

Oracle8i *interMedia* Audio, Image, Video

ユーザーズ・ガイドおよびリファレンス

リリース 8.1

ORACLE®

Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス リリース 8.1

部品番号 : A62988-1

第 1 版 1999 年 5 月 (第 1 刷)

原本名 : Oracle8i *interMedia* Audio, Image, and Video User's Guide and Reference, Release 8.1.5

原本部品番号 : A67299-01

原本著者 : Rod Ward and Max Chittister

原本協力者 : Alok Srivastava, Raja Chatterjee, Dan Mullen, Susan Mavris, Todd Rowell, Ashok Joshi, Susan Kotsovo, Rosanne Toohey, Bill Beauregard, Jeff Hebert, Chuck Murray, Susan Shepard, Brenda Silva, Deborah Laderoute

Copyright © 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラムの使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当ソフトウェア（プログラム）のリバース・エンジニアリングは禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Legend が適用されます。

Restricted Rights Legend

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに.....	xxi
-----------	-----

1 概要

1.1	Oracle8i <i>interMedia</i>	1-3
1.2	オーディオの概念.....	1-4
1.2.1	デジタル・オーディオ.....	1-4
1.2.2	オーディオ・コンポーネント.....	1-5
1.3	イメージの概念.....	1-5
1.3.1	デジタル・イメージ.....	1-5
1.3.2	イメージ・コンポーネント.....	1-6
1.4	ビデオの概念.....	1-7
1.4.1	デジタル・ビデオ.....	1-7
1.4.2	ビデオ・コンポーネント.....	1-7
1.5	オブジェクト・リレーショナル・テクノロジー.....	1-8
1.5.1	マルチメディア・オブジェクト型およびメソッド.....	1-8
1.5.2	ORDSource オブジェクト型およびメソッド.....	1-9
1.5.2.1	マルチメディア・データの格納.....	1-9
1.5.2.2	マルチメディア・データの問合せ.....	1-10
1.5.2.3	マルチメディア・データへのアクセス.....	1-10
1.6	Oracle8i <i>interMedia</i> の拡張.....	1-10
1.6.1	他の外部ソースおよび他のオーディオ、イメージ、およびビデオ・データ・フォーマットのサポート	1-11
1.6.2	オーディオ・データ処理のサポート.....	1-12
1.6.3	ビデオ・データ処理のサポート.....	1-12

2 *interMedia* の例

2.1	オーディオ・データの例.....	2-1
2.1.1	ソング・オブジェクトの定義.....	2-2
2.1.2	オブジェクト・テーブル SongsTable の作成.....	2-2
2.1.3	ソングへの参照リストを含むリスト・オブジェクトの作成.....	2-3
2.1.4	songList オブジェクトのインプリメンテーションの定義.....	2-3
2.1.5	CD オブジェクトと CD テーブルの作成.....	2-3
2.1.6	SongsTable テーブルへのソングの挿入	2-4
2.1.7	CdTable テーブルへの CD の挿入.....	2-5
2.1.8	SongsTable テーブルへのソングのロード	2-6
2.1.9	ソング・オブジェクトへの参照の、 CdTable テーブル内のソング・リストへの挿入.....	2-7
2.1.10	ソングへの CD 参照の追加.....	2-8
2.1.11	ソングに関するコメントの追加.....	2-9
2.1.12	CD 内のソングからのオーディオ・データの取得.....	2-9
2.1.13	CD 内のソングに関するすべてのコメントの表示.....	2-9
2.1.14	<i>interMedia</i> の拡張による新しいオーディオ・データ・フォーマットのサポート.....	2-10
2.1.15	新しいオブジェクト型を使用した <i>interMedia</i> の拡張.....	2-10
2.1.16	オブジェクト・ビューと共に使用するオーディオ型.....	2-11
2.2	イメージ・データの例.....	2-12
2.2.1	既存のテーブルへのイメージ型の追加.....	2-13
2.2.2	新規テーブルへのイメージ型の追加.....	2-13
2.2.3	BLOB イメージを使用した行の挿入	2-14
2.2.4	BLOB イメージを使用した行の移入	2-14
2.2.5	BFILE イメージを使用した行の挿入.....	2-15
2.2.6	BFILE イメージを使用した行の移入.....	2-16
2.2.7	行の問合せ.....	2-17
2.2.8	外部ファイルからデータベース内へのイメージのインポート.....	2-17
2.2.9	イメージのコピー	2-18
2.2.10	イメージ・フォーマットの変換.....	2-18
2.2.11	コピーおよび変換の一括操作.....	2-19
2.2.12	新しい型を使用した <i>interMedia</i> の拡張.....	2-19
2.2.13	オブジェクト・ビューと共に使用するイメージ型.....	2-21
2.2.14	各国語サポート (NLS) 問題への取組み.....	2-22
2.3	ビデオ・データの例.....	2-23
2.3.1	Clip オブジェクトの定義.....	2-24

