一覧

メソッド	機能
getOiid	操作対象フォルダの OIID を取得します。

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

メソッド	機能
setOiid	操作対象フォルダの OIID を設定します。
setParentOiid	フォルダ作成時の上位フォルダの OIID を設定します。
setFolderProp	フォルダのプロパティを設定します。
setFolderPerm	フォルダのアクセス権限を設定します。
setCapacityMax	フォルダの最大容量を設定します。
resetCapacityMax	フォルダの最大容量をリセットします。
createNodeFolder	ノードフォルダを作成します。
createDocFolder	文書フォルダを作成します。
createPubFolder	公開フォルダを作成します。
deleteFolder	フォルダを削除します。
fetchFolderProp	フォルダの最新バージョンのプロパティを取得します。
updateNodeFolderProp	ノードフォルダのプロパティを更新します。
updateDocFolderProp	文書フォルダのプロパティを更新します。
updatePubFolderProp	公開フォルダのプロパティを更新します。
fetchFolderPerm	フォルダのアクセス権限を取得します。
updateFolderPerm	フォルダのアクセス権限を更新します。
changeCapacityMode	フォルダ容量制限機能を変更します。
lockRead	フォルダに対する参照ロックを予約します。
lockWrite	フォルダに対する更新ロックを予約します。

表 2-15 DecmFolder クラスのフィールド一覧

定数	形式	意味
ACL_TYPE_PRIVATE	int 型	文書フォルダのアクセス権限方式：固有
ACL_TYPE_PUBLIC	int 型	文書フォルダのアクセス権限方式：共有
PUBFOL_ACL_TYPE_PUBLIC	int 型	公開フォルダのアクセス権限方式：全員に公開
PUBFOL_ACL_TYPE_PRIVATE	int 型	公開フォルダのアクセス権限方式：公開範囲限定
DOC_CREATE_NG	int 型	フォルダ下の文書の作成：不可
DOC_CREATE_OK	int 型	フォルダ下の文書の作成：可
UNIT_MB	int 型	フォルダ容量制限単位：メガバイト
UNIT_GB	int 型	フォルダ容量制限単位：ギガバイト
UNIT_TB	int 型	フォルダ容量制限単位：テラバイト

定数	形式	意味
CAP_NOUSE	int 型	フォルダ容量制限使用フラグ：使用しない
CAP_USE	int 型	フォルダ容量制限使用フラグ：使用する

(3) DecmFolderProp クラス

スーパークラスは DecmCommonProp クラスです。

(4) DecmFolderPerm クラス

スーパークラスは DecmCommonPerm クラスです。

2.6 検索操作パッケージ提供クラス

このパッケージは、DocumentBroker Standard GUI Development Kit 検索操作パッケージです。

検索操作パッケージのクラス、およびメソッドを一覧形式で説明します。

2.6.1 検索操作パッケージのクラス一覧

フォルダ操作パッケージのクラス一覧を次の表に示します。

表 2-16 検索操作パッケージのクラス一覧

クラス	説明
DecmSelect クラス	文書検索クラス
DecmCondition クラス	通常の検索条件の表現用クラス
DecmCondGroup クラス	組み合わせ検索条件の表現用クラス

2.6.2 検索操作パッケージのメソッド一覧

検索操作パッケージのクラスのメソッド一覧を示します。

(1) DecmSelect クラス

DecmSelect クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-17 DecmSelect クラスのメソッド一覧

メソッド	機能
setFolderOiid	検索対象となるフォルダの OIID を設定します。
setFetchCount	検索実行時に取得するレコード件数を設定します。
setSortKey	検索実行時のソート用カラム名を設定します。
setSortMode	検索実行時のソート方式（昇順、または降順）を設定します。
setTrashMode	検索対象とする文書の種別を設定します。
setFolderMode	フォルダ検索時に、下位フォルダを検索対象とするかどうかを設定します。
isSubFileSearchMode	関連ファイルの検索対象の指定有無を取得します。
setSubFileSearchMode	関連ファイルを検索対象とするかどうかを設定します。
addCondition	検索条件を追加します。
setSelectList	検索取得列を設定します。
setCondGroup	組み合わせ検索条件を設定します。

メソッド	機能
setKeyword	全文検索のキーワードを設定します。
setSearchExpanse	異表記展開指定の組み合わせを設定します。
setSearchDictionary	全文検索実行時の同義語辞書名称を設定します。
searchDoc	文書検索を実行します。
resetCondition	指定した検索条件をリセットします。

表 2-18 DecmSelect クラスのフィールド一覧

定数	形式	意味
SORT_MODE_UP	int 型	ソートの設定：昇順
SORT_MODE_DOWN	int 型	ソートの設定：降順
TRASH_MODE_NORMAL	int 型	検索対象とする文書：通常文書
TRASH_MODE_DELETED	int 型	検索対象とする文書：ごみ箱の文書
TRASH_MODE_ALL	int 型	検索対象とする文書：全ての文書

(2) DecmCondition クラス

DecmCondition クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-19 DecmCondition クラスのメソッド一覧

メソッド	機能
createStrCond	文字列型の検索条件を生成します。
createIntCond	整数型の検索条件を生成します。

表 2-20 DecmCondition クラスのフィールド一覧

定数	形式	意味
OP_EQ	int 型	検索条件のオペレータ：=
OP_NEQ	int 型	検索条件のオペレータ：<>
OP_LT	int 型	検索条件のオペレータ：<
OP_LTEQ	int 型	検索条件のオペレータ：<=
OP_GT	int 型	検索条件のオペレータ：>
OP_GTEQ	int 型	検索条件のオペレータ：>=
OP_PART	int 型	検索条件のオペレータ：部分一致
OP_STWITH	int 型	検索条件のオペレータ：前方一致
OP_ENDWITH	int 型	検索条件のオペレータ：後方一致

(3) DecmCondGroup クラス

DecmCondGroup クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-21 DecmCondGroup クラスのメソッド一覧

メソッド	機能
addCond	検索条件を追加します。
addCondGroup	組み合わせ検索条件を追加します。

表 2-22 DecmCondGroup クラスのフィールド一覧

定数	形式	意味
OP_AND	int 型	検索条件追加のオペレータ : AND
OP_OR	int 型	検索条件追加のオペレータ : OR

2.7 文書種別マスタ操作パッケージ提供クラス

このパッケージは、DocumentBroker Standard GUI Development Kit 文書種別マスタ操作パッケージです。

文書種別マスタ操作パッケージのクラス、およびメソッドを一覧形式で示します。

2.7.1 文書種別マスタ操作パッケージのクラス一覧

文書種別マスタ操作パッケージのクラス一覧を次の表に示します。

表 2-23 文書種別マスタ操作パッケージのクラス一覧

クラス	説明
DecmDocMasterList クラス	文書種別マスター一覧クラス
DecmDocMaster クラス	文書種別マスタ操作クラス
DecmDocMasterProp クラス	文書種別マスタプロパティクラス
DecmDocPropMaster クラス	属性値マスタ取得クラス
DecmDocPropMasterProp クラス	属性値マスタプロパティクラス

2.7.2 文書種別マスタ操作パッケージのメソッド一覧

文書種別マスタ操作パッケージのクラスのメソッド一覧を示します。

(1) DecmDocMasterList クラス

DecmDocMasterList クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-24 DecmDocMasterList クラスのメソッド一覧

メソッド	機能
getDocMasterList	すべての文書種別マスター一覧を取得します。

(2) DecmDocMaster クラス

DecmDocMaster クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-25 DecmDocMaster クラスのメソッド一覧

メソッド	機能
getOiid	操作対象文書種別マスタの OIID を取得します。

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

メソッド	機能
setOiid	操作対象文書種別マスタの OIID を設定します。
fetchDocMasterProp	文書種別マスタのプロパティを取得します。
getNewDocMaster	文書作成時に指定するプロパティ項目一覧を取得します。
getSearchDocMaster	検索に使用するプロパティ項目一覧を取得します。
getListDocMaster	一覧表示に使用するプロパティ項目一覧を取得します。

(3) DecmDocMasterProp クラス

スーパークラスは DecmCommonProp クラスです。

(4) DecmDocPropMaster クラス

DecmDocPropMaster クラスのメソッド一覧を次の表に示します。スーパークラスは DecmDBControl クラスです。

表 2-26 DecmDocPropMaster クラスのメソッド一覧

メソッド	機能
setParentOiid	上位属性値マスタの OIID を設定します。
getDocPropMasterList	下位属性値マスター一覧を取得します。

(5) DecmDocPropMasterProp クラス

スーパークラスは DecmCommonProp クラスです。

2.8 パッケージ提供クラス，およびメソッドの説明形式

パッケージ提供クラス，およびメソッドの説明形式について示します。各クラスの説明では，そのクラスに定義されるメソッドについてだけ説明しています。上位のクラスで定義されているメソッドについては，定義元のクラスを参照してください。

2.8.1 メソッドで説明する項目

クラスの説明に続けて，そのクラスのメソッドを説明しています。上位のクラスから継承されるメソッドの詳細については，継承元のクラスのメソッドの説明を参照してください。

(1) 機能

メソッドの機能の詳細を説明しています。

(2) 形式

メソッドの形式を説明しています。

(3) 引数

引数について説明しています。引数がないメソッドでは，「なし」と記述します。

(4) 戻り値

メソッドの戻り値を説明しています。戻り値を返却しないメソッドでは，「なし」と記述します。

(5) 例外

発生する例外について説明しています。発生する例外がないメソッドでは，「なし」と記述します。

なお，例外の詳細に以下の項目の記載がある場合は，該当する項目の説明を参照してください。

通常オブジェクトの設定チェック

オブジェクトの設定が次の場合，失敗と判定します。それ以外の場合は成功と判定します。

- オブジェクトに `null` を設定した場合

通常文字列の設定チェック

文字列の設定が次の場合，失敗と判定します。それ以外の場合は，成功と判定します。

- 文字列に `null` を設定した場合

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

- 文字列に「""」(空文字)を設定した場合

133 桁 OIID の設定チェック

OIID の設定が次の場合、失敗と判定します。それ以外の場合は、成功と判定します。

- OIID に null を設定した場合
- OIID に 133 桁以外の桁数を設定した場合

通常配列の設定チェック

配列の設定が次の場合、失敗と判定します。それ以外の場合は、成功と判定します。

- 配列に null を設定した場合
- 配列の項目数が 0 の場合

プロパティの設定規則チェック

プロパティの設定は、設定項目の最大長、省略可否などのチェックにより失敗を判定します。設定規則の詳細については、「付録 A DocumentBroker Standard GUI Development Kit で使用するプロパティ一覧」を参照してください。

2.9 メソッドの関係

この節では、DocumentBroker Standard GUI Development Kit で使用するメソッドの関係について説明します。

2.9.1 文書空間オブジェクトの操作時に影響する setter メソッド

DecmDoc クラスのメソッドを実行する場合に影響する setter メソッドの関係を次の表に示します。

なお、DecmSubFile クラスのメソッドは、setter メソッドではなく、DecmDoc#FetchSubFile メソッドの実行結果に影響されます。

表 2-27 DecmDoc クラスのメソッド実行時に影響する setter メソッド (1/2)

操作メソッド	setter メソッド					
	setOiid	setFolderOiid	setDocProp	setDocPerm	setFilePath	setTextPath
createDoc	-					
deleteDoc	2	-	-	-	-	-
trashDoc	2	-	-	-	-	-
trashBackDoc	2	-	-	-	-	-
fetchDocProp	2	-	-	-	-	-
updateDocProp	2	-	3	-	3	
fetchDocPerm	2	-	-	-	-	-
updateDocPerm	2	-	-		-	-
downloadContent	2	-	-	-	-	-
fetchDocHistory	2	-	-	-	-	-
deleteDocHistory	2	-	-	-	-	-
checkoutDoc	2	-	-	-	-	-
cancelCheckoutDoc	2	-	-	-	-	-
checkinDoc	2	-		-		
moveDoc	2	-	-	-	-	-
fetchSubFile	2	-	-	-	-	-

表 2-28 DecmDoc クラスのメソッド実行時に影響する setter メソッド (2/2)

操作メソッド	setter メソッド				
	setDocMaster	setRenditionType	setSubTraceMode	setSubFileList	
				DecmSubFile#setFilePath	DecmSubFile#setRenditionType
createDoc				1	
deleteDoc	-	-		-	-
trashDoc	-	-		-	-
trashBackDoc	-	-		-	-
fetchDocProp	-	-	-	-	-
updateDocProp				1	
fetchDocPerm	-	-	-	-	-
updateDocPerm	-	-	-	-	-
downloadContent	-	-	-	-	-
fetchDocHistory	-	-	-	-	-
deleteDocHistory	-	-		-	-
checkoutDoc	-	-		-	-
cancelCheckoutDoc	-	-		-	-
checkinDoc				1	
moveDoc	-	-		-	-
fetchSubFile	-	-	-	-	-

凡例

- : 必ず指定します。
- : 任意で指定します。
- : 指定の対象外です。

注 1

DecmDoc#setSubFileList メソッドは任意です。コールする場合、DecmSubFile#setFilePath メソッドのコールが必ず必要です。

注 2

操作対象のインスタンスが DecmDoc#createDoc を発行後の状態の場合は、必要ありません。

注 3

DecmDoc#setDocProp と DecmDoc#setFilePath は、必ずどちらかをコールする必要があります。

DecmFolder クラスのメソッドを実行する場合に影響する setter メソッドの関係を次の

表に示します。

表 2-29 DecmFolder クラスのメソッド実行時に影響する setter メソッド

操作メソッド	setter メソッド				
	setOiid	setParentOiid	seFolderProp	setFolderPerm	setCapacityMax
createNodeFolder	-			-	
createDocFolder	-				
createPubFolder	-				-
deleteFolder	1	-	-	-	-
fetchFolderProp	1	-	-	-	-
updateNodeFolderProp	1	-	2	-	2
updateDocFolderProp	1	-	2	-	2
updatePubFolderProp	1	-		-	-
fetchFolderPerm	1	-	-	-	-
updateFolderPerm	1	-	-		-
changeCapacityMode	1	-	-	-	-

凡例

- : 必ず指定します。
- : 任意で指定します。
- : 指定の対象外です。

注 1

操作対象インスタンスが DecmFolder#createNodeFolder, DecmFolder#createDocFolder, または DecmFolder#createPubFolder を発行後の状態の場合は, 必要ありません。

注 2

DecmFolder#setFolderProp と DecmFolder#setCapacityMax は, 必ずどちらかをコールしている必要があります。

2.9.2 文書空間オブジェクト操作時に必要なアクセス権限

文書空間オブジェクト操作時に必要なアクセス権限を次の表に示します。

なお, セキュリティ管理者は対象オブジェクトのアクセス権限に関係なく, すべての操作を実行できます。

表 2-30 文書空間オブジェクト操作時に必要なアクセス権限

クラス名	メソッド名	実行できるアクセス権限
DecmDoc	createDoc	登録先フォルダに対する「リンク」以上の権限を持つユーザ
	deleteDoc	文書に対する「全て」の権限を持つユーザ
	trashDoc	文書に対する「更新」以上の権限を持つユーザ
	trashBackDoc	文書に対する「更新」以上の権限を持つユーザ
	fetchDocProp	文書に対する「参照」以上の権限を持つユーザ
	updateDocProp	文書に対する「更新」以上の権限を持つユーザ
	fetchDocPerm	文書に対する「参照」以上の権限を持つユーザ
	updateDocPerm	文書に対する「所有者」の権限を持つユーザ
	downloadContent	文書に対する「参照」以上の権限を持つユーザ
	fetchDocHistory	文書に対する「参照」以上の権限を持つユーザ
	deleteDocHistory	文書に対する「全て」の権限を持つユーザ
	checkoutDoc	文書に対する「更新」以上の権限を持つユーザ
	cancelCheckoutDoc	文書に対する「更新」以上の権限を持つユーザ
	checkinDoc	文書に対する「更新」以上の権限を持ち、チェックアウトを実行したユーザ
moveDoc	次の条件を満たすユーザが実行できます。 <ul style="list-style-type: none"> 移動元文書に対する「所有者」の権限を持つユーザ 移動元、および移動先フォルダに対する「リンク」以上の権限を持つユーザ 	
fetchSubFile	文書に対する「参照」以上の権限を持つユーザ	
DecmSubFile	fetchFileProp	文書に対する「参照」以上の権限を持つユーザ
	downloadContent	文書に対する「参照」以上の権限を持つユーザ
DecmFolder	createNodeFolder	アクセス権限の指定はありません。
	createDocFolder	最上位フォルダの場合 <ul style="list-style-type: none"> アクセス権限の指定はありません。 下位フォルダの場合 <ul style="list-style-type: none"> 上位フォルダが文書フォルダの場合は、「リンク」以上の権限を持つユーザです。 上位フォルダがノードフォルダの場合は、アクセス権限の指定はありません。
	createPubFolder	最上位フォルダの場合 <ul style="list-style-type: none"> アクセス権限の指定はありません。 下位フォルダの場合 <ul style="list-style-type: none"> 上位フォルダが公開フォルダの場合は、公開されているユーザです。 上位フォルダがノードフォルダの場合は、アクセス権限の指定はありません。
	deleteFolder	フォルダに対する「全て」の権限を持つユーザ

クラス名	メソッド名	実行できるアクセス権限
	fetchFolderProp	フォルダに対する「参照」の権限を持つユーザ
	updateNodeFolderProp	フォルダに対する「更新」の権限を持つユーザ
	updateDocFolderProp	フォルダに対する「更新」の権限を持つユーザ
	updatePubFolderProp	フォルダに対する「更新」の権限を持つユーザ
	fetchFolderPerm	フォルダに対する「参照」の権限を持つユーザ
	updateFolderPerm	フォルダに対する「所有者」の権限を持つユーザ
	changeCapacityMode	フォルダに対する「更新」の権限を持つユーザ

2.9.3 文書の状態によるメソッドの実行可否

文書の状態によるメソッドの実行可否について、次の表に示します。

なお、操作を実行するユーザは、操作対象の文書に対して各メソッドが実行できるアクセス権限を持つことが前提となります。

表 2-31 文書の状態によるメソッドの実行可否

クラス	メソッド	一般文書				公開文書
		通常文書	チェックアウト中文書 ¹	ごみ箱の文書 ¹	リファレンスファイル文書化	
DecmDoc	createDoc		×	×	×	×
	deleteDoc		×		×	×
	trashDoc		×	×		×
	trashBackDoc		×			×
	fetchDocProp					
	updateDocProp		×	×	²	×
	fetchDocPerm					
	updateDocPerm		×	×		×
	downloadContent				×	
	fetchDocHistory					
	deleteDocHistory		×	×	×	×
	checkoutDoc		×	×	×	×
	cancelCheckoutDoc			×	×	×
	checkinDoc			×	×	×
moveDoc		×	×		×	

2. DocumentBroker Standard GUI Development Kit 提供のクラス一覧

クラス	メソッド	一般文書				公開文書
		通常文書	チェックアウト中文書 ¹	ごみ箱の文書 ¹	リファレンスファイル文書化	
	fetchSubFile					
DecmSubFile	fetchFileProp					
	downloadContent				x	

凡例

- : 実行できます。
- : 条件により実行できます。
- x : 実行できません。

注 1

チェックアウト中文書や、ごみ箱に存在する文書に対する操作は、実行したユーザに関係なく操作できます。

例えば、チェックアウトを実行したユーザ以外でも、「更新」以上のアクセス権限を持つユーザは、チェックアウトの取り消しを実行できます。

注 2

プロパティの更新はできますが、実体（コンテンツ）の更新はできません。

3

共通パッケージ提供クラスの 詳細

この章では、共通パッケージ
(`jp.co.hitachi_system.sContents.API.Common`) 提供のクラス、およびメソッドについて説明します。

3.1 DecmFactory クラス

3.2 DecmObject クラス

3.3 DecmCommonProp クラス

3.4 DecmCommonPerm クラス

3.5 DecmACLParm クラス

3.6 DecmDBControl クラス

3.1 DecmFactory クラス

DecmFactory クラスは、DocumentBroker Standard GUI Development Kit で提供するクラスのインスタンスを生成するための Factory クラスです。

このクラスのインスタンスは、文書空間とのセッションごとに一つ生成してください。サーバとのセッションに複数のオブジェクトを生成した場合、正しく動作しないおそれがあります。

次の項から、DecmFactory クラスのメソッドについて説明します。

3.1.1 DecmFactory

(1) 機能

コンストラクタです。

このクラスを利用して、DocumentBroker Standard GUI Development Kit が提供しているすべてのクラスのインスタンスが生成できます。ただし、例外クラスのインスタンスは生成できません。

(2) 形式

```
DecmFactory(  
    DbjDocSpace docSpace,  
    String   userId,  
    String   userName,  
    int      dbType  
)
```

(3) 引数

DbjDocSpace docSpace

ログインユーザに対する DbjDocSpace インターフェースを指定します。

String userId

ログインユーザのユーザ ID を指定します。

String userName

ログインユーザのユーザ名称を指定します。

int dbType

使用するデータベース種別を指定します。

指定できる値を次に示します。

- DecmFactory.DB_HIRDB

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 docSpace に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 userId の値に対する「通常文字列の設定チェック」に失敗した場合
- 引数 userName に対する「通常文字列の設定チェック」に失敗した場合
- 引数 dbType に対して、指定できる値以外の値を指定した場合

DecmApiDBException

- 文書管理サーバでエラーが発生した場合 (DbjException など)

3.1.2 createDecmACLPPerm

(1) 機能

アクセス権限表現用クラスのインスタンスを生成します。

このメソッドを利用して、DecmACLPPerm クラスのインスタンスを生成してください。

(2) 形式

```
DecmACLPPerm createDecmACLPPerm()
```

(3) 引数

なし

(4) 戻り値

アクセス権限表現用クラスのインスタンス

(5) 例外

なし

3.1.3 createDecmDoc

(1) 機能

文書操作クラスのインスタンスを生成します。

このメソッドを利用して、DecmDoc クラスのインスタンスを生成してください。

(2) 形式

```
DecmDoc createDecmDoc()
```

(3) 引数

なし

3. 共通パッケージ提供クラスの詳細

(4) 戻り値

文書操作クラスのインスタンス

(5) 例外

なし

3.1.4 createDecmSubFile

(1) 機能

関連ファイル操作クラスのインスタンスを生成します。

このメソッドを利用して、DecmSubFile クラスのインスタンスを生成してください。

(2) 形式

```
DecmSubFile createDecmSubFile()
```

(3) 引数

なし

(4) 戻り値

関連ファイル操作クラスのインスタンス

(5) 例外

なし

3.1.5 createDecmDocProp

(1) 機能

文書プロパティクラスのインスタンスを生成します。

このメソッドを利用して、DecmDocProp クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocProp createDecmDocProp()
```

(3) 引数

なし

(4) 戻り値

文書プロパティクラスのインスタンス

(5) 例外

なし

3.1.6 createDecmDocPerm

(1) 機能

文書アクセス権限クラスのインスタンスを生成します。

このメソッドを利用して、DecmDocPerm クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocPerm createDecmDocPerm()
```

(3) 引数

なし

(4) 戻り値

文書アクセス権限クラスのインスタンス

(5) 例外

なし

3.1.7 createDecmFolderTree

(1) 機能

フォルダツリー階層情報取得クラスのインスタンスを生成します。

このメソッドを利用して、DecmFolderTree クラスのインスタンスを生成してください。

(2) 形式

```
DecmFolderTree createDecmFolderTree()
```

(3) 引数

なし

(4) 戻り値

フォルダツリー階層情報取得クラスのインスタンス

(5) 例外

なし

3.1.8 createDecmFolder

(1) 機能

フォルダ操作クラスのインスタンスを生成します。

このメソッドを利用して、DecmFolder クラスのインスタンスを生成してください。

(2) 形式

```
DecmFolder createDecmFolder()
```

(3) 引数

なし

(4) 戻り値

フォルダ操作クラスのインスタンス

(5) 例外

なし

3.1.9 createDecmFolderProp

(1) 機能

フォルダプロパティクラスのインスタンスを生成します。

このメソッドを利用して、DecmFolderProp クラスのインスタンスを生成してください。

(2) 形式

```
DecmFolderProp createDecmFolderProp()
```

(3) 引数

なし

(4) 戻り値

フォルダプロパティクラスのインスタンス

(5) 例外

なし

3.1.10 createDecmFolderPerm

(1) 機能

フォルダアクセス権限クラスのインスタンスを生成します。

このメソッドを利用して、DecmFolderPerm クラスのインスタンスを生成してください。

(2) 形式

```
DecmFolderPerm createDecmFolderPerm()
```

(3) 引数

なし

(4) 戻り値

フォルダアクセス権限クラスのインスタンス

(5) 例外

なし

3.1.11 createDecmSelect

(1) 機能

文書検索クラスのインスタンスを生成します。

このメソッドを利用して、DecmSelect クラスのインスタンスを生成してください。

(2) 形式

```
DecmSelect createDecmSelect()
```

(3) 引数

なし

(4) 戻り値

文書検索クラスのインスタンス

(5) 例外

なし

3.1.12 createDecmCondition

(1) 機能

検索条件クラスのインスタンスを生成します。

このメソッドを利用して、DecmCondition クラスのインスタンスを生成してください。

(2) 形式

```
DecmCondition createDecmCondition()
```

(3) 引数

なし

(4) 戻り値

検索条件クラスのインスタンス

(5) 例外

なし

3.1.13 createDecmCondGroup

(1) 機能

組み合わせ検索条件クラスのインスタンス生成します。

このメソッドを利用して、DecmCondGroup クラスのインスタンスを生成してください。

(2) 形式

```
DecmCondGroup createDecmCondGroup()
```

(3) 引数

なし

(4) 戻り値

組み合わせ検索条件クラスのインスタンス

(5) 例外

なし

3.1.14 createDecmDocMasterList

(1) 機能

文書種別マスター一覧クラスのインスタンスを生成します。

このメソッドを利用して、DecmDocMasterList クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocMasterList createDecmDocMasterList()
```

(3) 引数

なし

(4) 戻り値

文書種別マスター一覧クラスのインスタンス

(5) 例外

なし

3.1.15 createDecmDocMaster

(1) 機能

文書種別マスタ操作作用クラスのインスタンスを生成します。

このメソッドを利用して、DecmDocMaster クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocMaster createDecmDocMaster()
```