2.3.2	オブジェクト・テーブル clipsTable の作成.....	2-24
2.3.3	クリップ・リストを含むリスト・オブジェクトの作成.....	2-24
2.3.4	clipList オブジェクトのインプリメンテーションの定義.....	2-25
2.3.5	ビデオ・オブジェクトとビデオ・テーブルの作成.....	2-25
2.3.6	clipsTable テーブルへのビデオ・クリップの挿入.....	2-26
2.3.7	videoTable テーブルへの行の挿入.....	2-26
2.3.8	clipsTable テーブルへのビデオのロード.....	2-27
2.3.9	クリップ・オブジェクトへの参照の、 videoTable テーブル内のクリップ・リストへの挿入	2-28
2.3.10	ビデオ・オブジェクトへの参照の、クリップへの挿入.....	2-29
2.3.11	videoTable テーブルからのビデオ・クリップの取得.....	2-30
2.3.12	interMedia の拡張による新しいビデオ・データ・フォーマットのサポート.....	2-30
2.3.13	新しいオブジェクト型を使用した interMedia の拡張.....	2-31
2.3.14	オブジェクト・ビューでのビデオ型の使用.....	2-31
2.4	interMedia の拡張による新しいデータ・ソースのサポート.....	2-32

3 ORDAudio 参照情報

3.1	オブジェクト型.....	3-2
	ORDAudio オブジェクト型.....	3-3
3.2	メソッド.....	3-8
3.2.1	例題テーブル定義.....	3-12
3.2.2	updateTime 属性と関連付けられた ORDAudio メソッド.....	3-13
	getUpdateTime メソッド.....	3-14
	setUpdateTime() メソッド.....	3-15
3.2.3	説明属性と関連付けられた ORDAudio メソッド.....	3-16
	setDescription() メソッド.....	3-17
	getDescription メソッド.....	3-19
3.2.4	mimeType 属性と関連付けられた ORDAudio メソッド.....	3-20
	setMimeType() メソッド.....	3-21
	getMimeType メソッド.....	3-23
3.2.5	ソース属性と関連付けられた ORDAudio メソッド.....	3-24
	processSourceCommand() メソッド.....	3-25
	isLocal メソッド.....	3-27
	setLocal メソッド.....	3-28

	clearLocal メソッド.....	3-29
	setSource() メソッド	3-30
	getSource メソッド	3-32
	getSourceType メソッド	3-33
	getSourceLocation メソッド.....	3-35
	getSourceName メソッド.....	3-36
	import() メソッド.....	3-37
	importFrom() メソッド	3-39
	export() メソッド.....	3-41
	getContentLength() メソッド.....	3-43
	getContentInLob() メソッド.....	3-44
	getContent メソッド	3-46
	deleteContent メソッド.....	3-47
3.2.6	ファイル関連操作と関連付けられた ORDAudio メソッド	3-48
	openSource() メソッド	3-49
	closeSource() メソッド	3-51
	trimSource() メソッド.....	3-53
	readFromSource() メソッド.....	3-55
	writeToSource() メソッド	3-57
3.2.7	コメント属性と関連付けられた ORDAudio メソッド	3-59
	appendToComments() メソッド.....	3-60
	writeToComments() メソッド	3-62
	readFromComments() メソッド	3-63
	locateInComments() メソッド.....	3-64
	trimComments() メソッド	3-65
	eraseFromComments() メソッド	3-66
	deleteComments メソッド.....	3-67
	loadCommentsFromFile() メソッド.....	3-68
	copyCommentsOut() メソッド	3-70
	compareComments() メソッド	3-72
	getCommentLength() メソッド	3-74
3.2.8	オーディオ属性アクセサと関連付けられた ORDAudio メソッド	3-75
	setFormat() メソッド	3-76

	<code>getFormat</code> メソッド	3-78
	<code>getFormat()</code> メソッド.....	3-79
	<code>setEncoding()</code> メソッド	3-81
	<code>getEncoding</code> メソッド	3-82
	<code>getEncoding()</code> メソッド.....	3-83
	<code>setNumberOfChannels()</code> メソッド	3-85
	<code>getNumberOfChannels</code> メソッド	3-86
	<code>getNumberOfChannels()</code> メソッド.....	3-87
	<code>setSamplingRate()</code> メソッド	3-89
	<code>getSamplingRate</code> メソッド	3-90
	<code>getSamplingRate()</code> メソッド.....	3-91
	<code>setSampleSize()</code> メソッド.....	3-93
	<code>getSampleSize</code> メソッド	3-94
	<code>getSampleSize()</code> メソッド	3-95
	<code>setCompressionType()</code> メソッド.....	3-97
	<code>getCompressionType</code> メソッド.....	3-98
	<code>getCompressionType()</code> メソッド	3-99
	<code>setAudioDuration()</code> メソッド.....	3-101
	<code>getAudioDuration</code> メソッド.....	3-102
	<code>getAudioDuration()</code> メソッド	3-103
	<code>setKnownAttributes()</code> メソッド	3-105
	<code>setProperties()</code> メソッド	3-107
	<code>checkProperties()</code> メソッド.....	3-109
	<code>getAttribute()</code> メソッド	3-111
	<code>getAllAttributes()</code> メソッド.....	3-113
3.2.9	オーディオ・データ処理と関連付けられた ORDAudio メソッド	3-115
	<code>processAudioCommand()</code> メソッド.....	3-116
3.3	パッケージまたは PL/SQL プラグイン.....	3-118
3.3.1	ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージ.....	3-118
3.3.2	ORDPLUGINS.ORDX_AUFF_AUDIO パッケージ.....	3-120
3.3.3	ORDPLUGINS.ORDX_AIFF_AUDIO パッケージ.....	3-123
3.3.4	ORDPLUGINS.ORDX_AIFC_AUDIO パッケージ.....	3-126
3.3.5	ORDPLUGINS.ORDX_WAVE_AUDIO パッケージ.....	3-128
3.3.6	<i>interMedia</i> の拡張による新しいオーディオ・データ・フォーマットのサポート	3-131