(3) 引数

なし

(4) 戻り値

文書種別マスタ操作作用クラスのインスタンス

(5) 例外

なし

3.1.16 createDecmDocMasterProp

(1) 機能

文書種別マスタプロパティクラスのインスタンスを生成します。

このメソッドを利用して、DecmDocMasterProp クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocMasterProp createDecmDocMasterProp()
```

(3) 引数

なし

(4) 戻り値

文書種別マスタプロパティクラスのインスタンス

(5) 例外

なし

3.1.17 createDecmDocPropMaster

(1) 機能

属性値マスタ取得クラスのインスタンスを生成します。

このメソッドを利用して、DecmDocPropMaster クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocPropMaster createDecmDocPropMaster()
```

(3) 引数

なし

(4) 戻り値

属性値マスタ取得クラスのインスタンス

(5) 例外

なし

3.1.18 createDecmDocPropMasterProp

(1) 機能

属性値マスタプロパティクラスのインスタンスを生成します。

このメソッドを利用して、DecmDocPropMasterProp クラスのインスタンスを生成してください。

(2) 形式

```
DecmDocPropMasterProp createDecmDocPropMasterProp()
```

(3) 引数

なし

(4) 戻り値

属性値マスタプロパティクラスのインスタンス

(5) 例外

なし

3.1.19 initUserProp (文字列型)

(1) 機能

ユーザプロパティ、および文書種別マスタに使用するプロパティで、文字列型のプロパティをシステムに登録します。

このメソッドを複数回コールして、使用する文字列型のプロパティをすべて登録する必要があります。

なお、DecmDoc#setDocProp() メソッドの設定値を有効にする場合は、必ずこのメソッドでプロパティを定義してください。定義していないプロパティの場合、DecmDoc#setDocProp() メソッドで値を設定しても、その値は有効となりません。

(2) 形式

```
void initUserProp(
    String propName,
    int type,
    int size
)
```

(3) 引数

String propName

プロパティ名を指定します。

3. 共通パッケージ提供クラスの詳細

int type

プロパティ名に対する型を指定します。

指定できる値を次に示します。

- DecmFactory.PROP_TYPE_STRING (文字列型)

int size

プロパティ長を正の整数で指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 propName に対する「通常文字列の設定チェック」に失敗した場合
- 引数 type に対して、指定できる値以外を指定した場合
- 引数 size に対して、指定できる値以外を指定した場合
- 引数 propName に対して、すでに登録済みの値を指定した場合

3.1.20 initUserProp (文字列型以外)

(1) 機能

ユーザプロパティ、および文書種別マスタに使用するプロパティで、文字列型以外のプロパティをシステムに登録します。

このメソッドを複数回コールして、使用する文字列型以外のプロパティをすべて登録する必要があります。

なお、DecmDoc#setDocProp() メソッドの設定値を有効にする場合は、必ずこのメソッドでプロパティを定義してください。定義していないプロパティの場合、DecmDoc#setDocProp() メソッドで値を設定しても、その値は有効となりません。

(2) 形式

```
void initUserProp(  
    String propName,  
    int type  
)
```

(3) 引数

String propName

プロパティ名を指定します。

int type

プロパティ名の型を指定します。

指定できる値を次に示します。

- DecmFactory.PROP_TYPE_INTEGER (整数型)
- DecmFactory.PROP_TYPE_DATE (日付型)
- DecmFactory.PROP_TYPE_MASTER (マスタ型)

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 propNam に対する「通常文字列の設定チェック」に失敗した場合
- 引数 type に対して、指定できる値以外を指定した場合
- 引数 propName に対して、すでに登録済みの値を指定した場合

3.1.21 isCapacityUse

(1) 機能

フォルダ下ファイル容量制限機能の使用の有無を取得します。

このメソッドは、接続先の DocumentBroker サーバのフォルダ下ファイル容量制限機能使用の有無をチェックします。

(2) 形式

```
boolean isCapacityUse()
```

(3) 引数

なし

(4) 戻り値

- true : 容量制限機能を使用できます。
- false : 容量制限機能を使用できません。

(5) 例外

なし

3.2 DecmObject クラス

DecmObject クラスは、DocumentBroker Standard GUI Development Kit で提供するすべてのクラスの基底クラスです。ただし、例外クラスは DecmObject クラスから派生しません。

このクラスに DocumentBroker サーバとの接続情報を格納し、派生クラスはこのクラスに格納している情報をもとに、DocumentBroker サーバとのアクセスを行います。

なお、DecmObject クラスにはメソッドはありません。

3.3 DecmCommonProp クラス

DecmCommonProp クラスは、DocumentBroker Standard GUI Development Kit のプロパティの基底クラスです。プロパティを示すすべてのクラスは、このクラスを継承しています。

次の項から、DecmCommonProp クラスのメソッドについて説明します。

3.3.1 getOiid

(1) 機能

対象となるオブジェクトの OIID 値を取得します。

(2) 形式

```
String getOiid()
```

(3) 引数

なし

(4) 戻り値

対象となるオブジェクトの OIID 文字列

(5) 例外

なし

3.3.2 getStrProp

(1) 機能

文字列型、またはマスタ型のプロパティに対するプロパティ値を取得します。

(2) 形式

```
String getStrProp(  
    String fieldName  
)
```

(3) 引数

String fieldName

取得するプロパティ値のプロパティ名を指定します。

定義されていないプロパティ名を指定した場合は、無視されます。

3. 共通パッケージ提供クラスの詳細

(4) 戻り値

プロパティ値

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.3.3 setStrProp

(1) 機能

文字列型、またはマスタ型のプロパティに対するプロパティ値を設定します。

(2) 形式

```
void setStrProp(  
    String fieldName,  
    String strValue  
)
```

(3) 引数

String fieldName

設定するプロパティ値のプロパティ名を指定します。

定義されていないプロパティ名を指定した場合は、無視されます。

String strValue

プロパティ名に対する設定値を指定します。

プロパティの更新処理で使用する場合、null を指定すると、該当するフィールドを更新しません。

マスタ型の値を設定する場合は、設定する属性値コードを最上位からハイフン(・)区切りで指定します。

(凡例)

TOP-SUB01-SUB011-SUB0111

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.3.4 getIntProp

(1) 機能

整数型, または日付型のプロパティに対するプロパティ値を取得します。

(2) 形式

```
int getIntProp(
    String fieldName
)
```

(3) 引数

String fieldName

取得するプロパティ値のプロパティ名を指定します。

定義されていないプロパティ名を指定した場合は, 無視されます。

(4) 戻り値

プロパティ値

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.3.5 setIntProp

(1) 機能

整数型, または日付型のプロパティに対するプロパティ値を設定します。

(2) 形式

```
void setIntProp(
    String fieldName,
    int intValue
)
```

(3) 引数

String fieldName

設定するプロパティ値の, プロパティ名を指定します。

定義されていないプロパティ名を指定した場合は, 無視されます。

int intValue

プロパティ名に対する設定値を指定します。

(4) 戻り値

なし

3. 共通パッケージ提供クラスの詳細

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.3.6 getLongProp

(1) 機能

整数型 (Long 型) のプロパティに対するプロパティ値を取得します。

(2) 形式

```
long getLongProp(  
    String fieldName  
)
```

(3) 引数

String fieldName

設定するプロパティ値のプロパティ名を指定します。

定義されていないプロパティ名を指定した場合は、無視されます。

(4) 戻り値

プロパティ値

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.3.7 setLongProp

(1) 機能

整数型 (Long 型) のプロパティに対するプロパティ値を設定します。

(2) 形式

```
void setLongProp(  
    String fieldName,  
    long longValue  
)
```

(3) 引数

String fieldName

設定するプロパティ値のプロパティ名を指定します。

定義されていないプロパティ名を指定した場合は、無視されます。

long longValue

プロパティ名に対する設定値を指定します

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 fieldName に対する「通常文字列の設定チェック」に失敗した場合

3.4 DecmCommonPerm クラス

DecmCommonPerm クラスは、DocumentBroker Standard GUI Development Kit のアクセス権限情報の基底クラスです。DocumentBroker Standard GUI Development Kit で提供するすべてのアクセス権限情報を示すクラスは、このクラスを継承しています。

次の項から、DecmCommonPerm クラスのメソッドについて説明します。

3.4.1 getOwnerId

(1) 機能

オブジェクトの所有者のユーザ ID を取得します。

(2) 形式

```
String getOwnerId()
```

(3) 引数

なし

(4) 戻り値

オブジェクトの所有者のユーザ ID

(5) 例外

なし

3.4.2 setOwnerId

(1) 機能

オブジェクトの所有者のユーザ ID を設定します。

(2) 形式

```
void setOwnerId(  
    String ownerId  
)
```

(3) 引数

```
String ownerId
```

オブジェクトの所有者のユーザ ID を指定します。

(4) 戻り値

なし

(5) 例外

`DecmApiInvalidParamException`

引数 `ownerId` に対する「通常文字列の設定チェック」に失敗した場合

3.4.3 `getOwnerName`

(1) 機能

オブジェクトの所有者名称を取得します。

(2) 形式

```
String getOwnerName()
```

(3) 引数

なし

(4) 戻り値

オブジェクトの所有者名称

(5) 例外

なし

3.4.4 `setOwnerName`

(1) 機能

オブジェクトの所有者名称を設定します。

(2) 形式

```
void setOwnerName(  
    String ownerName  
)
```

(3) 引数

```
String ownerName
```

オブジェクトの所有者名称を指定します。

(4) 戻り値

なし

(5) 例外

`DecmApiInvalidParamException`

引数 `ownerName` に対する「通常文字列の設定チェック」に失敗した場合

3.4.5 `getEveryonePerm`

(1) 機能

全ユーザに対するアクセス権限を取得します。

(2) 形式

```
int getEveryonePerm()
```

(3) 引数

なし

(4) 戻り値

全ユーザに対するアクセス権限

戻り値を次に示します。

- `DecmACLPPerm.PERM_NONE`
- `DecmACLPPerm.PERM_READ`
- `DecmACLPPerm.PERM_LINK`
- `DecmACLPPerm.PERM_UPDATE`
- `DecmACLPPerm.PERM_FULL`

(5) 例外

なし

3.4.6 `setEveryonePerm`

(1) 機能

全ユーザに対するアクセス権限を設定します。

全ユーザに対するアクセス権限の設定、または変更ができないオブジェクトに対して操作を行う場合、このメソッドで値を設定していると `DecmApiNotSupport` 例外がスローされます。

(2) 形式

```
void setEveryonePerm(  
    int everyonePerm  
)
```

(3) 引数

```
int everyonePerm
```

全ユーザに対するアクセス権限を指定します。

指定できる値を次に示します。

- DecmACLPerm.PERM_NONE
- DecmACLPerm.PERM_READ
- DecmACLPerm.PERM_LINK
- DecmACLPerm.PERM_UPDATE
- DecmACLPerm.PERM_FULL

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 everyonePerm に対して、指定できる値以外を指定した場合

3.4.7 getACLList

(1) 機能

アクセス権限のリストを取得します。

(2) 形式

```
Vector<DecmACLPerm> getACLList()
```

(3) 引数

なし

(4) 戻り値

アクセス権限のリスト

(5) 例外

なし

3.4.8 setACLList

(1) 機能

アクセス権限のリストを設定します。

アクセス権限リストを設定、または変更できないオブジェクトに対して操作を行う場合、このメソッドで値を設定していると DecmApiNotSupport 例外がスローされます。

(2) 形式

```
void setACLList(
```

3. 共通パッケージ提供クラスの詳細

```
    Vector<DecmACLPerm> aclList  
)
```

(3) 引数

```
Vector<DecmACLPerm> aclList  
    アクセス権限一覧を指定します。
```

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 aclList に対する「通常オブジェクトの設定チェック」に失敗した場合

3.5 DecmACLPerm クラス

DecmACLPerm クラスは、DocumentBroker Standard GUI Development Kit のアクセス権限表現用クラスです。

次の項から、DecmACLPerm クラスのメソッドについて説明します。

3.5.1 getSubject

(1) 機能

アクセス権限対象の ID (ユーザ ID, 組織 ID, グループ ID) を取得します。

(2) 形式

```
String getSubject()
```

(3) 引数

なし

(4) 戻り値

アクセス権限対象の ID (ユーザ ID, 組織 ID, グループ ID)

(5) 例外

なし

3.5.2 setSubject

(1) 機能

アクセス権限対象の ID (ユーザ ID, 組織 ID, グループ ID) を設定します。

(2) 形式

```
void setSubject(  
    String subject  
)
```

(3) 引数

```
String subject
```

アクセス権限対象の ID (ユーザ ID, 組織 ID, グループ ID) を指定します。

(4) 戻り値

なし

3. 共通パッケージ提供クラスの詳細

(5) 例外

DecmApiInvalidParamException

引数 subject に対する「通常文字列の設定チェック」に失敗した場合

3.5.3 getSubjectType

(1) 機能

アクセス権限対象の種別（ユーザ，組織，グループ）を取得します。

(2) 形式

```
int getSubjectType()
```

(3) 引数

なし

(4) 戻り値

アクセス権限対象の種別（ユーザ，組織，グループ）

(5) 例外

なし

3.5.4 setSubjectType

(1) 機能

アクセス権限対象の種別（ユーザ，組織，グループ）を設定します。

なお，組織 ID とグループ ID は同じ値を設定します。

(2) 形式

```
void setSubjectType(  
    int subjectType  
)
```

(3) 引数

int subjectType

アクセス権限対象の種別です。

指定できる値を次に示します。

- DecmACLParm.PERM_TYPE_USER
- DecmACLParm.PERM_TYPE_GROUP

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 subjectType に対して、指定できる値以外を指定した場合

3.5.5 getPermission

(1) 機能

アクセス権限を取得します。

(2) 形式

```
int getPermission()
```

(3) 引数

なし

(4) 戻り値

アクセス権限

戻り値を次に示します。

- DecmACLPPerm.PERM_NONE
- DecmACLPPerm.PERM_READ
- DecmACLPPerm.PERM_LINK
- DecmACLPPerm.PERM_UPDATE
- DecmACLPPerm.PERM_FULLL

(5) 例外

なし

3.5.6 setPermission

(1) 機能

アクセス権限を設定します。

(2) 形式

```
void setPermission(
    int permission
)
```

3. 共通パッケージ提供クラスの詳細

(3) 引数

int permission

ACL のパーミッションを指定します。

指定できる値を次に示します。

- DecmACLParm.PERM_NONE
- DecmACLParm.PERM_READ
- DecmACLParm.PERM_LINK
- DecmACLParm.PERM_UPDATE
- DecmACLParm.PERM_FULL

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 permission に対して、指定できる値以外を指定した場合

3.6 DecmDBControl クラス

DecmDBControl クラスは、DocumentBroker Standard GUI Development Kit のデータベース操作の基底クラスです。

DocumentBroker Standard GUI Development Kit で提供するすべてのデータベースを操作するクラスは、このクラスの派生クラスです。

なお、DecmDBControl クラスにはメソッドはありません。

4

文書操作パッケージ提供クラスの詳細

この章では、文書操作パッケージ (jp.co.hitachi_system.sContents.API.Document) 提供のクラス、およびメソッドについて説明します。

-
- 4.1 DecmDoc クラス

 - 4.2 DecmDocProp クラス

 - 4.3 DecmDocPerm クラス

 - 4.4 DecmSubFile クラス
-

4.1 DecmDoc クラス

DecmDoc クラスは、文書操作を扱うクラスです。このクラスのインスタンスで、1 文書のオブジェクトを表現します。

このクラスは以下の機能を提供します

- 文書の作成
- 文書の削除
- ごみ箱への移動
- ごみ箱から元に戻す処理
- 文書のプロパティ取得
- 文書のプロパティ更新
- 文書のアクセス権限取得
- 文書のアクセス権限更新
- 文書実体（コンテンツ）のダウンロード
- 文書の履歴取得
- 文書の履歴削除
- チェックアウト処理
- キャンセルチェックアウト処理
- チェックイン処理
- 文書の移動
- 関連ファイル一覧の取得

プロパティ一覧

DecmDoc クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 4-1 DecmDoc クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド	更新可能メソッド
		fetchDocProp() fetchDocHistory()	createDoc() updateDocProp() checkinDoc()
dmaProp_OIID	String		-
esysProp_Name	String		

プロパティ名	データ型	参照可能メソッド	更新可能メソッド
		fetchDocProp() fetchDocHistory()	createDoc() updateDocProp() checkinDoc()
dbrProp_OwnerId	String		-
dbuUserProp_OwnerName	String		-
esysProp_CreateTime	Integer		-
esysProp_Creator	String		-
esysProp_ModifyTime	Integer		-
esysProp_Modifier	String		-
dbuUserProp_ModifierName	String		-
dbuUserProp_ReservedUserId	String		-
dbuUserProp_ReservedUserName	String		-
esysProp_Comment	String		
ecmProp_DocTypeOIID	String		-
ecmProp_DocTypeOIID_FULL	String		-
ecmProp_CHOiid	String		-
ecmProp_CHOiid_FULL	String		-
ecmProp_DocType	Integer		-
ecmProp_FileName	String		-
ecmProp_FileSize	Integer		-
ecmProp_ParentContainer	String		-
ecmProp_ParentContainer_FULL	String		-
ecmProp_DeleteFlag	Integer		-
ecmProp_DeleteUserId	String		-
ecmProp_DeleteTime	Integer		-
ecmProp_DocNumber	String		-
ecmProp_UserInt1 ~ ecmProp_UserInt5	Integer		
ecmProp_UserStr1 ~ ecmProp_UserStr5	String		
ecmProp_DocTypePropInt1 ~ ecmProp_DocTypePropInt10	Integer		
ecmProp_DocTypePropStr1 ~ ecmProp_DocTypePropStr10	String		
ecmProp_DocTypePropLongStr1 ~ ecmProp_DocTypePropLongStr5	String		

(凡例)

4. 文書操作パッケージ提供クラスの詳細

- ：デフォルトで取得，および更新ができます。
- ：該当しません。

注

DecmFactory#initUserProp メソッド，DecmFactory#initUserProp メソッドを使用してプロパティ名を登録しておく必要があります。

次の項から，DecmDoc クラスのメソッドについて説明します。

4.1.1 getOiid

(1) 機能

操作対象文書のバージョンニングオブジェクトの OIID を取得します。

(2) 形式

```
String getOiid()
```

(3) 引数

なし

(4) 戻り値

操作対象文書のバージョンニングオブジェクトの OIID 値

(5) 例外

なし

4.1.2 setOiid

(1) 機能

操作対象文書のバージョンニングオブジェクトの OIID を設定します。

このメソッドは既存の文書にアクセスする場合に最初にコールしてください。なお，OIID に対する文書の存在チェックは行いません。

(2) 形式

```
void setOiid(  
    String oidd  
)
```

(3) 引数

String oidd

操作対象文書のバージョンニングオブジェクトの OIID を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 OIID に対する「133 桁 OIID の設定チェック」に失敗した場合

4.1.3 setFolderOiid

(1) 機能

文書の上位フォルダの OIID を設定します。

文書作成前に、必ずこのメソッドをコールしてください。なお、OIID に対するフォルダの存在チェックは行いません。

(2) 形式

```
void setFolderOiid(
    String folderOiid
)
```

(3) 引数

String folderOiid

文書を登録するフォルダの OIID を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 folderOiid に対する「133 桁 OIID の設定チェック」に失敗した場合

4.1.4 setDocProp

(1) 機能

文書のプロパティを設定します。

文書作成前に、必ずこのメソッドをコールしてください。文書の作成、プロパティの更新、およびチェックインの実行前にこのメソッドをコールすると、文書のプロパティ値の変更ができます。

文書種別マスタに定義されているプロパティ一覧と、docProp に設定しているプロパティの整合性についてはチェックを行いません。ユーザアプリケーション側で整合性を

4. 文書操作パッケージ提供クラスの詳細

確保してください。

なお、マスタ型のプロパティに対しては指定値の妥当性チェックを行います。

(2) 形式

```
void setDocProp(  
    DecmDocProp docProp  
)
```

(3) 引数

DecmDocProp docProp
 文書のプロパティを指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 docProp に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 docProp に対する「プロパティの設定規則チェック」に失敗した場合

4.1.5 setDocPerm

(1) 機能

文書のアクセス権限を設定します。

文書の作成、およびアクセス権限の変更の実行前にこのメソッドをコールすると、文書のアクセス権限の更新ができます。

このメソッドを使って、文書の所有者にメソッド実行ユーザと異なるユーザを設定する場合、セキュリティ管理者で実行してください。一般ユーザで実行した場合、DecmApiAccessException 例外が発生する場合があります。

なお、画面 GUI の定義である配置ディスクリプタ (web.xml) の app.ACLObjNum (アクセス権限リストの設定数) の設定値を超えるアクセス権限のリストを指定しても、エラーにはなりません。

配置ディスクリプタの詳細については、マニュアル「uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド」を参照してください。

(2) 形式

```
void setDocPerm(  
    DecmDocPerm docPerm  
)
```

(3) 引数

DecmDocPerm docPerm

文書のアクセス権限を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 docPerm に対する「通常オブジェクトの設定チェック」に失敗した場合

4.1.6 setFilePath

(1) 機能

文書の实体 (コンテンツ) ファイルのローカルディスク上でのパスを設定します。

このメソッドは、文書の作成、プロパティの更新、およびチェックイン実行の前にコールしてください。

(2) 形式

```
void setFilePath(  
    String filePath  
)
```

(3) 引数

String filePath

文書のコンテンツファイルのローカルディスク上でのパスを指定します。

null, または「」(空文字) を指定した場合は、空文書が作成されます。

ただし、チェックイン時には必ず文書ファイルを指定してください。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 filePath に指定したファイル名が 255 バイト (UTF-8) より大きい場合

4.1.7 setTextPath

(1) 機能

全文検索用のテキストファイルの、ローカルディスク上でのパスを設定します。

4. 文書操作パッケージ提供クラスの詳細

このメソッドは、文書の作成、プロパティの更新、およびチェックイン実行の前にコールしてください。

(2) 形式

```
void setTextPath(  
    String textPath  
)
```

(3) 引数

String textPath

全文検索用のテキストファイルの、ローカルディスク上でのパスを指定します。
null, または「」(空文字)を指定した場合は、全文検索用のテキストファイルを登録しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 textPath に指定したファイル名が 255 バイト (UTF-8) より大きい場合

4.1.8 setDocMaster

(1) 機能

文書種別マスタの OIID を設定します。

(2) 形式

```
void setDocMaster(  
    String docMasterOiid  
)
```

(3) 引数

String docMasterOiid

使用する文書種別マスタの OIID を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 docMasterOiid に対する「133 桁 OIID の設定チェック」に失敗した場合

4.1.9 setRenditionType

(1) 機能

登録する文書ファイルの参照用のファイル (PDF) を設定します。

このメソッドをコールすると、PDF の格納領域を作成します。

なお、PDF を作成する対象となる文書の拡張子は、uCosminexus DocumentBroker Version 3 Standard GUI の配置ディスクリプタ (web.xml) の app.RenderingExtList で設定している拡張子とあわせてください。