4 ORDIImage 参照情報

4.1	オブジェクト型.....	4-2
	ORDImage オブジェクト型.....	4-3
4.2	メソッド.....	4-6
4.2.1	表定義の例.....	4-8
4.2.2	Copy 操作に関連する ORDIImage メソッド	4-10
	copy() メソッド	4-11
4.2.3	Process 操作に関連する ORDIImage メソッド.....	4-13
	process() メソッド.....	4-14
	processCopy() メソッド	4-17
4.2.4	プロパティの設定および check 操作に関連する ORDIImage メソッド.....	4-19
	setProperties メソッド.....	4-20
	外部イメージの setProperties() メソッド	4-22
	checkProperties メソッド.....	4-25
4.2.5	イメージ属性に関連する ORDIImage メソッド.....	4-26
	getHeight メソッド	4-27
	getWidth メソッド.....	4-28
	getContentLength メソッド.....	4-29
	getFileFormat メソッド.....	4-30
	getContentFormat メソッド	4-31
	getCompressionFormat メソッド	4-32
4.2.6	ローカル属性に関連する ORDIImage メソッド.....	4-33
	setLocal メソッド	4-34
	clearLocal メソッド.....	4-35
	isLocal メソッド	4-36
4.2.7	日付属性に関連する ORDIImage メソッド.....	4-37
	getUpdateTime メソッド	4-38
	setUpdateTime() メソッド.....	4-39
4.2.8	mimeType 属性に関連する ORDIImage メソッド.....	4-40
	getMimeType メソッド.....	4-41
	setMimeType() メソッド.....	4-42
4.2.9	ソース属性に関連する ORDIImage メソッド.....	4-44
	getContent メソッド.....	4-45

	getBFILE メソッド	4-46
	deleteContent メソッド	4-47
	setSource() メソッド	4-48
	getSource メソッド	4-50
	getSourceType メソッド	4-51
	getSourceLocation メソッド	4-52
	getSourceName メソッド	4-53
	import() メソッド	4-54
	importFrom() メソッド	4-56
	export() メソッド	4-58
4.2.10	イメージ移行に関連する ORDImage メソッド	4-60
	migrateFromORDImgB() メソッド	4-61
	migrateFromORDImgF() メソッド	4-63

5 ORDVideo 参照情報

5.1	オブジェクト型	5-2
	ORDVideo オブジェクト型	5-3
5.2	メソッド	5-9
5.2.1	表定義の例	5-13
5.2.2	updateTime 属性と関連付けられた ORDVideo メソッド	5-14
	getUpdateTime メソッド	5-15
	setUpdateTime() メソッド	5-16
5.2.3	説明属性に関連付けられた ORDVideo メソッド	5-17
	setDescription() メソッド	5-18
	getDescription メソッド	5-20
5.2.4	mimeType 属性と関連付けられた ORDVideo メソッド	5-21
	setMimeType() メソッド	5-22
	getMimeType メソッド	5-24
5.2.5	ソース属性と関連付けられた ORDVideo メソッド	5-25
	processSourceCommand() メソッド	5-26
	isLocal メソッド	5-28
	setLocal メソッド	5-29
	clearLocal メソッド	5-30

	setSource() メソッド	5-31
	getSource メソッド	5-33
	getSourceType メソッド	5-34
	getSourceLocation メソッド	5-36
	getSourceName メソッド	5-37
	import() メソッド	5-38
	importFrom() メソッド	5-40
	export() メソッド	5-43
	getContentLength() メソッド	5-45
	getContentInLob() メソッド	5-46
	getContent メソッド	5-48
	deleteContent メソッド	5-50
5.2.6	ファイル関係の操作に関連付けられた ORDVide メソッド	5-51
	openSource() メソッド	5-52
	closeSource() メソッド	5-54
	trimSource() メソッド	5-56
	readFromSource() メソッド	5-58
	writeToSource() メソッド	5-60
5.2.7	コメント属性に関連する ORDVide メソッド	5-62
	appendToComments() メソッド	5-63
	writeToComments() メソッド	5-65
	readFromComments() メソッド	5-66
	locateInComments() メソッド	5-67
	trimComments() メソッド	5-68
	eraseFromComments() メソッド	5-69
	deleteComments() メソッド	5-70
	loadCommentsFromFile() メソッド	5-71
	copyCommentsOut() メソッド	5-73
	compareComments() メソッド	5-75
	getCommentLength() メソッド	5-77
5.2.8	ビデオ属性アクセサと関連付けられた ORDVide メソッド	5-78
	setFormat() メソッド	5-79
	getFormat メソッド	5-80

	getFormat() メソッド.....	5-81
	setFrameSize() メソッド.....	5-83
	getFrameSize() メソッド.....	5-85
	getFrameSize() メソッド.....	5-87
	setFrameResolution() メソッド.....	5-89
	getFrameResolution メソッド.....	5-90
	getFrameResolution() メソッド.....	5-91
	setFrameRate() メソッド.....	5-93
	getFrameRate メソッド.....	5-94
	getFrameRate() メソッド.....	5-95
	setVideoDuration() メソッド.....	5-97
	getVideoDuration メソッド.....	5-98
	getVideoDuration() メソッド.....	5-99
	setNumberOfFrames() メソッド.....	5-101
	getNumberOfFrames メソッド.....	5-102
	getNumberOfFrames() メソッド.....	5-103
	setCompressionType() メソッド.....	5-105
	getCompressionType メソッド.....	5-106
	getCompressionType() メソッド.....	5-107
	setNumberOfColors() メソッド.....	5-109
	getNumberOfColors メソッド.....	5-110
	getNumberOfColors() メソッド.....	5-111
	setBitRate() メソッド.....	5-113
	getBitRate メソッド.....	5-114
	getBitRate() メソッド.....	5-115
	setKnownAttributes() メソッド.....	5-117
	setProperties() メソッド.....	5-119
	checkProperties() メソッド.....	5-121
	getAttribute() メソッド.....	5-123
	getAllAttributes() メソッド.....	5-125
5.2.9	ビデオ・データの処理に関連付けられた ORDVideo メソッド.....	5-127
	processVideoCommand() メソッド.....	5-128
5.3	パッケージまたは PL/SQL プラグイン.....	5-130

5.3.1	ビデオ・フォーマット・パッケージ.....	5-130
5.3.2	ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージ.....	5-130
5.3.3	ORDPLUGINS.ORDX_AVI_VIDEO パッケージ.....	5-132
5.3.4	ORDPLUGINS.ORDX_MOOV_VIDEO パッケージ.....	5-135
5.3.5	ORDPLUGINS.ORDX_MPEG_VIDEO パッケージ.....	5-137
5.3.6	<i>interMedia</i> を拡張して新しいビデオ・データ・フォーマットをサポートする.....	5-139

6 ORDSource 参照情報

6.1	オブジェクト型.....	6-2
	ORDSource オブジェクト型.....	6-3
6.2	メソッド.....	6-7
6.2.1	例題テーブル定義.....	6-8
6.2.2	ローカル属性と関連付けられた ORDSource メソッド.....	6-9
	setLocal メソッド	6-10
	clearLocal メソッド.....	6-11
	isLocal メソッド	6-12
6.2.3	updateTime 属性と関連付けられた ORDSource メソッド	6-13
	getUpdateTime メソッド	6-14
	setUpdateTime() メソッド.....	6-15
6.2.4	srcType、srcLocation、およびsrcName 属性と関連付けられた ORDSource メソッド ..	6-16
	setSourceInformation() メソッド.....	6-17
	getSourceInformation メソッド	6-19
	getSourceType メソッド	6-20
	getSourceLocation メソッド.....	6-21
	getSourceName メソッド.....	6-22
	getBFile メソッド	6-23
6.2.5	インポートおよびエクスポート操作と関連付けられた ORDSource メソッド.....	6-24
	import() メソッド.....	6-25
	importFrom() メソッド	6-27
	export() メソッド.....	6-29
6.2.6	localData 属性と関連付けられた ORDSource メソッド	6-31
	getContentLength() メソッド.....	6-32
	getSourceAddress() メソッド.....	6-34
	getLocalContent メソッド.....	6-36

	getContentInTempLob() メソッド.....	6-37
	deleteLocalContent メソッド	6-39
6.2.7	ファイル操作と関連付けられた ORDSOURCE メソッド.....	6-40
	open() メソッド	6-41
	close() メソッド	6-43
	trim() メソッド	6-45
6.2.8	読み込み / 書き込み操作と関連付けられた ORDSOURCE メソッド	6-47
	read() メソッド	6-48
	write() メソッド	6-50
6.2.9	外部ソースへの処理コマンドと関連付けられた ORDSOURCE メソッド	6-52
	processCommand() メソッド	6-53
6.3	パッケージまたは PL/SQL プラグイン.....	6-55
6.3.1	ORDPLUGINS.ORDX_FILE_SOURCE パッケージ.....	6-55
6.3.2	ORDPLUGINS.ORDX_HTTP_SOURCE パッケージ	6-57
6.3.3	ORDPLUGINS.ORDX_<srcType>_SOURCE パッケージ	6-59
6.3.4	interMedia の拡張による新しいデータ・ソースのサポート.....	6-59

7 DBA のためのチューニング・ヒント

7.1	マルチメディア LOB データの INSERT パフォーマンスの向上	7-2
7.2	他の大量ロード方法.....	7-7
7.3	最大パフォーマンス実現のためのユーザー・ガイドライン.....	7-9
7.4	マルチメディア LOB データの検索および更新パフォーマンスの向上.....	7-10
7.5	オープン BFILE の最大数の設定.....	7-10
7.6	表パーティション内での LOB の使用.....	7-10

A オーディオ・ファイル形式および圧縮形式

A.1	サポートしているオーディオ・ファイル形式および圧縮形式.....	A-1
-----	----------------------------------	-----

B イメージ・ファイル形式および圧縮形式

B.1	サポートしているイメージ・ファイル形式および圧縮形式.....	B-1
-----	---------------------------------	-----