詳細については、マニュアル「uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド」を参照してください。

(2) 形式

```
void setRenditionType(
    String renditionType
)
```

(3) 引数

String renditionType

登録する文書ファイルの参照用のファイル (PDF) を設定する場合は、
「application/pdf」を指定してください。

null、または「」(空文字)を指定した場合は、文書を PDF に変換しません。

(4) 戻り値

なし

(5) 例外

なし

4.1.10 setSubFileList

(1) 機能

関連ファイルリストを設定します。

(2) 形式

```
void setSubFileList(
    Vector<DecmSubFile> subFileList
)
```

(3) 引数

Vector<DecmSubFile> subFileList

4. 文書操作パッケージ提供クラスの詳細

関連ファイルリストを指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 subFileList に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 subFileList に指定した要素に対して、ファイルのローカルディスク上でのパス (setFilePath メソッド) を発行してない場合
- 関連ファイル内で、ファイル名に同じ名称を指定している場合

4.1.11 isSubTraceMode

(1) 機能

関連ファイルの監査証跡ログの出力有無を取得します。

(2) 形式

```
boolean isSubTraceMode()
```

(3) 引数

なし

(4) 戻り値

関連ファイルの監査証跡ログの出力有無

(5) 例外

なし

4.1.12 setSubTraceMode

(1) 機能

主ファイルと関連ファイルについて操作するメソッドで、関連ファイルの監査証跡ログを出力するかどうかを設定します。

このメソッドを発行しない場合のデフォルト値は、false (出力しない) です。

DecmDoc クラス内で、引数に outputLog が必要なメソッドに対して false を指定した場合、このメソッドで true (出力する) を指定していても、関連ファイルの監査証跡ログは出力されません。

(2) 形式

```
void setSubTraceMode(
    boolean subTraceMode
)
```

(3) 引数

```
boolean subTraceMode
```

関連ファイルの監査証跡ログの出力するかどうかを指定します。

- true : 関連ファイルの監査証跡ログを出力します。
- false : 関連ファイルの監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

なし

4.1.13 createDoc

(1) 機能

一般文書を作成します。

事前に関連ファイルリストを設定している場合は、関連ファイル付きの文書を作成しません。

- 主ファイル

文書の实体 (コンテンツ) 付き文書、および空文書を作成できます。

アクセス権限情報を指定した場合、アクセス権限情報も更新されます。

指定しない場合は、登録フォルダに設定されているアクセス権限を適用します。

- 関連ファイル

文書のコンテンツ付きファイルを作成できます。

ファイルサイズとファイル名以外のプロパティ、アクセス権限などは主ファイルと同じ値が設定されます。

(2) 形式

```
String createDoc(
    boolean docNumber,
    boolean outputLog
)
```

(3) 引数

```
boolean docNumber
```

文書番号の採番の実施フラグを指定します。

なお、setDocMaster() が未発行の場合、この引数に true (採番する) を指定しても

4. 文書操作パッケージ提供クラスの詳細

指定は無視されます。

- true : 文書番号を採番します。
- false : 文書番号を採番しません。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

作成した一般文書のバージョンングオブジェクトの OIID 値

(5) 例外

DecmApiInvalidInitialException

- setFolderOiid メソッドを発行していない場合
- setDocProp メソッドを発行していない場合
- setDocProp メソッドを発行しているが、esysProp_Name を設定していない場合
- setFilePath メソッドで指定したファイルと、setSubFileList メソッドで指定したファイルのファイル名が重複している場合

DecmApiIOException

setFilePath メソッド、setTextPath メソッド、または setSubFileList メソッドで指定したファイルの読み込み中に、エラーが発生した場合

DecmApiFolderNotFoundException

上位フォルダが存在しない場合

DecmApiCapacityException

フォルダの容量制限を超えた場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiFolderNotCorrectException

- 上位フォルダに文書フォルダ以外のフォルダを指定した場合
- 文書が作成できない文書フォルダを指定した場合
- 上位フォルダの文書種別マスタと、文書の文書種別マスタが矛盾している場合

DecmApiDocNumNotCorrectException

引数 docNumber に true を指定した場合で、採番できる有効な文書種別マスタを指定していないとき

DecmApiAccessException

文書管理サーバでアクセス権限エラーが発生した場合
(DbjAccessControlException)

DecmApiInvalidPropValueException

指定した文書のプロパティ値が不正な場合

DecmApiInvalidFileException

登録するファイルが不正の場合、または0バイトのファイルを登録した場合

DecmApiACLOutOfRangeException

作成できる個数以上の ACL を指定している場合

DecmApiNotSupport

- アクセス権限情報の指定が不正な場合
- 指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.14 deleteDoc

(1) 機能

一般文書をデータベースから完全に削除します。

操作対象の一般文書に関連ファイルが存在している場合、関連ファイルも同時に削除します。

(2) 形式

```
void deleteDoc(
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

指定した一般文書がすでに削除された場合、または存在しない場合

4. 文書操作パッケージ提供クラスの詳細

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.15 trashDoc

(1) 機能

一般文書をごみ箱に移動します。

操作対象の一般文書に関連ファイルが存在している場合、関連ファイルも同時にごみ箱に移動します。

(2) 形式

```
void trashDoc(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

指定した一般文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.16 trashBackDoc

(1) 機能

ごみ箱から文書を戻します。

対象文書に関連ファイルが存在している場合、関連ファイルも同時にごみ箱から戻します。

(2) 形式

```
void trashBackDoc (
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

指定した文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.17 fetchDocProp

(1) 機能

文書の最新バージョンのプロパティを取得します。

対象文書に関連ファイルが存在しているかどうかにかかわらず、主ファイルのプロパティだけを取得します。

関連ファイルのプロパティを取得する場合は、DecmSubFile#fetchFileProp メソッドをコールしてください。

(2) 形式

```
DecmDocProp fetchDocProp(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書のプロパティ

(5) 例外

DecmApiInvalidInitialException
setOiid メソッドを発行していない場合

DecmApiDocNotFoundException
文書がすでに削除された場合、または存在しない場合

DecmApiAccessException
指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException
文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.18 updateDocProp

(1) 機能

一般文書の最新バージョンのプロパティを更新します。

操作対象の一般文書の实体 (コンテンツ) ファイルが登録されていない場合、

setFilePath メソッド, setTextPath メソッド, setRenditionType メソッドをコールすると、コンテンツファイルの追加ができます。

コンテンツファイルが登録されている場合、setFilePath メソッド, setTextPath メソッド, setRenditionType メソッドをコールすると、コンテンツファイルの差し替えができます。

文書種別マスタが設定されていない一般文書の場合、このメソッドをコールする前に setDocMaster メソッドをコールすると、文書種別マスタを設定できます。

操作対象の一般文書が文書番号を採番済みの場合、採番に使用しているプロパティの変更はできません。

事前に関連ファイルリストを設定している場合は、関連ファイルの文書のコンテンツを更新することができます。

(2) 形式

```
void updateDocProp(
    boolean docNumber,
    boolean outputLog
)
```

(3) 引数

boolean docNumber

文書採番の実施フラグを指定します。

採番済みの一般文書に対してこの引数を指定した場合は、無視されます。

- true : 文書番号を採番します。
- false : 文書番号を採番しません。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setDocProp メソッドと setFilePath メソッドのどちらも発行していない場合
- setFilePath メソッドで指定したファイルと、setSubFileList メソッドで指定したファイルのファイル名が重複している場合
- setSubFileList メソッドで指定したファイルが、更新対象の文書の関連ファイルに存在しない場合

4. 文書操作パッケージ提供クラスの詳細

DecmApiIOException

setFilePath メソッド, setTextPath メソッド, または setSubFileList メソッドで指定したファイルの読み込み中にエラーが発生した場合

DecmApiDocNotFoundException

指定した一般文書がすでに削除された場合, または存在しない場合

DecmApiDocNumNotCorrectException

引数 docNumber に true を指定した場合で, 採番できる有効な文書種別マスタを指定していないとき

DecmApiCapacityException

フォルダの容量制限を超えた場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiInvalidPropValueException

指定した一般文書のプロパティ値が不正な場合

DecmApiInvalidDocMasterException

指定した文書種別マスタが不正な場合

DecmApiInvalidFileException

登録するファイルが不正な場合, または 0 バイトのファイルを登録した場合

DecmApiNotSupport

- 採番済みの文書に対して, 採番に使用しているプロパティを更新しようとした場合
- 文書種別マスタを設定済みの文書に対して, 文書種別マスタを更新しようとした場合
- 指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.19 fetchDocPerm

(1) 機能

文書のアクセス権限情報を取得します。

(2) 形式

```
DecmDocPerm fetchDocPerm(  
    boolean outputLog
```

)

(3) 引数

boolean outputLog:

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書に対するアクセス権限情報

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.20 updateDocPerm

(1) 機能

文書のアクセス権限情報を更新します。

対象文書に関連ファイルが存在している場合、関連ファイルのアクセス権限も同時に更新します。

(2) 形式

```
void updateDocPerm(
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

4. 文書操作パッケージ提供クラスの詳細

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setDocPerm メソッドを発行していない場合

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiNotSupport

指定した文書に対するアクセス権限情報の更新をサポートしていない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiACLOutOfRangeException

作成できる個数以上の ACL を指定している場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.21 downloadContent

(1) 機能

文書の実体 (コンテンツ) をダウンロードします。

対象文書に関連ファイルが存在しているかどうかに関係なく、主ファイルのコンテンツだけをダウンロードします。

関連ファイルのコンテンツをダウンロードする場合は、
DecmSubFile#downloadContent メソッドをコールしてください。

(2) 形式

```
String downloadContent(  
    String dvOiid,  
    String filePath,  
    boolean bFileMode,  
    boolean outputLog  
)
```

(3) 引数

String dvOiid

対象文書のバージョンオブジェクトの OIID を指定します。

null, または「」(空文字)を指定した場合は, 最新バージョンの文書のコンテンツがダウンロードされます。

String filePath

文書のコンテンツのダウンロード先のファイルパスを, フルパスで指定します。

boolean bFileMode

文書のコンテンツをダウンロードするファイルを指定します。

- true : オリジナルファイルのコンテンツをダウンロードします。
- false : 参照用のファイル (PDF) に変換された文書のコンテンツをダウンロードします。

文書のコンテンツが存在しない場合は, DecmApiFileNotFoundException 例外をスローします。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書のコンテンツのダウンロード先ファイルのフルパス

(5) 例外

DecmApiInvalidParamException

- 引数 dvOiid が null, または「」(空文字)でなく「133 桁 OIID の設定チェック」に失敗した場合
- 引数 filePath に対する「通常文字列の設定チェック」に失敗した場合

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiIOException

引数 filePath で指定したファイルへの書き込み中にエラーが発生した場合

DecmApiDocNotFoundException

文書がすでに削除された場合, または存在しない場合

DecmApiAccessException

文書管理サーバでアクセス権限エラーが発生した場合
(DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiFileNotFoundException

4. 文書操作パッケージ提供クラスの詳細

文書のコンテンツが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.22 fetchDocHistory

(1) 機能

文書の履歴を取得します。

取得するのは、文書のプロパティです。

操作対象の文書に関連ファイルが存在しているかどうかにかかわらず、主ファイルの履歴だけを取得します。

取得した主ファイルの履歴は、バージョンの降順でソートします。

古い版 (バージョン) の関連ファイルの履歴を取得する場合は、fetchSubFile メソッドの引数に取得したい版 (バージョン) の主ファイルのバージョンオブジェクトの OIID を指定してください。

(2) 形式

```
Vector<DecmDocProp> fetchDocHistory(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書の履歴 (プロパティリスト形式)

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.23 deleteDocHistory

(1) 機能

一般文書の履歴を削除します。

操作対象の一般文書に関連ファイルが存在している場合、関連ファイルの履歴も同時に削除します。

(2) 形式

```
void deleteDocHistory(
    String dvOiid,
    boolean outputLog
)
```

(3) 引数

String dvOiid

操作対象の一般文書のバージョンオブジェクトの OIID を指定します。

null, または「」(空文字) を指定した場合、最新バージョンの一般文書の履歴を削除します。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 dvOiid が null, または「」(空文字) でなく「133桁 OIID の設定チェック」に失敗した場合

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiLastVersionException

指定した一般文書の最後の 1 バージョンを削除しようとした場合
(DbjLastVersionException)

DecmApiDocNotFoundException

4. 文書操作パッケージ提供クラスの詳細

指定した一般文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.24 checkoutDoc

(1) 機能

一般文書をチェックアウトします。

一般文書の実体 (コンテンツ) を取得する場合は、downloadContent メソッドをコールしてください。

操作対象の一般文書に関連ファイルが存在している場合、関連ファイルも同時にチェックアウトします。

(2) 形式

```
void checkoutDoc(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiAlreadyCheckoutException

文書管理サーバでチェックアウト状態不正エラー (すでにチェックアウト済み) が発生した場合 (DbjAlreadyCheckOutException)

DecmApiDocNotFoundException

指定した一般文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.25 cancelCheckoutDoc

(1) 機能

文書のチェックアウトをキャンセルします。

対象文書に関連ファイルが存在している場合、関連ファイルも同時にチェックアウトをキャンセルします。

(2) 形式

```
void cancelCheckoutDoc(
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiNotCheckoutException

文書管理サーバでチェックアウト状態不正エラー (未チェックアウト) が発生した場合 (DbjNotCheckOutException)

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

4. 文書操作パッケージ提供クラスの詳細

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.26 checkinDoc

(1) 機能

文書をチェックインします。

文書種別マスタが設定されていない文書の場合、このメソッドをコールする前に setDocMaster メソッドをコールすると、文書種別マスタを設定できます。

操作対象文書が文書番号を採番済みの場合、採番に使用しているプロパティの変更はできません。

事前に関連ファイルリストを設定している場合は、関連ファイルの文書の实体 (コンテント) を更新することができます。

(2) 形式

```
String checkinDoc(  
    boolean docNumber,  
    boolean outputLog  
)
```

(3) 引数

boolean docNumber

文書採番の実施フラグを指定します。

採番済みの文書に対して指定した場合は、無視されます。

- true : 文書番号を採番します。
- false : 文書番号を採番しません。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

チェックイン後の文書のバージョンオブジェクトの OIID

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setFilePath メソッドを発行していない場合
- setFilePath メソッドを発行しているが、null、または「」(空文字)を指定している場合
- setFilePath メソッドで指定したファイルと、setSubFileList メソッドで指定したファイルのファイル名が重複している場合

DecmApiIOException

setFilePath メソッド、setTextPath メソッド、または setSubFileList メソッドで指定したファイルの読み込み中にエラーが発生した場合

DecmApiNotCheckoutException

文書管理サーバでチェックアウト状態不正エラー（未チェックアウト）が発生した場合（DbjNotCheckOutException）

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiCapacityException

フォルダの容量制限を超えた場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在していない場合

DecmApiDocNumNotCorrectException

引数 docNumber に true を指定した場合で、採番可能な有効な文書種別マスタを指定していないとき

DecmApiAccessException

指定した文書に対してアクセス権限がない場合（DbjAccessControlException）

DecmApiInvalidPropValueException

指定した文書のプロパティ値が不正な場合

DecmApiInvalidFileException

登録するファイルが不正な場合、または0バイトのファイルを登録した場合

DecmApiNotSupport

- 採番済みの文書に対して、採番に使用しているプロパティを更新しようとした場合
- 文書種別マスタを設定済みの文書に対して、文書種別マスタを更新しようとした場合
- 指定した操作をサポートしていない場合

DecmApiInvalidDocMasterException

指定した文書種別マスタが不正な場合

4. 文書操作パッケージ提供クラスの詳細

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.27 moveDoc

(1) 機能

一般文書を移動します。

移動元の一般文書に対して「所有者」権限が必要です。

また、移動元のフォルダ、および移動先フォルダに対して、次のどれかのアクセス権限が必要です。

- すべて
- 更新
- リンク

移動元の一般文書には、チェックアウト中以外の一般文書を指定できます。

移動先フォルダには、一般文書の作成ができる文書フォルダを指定できます。

操作対象の一般文書に関連ファイルが存在している場合、関連ファイルも同時に移動します。

(2) 形式

```
void moveDoc(  
    String targetFolderOiid,  
    boolean outputLog  
)
```

(3) 引数

String targetFolderOiid

一般文書の移動先フォルダの OIID を指定します。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログ出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 targetFolderOiid に対する「133 桁 OIID の設定チェック」に失敗した場合

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

指定した一般文書がすでに削除された場合、または存在しない場合

DecmApiFolderNotFoundException

一般文書の移動先に指定したフォルダが存在しない場合

DecmApiCapacityException

一般文書の移動先に指定したフォルダの容量制限を超えた場合

DecmApiAccessException

指定した一般文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiNotSupport

- 公開文書を移動しようとした場合
- 移動先に指定できないフォルダを指定した場合
- 指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.28 fetchSubFile

(1) 機能

主ファイルのバージョンオブジェクトと関連付けられた、関連ファイルのバージョンオブジェクトのリストを取得します。

関連ファイルのプロパティ、および実体 (コンテンツ) を取得する場合は、このメソッドで取得した DecmSubFile クラスのインスタンスを使用してください。

取得した関連ファイルリストは、ファイル名の昇順でソートします。

(2) 形式

```
Vector<DecmSubFile> fetchSubFile(
    String dvOiid
)
```

(3) 引数

String dvOiid

対象文書のバージョンオブジェクトの OIID を指定します。

null、または「」(空文字) を指定した場合、最新のバージョンの主ファイルに関連付けられた関連ファイルリストを取得します。

4. 文書操作パッケージ提供クラスの詳細

(4) 戻り値

主ファイルと関連付けられた関連ファイルリスト

(5) 例外

DecmApiInvalidParamException

引数 dvOiid が、null、または「」(空文字)でなく「133桁 OIID の設定チェック」に失敗した場合

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

4.1.29 lockRead

(1) 機能

文書に対する参照ロックの予約を行います。

このメソッドのコール後に文書データに対するアクセスを行うと、参照ロックが取得されます。

なお、1 トランザクション中にこのメソッドと lockWrite メソッドを複数回コールした場合、最後にコールしたメソッドが有効となります。

(2) 形式

```
void lockRead()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

4.1.30 lockWrite

(1) 機能

文書に対する更新ロックの予約を行います。

このメソッドのコール後に文書データに対するアクセスを行うと、更新ロックが取得されます。

なお、1 トランザクション中にこのメソッドと lockRead メソッドを複数回コールした場合は、最後にコールしたメソッドが有効となります。

(2) 形式

```
void lockWrite()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

4.2 DecmDocProp クラス

DecmDocProp クラスは、文書プロパティを扱うクラスです。このクラスのインスタンスで、文書に設定されているプロパティを表現します。

以下の機能で、このクラスのインスタンスを使用します。

文書の作成

文書のプロパティ取得

文書のプロパティ更新

文書のチェックイン処理

スーパークラスは、DecmCommonProp クラスです。

なお、DecmCommonProp クラスにはメソッドはありません。

4.3 DecmDocPerm クラス

DecmDocPerm クラスは、文書アクセス権限を扱うクラスです。このクラスのインスタンスで、文書に設定されているアクセス権限情報を表現します。

以下の機能で、このクラスのインスタンスを使用します。

- 文書のアクセス権限取得

- 文書のアクセス権限更新

スーパークラスは、DecmCommonPerm クラスです。

なお、DecmDocProp クラスにはメソッドはありません。

4.4 DecmSubFile クラス

DecmSubFile クラスは、関連ファイルの操作を扱うクラスです。このクラスのインスタンスで、一つの文書の関連ファイルのオブジェクトを表現します。

以下の機能で、このクラスのインスタンスを使用します。

関連ファイルのプロパティ取得

関連ファイルの実体（コンテンツ）のダウンロード

スーパークラスは、DecmDBControl クラスです。

プロパティ一覧

DecmSubFile クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 4-2 DecmDoc クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド
		fetchFileProp()
ecmProp_FileName	String	
ecmProp_FileSize	Integer	

次の項から、DecmSubFile クラスのメソッドについて説明します。

4.4.1 setFilePath

(1) 機能

文書の実体（コンテンツ）ファイルのローカルディスク上でのパスを設定します。

このメソッドは、主ファイルに対する文書の作成、プロパティの更新、およびチェックインの実行前にコールしてください。

(2) 形式

```
void setFilePath(
    String filePath
)
```

(3) 引数

String filePath

文書のコンテンツファイルのローカルディスク上でのパスを指定します。

主ファイルとは異なり、関連ファイルに対して null、または「」（空文字）の指定

はできません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 filePath に指定したファイル名が 255 バイト (UTF-8) より大きい場合
- 引数 String filePath に, null, または「」(空文字)を指定している場合
- 引数 filePath に対する「通常文字列の設定チェック」に失敗した場合

4.4.2 setRenditionType

(1) 機能

登録する文書ファイルの参照用のファイル (PDF) を設定します。

このメソッドをコールすると,PDF の格納領域を作成します。

なお,PDF を作成する対象となる文書の拡張子は, uCosminexus DocumentBroker Version 3 Standard GUI の配置ディスクリプタ (web.xml) の app.RenderingExtList で設定している拡張子とあわせてください。

詳細については, マニュアル「uCosminexus DocumentBroker Version 3 Standard GUI システム導入・運用ガイド」を参照してください。

(2) 形式

```
void setRenditionType(
    String renditionType
)
```

(3) 引数

String renditionType

登録する文書ファイルの参照用のファイル (PDF) を設定する場合は, 「application/pdf」を指定してください。
null, または「」(空文字)を指定した場合は, 文書を PDF に変換しません。

(4) 戻り値

なし

(5) 例外

なし

4.4.3 fetchFileProp

(1) 機能

文書のプロパティを取得します。

関連ファイル一覧取得メソッド (DecmDoc#fetchSubFile) の第一引数で指定した主ファイルのバージョンと関連付けられたバージョンが、プロパティ取得の対象となります。

(2) 形式

```
DecmDocProp fetchFileProp(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書のプロパティ

(5) 例外

DecmApiInvalidInitialException

DecmDoc#fetchSubFile で取得したインスタンス以外のインスタンスを使用して、このメソッドをコールした場合

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

指定した文書に対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException:

文書管理サーバでエラーが発生した場合 (DbjException など)

4.4.4 downloadContent

(1) 機能

文書の实体 (コンテンツ) をダウンロードします。

関連ファイル一覧取得メソッド (DecmDoc#fetchSubFile メソッド) の第一引数で指定した主ファイルのバージョンと関連付けられたバージョンが、文書のコンテンツのダウ

ンロード対象となります。

(2) 形式

```
String downloadContent(
    String filePath,
    boolean bFileMode,
    boolean outputLog
)
```

(3) 引数

String filePath

文書のコンテンツのダウンロード時のファイルパスを、フルパスで指定します。

boolean bFileMode

文書のコンテンツをダウンロードするファイルを指定します。

- true : オリジナルファイルのコンテンツをダウンロードします。
- false : 参照用のファイル (PDF) に変換された文書のコンテンツをダウンロードします。

文書のコンテンツが存在しない場合、DecmApiFileNotFoundException 例外をスローします。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

文書のコンテンツのダウンロード先ファイルのフルパス

(5) 例外

DecmApiInvalidParamException

引数 filePath に対する「通常文字列の設定チェック」に失敗した場合

DecmApiInvalidInitialException

DecmDoc#fetchSubFile で取得したインスタンス以外のインスタンスを使用して、このメソッドをコールした場合

DecmApiIOException

引数 filePath で指定したファイルへの書き込み中にエラーが発生した場合 (java.io.IOException)

DecmApiDocNotFoundException

文書がすでに削除された場合、または存在しない場合

DecmApiAccessException

4. 文書操作パッケージ提供クラスの詳細

文書管理サーバでアクセス権限エラーが発生した場合
(DbjAccessControlException)

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiFileNotFoundExpection

文書のコンテンツが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5

フォルダ操作パッケージ提供クラスの詳細

この章では、フォルダ操作パッケージ (jp.co.hitachi_system.sContents.API.Folder) 提供のクラス、およびメソッドについて説明します。

5.1 DecmFolderTree クラス

5.2 DecmFolder クラス

5.3 DecmFolderProp クラス

5.4 DecmFolderPerm クラス

5.1 DecmFolderTree クラス

DecmFolderTree クラスは、フォルダツリー階層情報の取得を扱うクラスです。

上位フォルダの OIID を指定することで、下位フォルダの一覧を取得します。上位フォルダを指定しない場合は、最上位フォルダの一覧を取得します。

このクラスの提供メソッドをコールすると、フォルダのツリー階層を取得できます。

プロパティ一覧

DecmFolderTree クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 5-1 DecmFolderTree クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド
		getFolderList()
dmaProp_OIID	String	
esysProp_Name	String	
dbrProp_OwnerId	String	
dbuUserProp_OwnerName	String	
esysProp_CreateTime	Integer	
esysProp_Creator	String	
esysProp_ModifyTime	Integer	
esysProp_Modifier	String	
dbuUserProp_ModifierName	String	
esysProp_Comment	String	
ecmProp_FolderType	Integer	
ecmProp_FolderLevel	Integer	
ecmProp_DocCreateFlg	Integer	
ecmProp_AclType	Integer	
ecmProp_OpenPubACL	String	
ecmProp_OpenPubACL_FULL	String	
ecmProp_AclSucceed	Integer	
ecmProp_DocTypeOIID	String	
ecmProp_DocTypeOIID_FULL	String	
ecmProp_ExtInt1	Integer	
under	Integer	

(凡例) : デフォルトで取得, および更新ができます。

次の項から, DecmFolderTree クラスのメソッドについて説明します。

5.1.1 setParentOiid

(1) 機能

上位フォルダの OIID を設定します。

このメソッドを発行しない場合のデフォルトは, 「」(空文字) です。

(2) 形式

```
void setParentOiid(
    String parentOiid
)
```

(3) 引数

String parentOiid

上位フォルダの OIID を指定します。

「」(空文字) を設定した場合, 最上位のフォルダを取得します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 parentOiid に null, または 「」(空文字) を指定した場合
- 引数 parentOiid が null, または 「」(空文字) でなく 「133 桁 OIID の設定チェック」に失敗した場合

5.1.2 getFolderList

(1) 機能

下位フォルダ一覧を取得します。

取得した下位フォルダ一覧は「フォルダ種別」, 「フォルダ名の昇順」の順でソートします。

フォルダ種別のソート順を次に示します。

- 公開フォルダ
- 文書フォルダ
- ノードフォルダ

5. フォルダ操作パッケージ提供クラスの詳細

(2) 形式

`Vector<DecmFolderProp> getFolderList()`

(3) 引数

なし

(4) 戻り値

下位フォルダのプロパティ一覧

(5) 例外

`DecmApiFolderNotFoundException`

検索するフォルダが存在しない場合

`DecmApiAccessException`

文書管理サーバでアクセス権限エラーが発生した場合

(`DbjAccessControlException`)

`DecmApiDBException`

文書管理サーバでエラーが発生した場合 (`DbjException` など)

5.2 DecmFolder クラス

DecmFolder クラスは、フォルダ操作を扱うクラスです。このクラスのインスタンスで、1 フォルダのオブジェクトを表現します。

このクラスのメソッドで実行できる操作を、次に示します。

フォルダの作成

フォルダの削除

フォルダのプロパティ取得

フォルダのプロパティ更新

フォルダのアクセス権限取得

フォルダのアクセス権限更新

プロパティ一覧

DecmFolder クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 5-2 DecmFolder クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド	更新可能メソッド
		fetchFolderProp()	createNodeFolder() createDocFolder() createPubFolder() updateNodeFolderProp()) updateDocFolderProp() updatePubFolderProp()
dmaProp_OIID	String		-
esysProp_Name	String		
dbrProp_OwnerId	String		-
dbuUserProp_OwnerName	String		-
esysProp_CreateTime	Integer		-
esysProp_Creator	String		-
esysProp_ModifyTime	Integer		-
esysProp_Modifier	String		-
dbuUserProp_ModifierName	String		-
esysProp_Comment	String		
ecmProp_FolderType	Integer		-
ecmProp_FolderLevel	Integer		-

5. フォルダ操作パッケージ提供クラスの詳細

プロパティ名	データ型	参照可能メソッド	更新可能メソッド
		fetchFolderProp()	createNodeFolder() createDocFolder() createPubFolder() updateNodeFolderProp()) updateDocFolderProp() updatePubFolderProp()
ecmProp_DocCreateFlg	Integer		-
ecmProp_AclType	Integer		-
ecmProp_OpenPubACL	String		-
ecmProp_OpenPubACL_FULL	String		-
ecmProp_AclSucceed	Integer		-
ecmProp_DocTypeOIID	String		-
ecmProp_DocTypeOIID_FULL	String		-
ecmProp_ExtInt1	Integer		-
ecmProp_MaxCapacity	Integer		-
ecmProp_CapacityUnit	Integer		-
ecmProp_UseCapacity	Long		-

(凡例)

: デフォルトで取得, または更新ができます。

- : 該当しません。

次の項から, DecmFolder クラスのメソッドについて説明します。

5.2.1 getOiid

(1) 機能

操作対象フォルダの OIID を取得します。

(2) 形式

```
String getOiid()
```

(3) 引数

なし

(4) 戻り値

操作対象フォルダの OIID

(5) 例外

なし

5.2.2 setOiid

(1) 機能

操作対象フォルダの OIID を設定します。

このメソッドは既存のフォルダにアクセスする場合に、最初にコールしてください。なお、OIID に対するフォルダの存在チェックは行いません。

(2) 形式

```
void setOiid(  
    String oiid  
)
```

(3) 引数

String oiid

操作対象フォルダの OIID を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 oiid に対する「133 桁 OIID の設定チェック」に失敗した場合

5.2.3 setParentOiid

(1) 機能

フォルダ作成時の上位フォルダの OIID を設定します。

下位のフォルダを作成する場合、このメソッドを設定してください。

(2) 形式

```
void setParentOiid(  
    String parentOiid  
)
```

(3) 引数

String parentOiid

フォルダ作成時の上位フォルダの OIID を指定します。

5. フォルダ操作パッケージ提供クラスの詳細

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 parentOiid に対する「133 桁 OIID の設定チェック」に失敗した場合

5.2.4 setFolderProp

(1) 機能

フォルダのプロパティを設定します。

(2) 形式

```
void setFolderProp(  
    DecmFolderProp folderProp  
)
```

(3) 引数

DecmFolderProp folderProp

フォルダのプロパティを指定します

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 folderProp に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 folderProp に対する「プロパティに設定規則チェック」に失敗した場合

5.2.5 setFolderPerm

(1) 機能

フォルダのアクセス権限を設定します。

このメソッドを使って、フォルダの所有者にメソッド実行ユーザと異なるユーザを設定する場合、セキュリティ管理者で実行してください。一般ユーザで実行した場合、DecmApiAccessException 例外が発生する場合があります。

操作対象フォルダが公開範囲を限定した公開フォルダの場合、引数 folderPerm で DecmACLPerm クラスのリストに設定しているユーザのアクセス権限の指定は無視し、無条件で「更新 (DecmACLPerm . PERM_UPDATE)」権限を与えます。

(2) 形式

```
void setFolderPerm(
    DecmFolderPerm folderPerm
)
```

(3) 引数

DecmFolderPerm folderPerm
フォルダのアクセス権限を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException
引数 folderPerm に対する「通常オブジェクトの設定チェック」に失敗した場合

5.2.6 setCapacityMax

(1) 機能

フォルダの最大容量を設定します。

このメソッドで設定した値は、resetCapacityMax メソッドをコールするまで、または、このクラスのインスタンスが破棄されるまで保持します。

- フォルダ容量制限を設定する場合
createNodeFolder メソッド、または createDocFolder メソッドをコールする前に、このメソッドをコールしてください。
なお、下位のフォルダを作成する場合、このメソッドで設定した値は無視されます。
- フォルダ容量制限を変更する場合
updateNodeFolder メソッド、または updateDocFolder メソッドをコールする前に、このメソッドをコールしてください。
なお、下位のフォルダを更新する場合、および更新対象フォルダが容量制限をしていない場合、このメソッドで設定した値は無視されます。

(2) 形式

```
void setCapacityMax(
    int cap,
    int unit
)
```

(3) 引数

int cap
フォルダ最大容量を指定します。指定できる値の範囲は、1 ~ 9,999 です。

5. フォルダ操作パッケージ提供クラスの詳細

int unit

フォルダ容量の単位を指定します。
指定できる値を次に示します。

- DecmFolder.UNIT_MB
- DecmFolder.UNIT_GB
- DecmFolder.UNIT_TB

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 cap が 1 より小さい場合，または 9,999 より大きい場合
- 引数 unit に対して，指定できる値以外を指定した場合

5.2.7 resetCapacityMax

(1) 機能

フォルダの最大容量をリセットします。

(2) 形式

```
void resetCapacityMax()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

なし

5.2.8 createNodeFolder

(1) 機能

ノードフォルダを作成します。

(2) 形式

```
String createNodeFolder(  
    boolean outputLog  
)
```


(3) 引数

boolean outputLog:

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

作成したノードフォルダの OIID

(5) 例外

DecmApiInvalidInitialException

- setFolderProp メソッドを発行していない場合
- setFolderProp メソッドを発行しているが、esysProp_Name を設定していない場合

DecmApiFolderNotFoundException

上位フォルダが存在しない場合

DecmApiFolderNotCorrectException

上位フォルダにノードフォルダ以外のフォルダを指定した場合

DecmApiNotSupport

- 作成できるフォルダ階層の数を超えている場合
- アクセス権限情報を指定している場合
- 指定した操作をサポートしていない場合

DecmApiAccessException

指定したノードフォルダに対してアクセス権限がない場合

(DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.9 createDocFolder

(1) 機能

文書フォルダを作成します。

アクセス権限情報を指定した場合は、アクセス権限が設定されます。指定しない場合は、上位フォルダの権限が継承されます。

(2) 形式

String createDocFolder(

5. フォルダ操作パッケージ提供クラスの詳細

```
int aclType,  
int docCreateType,  
String docMasterOiid,  
boolean outputLog  
)
```

(3) 引数

int aclType

フォルダのアクセス権限タイプを指定します。
指定できる値を次に示します。

- DecmFolder.ACL_TYPE_PRIVATE
- DecmFolder.ACL_TYPE_PUBLIC

int docCreateType

フォルダ下の文書作成の可否を指定します。
指定できる値を次に示します。

- DecmFolder.DOC_CREATE_NG
- DecmFolder.DOC_CREATE_OK

String docMasterOiid

使用する文書種別マスタの OIID を指定します。文書種別マスタを使用しない場合は「」(空文字)を指定してください。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

作成した文書フォルダの OIID

(5) 例外

DecmApiInvalidParamException

- 引数 aclType に対して、指定できる値以外を指定した場合
- 引数 docCreateType に対して、指定できる値以外を指定した場合
- 引数 docMasterOiid に null を指定した場合

DecmApiInvalidInitialException

- setFolderProp メソッドを発行していない場合
- setFolderProp メソッドを発行しているが、esysProp_Name を設定していない場合

DecmApiFolderNotFoundException

上位フォルダが存在しない場合

DecmApiFolderNotCorrectException

- 上位フォルダにノードフォルダ、文書フォルダ以外のフォルダを指定した場合
- 上位フォルダのアクセス権限タイプと、aclType に指定した値が矛盾している場合

DecmApiACLOutOfRangeException

- パブリック ACL の個数が制限値を超えた場合
(DbjPublicACLOutOfRangeException 例外が発生した場合)

DecmApiMasterNotFoundExcepion

- 指定した文書種別マスタが存在しない場合

DecmApiNotSupport

- 作成できるフォルダ階層の数を超過している場合
- アクセス権限情報の指定が不正な場合
- 指定した操作をサポートしていない場合

DecmApiACLOutOfRangeException

- 作成できる個数以上の ACL を指定している場合

DecmApiAccessException

- 指定したフォルダに対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException

- 文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.10 createPubFolder

(1) 機能

公開フォルダを作成します。

アクセス権限情報を指定した場合、アクセス権限が設定されます。指定しない場合は、上位フォルダの権限が継承されます。

フォルダ最大容量を指定した場合は、無視されます。

(2) 形式

```
String createPubFolder(
    int aclType,
    boolean outputLog
)
```

(3) 引数

int aclType

公開フォルダのアクセス権限タイプを指定します。

指定できる値を次に示します。

- DecmFolder.PUBFOL_ACL_TYPE_PUBLIC

5. フォルダ操作パッケージ提供クラスの詳細

- DecmFolder.PUBFOL_ACL_TYPE_PRIVATE

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

作成した公開フォルダの OIID

(5) 例外

DecmApiInvalidParamException

引数 aclType に対して、指定できる値以外を指定した場合

DecmApiInvalidInitialException

- setFolderProp メソッドを発行していない場合
- setFolderProp メソッドを発行しているが、esysProp_Name を設定していない場合

DecmApiFolderNotFoundException

上位フォルダが存在しない場合

DecmApiFolderNotCorrectException

上位フォルダに公開フォルダ、ノードフォルダ以外のフォルダを指定した場合

DecmApiACLOutOfRangeException

パブリック ACL の個数が制限値を超えた場合

(DbjPublicACLOutOfRangeException 例外を発生した場合)

DecmApiACLOutOfRangeException

作成できる個数以上の ACL を指定している場合

DecmApiNotSupport

- 作成できるフォルダ階層の数を超えている場合
- アクセス権限情報の指定が不正な場合
- 指定した操作をサポートしていない場合

DecmApiAccessException

指定した公開フォルダに対してアクセス権限がない場合

(DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.11 deleteFolder

(1) 機能

フォルダを削除します。

(2) 形式

```
void deleteFolder(
    boolean outputLog
)
```

(3) 引数

boolean outputLog:

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiFolderNotFoundException

削除を指定したフォルダが存在しない場合

DecmApiAccessException

削除を指定したフォルダに対してアクセス権限がない場合

(DbjAccessControlException)

DecmApiNotSupport

- 削除を指定したフォルダ配下に文書, またはフォルダが存在する場合
- 削除を指定したフォルダに対する審査・承認案件が存在する場合
- 削除を指定したフォルダに対する承認ルートが存在する場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.12 fetchFolderProp

(1) 機能

フォルダの最新バージョンのプロパティを取得します。

5. フォルダ操作パッケージ提供クラスの詳細

(2) 形式

```
DecmFolderProp fetchFolderProp(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログを出力フラグに指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

フォルダのプロパティ

(5) 例外

DecmApiInvalidInitialException
setOiid メソッドを発行していない場合

DecmApiFolderNotFoundException
指定したフォルダが存在しない場合

DecmApiAccessException
指定したフォルダに対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException
文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.13 updateNodeFolderProp

(1) 機能

ノードフォルダのプロパティを更新します。

(2) 形式

```
void updateNodeFolderProp(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setFolderProp メソッドと、setCapacityMax メソッドのどちらも発行していない場合

DecmApiFolderNotFoundException

指定したノードフォルダが存在しない場合

DecmApiCapacityException

フォルダ最大容量を、すでに登録済みのサイズ以下に変更しようとした場合

DecmApiAccessException

指定したノードフォルダに対してアクセス権限がない場合
(DbjAccessControlException)

DecmApiInvalidFolder

フォルダ種別が不正な場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.14 updateDocFolderProp

(1) 機能

文書フォルダのプロパティを更新します。

(2) 形式

```
void updateDocFolderProp(
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setFolderProp メソッドと、setCapacityMax メソッドのどちらも発行していない場合

DecmApiFolderNotFoundException

指定した文書フォルダが存在しない場合

DecmApiCapacityException

フォルダ最大容量を、すでに登録済みのサイズ以下に変更しようとした場合

DecmApiAccessException

指定した文書フォルダに対してアクセス権限がない場合
(DbjAccessControlException)

DecmApiInvalidFolder

フォルダ種別が不正な場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.15 updatePubFolderProp

(1) 機能

公開フォルダのプロパティを更新します。

フォルダ最大容量を指定した場合は、無視されます。

(2) 形式

```
void updatePubFolderProp(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setFolderProp メソッドを発行していない場合

DecmApiFolderNotFoundException

指定した公開フォルダが存在しない場合

DecmApiAccessException

文書管理サーバでアクセス権限エラーが発生した場合
(DbjAccessControlException)

DecmApiInvalidFolder

フォルダ種別が不正な場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.16 fetchFolderPerm

(1) 機能

フォルダのアクセス権限を取得します。

(2) 形式

```
DecmFolderPerm fetchFolderPerm(
    boolean outputLog
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

フォルダに対するアクセス権限

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiFolderNotFoundException

指定したフォルダが存在しない場合

5. フォルダ操作パッケージ提供クラスの詳細

DecmApiAccessException

指定したフォルダに対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.17 updateFolderPerm

(1) 機能

フォルダのアクセス権限を更新します。

(2) 形式

```
void updateFolderPerm(  
    boolean outputLog  
)
```

(3) 引数

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

- setOiid メソッドを発行していない場合
- setFolderPerm メソッドを発行していない場合

DecmApiFolderNotFoundException

指定したフォルダが存在しない場合

DecmApiAccessException

指定したフォルダに対してアクセス権限がない場合 (DbjAccessControlException)

DecmApiInvalidFolder

フォルダ種別が不正な場合

DecmApiACLOutOfRangeException

作成できる個数以上の ACL を指定している場合

DecmApiNotSupport

指定した操作をサポートしていない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.18 changeCapacityMode

(1) 機能

フォルダ容量制限機能を変更します。

なお、操作対象フォルダが下位フォルダの場合や、引数にメソッド発行前と同じ状態を指定したときは、このメソッドをコールしても何もしません。

(2) 形式

```
void changeCapacityMode(
    int capMode,
    boolean outputLog
)
```

(3) 引数

int capMode

フォルダ容量制限機能の使用有無を指定します。

指定できる値を次に示します。

- DecmFolder.CAP_NOUSE : フォルダ容量制限機能を使用しません。
- DecmFolder.CAP_USE : フォルダ容量制限機能を使用します。

boolean outputLog

監査証跡ログの出力フラグを指定します。

- true : 監査証跡ログを出力します。
- false : 監査証跡ログを出力しません。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 capMode に対して、指定できる値以外を指定した場合

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiFolderNotFoundException

指定したフォルダが存在しない場合

DecmApiAccessException

指定したフォルダに対してアクセス権限がない場合 (DbjAccessControlException)

5. フォルダ操作パッケージ提供クラスの詳細

DecmApiInvalidFolder

フォルダ種別が不正な場合

DecmApiCapacityException

フォルダ容量制限機能を「使用しない」から「使用する」に変更しようとした場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

5.2.19 lockRead

(1) 機能

フォルダに対する参照ロックの予約を行います。

このメソッドのコール後にフォルダデータに対するアクセスを行うと、参照ロックが取得されます。

なお、1 トランザクション中にこのメソッドと lockWrite メソッドを複数回コールした場合、最後にコールしたメソッドが有効となります。

(2) 形式

```
void lockRead()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

5.2.20 lockWrite

(1) 機能

フォルダに対する更新ロックの予約を行います。

このメソッドのコール後にフォルダに対するアクセスを行うと、更新ロックが取得されます。

なお、1 トランザクション中にこのメソッドと lockRead メソッドを複数回コールした場合は、最後にコールしたメソッドが有効となります。

(2) 形式

```
void lockWrite()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

5.3 DecmFolderProp クラス

DecmFolderProp クラスは、フォルダのプロパティを扱うクラスです。このクラスのインスタンスで、フォルダに設定されているプロパティを表現します。

次の操作を行う場合に、このクラスのインスタンスを使用します。

- フォルダの作成
- フォルダのプロパティ取得
- フォルダのプロパティ更新

なお、DecmFolderProp クラスにはメソッドはありません。

5.4 DecmFolderPerm クラス

DecmFolderPerm クラスは、フォルダのアクセス権限を扱うクラスです。このクラスのインスタンスで、フォルダに設定されているアクセス権限を表現します。

次の操作を行う場合に、このクラスのインスタンスを使用します。

- フォルダのアクセス権限取得
- フォルダのアクセス権限更新

なお、DecmFolderPerm クラスにはメソッドはありません。

6

検索操作パッケージ提供クラスの詳細

この章では、検索操作パッケージ (jp.co.hitachi_system.sContents.API.Select) 提供のクラス、およびメソッドについて説明します。

6.1 DecmSelect クラス

6.2 DecmCondition クラス

6.3 DecmCondGroup クラス

6.1 DecmSelect クラス

DecmSelect クラスは、文書検索を扱うクラスです。

このクラスで、次の機能が実行できます。

フォルダ下の文書一覧の取得

全文検索結果一覧の取得

プロパティ検索結果一覧の取得

ごみ箱の文書一覧の取得

プロパティ一覧

DecmSelect クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 6-1 DecmSelect クラスで扱うプロパティ

プロパティ名	データ型	検索条件 (WHERE) への指定可否	参照可能メソッド
			searchDoc()
dmaProp_OIID	String	可	
esysProp_Name	String	可	
edmProp_OwnerId	String	可	
dbuUserProp_OwnerName	String	可	
esysProp_CreateTime	Integer	可	
esysProp_Creator	String	可	
esysProp_ModifyTime	Integer	可	
esysProp_Modifier	String	可	
dbuUserProp_ModifierName	String	可	
dbuUserProp_ReservedUserId	String	可	
dbuUserProp_ReservedUserName	String	可	
esysProp_Comment	String	可	
ecmProp_DocTypeOIID	String	可	
ecmProp_DocTypeOIID_FULLL	String	不可	
ecmProp_CHOiid	String	可	
ecmProp_CHOiid_FULLL	String	不可	
ecmProp_DocType	Integer	可	
ecmProp_FileName	String	可	

プロパティ名	データ型	検索条件 (WHERE) への指定可否	参照可能メソッド
			searchDoc()
ecmProp_FileSize	Integer	可	
ecmProp_ParentContainer	String	不可	
ecmProp_ParentContainer_FULL	String	不可	
ecmProp_DeleteFlag	Integer	不可	
ecmProp_DeleteUserId	String	可	
ecmProp_DeleteTime	Integer	可	
ecmProp_DocNumber	String	可	
ecmProp_UserInt1 ~ ecmProp_UserInt5	Integer	可	
ecmProp_UserStr1 ~ ecmProp_UserStr5	String	可	
ecmProp_DocTypePropInt1 ~ ecmProp_DocTypePropInt10	Integer	可	
ecmProp_DocTypePropStr1 ~ ecmProp_DocTypePropStr10	String	可	
ecmProp_DocTypePropLongStr1 ~ ecmProp_DocTypePropLongStr5	String	可	
esysProp_FileType	Integer	不可	
esysProp_TopDVOIID_FULL	String	不可	

(凡例)

可：指定できます。

不可：指定できません。

：デフォルトで取得、および更新ができます。

：DecmSelect#setSelectList メソッドをコールすると取得できます。

注

DecmFactory#initUserProp メソッド、DecmFactory#initUserProp メソッドを使用してプロパティ名を登録しておく必要があります。

次の項から、DecmSelect クラスのメソッドについて説明します。

6.1.1 setFolderOiid

(1) 機能

検索対象となるフォルダの OIID を設定します。

このメソッドを発行しない場合のデフォルトは、「」(空文字)です。

6. 検索操作パッケージ提供クラスの詳細

(2) 形式

```
void setFolderOiid(  
    String folderOiid  
)
```

(3) 引数

String folderOiid

検索対象となるフォルダの OIID を指定します。

null, または「」(空文字) を指定した場合, 登録されている文書すべてが検索対象となります。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 folderOiid に null, または「」(空文字) を指定した場合
- 引数 folderOiid が null, または「」(空文字) でなく「133 桁 OIID の設定チェック」に失敗した場合

6.1.2 setFetchCount

(1) 機能

検索実行時に取得するレコードの件数を設定します。

このメソッドを発行しない場合のデフォルトの取得件数は, 200 件です。

(2) 形式

```
void setFetchCount(  
    int fetchCount  
)
```

(3) 引数

int fetchCount

検索実行時に取得するレコードの件数を指定します。

指定できる値は, 1 ~ 2,147,483,647 の範囲です。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 `fetchCount` が正の整数以外の場合

6.1.3 setSortKey

(1) 機能

検索実行時のソート用カラム名を設定します。

`setSelectList` メソッドに指定可能なプロパティ名を指定できます。

このメソッドで指定したソートキーとなるプロパティ値は、検索結果のプロパティ一覧に取得されます。

なお、このメソッドを発行しない場合のデフォルトは `esysProp_ModifyTime` (更新日時) です。

(2) 形式

```
void setSortKey(
    String sortKey
)
```

(3) 引数

String sortKey

検索実行時のソート用カラム名を指定します。

`null`, または「`""`」(空文字)を指定した場合は、デフォルトの `esysProp_ModifyTime` (更新日時) が設定されます。

(4) 戻り値

なし

(5) 例外

なし

6.1.4 setSortMode

(1) 機能

検索実行時のソート方式 (昇順, または降順) を設定します。

このメソッドを発行しない場合のデフォルトは降順です。

(2) 形式

```
void setSortMode(
    int sortMode
)
```

(3) 引数

int sortMode

検索実行時にソート方式（昇順，または降順）を指定します。
指定できる値を次に示します。

- DecmSelect.SORT_MODE_UP
- DecmSelect.SORT_MODE_DOWN

指定できる値以外を指定した場合は，デフォルトの降順が設定されます。

(4) 戻り値

なし

(5) 例外

なし

6.1.5 setTrashMode

(1) 機能

検索対象とする文書を「通常文書」，「ごみ箱の文書」，「全ての文書」から設定します。

このメソッドを発行しない場合のデフォルトは，
DecmSelect.TRASH_MODE_NORMAL（通常文書）です。

なお，参照以上の権限があるごみ箱の文書が検索対照となります。

(2) 形式