C Process() および ProcessCopy() のイメージ演算子

C.1	共通概念.....	C-1
C.1.1	ソース・イメージおよび宛先イメージ.....	C-1

C.1.2	process() および processCopy()	C-1
C.1.3	演算子および値	C-2
C.1.4	演算子の組合せ	C-2
C.2	イメージ・フォーマット演算子	C-2
C.2.1	FileFormat	C-3
C.2.2	ContentFormat	C-3
C.2.3	CompressionFormat	C-4
C.2.4	CompressionQuality	C-5
C.3	イメージ処理演算子	C-5
C.3.1	Cut	C-5
C.3.2	Scale	C-5
C.3.3	XScale	C-6
C.3.4	YScale	C-6
C.3.5	FixedScale	C-6
C.3.6	MaxScale	C-7
C.4	形式固有の演算子	C-7
C.4.1	ChannelOrder	C-7
C.4.2	Interleave	C-8
C.4.3	PixelOrder	C-8
C.4.4	ScanlineOrder	C-8
C.4.5	InputChannels	C-8

D ロー・ピクセル・イメージ・フォーマット

D.1	ロー・ピクセルの概要	D-1
D.2	ロー・ピクセル・イメージの構造	D-2
D.3	ロー・ピクセル・ヘッダー・フィールドの説明	D-3
D.4	ロー・ピクセル・ポストヘッダー・ギャップ	D-7
D.5	ロー・ピクセル・データ・セクションおよびピクセル・データ・フォーマット	D-8
D.5.1	走査線の順序付け	D-8
D.5.2	ピクセルの順序付け	D-8
D.5.3	バンド・インターリーブ	D-9
D.5.4	N バンド・データ	D-10
D.6	ロー・ピクセル・ヘッダーの C 構造体	D-11
D.7	ロー・ピクセル・ヘッダーの C 定数	D-12
D.8	ロー・ピクセル PL/SQL 定数	D-13

D.9	CCITT 圧縮を使用したロー・ピクセル・イメージ.....	D-13
D.10	外部イメージ・サポートおよびそのロー・ピクセル・フォーマット.....	D-14

E サンプル・プログラム

E.1	サンプル・オーディオ・スクリプト.....	E-1
E.2	イメージの変更またはインストール・テストのサンプル・プログラム.....	E-2
E.2.1	デモンストレーション（デモ）のインストール・ステップ.....	E-2
E.2.2	デモの実行.....	E-3
E.3	サンプル・ビデオ・スクリプト.....	E-4
E.4	Java デモ	E-5

F 参照情報

F.1	オブジェクト型.....	F-1
	ORDAnnotations オブジェクト型.....	F-2
F.2	メソッド.....	F-3
F.2.1	注釈属性に関連付けられた ORDAnnotations メソッド.....	F-3
	setAnnotation() メソッド	F-4
	getAnnotation() メソッド	F-5
	deleteAnnotation() メソッド	F-7
	appendToAnnotation() メソッド.....	F-8
	getAnnotationCount メソッド	F-9
	getAnnotationObjByPosition() メソッド.....	F-10
	getAnnotationKeyByPosition() メソッド	F-12
	getAnnotationValueByPosition() メソッド.....	F-14

G よく寄せられる質問

H 例外およびエラー・メッセージ

H.1	例外.....	H-1
H.1.1	ORDAudioExceptions 例外	H-1
H.1.2	ORDImageExceptions 例外.....	H-2
H.1.3	ORDVideoExceptions 例外.....	H-3
H.1.4	ORDSourceExceptions 例外.....	H-4
H.2	ORDAudio エラー・メッセージ	H-5

H.3	ORDImage エラー・メッセージ	H-7
H.4	ORDVideo エラー・メッセージ	H-12

I 使用されなくなったイメージ・オブジェクト型およびメソッド

ORDImgB オブジェクト型	I-4
ORDImgF オブジェクト型	I-6
checkProperties メソッド	I-8
copyContent メソッド	I-9
deleteContent メソッド	I-10
getCompressionFormat メソッド	I-11
getContent メソッド	I-12
getContentFormat メソッド	I-13
getContentLength メソッド	I-14
getFileFormat メソッド	I-15
getHeight メソッド	I-16
getMimeType メソッド	I-17
getWidth メソッド	I-18
process メソッド	I-19
processCopy メソッド	I-22
setProperties メソッド	I-23
外部イメージの setProperties() メソッド	I-25