```
void setTrashMode(  
    int trashMode  
)
```

(3) 引数

int trashMode

検索対象とする文書を指定します。

指定できる値を次に示します。

- DecmSelect.TRASH_MODE_NORMAL（通常文書）
- DecmSelect.TRASH_MODE_DELETED（ごみ箱の文書）
- DecmSelect.TRASH_MODE_ALL（全ての文書）

指定できる値以外を指定した場合，デフォルトの
DecmSelect.TRASH_MODE_NORMAL（通常文書）が設定されます。

(4) 戻り値

なし

(5) 例外

なし

6.1.6 setFolderMode

(1) 機能

フォルダ検索を行う場合に下位フォルダを検索対象とするかどうかを設定します。

このメソッドを設定しない場合のデフォルトは、true (下位フォルダを検索対象とする) です。

(2) 形式

```
void setFolderMode(
    boolean folderMode
)
```

(3) 引数

boolean folderMode

下位フォルダを検索対象とするかどうかを指定します。

- true : 指定したフォルダの下位フォルダも検索対象とします。
- false : 指定したフォルダの下位フォルダは検索対象としません。

(4) 戻り値

なし

(5) 例外

なし

6.1.7 isSubFileSearchMode

(1) 機能

関連ファイルの検索対象の指定有無を取得します。

(2) 形式

```
boolean isSubFileSearchMode()
```

(3) 引数

なし

(4) 戻り値

関連ファイルの検索対象の指定有無

(5) 例外

なし

6.1.8 setSubFileSearchMode

(1) 機能

関連ファイルを検索対象とするかどうかを設定します。

このメソッドを設定しない場合のデフォルトは、false (関連ファイルを検索の対象としない) です。

(2) 形式

```
void setSubFileSearchMode(  
    boolean subFileSearchMode  
)
```

(3) 引数

boolean subFileSearchMode:

関連ファイルを検索対象とするかどうかを指定します。

- true : 関連ファイルを検索の対象とします。
- false : 関連ファイルを検索の対象としません。

(4) 戻り値

なし

(5) 例外

なし

6.1.9 addCondition

(1) 機能

検索条件を追加します。

このメソッドを複数回コールして、指定した条件を AND で追加します。

検索条件を追加するごとに、DecmCondition クラスのインスタンスを DecmFactory#createDecmCondition メソッドで生成してください。

(2) 形式

```
void addCondition(  
    DecmCondition condition  
)
```


(3) 引数

DecmCondition condition

通常の検索条件を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 condition に対する「通常オブジェクトの設定チェック」に失敗した場合

6.1.10 setSelectList

(1) 機能

検索取得列を設定します。

検索結果として取得する列（プロパティ名）を指定します。

(2) 形式

```
void setSelectList(  
    Vector<String> selectList  
)
```

(3) 引数

Vector<String> selectList

検索カラム名を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 selectList に対する「通常配列の設定チェック」に失敗した場合

6.1.11 setCondGroup

(1) 機能

組み合わせ検索条件を設定します。

このメソッドを複数回コールした場合、最後に指定した条件だけが有効となります。

組み合わせ検索条件の複数指定を設定する場合は、DecmCondGroup クラスのメソッド

6. 検索操作パッケージ提供クラスの詳細

で実行してください。

また、このメソッドで設定した検索条件と、その他の検索条件は AND 検索されます。

なお、このメソッドに指定する DecmCondGroup クラスのインスタンスは、DecmFactory#createDecmCondGroup メソッドで生成してください。

(2) 形式

```
void setCondGroup(  
    DecmCondGroup condGroup  
)
```

(3) 引数

DecmCondGroup condGroup
組み合わせ検索条件を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException
引数 condGroup に対する「通常オブジェクトの設定チェック」に失敗した場合

6.1.12 setKeyword

(1) 機能

全文検索のキーワードを設定します。

キーワードとキーワードは AND 検索されます。

また、このメソッドで設定した検索条件と、その他の検索条件も AND 検索されます。

このメソッドを発行しない場合、全文検索機能は使用しない設定となります。

(2) 形式

```
void setKeyword(  
    Vector<String> keyword  
)
```

(3) 引数

Vector<String> keyword
全文検索のキーワードを設定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 keyword に対する「通常配列の設定チェック」に失敗した場合
- 引数 keyword の項目数が 11 個以上の場合

6.1.13 setSearchExpanse

(1) 機能

異表記展開指定の組み合わせを設定します。

(2) 形式

```
void setSearchExpanse(
    String searchExpanse
)
```

(3) 引数

String searchExpanse

異表記展開指定の組み合わせを指定します。

null, または「」(空文字)を指定した場合, この機能を使用しない設定となります。

なお, 指定範囲外の値を設定した場合, 不正な動作をするおそれがあります。

指定範囲

- カタカナ異表記: K
- アルファベット異表記: A
- 全角半角異表記: E

使用する異表記展開により, 上記の値を組み合わせで指定できます。

例えば, カタカナ異表記とアルファベット異表記を指定する場合は, 「KA」と指定してください。

(4) 戻り値

なし

(5) 例外

なし

6.1.14 setSearchDictionary

(1) 機能

全文検索実行時の同義語辞書名称を設定します。

6. 検索操作パッケージ提供クラスの詳細

(2) 形式

```
void setSearchDictionary(  
    String searchDictionary  
)
```

(3) 引数

String searchDictionary

全文検索実行時の同義語辞書名称を指定します。

null, または「」(空文字)を指定した場合, この機能を使用しない設定となります。

(4) 戻り値

なし

(5) 例外

なし

6.1.15 searchDoc

(1) 機能

文書検索を実行します。

addCondition メソッド, setCondGroup メソッド, setKeyword メソッドに指定した条件で AND 検索を実行します。

どのメソッドもコールしない場合, 無条件で検索します。

フォルダ指定検索, プロパティ検索, 全文検索, およびごみ箱検索を行います。

なお, アクセス権限のないフォルダに対して検索を実行した場合も, エラーとはなりません。

(2) 形式

```
Vector<DecmDocProp> searchDoc()
```

(3) 引数

なし

(4) 戻り値

文書プロパティの一覧

なお, 関連ファイルを検索対象とした場合 (setSubFileSearchMode で true を指定した場合), 複数ファイル文書単位で検索結果をソートします。

その際, 主ファイルは指定のソートキーでソートを実行しますが, 関連ファイルはファ

イル名で昇順にソートされます。

(5) 例外

DecmApiDBException

文書管理サーバでエラーが発生した場合

6.1.16 resetCondition

(1) 機能

指定した検索条件をリセットします。

このメソッドをコールした場合、DecmSelect メソッドのコンストラクタを発行したあとの状態になります。

このため、次に示すメソッドによって設定した値は、すべてクリアされます。

- setFolderOiid
- setFetchCount
- setSortKey
- setSortMode
- setTrashMode
- setFolderMode
- setSubFileSearchMode
- addCondition
- setSelectList
- setCondGroup
- setKeyword
- setSearchExpanse
- setSearchDictionary

(2) 形式

```
void resetCondition()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

なし

6.2 DecmCondition クラス

DecmCondition クラスは、通常の検索条件の表現用クラスです。通常の検索条件の生成機能を提供します。

次の項から、DecmCondition クラスのメソッドについて説明します。

6.2.1 createStrCond

(1) 機能

文字列型の検索条件を生成します。

このメソッドをコールすると createIntCond メソッドは無効となり、createIntCond メソッドでセットした Condition はクリアされます。

(2) 形式

```
void createStrCond(  
    String field,  
    int operator,  
    String value  
)
```

(3) 引数

String field

プロパティ名を指定します。

int operator

検索のオペレータを指定します。
指定できる値を次に示します。

- DecmCondition.OP_EQ
- DecmCondition.OP_NEQ
- DecmCondition.OP_PART
- DecmCondition.OP_STWITH
- DecmCondition.OP_ENDWITH

String value

文字列型の検索条件の値を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 field に対する「通常文字列の設定チェック」に失敗した場合
- 引数 operator に対して、指定できる値以外を指定した場合
- 引数 value に null を指定した場合

6.2.2 createIntCond

(1) 機能

数値型の検索条件を生成します。

検索条件を開始する値 (beginValue) と検索条件を終了する値 (endValue) に同じ値を設定した場合は、指定した値と一致する field を検索条件とします。

このメソッドをコールすると createStrCond メソッドは無効となり、createStrCond メソッドでセットした Condition はクリアされます。

(2) 形式

```
void createIntCond(
    String field,
    int beginValue,
    int endValue
)
```

(3) 引数

String field

プロパティ名を指定します。

int beginValue

数値型の検索条件を開始する値を指定します。

int endValue

数値型の検索条件を終了する値を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 field に対する「通常文字列の設定チェック」に失敗した場合

6.3 DecmCondGroup クラス

DecmCondGroup クラスは、組み合わせ検索条件の表現用クラスです。

次の項から、DecmCondGroup クラスのメソッドについて説明します。

6.3.1 addCond

(1) 機能

検索条件を追加します。

なお、検索条件を追加するごとに、DecmCondition クラスのインスタンスを DecmFactory#createDecmCondition メソッドで生成してください。

(2) 形式

```
void addCond(  
    DecmCondition cond,  
    int operator  
)
```

(3) 引数

DecmCondition cond

検索条件を指定します。

int operator

検索条件追加のオペレータを指定します。

初めに指定したオペレータは無視されます。

なお、指定範囲外の値を設定した場合は、AND 検索と判断されます。

指定できる値を次に示します。

- DecmCondGroup.OP_AND
- DecmCondGroup.OP_OR

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 cond に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 operator に対して、指定できる値以外を指定した場合

6.3.2 addCondGroup

(1) 機能

組み合わせ検索条件を追加します。

(2) 形式

```
void addCondGroup(  
    DecmCondGroup condGroup,  
    int operator  
)
```

(3) 引数

DecmCondGroup condGroup

組み合わせ検索条件を指定します。

int operator

検索条件追加のオペレータを指定します。

初めに指定したオペレータは無視されます。

なお、指定範囲外の値を設定した場合は、AND 検索と判断されます。

指定できる値を次に示します。

- DecmCondGroup.OP_AND
- DecmCondGroup.OP_OR

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

- 引数 cond に対する「通常オブジェクトの設定チェック」に失敗した場合
- 引数 operator に対して、指定できる値以外を指定した場合

7

文書種別マスタ操作パッケージ提供クラスの詳細

この章では、文書種別マスタ操作パッケージ (jp.co.hitachi_system.sContents.API.DocMaster) パッケージ提供のクラス、およびメソッドについて説明します。

-
- 7.1 DecmDocMasterList クラス

 - 7.2 DecmDocMaster クラス

 - 7.3 DecmDocMasterProp クラス

 - 7.4 DecmDocPropMaster クラス

 - 7.5 DecmDocPropMasterProp クラス
-

7.1 DecmDocMasterList クラス

DecmDocMasterList クラスは、文書種別マスター一覧を扱うクラスです。

このクラスのインスタンスで、文書種別マスタの一覧情報を取得します。

プロパティ一覧

DecmDocMasterList クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 7-1 DecmDocMasterList クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド
		getDocMasterList()
dmaProp_OIID	String	
esysProp_Name	String	
ecmProp_DocNumProp	String	
ecmProp_DocNumDelimiter	String	
ecmProp_DocNumSuffix	String	

(凡例) : デフォルトで取得、または更新ができます。

次の項から、DecmDocMasterList クラスのメソッドについて説明します。

7.1.1 getDocMasterList

(1) 機能

すべての文書種別マスター一覧を取得します。

取得した文書種別マスター一覧は、更新日時の降順でソートします。

(2) 形式

```
Vector<DecmDocMasterProp> getDocMasterList()
```

(3) 引数

なし

(4) 戻り値

すべての文書種別マスター一覧のプロパティ一覧

(5) 例外

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.2 DecmDocMaster クラス

DecmDocMaster クラスは、文書種別マスタ操作を扱うクラスです。このクラスのインスタンスで、文書種別マスタに対する操作を行います。

このクラスのメソッドで実行できる操作を、次に示します。

基本情報、および採番情報の取得

文書種別マスタに定義されている文書新規作成時のプロパティ項目の取得

文書種別マスタに定義されている検索に使用するプロパティ項目の取得

文書種別マスタに定義されている一覧表示に使用するプロパティ項目の取得

プロパティ一覧

DecmDocMaster クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 7-2 DecmDocMaster クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド
		fetchDocMasterProp()
dmaProp_OIID	String	
esysProp_Name	String	
ecmProp_DocNumProp	String	
ecmProp_DocNumDelimiter	String	
ecmProp_DocNumSuffix	String	

(凡例) : デフォルトで取得、または更新ができます。

次の項から、DecmDocMaster クラスのメソッドについて説明します。

7.2.1 getOiid

(1) 機能

操作対象の文書種別マスタの OIID を取得します。

(2) 形式

```
String getOiid()
```

(3) 引数

なし

(4) 戻り値

操作対象の文書種別マスタの OIID

(5) 例外

なし

7.2.2 setOiid

(1) 機能

操作対象の文書種別マスタの OIID を設定します。

(2) 形式

```
void setOiid(  
    String oidd  
)
```

(3) 引数

String oidd

操作対象の文書種別マスタの OIID を指定します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 oidd に対する「133 桁 OIID の設定チェック」に失敗した場合

7.2.3 fetchDocMasterProp

(1) 機能

文書種別マスタのプロパティを取得します。

(2) 形式

```
DecmDocMasterProp fetchDocMasterProp()
```

(3) 引数

なし

(4) 戻り値

文書種別マスタのプロパティ

7. 文書種別マスタ操作パッケージ提供クラスの詳細

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.2.4 getNewDocMaster

(1) 機能

文書作成時に指定するプロパティ項目一覧を取得します。

(2) 形式

```
Vector<String> getNewDocMaster()
```

(3) 引数

なし

(4) 戻り値

文書作成時に指定するプロパティ項目一覧

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.2.5 getSearchDocMaster

(1) 機能

検索に使用するプロパティ項目一覧を取得します。

(2) 形式

```
Vector<String> getSearchDocMaster()
```


(3) 引数

なし

(4) 戻り値

検索に使用するプロパティ項目一覧

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.2.6 getListDocMaster

(1) 機能

一覧表示に使用するプロパティ項目一覧を取得します。

(2) 形式

```
Vector<String> getListDocMaster()
```

(3) 引数

なし

(4) 戻り値

一覧表示に使用するプロパティ項目一覧

(5) 例外

DecmApiInvalidInitialException

setOiid メソッドを発行していない場合

DecmApiMasterNotFoundException

指定した文書種別マスタが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.3 DecmDocMasterProp クラス

DecmDocMasterProp クラスは、文書種別マスタを扱うクラスです。このクラスのインスタンスで、文書種別マスタに定義されている情報を表現します。文書種別マスタの基本情報、および採番情報の参照ができます。

なお、DecmDocMasterProp クラスにメソッドはありません。

7.4 DecmDocPropMaster クラス

DecmDocPropMaster クラスは、属性値マスタの取得を扱うクラスです。

このクラスのインスタンスで、属性値マスタの階層構造を取得します。

プロパティ一覧

DecmDocPropMaster クラスで扱うプロパティの一覧を次の表に示します。各プロパティの詳細は、「付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ」を参照してください。

表 7-3 DecmDocPropMaster クラスで扱うプロパティ

プロパティ名	データ型	参照可能メソッド
		getDocPropMasterList()
dmaProp_OIID	String	
esysProp_Name	String	
ecmProp_Level	Integer	
ecmProp_DocTypePropName	String	
ecmProp_DocTypePropCode	String	
ecmProp_DocTypeParentCode	String	
ecmProp_SortNumber	Integer	

(凡例) : デフォルトで取得、または更新ができます。

次の項から、DecmDocPropMaster クラスのメソッドについて説明します。

7.4.1 setParentOiid

(1) 機能

上位属性値マスタの OIID を設定します。

このメソッドを発行しない場合のデフォルトは、「」(空文字)です。

(2) 形式

```
void setParentOiid(
    String parentOiid
)
```

(3) 引数

String parentOiid

上位属性値マスタの OIID を指定します。

7. 文書種別マスタ操作パッケージ提供クラスの詳細

「'''」(空文字)を指定した場合、最上位の属性値マスタを示します。

(4) 戻り値

なし

(5) 例外

DecmApiInvalidParamException

引数 parentOiid に null を指定した場合

7.4.2 getDocPropMasterList

(1) 機能

下位属性値マスター一覧を取得します。

取得した下位属性値マスター一覧は、ソート用番号の昇順でソートします。

(2) 形式

```
Vector<DecmDocPropMasterProp> getDocPropMasterList()
```

(3) 引数

なし

(4) 戻り値

下位属性値マスタのプロパティ一覧

(5) 例外

DecmApiPropMasterNotFoundException

上位属性値マスタが存在しない場合

DecmApiDBException

文書管理サーバでエラーが発生した場合 (DbjException など)

7.5 DecmDocPropMasterProp クラス

DecmDocPropMasterProp クラスは、属性値マスタのプロパティを扱うクラスです。

このクラスのインスタンスで、属性値マスタに定義されている情報を表現します。

なお、DecmDocPropMasterProp クラスにメソッドはありません。

8

DocumentBroker で提供されるクラスライブラリ

この章では、DocumentBroker Standard GUI Development Kit で使用する DocumentBroker のクラス、およびメソッドについて説明します。

8.1 DocumentBroker で提供されるクラスライブラリ

8.2 DbjFactory0200 クラス

8.3 DbjFactory インターフェース

8.4 DbjSession インターフェース

8.5 DbjDocSpace インターフェース

8.1 DocumentBroker で提供されるクラスライブラリ

DocumentBroker Standard GUI Development Kit は、DocumentBroker Development Kit、または DocumentBroker Runtime で提供される Java クラスライブラリの使用が前提となります。

DocumentBroker Development Kit、または DocumentBroker Runtime で提供している Java クラスライブラリでのクラス、インターフェースおよびメソッドのうち、DocumentBroker Standard GUI Development Kit で使用するクラス、インターフェースおよびメソッドを一覧形式で示します。

8.1.1 使用するクラス、およびインターフェースの一覧

DocumentBroker Development Kit、または DocumentBroker Runtime で提供している Java クラスライブラリでのクラス、およびインターフェースのうち、DocumentBroker Standard GUI Development Kit で使用するクラス、およびインターフェースを次の表に示します。

表 8-1 DocumentBroker Development Kit、Runtime Java クラスライブラリのクラス、およびインターフェース一覧

クラス / インターフェース	説明
DbjFactory0200 クラス	DocumentBroker にアクセスするためのファクトリインターフェースを取得します。
DbjFactory インターフェース	指定した文書空間に対する Java クラスライブラリのセッションオブジェクトを作成します。
DbjSession インターフェース	文書空間への一つの接続に対応し、セッション内でのトランザクション制御などを行うためのインターフェースです。
DbjDocSpace インターフェース	DbjSession インターフェースの login() メソッドをコールして、このインターフェースを取得します。

注 DocumentBroker Standard GUI Development Kit におけるファクトリクラス (DecmFactory) には、このインターフェースを渡す必要があります。

8.1.2 使用するメソッド一覧

DocumentBroker Development Kit、または DocumentBroker Runtime で提供している Java クラスライブラリでのクラス、インターフェースのメソッドのうち、DocumentBroker Standard GUI Development Kit で使用するものを示します。

(1) DbjFactory0200 クラス

DbjFactory0200 クラスのメソッド一覧を次の表に示します。

表 8-2 DbjFactory0200 クラスのメソッド一覧

メソッド	機能
getFactory	ファクトリインターフェースを取得します。

(2) DbjFactory インターフェース

DbjFactory インターフェースのメソッド一覧を次の表に示します。

表 8-3 DbjFactory インターフェースのメソッド一覧

メソッド	機能
createSession	指定した文書空間に対する Java クラスライブラリのセッションオブジェクトを作成し、DbjSession インターフェースを取得します。

(3) DbjSession インターフェース

DbjSession インターフェースのメソッド一覧を次の表に示します。

表 8-4 DbjSession インターフェースのメソッド一覧

メソッド	機能
login	文書空間へのログイン, および文書空間インターフェース (DbjDocSpace インターフェース) を取得します。
logout	文書空間からログアウトしてセッションを切断します。
begin	トランザクションを開始します。
rollback	トランザクションを取り消します。
commit	トランザクションを確定します。

8.2 DbjFactory0200 クラス

DbjFactory0200 クラスは、Java クラスライブラリのファクトリインターフェース、およびメタマネージャーインターフェースを取得するクラスです。なお、このクラスはインスタンスを生成できません。コンストラクタは `private` 属性です。

次の項から、DbjFactory0200 クラスのメソッドについて説明します。

8.2.1 getFactory

(1) 機能

ファクトリインターフェースを取得します。

(2) 形式

```
static DbjFactory getFactory()
```

(3) 引数

なし

(4) 戻り値

ファクトリインターフェース (DbjFactory インターフェース)

(5) 例外

なし

8.3 DbjFactory インターフェース

DbjFactory インターフェースは、Java クラスライブラリのオブジェクトを生成して、そのインターフェースを取得する機能を持つファクトリインターフェースです。

次の項から、DbjFactory インターフェースのメソッドについて説明します。

8.3.1 createSession

(1) 機能

指定した文書空間に対する Java クラスライブラリのセッションオブジェクトを作成し、そのインターフェース (DbjSession インターフェース) を取得します。

ただし、DbjSession インターフェースを取得した時点では文書空間にアクセスしません。

形式 1 では、セッションを開始する文書空間の文書空間識別子を指定します。

形式 2 では、動作環境定義ファイルに指定したデフォルト文書空間 (コンフィギュレーションキーは DefaultDocSpaceId) に接続するためのセッションオブジェクトを作成し、そのインターフェース (DbjSession インターフェース) を取得します。

動作環境定義ファイルの詳細についてはマニュアル「uCosminexus DocumentBroker Version 3 クラスライブラリ Java 解説」を参照してください。

(2) 形式

(a) 形式 1

```
DbjSession createSession(
    String      docspaceId
)
```

(b) 形式 2

```
DbjSession createSession()
```

(3) 引数

docspaceId

文書空間識別子を GUID 文字列で指定します。

GUID 文字列は、16 進数「X」によって、

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

「X」は、0 ~ 9, a ~ f (小文字), および A ~ F (大文字) のどれかです。null, または「」(空文字) を指定した場合は、例外がスローされます。

8. DocumentBroker で提供されるクラスライブラリ

(4) 戻り値

Java クラスライブラリのセッションインターフェース (DbjSession インターフェース)

(5) 例外

NullPointerException

引数 docspaceId に null , または 「 "" 」 (空文字) を指定した場合

8.4 DbjSession インターフェース

DbjSession インターフェースは、Java クラスライブラリのセッションを扱います。Java クラスライブラリの一つのセッションオブジェクトは文書空間への一つの接続に対応し、その接続している期間をセッションといいます。DbjSession インターフェースは、同時に一つのセッションのセッション管理をサポートします。

また、DocumentBroker Standard GUI Development Kit を使用して、DocumentBroker Standard GUI 環境へのオブジェクト操作に対するトランザクション制御を行う場合は、このインターフェースのメソッドをコールし、ユーザアプリケーション側で実装する必要があります。

次の項から、DbjSession インターフェースのメソッドについて説明します。

8.4.1 login

(1) 機能

文書空間にログインしてセッションを確立し、同時にユーザ識別子およびパスワードの認証を実行して、文書空間インターフェース (DbjDocSpace インターフェース) を取得します。

このメソッドは、一つの文書空間にアクセスするほかのメソッドの実行前に実行する必要があります。

確立しているセッションの有効期限は、DbjSession#logout メソッドを実行するまで、または文書空間への接続をタイムアウト (DocumentBroker サーバで設定される) するまでです。

なお、ユーザ認証の失敗、および二重のログインがあった場合は、例外がスローされません。

(2) 形式

```
DbjDocSpace login(
    String      userName,
    String      passWord
)
```

(3) 引数

userName
ログインするユーザのユーザ識別子を指定します。

passWord
ログインするユーザのパスワードを指定します。

8. DocumentBroker で提供されるクラスライブラリ

(4) 戻り値

文書空間オブジェクトアクセスインターフェース (DbjDocSpace インターフェース)

(5) 例外

DbjCORBAException

CORBA エラーの場合

DbjDBException

DB エラーの場合

DbjException

Java クラスライブラリ固有のエラーの場合

DbjIllegalDocSpaceIdException

接続先の文書空間識別子が不正の場合

DbjNotAuthenticatedException

認証エラーの場合

DbjSessionOverflowException

セッション数が同時に接続できる数を超えた場合

IllegalStateException

すでにログインしていた場合

8.4.2 logout

(1) 機能

文書空間からログアウトしてセッションを切断します。

セッションの切断によって、ログインセッション内で継続して使用される内部データ (検索結果キャッシュなど) は消去されます。

また、切断するセッションに未確定のトランザクションがある場合は、自動的にトランザクションが取り消され、最後にトランザクションが確定した状態まで戻ります。

なお、このメソッドのエラーは、エラーログファイルに出力されます。

(2) 形式

```
void logout()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

なし

8.4.3 begin

(1) 機能

トランザクションを開始します。

このメソッドを二重に実行した場合、例外がスローされます。

(2) 形式

```
void begin()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DbjCORBAException
CORBA エラーの場合

DbjDBException
DB エラーの場合

DbjException
Java クラスライブラリ固有のエラーの場合

DbjSessionNotConnectException
セッションが接続されていない場合

IllegalStateException
DbjSession#begin メソッドが二重に実行された場合

8.4.4 rollback

(1) 機能

トランザクションを取り消して、最後に DbjSession#commit メソッドが実行された状態

8. DocumentBroker で提供されるクラスライブラリ

まで戻します。

トランザクションが開始されていない場合も、正常終了します。

(2) 形式

```
void rollback()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DbjCORBAException
CORBA エラーの場合

DbjDBException
DB エラーの場合

DbjException
Java クラスライブラリ固有のエラーの場合

DbjSessionNotConnectException
セッションが接続されていない場合

8.4.5 commit

(1) 機能

トランザクションを確定します。

トランザクションが開始されていない場合、例外がスローされます。

(2) 形式

```
void commit()
```

(3) 引数

なし

(4) 戻り値

なし

(5) 例外

DbjCORBAException

CORBA エラーの場合

DbjDBException

DB エラーの場合

DbjException

Java クラスライブラリ固有のエラーの場合

DbjSessionNotConnectException

セッションが接続されていない場合

IllegalStateException

DbjSession#begin メソッドの実行前に DbjSession#commit メソッドが実行された
場合

8.5 DbjDocSpace インターフェース

DbjDocSpace インターフェースは、文書空間へのアクセスを扱うインターフェースです。

DocumentBroker Standard GUI Development Kit におけるファクトリクラス (DecmFactory) には、このインターフェースを渡す必要があります。

DbjDocSpace インターフェースには、メソッドはありません。

9

コーディング例

ここでは、クラスライブラリを使用したコーディング例について説明します。

-
- 9.1 作成するコーディング例

 - 9.2 共通操作パッケージのコーディング例

 - 9.3 文書操作パッケージのコーディング例

 - 9.4 フォルダ操作パッケージのコーディング例

 - 9.5 検索操作パッケージのコーディング例

 - 9.6 文書種別マスタ操作パッケージのコーディング例
-

9.1 作成するコーディング例

ここでは、DocumentBroker Standard GUI Development Kit で提供しているパッケージのコーディング例について説明します。

このコーディング例では、クラスライブラリのメソッドのコールシーケンスを例示します。また、クラスライブラリの各メソッドの使用方法を例示します。

このコーディング例を参考にして、実際のアプリケーションプログラムを作成してください。

9.1.1 前提条件

このコーディング例の前提条件を説明します。

このコーディング例では、アプリケーションプログラムのうち、クラスライブラリを使用したインターフェースの箇所だけを記述します。GUI や定型業務に依存した箇所については説明しません。

また、このコーディング例は、クラスライブラリのすべてのメソッドを使用しているものではありません。

9.2 共通操作パッケージのコーディング例

共通操作パッケージ (jp.co.hitachi_system.sContents.API.Common) のコーディング例を説明します。

9.2.1 DecmFactory インスタンスの作成

DecmFactory クラスのインスタンスを作成する場合のコーディング例を、次に示します。

```

1  //////////////////////////////////////
2  // dbjDocSpace: DbjDocSpaceインスタンス
3  // loginId      : DocumentBroker Serverログイン用ユーザID
4  // loginPw      : DocumentBroker Serverログイン用パスワード
5  // loginNm      : DocumentBroker Serverログイン用ユーザ名
6  //////////////////////////////////////
7
8  // DocumentBroker Serverログイン
9  dbjDocSpace = dbjSession.login(loginId, loginPw);
10
11 // API操作用パラメタを指定
12 String userId = loginId;
13 String userNm = loginNm;
14
15 // DecmFactoryインスタンスを取得する
16 DecmFactory decmFactory = new DecmFactory(dbjDocSpace, userId,
userNm, DecmFactory.DB_HIRDB);

```

9.3 文書操作パッケージのコーディング例

文書操作パッケージ (jp.co.hitachi_system.sContents.API.Document) のコーディング例を説明します。

9.3.1 文書の作成・取得・削除

文書を作成, 取得, 削除する場合のコーディング例を, 次に示します。

(1) 主ファイルだけで構成する文書を作成後ダウンロードする

```

1  //////////////////////////////////////
2  // decmFactory : DecmFactoryインスタンス
3  // fldOiid    : 登録先フォルダOIID
4  // filePath   : 登録するローカルファイルパス
5  // localPath  : 取得先ローカルパス
6  //////////////////////////////////////
7
8  // 文書操作のインスタンスを取得
9  DecmDoc doc = decmFactory.createDecmDoc();
10
11 // プロパティ操作インスタンスを取得
12 DecmDocProp docProp = decmFactory.createDecmDocProp();
13
14 // プロパティを設定
15 docProp.setStrProp( "esysProp_Name", "会議資料" ); doc.setDocProp(
16 docProp );
17 // 登録先フォルダ設定
18 doc.setFolderOiid( fldOiid );
19
20 // ファイル設定
21 doc.setFilePath( filePath );
22
23 // 文書作成
24 String oiid = doc.createDoc( false, true );
26 // ダウンロード
27 doc.downloadContent( null,
28                      localPath,
29                      true,
30                      true );

```

(2) 主ファイルと関連ファイルで構成する文書を作成後ダウンロードする

```

1  //////////////////////////////////////
2  // decmFactory : DecmFactoryインスタンス
3  // fldOiid    : 登録先フォルダOIID
4  // filePath   : 主ファイルに登録するローカルファイルパス
5  // subFilePath1 : 関連ファイルに登録するローカルファイルパス
6  // subFilePath2 : 関連ファイルに登録するローカルファイルパス
7  // downloadMasterPath : 主ファイルの取得先ローカルパス
8  // downloadSubPath  : 関連ファイルの取得先ローカルパス
9  //////////////////////////////////////
10

```

```

11 // 文書操作用のインスタンスを取得
12 DecmDoc doc = decmFactory.createDecmDoc();
13 // プロパティ操作インスタンスを取得
14 DecmDocProp docProp = decmFactory.createDecmDocProp();
15
16 // プロパティを設定
17 docProp.setStrProp( "esysProp_Name", "会議資料" );
18 doc.setDocProp( docProp );
19
20 // 登録先フォルダ設定
21 doc.setFolderOiid( fldOiid );
22
23 // ファイル設定
24 doc.setFilePath( masterFilePath );
25
26 // 関連ファイル操作インスタンスを取得
27 DecmSubFile subFile1 = decmFactory.createDecmSubFile();
28 DecmSubFile subFile2 = decmFactory.createDecmSubFile();
29
30 // 関連ファイルリストを設定
31 subFile1.setFilePath( subFilePath1 );
32 subFile2.setFilePath( subFilePath2 );
33 Vector<DecmSubFile> subFileList = new Vector<DecmSubFile>();
34 subFileList.add( subFile1 );
35 subFileList.add( subFile2 );
36 doc.setSubFileList( subFileList );
37
38
39 // 文書作成
40 String oiid = doc.createDoc( false,
41                             true );
42
43 // 主ファイルをダウンロード
44 doc.downloadContent( null,
45                     downloadMasterPath,
46                     true,
47                     true );
48
49 // 関連ファイルをダウンロード
50 Vector<DecmSubFile> subFileList = doc.fetchSubFile( null );
51 subFileList.get( 0 ).downloadContent( downloadSubPath,
52                                     true,
53                                     true );

```

(3) 文書を削除する

```

1  //////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  // docOiid          : 操作対象文書のバージョンOIID
4  //////////////////////////////////////
5
6  // 文書操作用のインスタンスを取得
7  DecmDoc doc = decmFactory.createDecmDoc();
8
9  // 文書に接続
10 doc.setOiid( docOiid );
11
12 // 文書削除
13 doc.deleteDoc( true );

```

9.3.2 文書のプロパティの更新・取得

文書のプロパティを更新，取得する場合のコーディング例を次に示します。

(1) 主ファイルのプロパティを更新後に取得する

```

1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOID
4 // filePath         : 更新後のファイルに登録するローカルパス
5 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
6
7 // 文書操作のインスタンスを取得
8 DecmDoc doc = decmFactory.createDecmDoc();
9
10 // 文書に接続
11 doc.setOiid( docOiid );
12
13 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
14 // 更新
15 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
16
17 // ファイルを設定
18 doc.setFilePath( filePath );
19
20 // プロパティ操作のインスタンスを取得
21 DecmDocProp docProp = decmFactory.createDecmDocProp();
22
23 // プロパティを設定して更新
24 docProp.setStrProp( "esysProp_Name", "会議資料_改訂" );
25 docProp.setStrProp( "esysProp_Comment", "誤記訂正" );
26
27 doc.setDocProp( docProp );
28
29 doc.updateDocProp( false,
30                   true );
31
32 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
33 // 取得
34 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
35 DecmDocProp docProp = doc.fetchDocProp( true );

```

(2) 関連ファイルのコンテンツを更新後にプロパティを取得する

```

1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOID
4 // subFilePath1     : 更新後のファイルに登録するローカルパス
5 // subFilePath2     : 更新後のファイルに登録するローカルパス
6 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
7
8 // 文書操作のインスタンスを取得
9 DecmDoc doc = decmFactory.createDecmDoc();
10
11 // 文書に接続
12 doc.setOiid( docOiid );
13
14 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```



```

15 // 更新
16 ///////////////////////////////////////////////////////////////////
17
18 // 関連ファイルリストを取得
19 Vector<DecmSubFile> subFileList = doc.fetchSubFile( null );
20
21 // ファイルを設定
22 subFileList.get( 0 ).setFilePath( subFilePath1 );
23 subFileList.get( 1 ).setFilePath( subFilePath2 );
24
25 // 関連ファイルリストを設定して更新
26 doc.setSubFileList( subFileList );
27
28 doc.updateDocProp( false,
29                   true );
30
31 ///////////////////////////////////////////////////////////////////
32 // 取得
33 ///////////////////////////////////////////////////////////////////
34
35 // 関連ファイルリストを取得
36 Vector<DecmSubFile> subFileList = doc.fetchSubFile( null );
37
38 // プロパティ取得
39 DecmDocProp docProp1 = subFileList.get( 0 ).fetchFileProp( true
40 );
41 DecmDocProp docProp2 = subFileList.get( 1 ).fetchFileProp( true
42 );

```

9.3.3 文書のアクセス権限の更新・取得

文書のアクセス権限を更新、取得する場合のコーディング例を次に示します。

```

1 ///////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOIID
4 ///////////////////////////////////////////////////////////////////
5
6 // 文書操作作用のインスタンスを取得
7 DecmDoc doc = decmFactory.createDecmDoc();
8
9 // 文書に接続
10 doc.setOiid( docOiid );
11
12 ///////////////////////////////////////////////////////////////////
13 // 更新
14 ///////////////////////////////////////////////////////////////////
15
16 // アクセス権限操作作用インスタンスを取得
17 DecmDocPerm docPerm = decmFactory.createDecmDocPerm();
18
19 // アクセス権限を設定し更新
20 docPerm.setEveryonePerm( DecmACLPPerm.PERM_READ );
21 doc.setDocPerm( docPerm );
22
23 doc.updateDocPerm( true );

```

9. コーディング例

```
24
25 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
26 // 取得
27 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
28
29 DecmDocPerm docPerm = doc.fetchDocPerm( true );
```

9.3.4 文書の履歴の取得・削除・更新

文書の履歴を取得，削除，更新する場合のコーディング例を次に示します。

(1) 文書の履歴を取得後に削除する

```
1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  ///
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOIID
4 // dvOiid           : 操作対象文書のバージョンオブジェクトのOIID
5 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  ///
6
7 // 文書操作作用のインスタンスを取得
8 DecmDoc doc = decmFactory.createDecmDoc();
9
10 // 文書に接続
11 doc.setOiid( docOiid );
12
13 // 履歴取得
14 Vector<DecmDocProp> verPropList = doc.fetchDocHistory( true );
15
16 // 履歴削除
17 doc.deleteDocHistory( dvOiid,
18                       true );
```

(2) 関連ファイルの過去のバージョンのプロパティを取得する

```
1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOIID
4 // dvOiid           : 取得対象となる主ファイルのバージョンオブジェクトのOIID
5 //////////////////////////////////////////////////////////////////
  //////////////////////////////////////////////////////////////////
6
7 // 文書操作作用のインスタンスを取得
8 DecmDoc doc = decmFactory.createDecmDoc();
9
10 // 文書に接続
11 doc.setOiid( docOiid );
12
13 // 関連ファイルリストを取得
14 Vector<DecmSubFile> subFileList = doc.fetchSubFile( dvOiid );
15
16 // プロパティを取得
17 DecmDocProp docProp = subFileList.get( 0 ).fetchFileProp( true );
```

(3) 文書の履歴を更新する (チェックアウトしてからチェックインする)

```

1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
3 // decmFactory      : DecmFactoryインスタンス
4 // docOiid          : 操作対象文書のバージョンingOIID
5 // dvOiid           : 取得対象となる主ファイルのバージョンオブジェクトの
6 // OIID
7 // updateMasterFile : 更新後の主ファイルに登録するローカルパス
8 // updateSubFile1   : 更新後の関連ファイルに登録するローカルパス
9 // updateSubFile2   : 更新後の関連ファイルに登録するローカルパス
10 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
11 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
12 // 文書操作作用のインスタンスを取得
13 DecmDoc doc = decmFactory.createDecmDoc();
14
15 // 文書に接続
16 doc.setOiid( docOiid );
17
18 // チェックアウト
19 doc.checkoutDoc( true );
20
21 // 更新するファイルを設定(主ファイル)
22 doc.setFilePath( updateMasterFile );
23
24 // 関連ファイルを取得
25 Vector<DecmSubFile> subFileList = doc.fetchSubFile( dvOiid );
26
27 // 更新するファイルを設定(関連ファイル)
28 subFileList.get( 0 ).setFilePath( updateSubFile1 );
29 subFileList.get( 1 ).setFilePath( updateSubFile2 );
30
31 // 関連ファイルを設定
32 doc.setSubFileList( subFileList );
33
34 // チェックイン
35 doc.checkinDoc( false,
36                true );

```

9.3.5 文書をごみ箱へ移動したあと元に戻す

文書をごみ箱へ移動後、元に戻す場合のコーディング例を次に示します。

```

1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
3 // decmFactory      : DecmFactoryインスタンス
4 // docOiid          : 操作対象文書のバージョンingOIID
5 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
6 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
7 // 文書操作作用のインスタンスを取得
8 DecmDoc doc = decmFactory.createDecmDoc();
9
10 // 文書に接続
11 doc.setOiid( docOiid );
12
13 // ごみ箱へ移動

```

9. コーディング例

```
13 doc.trashDoc( true );
14
15 // ごみ箱から元に戻す
16 doc.trashBackDoc( true );
```

9.3.6 文書の移動

文書を移動する場合のコーディング例を次に示します。

```
1 ///////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // docOiid          : 操作対象文書のバージョンingOIID
4 // fldOiid          : 移動先フォルダのOIID
5 ///////////////////////////////////////////////////////////////////
6
7 // 文書操作のインスタンスを取得
8 DecmDoc doc = decmFactory.createDecmDoc();
9
10 // 文書に接続
11 oc.setOiid( docOiid );
12
13 // 移動
14 doc.moveDoc( fldOiid, true );
```

9.4 フォルダ操作パッケージのコーディング例

フォルダ操作パッケージ (jp.co.hitachi_system.sContents.API.Folder) のコーディング例を説明します。

9.4.1 フォルダの作成・削除

フォルダを作成，削除する場合のコーディング例を次に示します。

(1) フォルダを作成する

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5  // フォルダ操作用インスタンスを取得
6  DecmFolder fld = decmFactory.createDecmFolder();
7
8  // プロパティ操作用インスタンスを取得
9  DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
10
11 // プロパティを設定
12 fldProp.setStrProp( "esysProp_Name", "文書フォルダ" );
13 fld.setFolderProp( fldProp );
14
15 // フォルダ作成
16 fld.createDocFolder( DecmFolder.ACL_TYPE_PRIVATE,
17                    DecmFolder.DOC_CREATE_OK,
18                    "",
19                    true );

```

(2) ノードフォルダを作成する

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5  // フォルダ操作用インスタンスを取得
6  DecmFolder fld = decmFactory.createDecmFolder();
7
8  // プロパティ操作用インスタンスを取得
9  DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
10
11 // プロパティを設定
12 fldProp.setStrProp( "esysProp_Name", "ノードフォルダ" );
13 fld.setFolderProp( fldProp );
14
15 // フォルダ作成
16 fld.createNodeFolder( true );

```

(3) 公開フォルダを作成する

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4

```

9. コーディング例

```
5 // フォルダ操作用インスタンスを取得
6 DecmFolder fld = decmFactory.createDecmFolder();
7
8 // プロパティ操作用インスタンスを取得
9 DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
10
11 // プロパティを設定
12 fldProp.setStrProp( "esysProp_Name", "公開フォルダ" );
13 fld.setFolderProp( fldProp );
14
15 // フォルダ作成
16 fld.createPubFolder( DecmFolder.ACL_TYPE_PRIVATE,
17                     true );
```

(4) フォルダを削除する

```
1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // fldOiid          : 操作対象フォルダのOIID
4 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6 //フォルダ操作用インスタンスを取得
7 DecmFolder fld = decmFactory.createDecmFolder();
8
9 //フォルダに接続
10 fld.setOiid( fldOiid );
11
12 // 削除
13 fld.deleteFolder( true );
```

9.4.2 フォルダのプロパティの作成・更新

フォルダのプロパティを作成，更新する場合のコーディング例を次に示します。

(1) 文書フォルダのプロパティを更新後に取得する

```
1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // fldOiid          : 操作対象フォルダのOIID
4 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6 // フォルダ操作用インスタンスを取得
7 DecmFolder fld = decmFactory.createDecmFolder();
8
9 // フォルダに接続
10 fld.setOiid( fldOiid );
11
12 // プロパティ操作用インスタンスを取得
13 DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
14
15 // プロパティを設定
16 fldProp.setStrProp( "esysProp_Name", "文書フォルダ_改訂" );
17 fldProp.setStrProp( "esysProp_Comment", "誤記訂正" );
18 fld.setFolderProp( fldProp );
19
20 // プロパティ更新
21 fld.updateDocFolderProp( true );
```

```

22
23 // プロパティ取得
24 DecmFolderProp getProp = fld.fetchFolderProp( false );

```

(2) ノードフォルダのプロパティを更新後に取得する

```

1  ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  // fldOiid         : 操作対象フォルダのOIID
4  ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6  // フォルダ操作用インスタンスを取得
7  DecmFolder fld = decmFactory.createDecmFolder();
8
9  // フォルダに接続
10 fld.setOiid( fldOiid );
11
12 // プロパティ操作用インスタンスを取得
13 DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
14
15 // プロパティを設定
16 fldProp.setStrProp( "esysProp_Name", "ノードフォルダ_改訂" );
17 fldProp.setStrProp( "esysProp_Comment", "誤記訂正" );
18 fld.setFolderProp( fldProp );
19
20 // プロパティ更新
21 fld.updateNodeFolderProp( true );
22
23 // プロパティ取得
24 DecmFolderProp getProp = fld.fetchFolderProp( false );

```

(3) 公開フォルダのプロパティを更新後に取得する

```

1  ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  // fldOiid         : 操作対象フォルダのOIID
4  ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6  // フォルダ操作用インスタンスを取得
7  DecmFolder fld = decmFactory.createDecmFolder();
8
9  // フォルダに接続
10 fld.setOiid( fldOiid );
11
12 // プロパティ操作用インスタンスを取得
13 DecmFolderProp fldProp = decmFactory.createDecmFolderProp();
14
15 // プロパティを設定
16 fldProp.setStrProp( "esysProp_Name", "ノードフォルダ_改訂" );
17 fldProp.setStrProp( "esysProp_Comment", "誤記訂正" );
18 fld.setFolderProp( fldProp );
19
20 // プロパティ更新
21 fld.updatePubFolderProp( true );
22
23 // プロパティ取得
24 DecmFolderProp getProp = fld.fetchFolderProp( false );

```

9.4.3 フォルダのアクセス権限の更新・取得

フォルダのアクセス権限を更新・取得する場合のコーディング例を次に示します。

```
1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 // fldOiid          : 操作対象フォルダのOIID
4 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6 // フォルダ操作作用インスタンスを取得
7 DecmFolder fld = decmFactory.createDecmFolder();
8
9 // フォルダに接続
10 fld.setOiid( fldOiid );
11
12 // アクセス権限操作作用インスタンスを取得
13 DecmFolderPerm fldPerm = decmFactory.createDecmFolderPerm();
14
15 // アクセス権限を設定
16 fldPerm.setEveryonePerm( DecmACLPerm.PERM_READ );
17 fld.setFolderPerm( fldPerm );
18
19 // アクセス権限更新
20 fld.updateFolderPerm( true );
21
22 // アクセス権限取得
23 DecmFolderPerm getPerm = fld.fetchFolderPerm( true );
```


9.5 検索操作パッケージのコーディング例

検索操作パッケージ (jp.co.hitachi_system.sContents.API.Select) のコーディング例を説明します。

9.5.1 文書の検索

文書を検索する場合のコーディング例を、次に示します。

コーディング例での検索の概要は次のとおりです。

実行する操作

文書名が「営業」で始まる文書を取得する

検索条件

- 文書名、および更新者の一覧を取得する
- 関連ファイルを検索対象とする

生成する EQL

- SELECT esysProp_Name, esysProp_ModifierFROM
ecmClass_FiledocDVWHERE esysProp_Name XLIKE '営業 %'

```

1  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5  // 文書検索用インスタンスを取得
6  DecmSelect decmSelect = decmFactory.createDecmSelect();
7
8  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9  // 検索パラメタ指定
10 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
11
12 // 検索取得列の指定
13 Vector<String> selectList = new Vector<String>();
14 selectList.add(DecmTableDef.ESYSPROP_NAME);
15 selectList.add(DecmTableDef.ESYSPROP_MODIFIER);
16 decmSelect.setSelectList(selectList);
17
18 // 関連ファイルを検索対象とする
19 decmSelect.setSubFileSearchMode(true);
20
21 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
22 // 検索条件を指定
23 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
24
25 //検索条件用のインスタンスを取得
26 DecmCondition condition = decmFactory.createDecmCondition();
27
28 // 検索条件作成
29 condition.createStrCond(DecmTableDef.ESYSPROP_NAME,
DecmCondition.OP_STWITH, "営業");

```

9. コーディング例

```
30
31 // 検索条件を指定
32 decmSelect.addCondition(condition);
33
34 ///////////////////////////////////////////////////////////////////
35 // 検索結果を取得
36 ///////////////////////////////////////////////////////////////////
37 Vector<DecmDocProp> ret = decmSelect.searchDoc();
```

9.5.2 組み合わせ検索条件を指定した文書の検索

組み合わせ検索条件を指定して文書を検索する場合のコーディング例を、次に示します。

コーディング例での検索の概要は次のとおりです。

実行する操作

文書名が「営業」か「経営」で始まり、更新者が「user01」の文書を取得する

検索条件

- 文書名、および更新者の一覧を取得する
- 関連ファイルを検索対象とする

生成する EQL

```
• SELECT esysProp_Name, esysProp_Modifier FROM
  ecmClass_FiledocDVWHERE (esysProp_Name XLIKE '営業 %' OR
  esysProp_Name XLIKE '経営 %') AND esysProp_Modifier == 'user01'
```

```
1 ///////////////////////////////////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス
3 ///////////////////////////////////////////////////////////////////
4
5 // 文書検索用インスタンスを取得
6 DecmSelect decmSelect = decmFactory.createDecmSelect();
7
8 ///////////////////////////////////////////////////////////////////
9 // 検索パラメタ指定
10 ///////////////////////////////////////////////////////////////////
11
12 // 検索取得列の指定
13 Vector<String> selectList = new Vector<String>();
14 selectList.add(DecmTableDef.ESYSPROP_NAME);
15 selectList.add(DecmTableDef.ESYSPROP_MODIFIER);
16 decmSelect.setSelectList(selectList);
17
18 // 関連ファイルを検索対象とする
19 decmSelect.setSubFileSearchMode(true);
20
21 ///////////////////////////////////////////////////////////////////
22 // 検索条件を指定
23 ///////////////////////////////////////////////////////////////////
24
25 //検索条件用のインスタンスを取得
26 DecmCondition condition_A = decmFactory.createDecmCondition();
```