用語集

索引

例一覧

2-1	ソング・オブジェクトの定義.....	2-2
2-2	SongsTable テーブルの作成.....	2-2
2-3	ソングへの参照リストを含むリスト・オブジェクトの作成.....	2-3
2-4	songList オブジェクトのインプリメンテーションの定義.....	2-3
2-5	CD 情報を格納する CD テーブルの作成.....	2-4
2-6	SongsTable テーブルへのソングの挿入.....	2-4
2-7	CdTable テーブルへの CD の挿入.....	2-5
2-8	SongsTable テーブルへのソングのロード.....	2-6
2-9	ソング・オブジェクトへの参照の、CdTable テーブル内のソング・リストへの挿入.....	2-7
2-10	ソングへの CD 参照の追加.....	2-8
2-11	ソングに関するコメントの追加.....	2-9
2-12	CD 内のソングからのオーディオ・データの取得.....	2-9
2-13	CD 内のソングに関するすべてのコメントの表示.....	2-9
2-14	ORDImage 型の新規列の emp テーブルへの追加.....	2-13
2-15	ORDImage 型の新規テーブルへの追加.....	2-13
2-16	ORDImage 型列のデータが空白である行のテーブルへの挿入.....	2-14
2-17	ORDImage BLOB データを使用した行の移入.....	2-14
2-18	ORDImage 型列にイメージを含む行の、テーブルへの挿入.....	2-15
2-19	ORDImage 外部ファイル・データを使用した行の移入.....	2-16
2-20	任意のコンテンツ長の ORDImage データを保持する行の問合せ.....	2-17
2-21	特定のコンテンツ長の ORDImage データを保持する行の問合せ.....	2-17
2-22	外部ファイルからのイメージのインポート.....	2-17
2-23	イメージのコピー.....	2-18
2-24	イメージ形式の変換.....	2-18
2-25	イメージ形式のコピーと変換.....	2-19
2-26	新しいオブジェクト型を使用した Oracle8i interMedia イメージの拡張.....	2-19
2-27	フラット・テーブル上でのオブジェクト・ビューの作成.....	2-22
2-28	各国語サポート問題への対処.....	2-22
2-29	クリップ・オブジェクトの定義.....	2-24
2-30	clipsTable テーブルの作成.....	2-24
2-31	クリップ・リストを含むリスト・オブジェクトの作成.....	2-24
2-32	clipList オブジェクトのインプリメンテーションの定義.....	2-25
2-33	ビデオ情報を格納するビデオ・テーブルの作成.....	2-25
2-34	clipsTable テーブルへのビデオ・クリップの挿入.....	2-26
2-35	videoTable テーブルへの行の挿入.....	2-26
2-36	clipsTable テーブルへのビデオのロード.....	2-27
2-37	クリップ・オブジェクトへの参照の、videoTable テーブル内のクリップ・リストへの挿入.....	2-28
2-38	ビデオ・オブジェクトへの参照の、クリップへの挿入.....	2-29
2-39	ビデオ・クリップの取得.....	2-30
3-1	新規オーディオ・データ・フォーマットへのサポートを拡張する、パッケージ本体リスト.....	3-131
5-1	新しいビデオ・データ・フォーマットを拡張サポートするためのパッケージ本体のリスト.....	5-140
6-1	新規データ・ソースへのサポートを拡張する、パッケージ本体リスト.....	6-59
7-1	LOB データを格納する個別の表領域の作成.....	7-6

E-1	コマンド行からデモを実行する.....	E-3
-----	---------------------	-----

表一覧

3-1	ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージでサポートされるメソッド	3-119
3-2	ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド	3-122
3-3	ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド	3-124
3-4	ORDPLUGINS.ORDX_AIFC_AUDIO パッケージでサポートされるメソッド	3-127
3-5	ORDPLUGINS.ORDX_WAVE_AUDIO パッケージでサポートされるメソッド	3-130
4-1	イメージ処理演算子	4-14
4-2	ロー・ピクセルおよび外部イメージの追加イメージ処理演算子	4-15
4-3	外部（ヘッダーなし）ファイルのイメージ特性	4-23
5-1	ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージでサポートされているメソッド	5-131
5-2	ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド	5-134
5-3	ORDPLUGINS.ORDX_MOOV_VIDEO パッケージでサポートされているメソッド	5-136
5-4	ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド	5-138
6-1	ORDPLUGINS.ORDX_FILE_SOURCE パッケージがサポートするメソッド	6-56
6-2	ORDPLUGINS.ORDX_HTTP_SOURCE パッケージがサポートするメソッド	6-58
A-1	AIFF データ形式.....	A-1
A-2	AIFF-C データ形式.....	A-2
A-3	AU データ形式.....	A-2
A-4	WAV データ形式.....	A-3
B-1	BMP データ形式	B-1
B-2	CALS ラスター・データ形式.....	B-2
B-3	GIF データ形式	B-2
B-4	JFIF データ形式	B-3
B-5	PCX データ形式	B-3
B-6	PICT データ形式	B-4
B-7	ロー・ピクセル・データ形式	B-4
B-8	Sun ラスター・データ形式	B-5
B-9	Targa データ形式	B-5
B-10	TIFF データ形式	B-6
I-1	ファンクションおよびプロシージャ	I-1
I-2	イメージ処理オペレータ	I-19
I-3	ロー・ピクセル・イメージおよび外部イメージの追加イメージ処理オペレータ.....	I-20
I-4	ヘッダーなしファイルのイメージ特性	I-25

はじめに

本ガイドでは、Oracle8i *interMedia* Audio, Image, Video の使用方法について説明します。
Oracle8i *interMedia* Audio, Image, Video を使用するには、Oracle8i または Oracle8i Enterprise Edition が必要です。

対象読者

本ガイドの対象読者は、アプリケーション開発者、および Oracle データベースでのオーディオ、イメージ、ビデオ・データの格納、取得、操作に関心を持つデータベース開発者（オーディオ、イメージ、ビデオの特殊オプション開発者を含む）です。