```

27 DecmCondition condition_B = decmFactory.createDecmCondition();
28 DecmCondition condition_C = decmFactory.createDecmCondition();
29
30 // 組み合わせ検索条件用のインスタンスを取得
31 DecmCondGroup condGroupTop = decmFactory.createDecmCondGroup();
32 DecmCondGroup condGroupSub = decmFactory.createDecmCondGroup();
33
34 // 検索条件作成
35 condition_A.createStrCond(DecmTableDef.ESYSPROP_NAME,
DecmCondition.OP_STWITH, "営業"); /* A */
36 condition_B.createStrCond(DecmTableDef.ESYSPROP_NAME,
DecmCondition.OP_STWITH, "経営"); /* B */
37 condition_C.createStrCond(DecmTableDef.ESYSPROP_MODIFIER,
DecmCondition.OP_EQ, "user01"); /* C */
38
39 // 組み合わせ検索条件を作成
40 // A or B
41 condGroupSub.addCond(condition_A, DecmCondGroup.OP_AND); // 第二
引数の指定値は無視される
42 condGroupSub.addCond(condition_B, DecmCondGroup.OP_OR);
43
44 // (A or B) and C
45 condGroupTop.addCondGroup(condGroupSub, DecmCondGroup.OP_AND); /
/ 第二引数の指定値は無視される
46 condGroupTop.addCond(condition_C, DecmCondGroup.OP_AND);
47
48 // 検索条件を指定
49 decmSelect.setCondGroup(condGroupTop);
50
51 //////////////////////////////////////
52 // 検索結果を取得
53 //////////////////////////////////////
54 Vector<DecmDocProp> ret = decmSelect.searchDoc();

```

9.5.3 全文検索を利用した文書の検索

全文検索を利用して文書を検索する場合のコーディング例を、次に示します。

コーディング例での検索の概要は次のとおりです。

実行する操作

登録ファイルの内容に「営業」という単語が含まれる文書を取得する

検索条件

- 文書名、および更新者の一覧を取得する
- 関連ファイルを検索対象とする

生成する EQL

```

• SELECT esysProp_Name, esysProp_ModifierFROM
ecmClass_FiledocDVWHERE contains_with_score( edmProp_TextIndex, {" 営業
" }) IS TRUE

```

```

1 //////////////////////////////////////
2 // decmFactory      : DecmFactoryインスタンス

```

9. コーディング例

```
3 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
4  
5 // 文書検索用インスタンスを取得  
6 DecmSelect decmSelect = decmFactory.createDecmSelect();  
7  
8 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
9 // 検索パラメタ指定  
10 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
11  
12 // 検索取得列の指定  
13 Vector<String> selectList = new Vector<String>();  
14 selectList.add(DecmTableDef.ESYSPROP_NAME);  
15 selectList.add(DecmTableDef.ESYSPROP_MODIFIER);  
16 decmSelect.setSelectList(selectList);  
17  
18 // 関連ファイルを検索対象とする  
19 decmSelect.setSubFileSearchMode(true);  
20  
21 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
22 // 検索条件を指定  
23 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
24  
25 // 全文検索キーワードを指定  
26 Vector<String> keywordList = new Vector<String>();  
27 keywordList.add("営業");  
28 decmSelect.setKeyword(keywordList);  
29  
30 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
31 // 検索結果を取得  
32 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
33 Vector<DecmDocProp> ret = decmSelect.searchDoc();
```

9.6 文書種別マスタ操作パッケージのコーディング例

文書種別マスタ操作パッケージ (jp.co.hitachi_system.sContents.API.DocMaster) のコーディング例を説明します。

9.6.1 文書種別マスター一覧の取得

文書種別マスター一覧を取得する場合のコーディング例を、次に示します。

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5  // 文書種別マスター一覧取得用インスタンスを取得
6  DecmDocMasterList docMasterList =
decmFactory.createDecmDocMasterList();
7
8  // 文書種別マスター一覧を取得
9  Vector<DecmDocMasterProp> ret =
docMasterList.getDocMasterList();

```

9.6.2 特定の文書種別マスタの OIID の取得

特定の文書種別マスタの OIID を取得する場合のコーディング例を、次に示します。

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  // masterOiid      : 操作対象マスタの OIID
4  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
5
6  // 文書種別マスタ操作用インスタンスを取得
7  DecmDocMaster docMaster = decmFactory.createDecmDocMaster();
8
9  // 操作対象マスタに接続
10 docMaster.setOiid(masterOiid);
11
12 // 文書種別マスタの OIID を取得
13 Vector<DecmDocMasterProp> ret = docMaster.fetchDocMasterProp();

```

9.6.3 属性値マスター一覧の取得

属性値マスター一覧を取得する場合のコーディング例を、次に示します。

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  // decmFactory      : DecmFactoryインスタンス
3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
4
5  // 属性値マスタ操作用インスタンスを取得
6  DecmDocPropMaster docPropMaster =
decmFactory.createDecmDocPropMaster();

```

9. コーディング例

```
7  
8 // 属性値マスター一覧を取得  
9 Vector<DecmDocPropMasterProp> ret =  
docPropMaster.getDocPropMasterList();
```

10 DocumentBroker Standard GUI Development Kit の環境 構築

DocumentBroker Standard GUI Development Kit のインストール，および環境構築の手順について説明します。

10.1 DocumentBroker Standard GUI Development Kit のインストール，および環境構築手順

10.1 DocumentBroker Standard GUI Development Kit のインストール，および環境構築手順

ここでは，DocumentBroker Standard GUI Development Kit のインストール，および環境構築の手順について説明します。

10.1.1 DocumentBroker Standard GUI Development Kit のインストール

この節では，DocumentBroker Standard GUI Development Kit のインストールの方法について説明します。

DocumentBroker Standard GUI Development Kit は，次に示す手順でインストールします。

1. システム管理者（Administrators）の権限を持つユーザ ID でログインします。
2. DocumentBroker Standard GUI Development Kit のインストーラを起動します。
3. インストーラの画面の指示に従って，DocumentBroker Standard GUI Development Kit をインストールしてください。

上書きインストールをするときは，DocumentBroker Standard GUI Development Kit のファイルとライブラリを使用しているプログラムをすべて停止してからインストールをしてください。

上書きインストールをすると，インストールフォルダ以下のすべてのファイルが上書きされます。

10.1.2 Java 実行環境の設定

Java クラスライブラリのパッケージの JAR ファイルは，次のディレクトリに格納されます。

<インストールディレクトリ> ¥java¥lib¥StdGuiApi.jar

注

DocumentBroker Runtime または，DocumentBroker Development Kit のインストールディレクトリです。

上記の JAR ファイルのパスを，Java 実行環境に設定します。使用する環境に合わせてパスを追加してください。

10.1.3 uCosminexus で使用する場合の設定

DocumentBroker Standard GUI Development Kit は、uCosminexus Application Server の J2EE モードでの動作を前提としています。このため、uCosminexus Application Server は、J2EE モードで動作するように環境を構築してください。

なお、uCosminexus Application Server の環境構築の詳細は、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(1) DocumentBroker Runtime , または DocumentBroker Development Kit の設定

DocumentBroker Runtime , または DocumentBroker Development Kit での設定を実施してください。

(2) uCosminexus への設定の追加

Cosminexus Component Container の usrconf.cfg ファイルに、次のパスを追加設定します。

```
add.class.path=<インストールディレクトリ> ¥java¥lib¥StdGuiApi.jar
```

注

DocumentBroker Runtime または、DocumentBroker Development Kit のインストールディレクトリです。

付録

付録 A DocumentBroker Standard GUI Development Kit で使用するプロパ
ティ一覧

付録 B 用語解説

付録 A DocumentBroker Standard GUI Development Kit で使用するプロパティ一覧

ここでは、DocumentBroker Standard GUI Development Kit で使用するプロパティについて説明します。

付録 A.1 DocumentBroker Standard GUI Development Kit で使用するプロパティ

DocumentBroker Standard GUI Development Kit で使用するプロパティの一覧を次の表に示します。

表 A-1 DocumentBroker Standard GUI Development Kit で使用するプロパティ

プロパティ名称	説明	更新	指定の省略
dmaProp_OIID	オブジェクトの OIID 値が表示されます。 文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクトで使用されます。	×	-
esysProp_Name	次に示すオブジェクトの名前、または表示名のプロパティです。 <ul style="list-style-type: none"> 文書、およびフォルダ：「名前」のプロパティを示します。 文書種別マスタ：「文書種別マスタ名」のプロパティを示します。 属性値マスタ：「表示名」のプロパティを示します。 指定できるバイト数は 255 バイトまでです。		作成時： × 更新時：
edmProp_OwnerId	文書、またはフォルダオブジェクトの検索時用所有者 ID プロパティが表示されます。 255 バイトまでの値が設定されます。	×	-
dbrProp_OwnerId	文書、またはフォルダオブジェクトのプロパティ値取得時の所有者 ID プロパティが表示されます。 255 バイトまでの値が設定されます。	×	-
dbuUserProp_OwnerName	文書、またはフォルダオブジェクトの所有者名のプロパティが表示されます。 255 バイトまでの値が設定されます。	×	-

プロパティ名称	説明	更新	指定の省略
esysProp_CreateTime	文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクト作成日時プロパティが表示されます。表示されるのは、1970/01/01からの通算秒値、または time_t 型の値です。255 バイトまでの値が設定されます。	×	-
esysProp_Creator	文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクトに対する作成者 ID プロパティが表示されます。255 バイトまでの値が設定されます。	×	-
esysProp_ModifyTime	文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクトの更新日時プロパティが表示されます。表示されるのは、1970/01/01からの通算秒値、または time_t 型の値です。	×	-
esysProp_Modifier	文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクトに対する更新者 ID プロパティが表示されます。255 バイトまでの値が設定されます。	×	-
dbuUserProp_ModifierName	文書、フォルダ、文書種別マスタ、属性値マスタのオブジェクトに対する更新者名プロパティが表示されます。255 バイトまでの値が設定されます。	×	-
dbuUserProp_ReservedUserId	フォルダオブジェクトに対するチェックアウト中ユーザ ID プロパティが表示されます。255 バイトまでの値が設定されます。	×	-
dbuUserProp_ReservedUserName	フォルダオブジェクトに対するチェックアウト中ユーザ名プロパティが表示されます。255 バイトまでの値が設定されます。	×	-
esysProp_Comment	文書、またはフォルダのオブジェクトに対するコメントプロパティです。指定できるバイト数は 1,024 バイトまでです。		
ecmProp_FolderType	フォルダ種別が表示されます。次の値でフォルダ種別を判断します。 <ul style="list-style-type: none"> • 1：ノードフォルダ • 2：文書フォルダ • 3：公開フォルダ 	×	-

プロパティ名称	説明	更新	指定の省略
ecmProp_FolderLevel	フォルダ階層のレベルのプロパティが表示されます。 次の値で、フォルダ階層のレベルのプロパティを判断します。 ・ 1：最上位フォルダ ・ 2：下位フォルダ	×	-
ecmProp_DocCreateFlg	フォルダ下の文書の作成可否を示すプロパティが表示されます。 次の値で作成の可否を判断します。 ・ 0：文書作成不可 ・ 1：文書作成可	×	-
ecmProp_AclType	文書フォルダ、または公開フォルダに対して、アクセス権限の設定方式を示すプロパティが表示されます。 次の値で文書フォルダ、または公開フォルダのアクセス権限の設定方式を判断します。 文書フォルダの場合 ・ 1：個別に設定 ・ 2：同じ権限を共有 公開フォルダの場合 ・ 1：全員に公開 ・ 2：公開範囲を限定	×	-
ecmProp_OpenPubACL	アクセス権限の設定方式で、文書フォルダに「同じ権限を共有」、公開フォルダに「公開範囲を限定」とした場合に設定しているパブリックACLの oiid 値が表示されます。 フォルダオブジェクトで使用されません。 返却値は、OIID 文字列の下位 52 バイトです。	×	-
ecmProp_OpenPubACL_FULLL	アクセス権限の設定方式で、文書フォルダに「同じ権限を共有」、公開フォルダに「公開範囲を限定」とした場合に設定しているパブリックACLの oiid 値が表示されます。 フォルダオブジェクトで使用されません。 返却値は、OIID 文字列の下位 133 バイトです。	×	-

プロパティ名称	説明	更新	指定の省略
ecmProp_AclSucceed	<p>上位フォルダからのアクセス権限の継承を示すプロパティが表示されます。</p> <p>次の値で、アクセス権限の継承を判断します。</p> <ul style="list-style-type: none"> • 0：上位フォルダからのアクセス権限継承する • 1：上位フォルダからのアクセス権限継承しない 	×	-
ecmProp_DocTypeOIID	<p>フォルダ、または文書に設定されている文書種別マスタの OIID を示すプロパティが表示されます。</p> <p>返却値は、OIID 文字列の下位 52 バイトです。</p>	×	-
ecmProp_DocTypeOIID_FULL	<p>フォルダ、または文書に設定されている文書種別マスタの OIID を示すプロパティが表示されます。</p> <p>返却値は、OIID 文字列の下位 133 バイトです。</p>	×	-
ecmProp_ExtInt1	<p>フォルダ下ファイル容量制限の使用有無を示すプロパティが表示されます。</p> <p>次の値で、ファイル容量制限の使用有無を判断します。</p> <ul style="list-style-type: none"> • 0：フォルダ下容量制限を使用しない • 1：フォルダ下容量制限を使用する <p>なお、このプロパティは最上位のフォルダにだけ設定されます。</p>	×	-
ecmProp_MaxCapacity	<p>フォルダ下ファイル容量制限機能する場合の最大容量（数値部分）を 1 ~ 9,999 の範囲で示すプロパティが表示されます。</p> <p>なお、このプロパティは最上位のフォルダにだけ設定されます。</p>	×	-

プロパティ名称	説明	更新	指定の省略
ecmProp_CapacityUnit	<p>フォルダ下ファイル容量制限機能する場合の最大容量（単位部分）を示すプロパティが表示されます。</p> <p>「ecmProp_MaxCapacity」が示す数値に対する単位が表示されます。</p> <p>次の値で、フォルダ下ファイル容量制限機能する場合の最大容量（単位部分）を判断します。</p> <ul style="list-style-type: none"> • 2：メガバイト • 3：ギガバイト • 4：テラバイト <p>なお、このプロパティは最上位のフォルダにだけ設定されます。</p>	×	-
ecmProp_UseCapacity	<p>フォルダ下ファイル容量制限機能する場合の使用済み容量を示すプロパティが表示されます。</p> <p>なお、このプロパティは最上位のフォルダにだけ設定されます。</p>	×	-
ecmProp_CHOiid	<p>文書のバージョンングオブジェクトの OIID を示すプロパティが表示されます。</p> <p>返却値は、OIID 文字列の下位 52 バイトです。なお、関連ファイルの場合は、主ファイルのバージョンングオブジェクトの OIID が表示されません。</p>	×	-
ecmProp_CHOiid_FULL	<p>文書のバージョンングオブジェクトの OIID 値を示すプロパティが表示されます。</p> <p>返却値は、OIID 文字列の下位 133 バイトです。</p>	×	-
ecmProp_DocType	<p>文書の種類（一般文書、公開文書）を示すプロパティが表示されます。</p> <p>次の値で文書の種類を判断します。</p> <ul style="list-style-type: none"> • 1：一般文書 • 2：審査・承認機能による公開文書 	×	-
ecmProp_FileName	<p>登録されている文書のファイル名称を示すプロパティが表示されます。</p> <p>255 バイトまでの値が設定されます。</p>	×	-
ecmProp_FileSize	<p>登録されている文書のファイルサイズ（バイト数）を示すプロパティが表示されます。</p>	×	-

プロパティ名称	説明	更新	指定の省略
ecmProp_ParentContainer	文書が所属しているフォルダの OIID 値を示すプロパティが表示されます。返却値は、OIID 文字列の下位 52 バイトです。	×	-
ecmProp_ParentContainer_FULLL	文書が所属しているフォルダの OIID 値を示すプロパティが表示されます。返却値は、OIID 文字列の下位 133 バイトです。	×	-
ecmProp_DeleteFlag	文書削除フラグ（ごみ箱の文書）を示すプロパティが表示されます。次の値で、ごみ箱への削除状態を判断します。 <ul style="list-style-type: none"> • 0：通常 • 1：ごみ箱へ削除した状態 	×	-
ecmProp_DeleteUserId	文書削除を実行したユーザの ID を示すプロパティが表示されます。255 バイトまでの値が設定されます。	×	-
ecmProp_DeleteTime	ごみ箱への削除を実行した日時を示すプロパティが表示されます。time_t 型の値で表示されます。	×	-
ecmProp_DocNumber	採番した文書番号を示すプロパティが表示されます。255 バイトまでの値が設定されます。	×	-
ecmProp_UserInt1 ~ ecmProp_UserInt5	ユーザプロパティです。数値型、または日付型に使用します。次の値で数値型、または日付型を示します。 <ul style="list-style-type: none"> • 数値型の場合：INT 型の数値 • 日付型の場合：「YYYYMMDD」形式の数値 例えば、2008 年 1 月 18 日の場合、20080118 と指定します。 		
ecmProp_UserStr1 ~ ecmProp_UserStr5	ユーザプロパティです。文字列型に使用します。指定できるバイト数は 255 バイトまでです。		
ecmProp_DocTypePropInt1 ~ ecmProp_DocTypePropInt10	文書種別マスタ用プロパティ（数値）です。数値型、または日付型に使用します。		

プロパティ名称	説明	更新	指定の省略
ecmProp_DocTypePropStr1 ~ ecmProp_DocTypePropStr10	文書種別マスタ用プロパティ（文字列）です。 文字列型、またはマスタ型に使用します。 指定できるバイト数は 255 バイトまでです。		
ecmProp_DocTypePropLongStr1 ~ ecmProp_DocTypePropLongStr5	文書種別マスタ用プロパティ（文字列）です。 文字列型に使用します。 指定できるバイト数は 4,096 バイトまでです。		
ecmProp_DocNumProp	採番に使用するプロパティ名を示すプロパティが表示されます。 文書種別マスタオブジェクトで使用されます。 255 バイトまでの値が設定されます。	×	-
ecmProp_DocNumDelimiter	採番用の区切り文字を示すプロパティが表示されます。 文書種別マスタオブジェクトで使用されます。 255 バイトまでの値が設定されます。	×	-
ecmProp_DocNumSuffix	採番の接尾辞を示すプロパティが表示されます。 文書種別マスタオブジェクトで使用されます。 255 バイトまでの値が設定されます。	×	-
ecmProp_Level	属性値マスタの階層を示すプロパティが表示されます。 255 バイトまでの値が設定されます。 次の値で属性値階層の階層を判断します。 • 1：最上位 • 2～10：下位	×	-
ecmProp_DocTypePropName	属性値マスタを使用する文書種別マスタ用のプロパティ名を示すプロパティが表示されます。 255 バイトまでの値が設定されます。 最上位のオブジェクトにだけ設定されます。	×	-
ecmProp_DocTypePropCode	属性値マスタの階層を示すプロパティ表示名のコード値を示すプロパティが表示されます。 255 バイトまでの値が設定されます。	×	-

プロパティ名称	説明	更新	指定の省略
ecmProp_DocTypeParentCode	上位属性値階層の属性値コードを示すプロパティが表示されます。255 バイトまでの値が設定されます。指定例を次に示します。 A (最上位) B C D の階層を持つ場合 B の設定値 : A C の設定値 : A-B D の設定値 : A-B-C 自分の最上位の属性値コードから、一つ上の親コードまでを「-」で区切った属性値コードで設定されます (階層情報を持たせません)。最上位の属性値は空白が指定されます。	×	-
ecmProp_SortNumber	同一階層の属性値コードをソートするためのソート用の番号を示すプロパティが表示されます。属性値マスタオブジェクトで使用されます。	×	-
under	取得したフォルダに対して、下位フォルダの存在有無を示すプロパティが表示されます。次の値で、下位フォルダの存在有無を判断します。 • 0 : 下位フォルダは存在しない • 1 : 下位フォルダが存在する	×	-
esysProp_FileType	文書が、主ファイルが関連ファイルかを示すフラグが表示されます。次の値で、主ファイル、関連ファイルの別を判断します。 • 0 : 主ファイル • 1 : 関連ファイル	×	-
esysProp_TopDVOIID_FULL	関連ファイルが対応する主ファイルのバージョンオブジェクトの OIID が表示されます。値を次に示します。 • 主ファイルの場合 : null, または「」空文字 • 関連ファイルの場合 : 関連ファイルと関連付けられた主ファイルのバージョンオブジェクトの OIID (133 桁)	×	-

凡例

- : 更新, または省略ができます。
 - ×
- × : 更新, または省略ができません。

- : 更新, または省略の対象外です。

注

指定できるバイト数は, 文字コード種別が UTF-8 の場合のバイト数です。
バイト数を確認するときは, UTF-8 形式で文字列長を調べてください。

付録 B 用語解説

ここでは、DocumentBroker Standard GUI Development Kit で使用する用語について説明しています。

(英字)

ACL (Access Control List)

アクセス制御エレメントのことです。

AND 検索

検索条件同士の論理積を求める結合条件です。例えば、著者が「日立太郎」で、文書の実体（コンテンツ）中に「コンピュータ」という文字列を含む文書を検索する、というような場合に使用できません。

API (Application Programming Interface)

アプリケーションプログラムとのインターフェースです。

CORBA (Common Object Request Broker Architecture)

OMG (Object Management Group) が提唱するオブジェクト間の通信メカニズムを提唱する ORB (Object Request Broker) の標準アーキテクチャです。

DMA (Document Management Alliance)

文書管理インターフェースの標準化を図る団体 AIIM (Association for Information and Image Management International) によって定義される共通インターフェースです。

GUID (Global Unique Identifier)

DMA のクラス、プロパティなどに与えるユニークな識別子です。GUID は、「X」を 0 ~ 9, a ~ f (小文字), および A ~ F (大文字) で表される 16 進数とした「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (8 けた -4 けた -4 けた -4 けた -12 けた)」の形式で表されます。

Java クラスライブラリ

DocumentBroker サーバにアクセスするユーザアプリケーションプログラムを Java 言語で作成するために、DocumentBroker Development Kit および DocumentBroker Runtime が提供しているインターフェースです。

OIID (Object Instance Identifier)

データベースに格納されたオブジェクトの存在や格納位置などを明確にするために、各オブジェクトに与えるユニークな識別子です。

Web サービス

DocumentBroker Standard GUI を使用した文書管理サービスを、クライアントに対して提供するサーバです。

(ア行)

アクセス権限

文書空間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。

一般文書

作成済みの文書で、公開していない文書のことです。

イメージ文書登録サーバ

デジタル複合機でスキャンした紙の文書や受信した FAX を、DocumentBroker Standard GUI のイメージ文書として登録するサーバです。

インターフェース

Java クラスライブラリの機能を実行するためのメソッドが定義されている、クラスの仕様です。

Java クラスライブラリでは、オブジェクトを操作する時、まず、インターフェースを取得して、そのインターフェースで定義されているメソッドを実行する、という手順で操作します。

案件

文書に対して審査・承認を依頼してから、その文書が審査・承認を経て、公開されるまでの処理単位です。

オリジナルファイル

文書を PDF 形式に変換する設定の場合、PDF 形式に変換される前の文書ファイルのことです。

(カ行)

監査証跡ログ

フォルダや文書などに対するユーザの操作を記録した履歴のことです。監査証跡ログを参照することで、システムの使用状況や不正なアクセスがないかどうかを確認できます。

関連ファイル

文書に登録された文書ファイルのうち、主ファイル以外の文書ファイルのことです。関連ファイルは複数登録できます。関連ファイルには、一般文書に登録された一般関連ファイルと、公開文書に登録された公開関連ファイルがあります。

継承

既存のクラスを利用して、新しいクラスを定義するオブジェクト指向の技術です。

検索結果キャッシュ

キャッシュ付き検索実行時に作成される検索結果のキャッシュです。キャッシュ付き検索では、例えば、10,000 件の検索結果がある場合に、これを検索結果キャッシュとして保存しておき、ユーザアプリケーションプログラムではこれを 100 件ずつ取得する、という検索ができます。

公開フォルダ

公開文書を格納するフォルダのことです。

公開文書

審査・承認後に公開された文書のことです。

ごみ箱

削除した文書を一時的に退避する場所です。

ごみ箱検索

ごみ箱を指定した検索です。

コンテンツ

文書のデータ部分を指します。DMA で規定されているコンテンツ管理モデルに従ってアクセスされるオブジェクトの実体（例えば、report.doc、document.htm など）です。

(サ行)

主ファイル

文書に一つだけ登録される文書ファイルのことです。主ファイルに対する操作が文書ファイル全体に反映されます。

システム管理者

DocumentBroker Standard GUI Development Kit の運用、管理および保守をするユーザです。
DocumentBroker Standard GUI Development Kit の実行環境を設定することができます。