構成

本ガイドに含まれる章および付録は、次のとおりです。

第 1 章	マルチメディアおよび Oracle8i <i>interMedia</i> の概要を説明します。マルチメディア関連の概念について説明します。
第 2 章	Oracle8i <i>interMedia</i> オブジェクト型およびメソッドの基本的な使用例を示します。
第 3 章	Oracle8i <i>interMedia</i> ORDAudio オブジェクト型およびメソッドの参照情報を提供します。
第 4 章	Oracle8i <i>interMedia</i> ORDImage オブジェクト型およびメソッドの参照情報を提供します。
第 5 章	Oracle8i <i>interMedia</i> ORDVideo オブジェクト型およびメソッドの参照情報を提供します。
第 6 章	Oracle8i <i>interMedia</i> ORDSrc ource オブジェクト型およびメソッドの参照情報を提供します。
第 7 章	DBA をチューニングして、マルチメディア・データの格納効率を高めるためのヒントを提供します。
付録 A	サポートするオーディオ・データ・フォーマットについて説明します。

付録 B	サポートするイメージ・データ・フォーマットについて説明します。
付録 C	<code>process</code> および <code>processCopy</code> 演算子について説明します。
付録 D	ロー・ピクセル・フォーマットについて説明します。
付録 E	サンプル・アプリケーションの実行方法について説明します。サンプル・アプリケーションのソース・リストも含みます。
付録 F	デモ・プログラムに含まれるオブジェクト型およびメソッドに関する参照情報を提供します。
付録 G	オンライン FAQ に含まれるいくつかの重要項目について説明します。
付録 H	例外および発生する可能性のあるエラーについて、その原因と対処方法を示します。
付録 I	使用が推奨されていないイメージ・オブジェクト型およびメソッドについて説明します。
用語集	データ・オプションおよびマルチメディア情報に関連した重要な用語を説明します。

関連ドキュメント

注意: 本ガイドのリリース後の追加情報については、`ORACLE_HOME` ディレクトリに格納されているオンラインの `README.txt` ファイルを参照してください。オペレーティング・システムによっては、このファイルは次のディレクトリに格納されている場合もあります。

`ORACLE_HOME/ord/img/admin/README.txt`

詳細は、オペレーティング・システム固有のインストール・ガイドを参照してください。

開発環境での *interMedia* の使用方法の詳細は、リリース **8.1.5 Oracle** データベース・サーバー・ドキュメント・セット内の次のドキュメントを参照してください。

- Oracle8i コール・インタフェース・プログラマーズ・ガイド
- Oracle8i アプリケーション開発者ガイド 基礎編
- Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト
- Oracle8i 概要
- PL/SQL ユーザーズ・ガイドおよびリファレンス
- Oracle8i *interMedia* Audio, Image, Video Java Client ユーザーズ・ガイドおよびリファレンス

規則

本ガイドでは、*interMedia* は、**Oracle8i interMedia** のことを指します。

コード例では、特に指定がない限り、各行の最後にキャリッジ・リターンが含まれます。各入力行の最後で、**[Enter]** キーを押してください。

本ガイドで使用する他の規則は、次のとおりです。

規則	説明
.	コード例で使用する縦の省略記号は、その例に直接関係しない情報が省略されていることを示します。
...	文やコマンド内で使用される横の省略記号は、その例に直接関係しない文やコマンドが省略されていることを示します。
太字	太字は、その語がテキスト、用語集、またはその両方で定義された用語であることを示します。
< >	山カッコは、ユーザーが指定する名前を示します。
[]	大カッコは、その中の 1 つを選択しても、何も選択しなくてもかまわないオプション句を示します。

Oracle8i *interMedia* Audio, Image, Video は、Oracle8i *interMedia* 製品のコンポーネントです。Oracle8i *interMedia* Audio, Image, Video を使用することにより、Oracle8i でオブジェクト型を利用してイメージ、オーディオ、およびビデオを他の企業情報と共に管理できます。Oracle8i *interMedia* は、Oracle8i の信頼性、可用性、およびデータ管理機能を拡張して、インターネット、電子商取引、および多様なメディアを利用するアプリケーションでマルチメディア・コンテンツの利用を可能にします。

interMedia Audio, Image, Video 機能は、Oracle8i が管理するマルチメディア・データの格納、取得、管理、および操作機能を含みます。*interMedia* Audio, Image, Video は、次のマルチメディア・データの格納、取得、および管理機能をサポートします。

- Oracle8i でローカルに格納され、オーディオ、イメージ、またはビデオ・データを含むバイナリ・ラージ・オブジェクト (BLOB)
- オペレーティング・システム固有のファイル・システム内でローカルに格納され、オーディオ、イメージ、またはビデオ・データを含むファイルベースのラージ・オブジェクト (BFILE)
- Oracle Application Server、Netscape Application Server、Microsoft Internet Information Server、Apache HTTPD サーバー、および Spyglass サーバーなどの HTTP サーバーに格納されたオーディオ、イメージ、またはビデオ・データへのリンクを示す URL
- Oracle Video Server、Real Audio Server、および Real Video Server などの専用メディア・サーバーに格納されたストリーミング・オーディオまたはビデオ・データ
- BLOB および他のサーバー上の任意のユーザー定義ソースに格納された特殊フォーマットのマルチメディア・データすべて

interMedia は、拡張可能な設計がなされています。これは、一般的なオーディオ、イメージ、およびビデオ・データ基本セットのマルチメディア処理をサポートします。また、マルチメディア処理を拡張して、他のフォーマット、新たなデジタル圧縮および展開スキーム (codecs)、データ・ソース、およびオーディオとビデオ・データの特種なデータ処理アルゴリズムをサポートすることも可能です。

interMedia は、エンドユーザー・アプリケーションとしてではなく、さまざまなマルチメディア・アプリケーションの構成部品として機能します。その中には、オブジェクト型、およびマルチメディア・データ管理および処理関連のメソッドが含まれます。*interMedia* Audio, Image, Video を活用したアプリケーションの例を次に示します。