所有者

文書空間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。

スーパークラス

あるクラスのクラス定義に使用されたクラスを、派生したクラスのスーパークラスといいます。

セッション

文書空間に接続している間のことです。文書空間に接続することを、セッションの確立といいます。
文書空間との接続を解除することを、セッションの切断といいます。

セッションオブジェクト

セッションを確立する機能と、セッション内のトランザクションを制御する機能を持つ、Java クラスライブラリのオブジェクトです。DbjSession インターフェースの機能を実行できます。

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。

全文検索テキストインデクス登録サーバ

文書から全文検索用のテキストデータを抽出し、データベースに対してテキストインデクス情報を登録するためのサーバです。

(タ行)

チェックアウト

文書を更新するために、文書ファイルの実体をダウンロードして、ほかのユーザによって更新されないようにすることです。

チェックイン

チェックアウトした文書を更新し、新しい版（バージョン）の文書をフォルダに再登録することです。

同義語辞書

全文検索実行時に指定する検索タームと同じ意味を持つ単語（同義語）を定義する辞書です。

トランザクション

文書空間オブジェクトの処理単位です。トランザクション単位で、文書空間オブジェクトへの操作を確定、または取り消すことができます。

トランザクションの単位は、ユーザが明示的に指定できます。指定しなかった場合は、メソッド単位でトランザクションが分割されます。

(ナ行)

ノードフォルダ

フォルダ体系に対して階層の分類を整理するために使用するフォルダです。ノードフォルダには文書を格納できません。

(ハ行)

バージョンングオブジェクト

文書の一連のバージョンを統括する文書空間オブジェクトです。

バージョンオブジェクト

文書の個々のバージョンを表現する文書空間オブジェクトです。

パーミッション

文書空間オブジェクトの作成、プロパティ参照、コンテンツ更新などの実行できる操作の範囲を表す値です。オブジェクト作成権限を与えるパーミッション、オブジェクトの操作の範囲を定めるパーミッションがあります。

パブリック ACL

文書空間オブジェクトとして存在するアクセス制御情報です。複数の文書空間オブジェクトで共有できます。

ファクトリクラス

パラメタクラスのオブジェクトの生成、セッションオブジェクトの生成、および文書空間メタ情報アクセスインターフェースの取得を実行する、クラスならびにインターフェースです。

フォルダ

フォルダのプロパティと、フォルダに格納するファイルとの関連づけについての情報で構成される概念です。これらの情報はデータベースに格納されて管理されます。

DocumentBroker Standard GUI Development Kit で扱うフォルダには、文書フォルダ、公開フォルダ、およびノードフォルダがあります。

フォルダ指定検索

指定したフォルダに格納されている文書を検索する方法です。

プロパティ

オブジェクトに関する付加情報です。プロパティに設定される値をプロパティ値といいます。

プロパティ検索

文書のプロパティ（作成日やファイルサイズなど）を検索条件として検索する方法です。

文書

Word やテキストエディターなどのアプリケーションプログラムで作成したファイルの実体と、そのファイルのプロパティを合わせて文書といいます。主ファイルと関連ファイルの複数のファイルを持つことができます。また、複数の版（バージョン）を持つことができ、それぞれの版にファイルを持つことができます。文書には一般文書と公開文書があります。

文書空間

DMA オブジェクトモデルを実装するリポジトリです。

文書管理サーバ

文書の管理機能を提供するサーバです。

文書空間オブジェクト

DocumentBroker Standard GUI Development Kit の文書管理で使用する、文書空間に存在するオブジェクトの総称です。バージョンなし文書、バージョン付き文書、バージョンなしフォルダ、バージョン付きフォルダ、独立データおよびパブリック ACL があります。

文書空間アクセスオブジェクト

セッションを確立した文書空間を、概念的なオブジェクトとして扱うためのオブジェクトです。DbjDocSpace インターフェースの機能を実行できます。

文書空間識別子

セッションを確立する文書空間を特定するための識別子です。GUID 文字列で表されます。

文書種別

文書の種類を表す定義です。

文書フォルダ

一般文書を格納するフォルダのことです。文書の登録の可否が指定できます。

(マ行)

メソッド

データを操作するために定義されている方法です。Java クラスライブラリでは、インターフェースおよびクラスごとにメソッドが定義されています。

(ラ行)

例外クラス

Java クラスライブラリで発生する例外のうち、Java クラスライブラリ固有の例外を扱うクラスの総称です。java.lang.Exception クラスを継承しています。

レンディション

文書の実体（コンテンツ）の形式、およびそのコンテンツを併せた概念です。

ログアウト

文書空間とのセッションを切断することです。

ログイン

文書空間とのセッションを確立することです。ログインするときには、ユーザ識別子とパスワードによる認証処理も実行されます。

ロック

排他制御を実行するために設定する、ロックの種類です。参照ロックと更新ロックがあります。

索引

A

ACL (Access Control List)〔用語解説〕 215
ACL_TYPE_PRIVATE 30
ACL_TYPE_PUBLIC 30
addcond 154
addCondGroup 155
addCondition 146
AND 検索〔用語解説〕 215
API (Application Programming Interface)
〔用語解説〕 215

B

begin 177

C

cancelCheckoutDoc 99
CAP_NOUSE 31
CAP_USE 31
changeCapacityMode 133
checkinDoc 100
checkoutDoc 98
commit 178
CORBA (Common Object Request Broker
Architecture)〔用語解説〕 215
createDecmACLPerm 47
createDecmCondGroup 52
createDecmCondition 52
createDecmDoc 47
createDecmDocMaster 53
createDecmDocMasterList 53
createDecmDocMasterProp 54
createDecmDocPerm 49
createDecmDocProp 48
createDecmDocPropMaster 54
createDecmDocPropMasterProp 55
createDecmFolder 50
createDecmFolderPerm 51
createDecmFolderProp 50
createDecmFolderTree 49

createDecmSelect 51
createDecmSubFile 48
createDoc 85
createDocFolder 123
createIntCond 153
createNodeFolder 122
createPubFolder 125
createSession 173
createStrCond 152

D

DB_HIRDB 21
DbjDocSpace インターフェース 180
DbjFactory0200 クラス 171, 172
DbjFactory0200 クラスのメソッド一覧 171
DbjFactory インターフェース 171, 173
DbjFactory インターフェースのメソッド一
覧 171
DbjSession インターフェース 171, 175
DbjSession インターフェースのメソッド一
覧 171
dbrProp_OwnerId 206
dbuUserProp_ModifierName 207
dbuUserProp_OwnerName 206
dbuUserProp_ReservedUserId 207
dbuUserProp_ReservedUserName 207
DecmACLPerm クラス 69
DecmACLPerm クラスのフィールド一覧 23
DecmACLPerm クラスのメソッド一覧 22
DecmApiAccessException クラス 24
DecmApiACLOutOfRangeException クラス
25
DecmApiAlreadyCheckoutException クラス
24
DecmApiCapacityException クラス 25
DecmApiDBException クラス 24
DecmApiDocNotFoundException クラス 24
DecmApiDocNumNotCorrectException クラ
ス 25
DecmApiException クラス 24

- DecmApiFileNotFoundException クラス 25
DecmApiFolderNotCorrectException クラス 25
DecmApiFolderNotFoundException クラス 24
DecmApiInvalidDocMasterException クラス 25
DecmApiInvalidFileException クラス 25
DecmApiInvalidFolder クラス 25
DecmApiInvalidInitialException クラス 24
DecmApiInvalidParamException クラス 24
DecmApiInvalidPropValueException クラス 25
DecmApiIOException クラス 24
DecmApiLastVersionException クラス 24
DecmApiMasterNotFoundException クラス 24
DecmApiNotCheckoutException クラス 24
DecmApiNotSupport クラス 25
DecmApiPropMasterNotFoundException クラス 24
DecmApiReferenceException クラス 25
DecmCommonPerm クラス 64
DecmCommonPerm クラスのメソッド一覧 22
DecmCommonProp クラス 59
DecmCommonProp クラスのメソッド一覧 21
DecmCondGroup クラス 34, 154
DecmCondGroup クラスのフィールド一覧 34
DecmCondGroup クラスのメソッド一覧 34
DecmCondition クラス 33, 152
DecmCondition クラスのフィールド一覧 33
DecmCondition クラスのメソッド一覧 33
DecmDBControl クラス 23, 73
DecmDocMasterList クラス 35, 158
DecmDocMasterList クラスで扱うプロパティ 158
DecmDocMasterList クラスのメソッド一覧 35
DecmDocMasterProp クラス 36, 164
DecmDocMaster クラス 35, 160
DecmDocMaster クラスで扱うプロパティ 160
DecmDocMaster クラスのメソッド一覧 35
DecmDocPerm クラス 27, 107
DecmDocPropMasterProp クラス 36, 167
DecmDocPropMaster クラス 36, 165
DecmDocPropMaster クラスで扱うプロパティ 165
DecmDocPropMaster クラスのメソッド一覧 36
DecmDocProp クラス 27, 106
DecmDoc クラス 26, 76
DecmDoc クラスで扱うプロパティ 76, 108
DecmDoc クラスのメソッド一覧 26
DecmFactory 46
DecmFactory インスタンスの作成 183
DecmFactory クラス 46
DecmFactory クラスのフィールド一覧 21
DecmFactory クラスのメソッド一覧 20
DecmFolderPerm クラス 31, 137
DecmFolderProp クラス 31, 136
DecmFolderTree クラス 29, 114
DecmFolderTree クラスで扱うプロパティ 114
DecmFolderTree クラスのメソッド一覧 29
DecmFolder クラス 29, 117
DecmFolder クラスで扱うプロパティ 117
DecmFolder クラスのフィールド一覧 30
DecmFolder クラスのメソッド一覧 29
DecmObject クラス 21, 58
DecmSelect クラス 32, 140
DecmSelect クラスで扱うプロパティ 140
DecmSelect クラスのフィールド一覧 33
DecmSelect クラスのメソッド一覧 32
DecmSubFile クラス 27, 108
DecmSubFile クラスのメソッド一覧 28
deleteDoc 87
deleteDocHistory 97
deleteFolder 127
DMA (Document Management Alliance)
〔用語解説〕 215
dmaProp_OIID 206
DOC_CREATE_NG 30

DOC_CREATE_OK 30
 DocumentBroker Runtime , または
 DocumentBroker Development Kit の設定
 203
 DocumentBroker Standard GUI
 Development Kit 提供のクラス一覧 17
 DocumentBroker Standard GUI
 Development Kit で実現する機能 3
 DocumentBroker Standard GUI
 Development Kit で使用するプロパティ 206
 DocumentBroker Standard GUI
 Development Kit で使用するプロパティ一覧
 206
 DocumentBroker Standard GUI
 Development Kit で使用するユーザ種別 15
 DocumentBroker Standard GUI
 Development Kit とは 2
 DocumentBroker Standard GUI
 Development Kit の概要 1
 DocumentBroker Standard GUI
 Development Kit の環境構築 201
 DocumentBroker Standard GUI
 Development Kit のシステム構成 5
 DocumentBroker Standard GUI
 Development Kit のシステム構成例 5
 DocumentBroker Standard GUI
 Development Kit のシステム内の前提ソフト
 ウェア 8
 DocumentBroker Standard GUI
 Development Kit の前提 OS 7
 DocumentBroker Standard GUI
 Development Kit の前提プログラム 7
 DocumentBroker Standard GUI
 Development Kit の同一装置内の前提ソフト
 ウェア 7
 DocumentBroker Standard GUI
 Development Kit のパッケージ一覧 10
 DocumentBroker Standard GUI
 Development Kit パッケージが提供するクラ
 ス 18
 DocumentBroker Standard GUI
 Development Kit を使用してユーザアプリ
 ケーションを開発する場合 7

DocumentBroker Standard GUI
 Development Kit を使用してユーザアプリ
 ケーションを実行する場合 8
 DocumentBroker で提供されるクラスライブ
 ラリ 169
 downloadContent 94, 110

E

ecmProp_AclSucceed 209
 ecmProp_AclType 208
 ecmProp_CapacityUnit 210
 ecmProp_CHOiid 210
 ecmProp_CHOiid_FULLL 210
 ecmProp_DeleteFlag 211
 ecmProp_DeleteTime 211
 ecmProp_DeleteUserId 211
 ecmProp_DocCreateFlg 208
 ecmProp_DocNumber 211
 ecmProp_DocNumDelimiter 212
 ecmProp_DocNumProp 212
 ecmProp_DocNumSuffix 212
 ecmProp_DocType 210
 ecmProp_DocTypeOIID 209
 ecmProp_DocTypeOIID_FULLL 209
 ecmProp_DocTypeParentCode 213
 ecmProp_DocTypePropCode 212
 ecmProp_DocTypePropInt1 ~
 ecmProp_DocTypePropInt10 211
 ecmProp_DocTypePropLongStr1 ~
 ecmProp_DocTypePropLongStr5 212
 ecmProp_DocTypePropName 212
 ecmProp_DocTypePropStr1 ~
 ecmProp_DocTypePropStr10 212
 ecmProp_ExtInt1 209
 ecmProp_FileName 210
 ecmProp_FileSize 210
 ecmProp_FolderLevel 208
 ecmProp_FolderType 207
 ecmProp_Level 212
 ecmProp_MaxCapacity 209
 ecmProp_OpenPubACL 208
 ecmProp_OpenPubACL_FULLL 208
 ecmProp_ParentContainer 211

ecmProp_ParentContainer_FULLL 211
 ecmProp_SortNumber 213
 ecmProp_UseCapacity 210
 ecmProp_UserInt1 ~ ecmProp_UserInt5
 211
 ecmProp_UserStr1 ~ ecmProp_UserStr5
 211
 edmProp_OwnerId 206
 esysProp_Comment 207
 esysProp_CreateTime 207
 esysProp_Creator 207
 esysProp_FileType 213
 esysProp_Modifier 207
 esysProp_ModifyTime 207
 esysProp_Name 206
 esysProp_TopDVOIID_FULLL 213

F

fetchDocHistory 96
 fetchDocMasterProp 161
 fetchDocPerm 92
 fetchDocProp 90
 fetchFileProp 110
 fetchFolderPerm 131
 fetchFolderProp 127
 fetchSubFile 103

G

getACLList 67
 getDocMasterList 158
 getDocPropMasterList 166
 getEveryonePerm 66
 getFactory 172
 getFolderList 115
 getIntProp 61
 getListDocMaster 163
 getLongProp 62
 getNewDocMaster 162
 getOiid 59, 78, 118, 160
 getOwnerId 64
 getOwnerName 65
 getPermission 71

getSearchDocMaster 162
 getStrProp 59
 getSubject 69
 getSubjectType 70
 GUID (Gllobary Unique Idenifier) (用語解
 説) 215

I

initUserProp (文字列型) 55
 initUserProp (文字列型以外) 56
 isCapacityUse 57
 isSubFileSearchMode 145
 isSubTraceMode 84

J

Java クラスライブラリ (用語解説) 215
 Java 実行環境の設定 202

L

lockRead 104, 134
 lockWrite 105, 134
 login 175
 logout 176

M

moveDoc 102

O

OIID (Object Instance Identifier) (用語解
 説) 215
 OP_AND 34
 OP_ENDWITH 33
 OP_EQ 33
 OP_GT 33
 OP_GTEQ 33
 OP_LT 33
 OP_LTEQ 33
 OP_NEQ 33
 OP_OR 34
 OP_PART 33
 OP_STWITH 33

P

PERM_TYPE_GROUP 23
 PERM_TYPE_USER 23
 PERM_FULL 23
 PERM_LINK 23
 PERM_NONE 23
 PERM_READ 23
 PERM_UPDATE 23
 PROP_TYPE_DAT 21
 PROP_TYPE_INTEGER 21
 PROP_TYPE_MASTER 21
 PROP_TYPE_STRING 21
 PUBFOL_ACL_TYPE_PRIVATE 30
 PUBFOL_ACL_TYPE_PUBLIC 30

R

resetCapacityMax 122
 resetCondition 151
 rollback 177

S

searchDoc 150
 setACLList 67
 setCapacityMax 121
 setCondGroup 147
 setDocMaster 82
 setDocPerm 80
 setDocProp 79
 setEveryonePerm 66
 setFetchCount 142
 setFilePath 81, 108
 setFolderMode 145
 setFolderOiid 79, 141
 setFolderPerm 120
 setFolderProp 120
 setIntProp 61
 setKeyword 148
 setLongProp 62
 setOiid 78, 119, 161
 setOwnerId 64
 setOwnerName 65
 setParentOiid 115, 119, 165

setPermission 71
 setRenditionType 83, 109
 setSearchDictionary 149
 setSearchExpanse 149
 setSelectList 147
 setSortKey 143
 setSortMode 143
 setStrProp 60
 setSubFileList 83
 setSubFileSearchMode 146
 setSubject 69
 setSubjectType 70
 setSubTraceMode 84
 setTextPath 81
 setTrashMode 144
 SORT_MODE_DOWN 33
 SORT_MODE_UP 33

T

TRASH_MODE_ALL 33
 TRASH_MODE_DELETED 33
 TRASH_MODE_NORMAL 33
 trashBackDoc 89
 trashDoc 88

U

uCosminexus で使用する場合の設定 203
 under 213
 UNIT_GB 30
 UNIT_MB 30
 UNIT_TB 30
 updateDocFolderProp 129
 updateDocPerm 93
 updateDocProp 90
 updateFolderPerm 132
 updateNodeFolderProp 128
 updatePubFolderProp 130

W

Web サービス〔用語解説〕 215

あ

アクセス権限〔用語解説〕 216
案件〔用語解説〕 216

い

一般文書〔用語解説〕 216
イメージ文書登録サーバ〔用語解説〕 216
インターフェース〔用語解説〕 216
インターフェースの処理の流れ 11

お

オリジナルファイル〔用語解説〕 216

か

監査証拠ログ〔用語解説〕 216
関連ファイル〔用語解説〕 216

き

共通処理機能 3
共通操作パッケージのクラス 20
共通操作パッケージのコーディング例 183
共通操作パッケージのメソッド一覧 20
共通操作パッケージの例外クラス 24
共通パッケージ提供クラスの詳細 45

く

組み合わせ検索条件を指定した文書の検索
196

け

継承〔用語解説〕 216
検索結果キャッシュ〔用語解説〕 216
検索操作機能 4
検索操作パッケージ提供クラス 32
検索操作パッケージ提供クラスの詳細 139
検索操作パッケージのクラス一覧 32
検索操作パッケージのコーディング例 195
検索操作パッケージのメソッド一覧 32

こ

公開フォルダ〔用語解説〕 216
公開文書〔用語解説〕 217
コーディング時の注意事項 14
コーディング例 181
ごみ箱〔用語解説〕 217
ごみ箱検索〔用語解説〕 217
コンテンツ〔用語解説〕 217

さ

作成するコーディング例 182

し

システム管理者〔用語解説〕 217
主ファイル〔用語解説〕 217
使用するクラス, およびインターフェースの
一覧 170
使用するメソッド一覧 170
使用できる文字コード種別 14
所有者〔用語解説〕 217

す

スーパークラス〔用語解説〕 217

せ

セッション〔用語解説〕 217
セッションオブジェクト〔用語解説〕 217
前提条件 182
全文検索〔用語解説〕 217
全文検索テキストインデクス登録サーバ〔用
語解説〕 217
全文検索を利用した文書の検索 197

そ

属性値マスター一覧の取得 199

ち

チェックアウト〔用語解説〕 218
チェックイン〔用語解説〕 218

と

同義語辞書〔用語解説〕 218
 特定の文書種別マスタの OIID の取得 199
 トランザクション〔用語解説〕 218
 トランザクションの開始と終了 16

の

ノードフォルダ〔用語解説〕 218

は

バージョンングオブジェクト〔用語解説〕
 218
 バージョンオブジェクト〔用語解説〕 218
 パーミッション〔用語解説〕 218
 パッケージ提供クラス, およびメソッドの説
 明形式 37
 パブリック ACL〔用語解説〕 218

ふ

ファクトリクラス〔用語解説〕 218
 フォルダ〔用語解説〕 219
 フォルダ指定検索〔用語解説〕 219
 フォルダ操作機能 3
 フォルダ操作パッケージ提供クラス 29
 フォルダ操作パッケージ提供クラスの詳細
 113
 フォルダ操作パッケージのクラス一覧 29
 フォルダ操作パッケージのコーディング例
 191
 フォルダ操作パッケージのメソッド一覧 29
 フォルダのアクセス権限の更新・取得 194
 フォルダの作成・削除 191
 フォルダのプロパティの作成・更新 192
 プロパティ〔用語解説〕 219
 プロパティ検索〔用語解説〕 219
 文書〔用語解説〕 219
 文書管理サーバ〔用語解説〕 219
 文書空間〔用語解説〕 219
 文書空間アクセスオブジェクト〔用語解説〕
 219
 文書空間オブジェクト〔用語解説〕 219

文書空間オブジェクト操作時に必要なアクセ
 ス権限 41
 文書空間オブジェクトの操作時に影響する
 setter メソッド 39
 文書空間識別子〔用語解説〕 219
 文書種別〔用語解説〕 219
 文書種別マスター一覧の取得 199
 文書種別マスタ操作パッケージ提供クラス
 35
 文書種別マスタ操作パッケージ提供クラスの
 詳細 157
 文書種別マスタ操作パッケージのクラス一覧
 35
 文書種別マスタ操作パッケージのメソッド一
 覧 35
 文書種別マスタの操作機能 4
 文書種別マスタ操作パッケージのコーディン
 グ例 199
 文書操作機能 3
 文書操作パッケージ提供クラス 26
 文書操作パッケージ提供クラスの詳細 75
 文書操作パッケージのクラス 26
 文書操作パッケージのコーディング例 184
 文書操作パッケージのメソッド一覧 26
 文書のアクセス権限の更新・取得 187
 文書の移動 190
 文書の検索 195
 文書の作成・取得・削除 184
 文書の状態によるメソッドの実行可否 43
 文書のプロパティの更新・取得 186
 文書の履歴の取得・削除・更新 188
 文書フォルダ〔用語解説〕 219
 文書をごみ箱へ移動したあと元に戻す 189

ま

マルチスレッド環境での注意事項 14

め

メソッド〔用語解説〕 220
 メソッドで説明する項目 37
 メソッドの関係 39

よ

用語解説 215

れ

例外クラス〔用語解説〕 220

レンディション〔用語解説〕 220

ろ

ログアウト〔用語解説〕 220

ログイン〔用語解説〕 220

ロック〔用語解説〕 220

ソフトウェアマニュアルのサービス ご案内

ソフトウェアマニュアルについて、3種類のサービスをご案内します。ご活用ください。

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

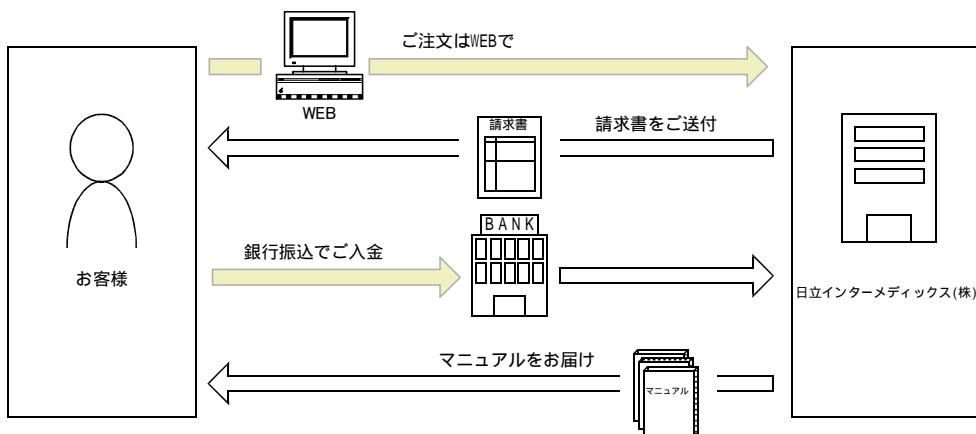
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

ソフトウェアサポートサービスの契約をしていただくと、インターネットでマニュアルを参照できます。本サービスの対象となる契約の種別、及び参照できるマニュアルは、マニュアル情報ホームページでご確認ください。なお、ソフトウェアサポートサービスは、マニュアル参照だけでなく、対象製品に対するご質問への回答、問題解決支援、バージョン更新版の提供など、お客様のシステムの安定的な稼働のためのサービスをご提供しています。まだご契約いただいていない場合は、ぜひご契約いただくことをお勧めします。

3. マニュアルのご注文

日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からご注文ください。



下記 URL にアクセスして必要事項を入力してください。

URL http://www2.himdx.net/manual/privacy.asp?purchase_flag=1

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

入金確認後、7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。