- CD 並みの音質で音楽を提供するインターネット・ミュージック・ストア
- デジタル・サウンドのリポジトリ
- 口述筆記および電話による会話のリポジトリ
- オーディオの保管および収集（音楽家向け）
- デジタル・アート・ギャラリー
- 不動産のマーケティング
- 文書の電子化
- 写真のコレクション（プロの写真家向け）
- インターネット・ビデオ・ストアおよびデジタル・ビデオクリップ・プレビュー
- ストーリーミング・ビデオ配信システム用のデジタル・ビデオ・ソース
- デジタル・ビデオ・ライブラリ、アーカイブ、およびリポジトリ
- デジタル・ビデオ・トレーニング・プログラムのライブラリ
- デジタル・ビデオ・リポジトリ（動画制作、テレビ放送、ドキュメンタリ、広告向け）

これらのアプリケーションすべてに共通した要件もあれば、アプリケーション固有の要件もあります。**Oracle8i *interMedia*** オーディオ、イメージ、およびビデオ・オブジェクト型は、アプリケーションすべてに共通した要件をサポートします。また、拡張することにより、各アプリケーション固有の要件への対応が可能です。**Oracle8i *interMedia*** を使用することにより、マルチメディア・データを一般的な属性データと同じくらい簡単に処理できます。

Oracle8i *interMedia* からアプリケーションへのアクセスは、リレーショナルおよびオブジェクト・インタフェースのどちらからでも可能です。**JAVA**、**C++**、また従来の **3GL** で記述されたデータベース・アプリケーションは、モダン・クラス・ライブラリ・インタフェース、または **PL/SQL** および **Oracle** コール・インタフェース (**OCI**) を使用して *interMedia* へアクセスできます。

interMedia は、多くの一般的なファイル形式での格納をサポートしています。その中には、**Oracle8i** データベースで利用可能な **DTP** イメージおよびオーディオやビデオ・フォーマットも含まれます。*interMedia* は、オーディオ、イメージ、およびビデオ列やオブジェクトを既存の表に追加したり、マルチメディア・データを挿入および取得する手段を提供します。また、アプリケーション形式間の制限付き変換機能も提供します。これにより、データベース設計者は、既存のアプリケーション・データベースをマルチメディア・データを使用して拡張したり、新しいエンドユーザー向けマルチメディア・データベース・アプリケーションを構築することができます。*interMedia* 開発者は、ここで提供される基本機能を使用して、特別なマルチメディア・アプリケーションを構築できます。

1.1 Oracle8i interMedia

Oracle8i interMedia は、JAVA や C++ のクラスに類似したオブジェクト型を使用してオーディオ、イメージ、およびビデオ・データを表現します。これらのオブジェクト型は、ORDAudio、ORDImage、および ORDVideo と呼ばれます。これらのオブジェクト型のインスタンスは属性で構成されており、メタデータ、メディア・データ、およびメソッドが含まれます。メディア・データは、実際のオーディオ、イメージ、またはビデオ・データです。メタデータは、オブジェクトの長さ、圧縮、または形式などのデータに関する情報です。メソッドは、getContent() や process() などの、オブジェクトに対して実行可能なプロシージャです。

interMedia オブジェクト (ORDAudio、ORDImage、および ORDVideo) は、共通メディア・データ格納モデルを保持します。これらのオブジェクトのメディア・データ・コンポーネントは、データベース内、トランザクション制御下のバイナリ・ラージ・オブジェクト (BLOB) に格納可能です。メディア・データは、トランザクション制御なしでデータベースの外部に格納することも可能です。この場合、ポインタはデータベース内、トランザクション制御下に格納され、メディア・データは次の場所に格納されます。

- 外部バイナリ・ファイル (BFILE)
- HTTP サーバーベースの URL
- Oracle Video Server、Real Networks ストリーミング・サーバー、また Live Pictures Image Server などの専用メディア・データ・サーバー上のソース
- ユーザーが定義した他のサーバー上のソース

データベース外に格納されたメディア・データが提供する管理メカニズムを利用すると、消去可能または読み専用メディア上にフラット・ファイルとして配置される、既存または新規の大規模メディア・リポジトリを効率良く管理できます。このデータは、トランザクション制御用としていつでも BLOB にインポートできます。

オブジェクトのメタデータおよびメソッドは、Oracle8i interMedia による制御の下に、常にデータベース内に格納されます。メディア・データがデータベースの内部または外部のどちらに格納されている場合でも、interMedia はすべてのメディア・タイプのメタデータを管理し、オーディオ、イメージ、およびビデオ用のメタデータを自動的に抽出することができます。このメタデータには、次の属性が含まれます。

- データベース内のローカル (オーディオ、イメージ、およびビデオ) データ
- ソース・タイプ、位置、ソース名、およびデータがローカル (データベース内) または外部のどちらに格納されているかなどの情報を含む、オーディオ、イメージ、およびビデオ・データの格納情報
- オーディオ、イメージ、およびビデオ・データ更新時のタイム・スタンプ
- オーディオおよびビデオ・データに関する説明
- オーディオ、イメージ、およびビデオ・データの形式
- オーディオ、イメージ、およびビデオ・データの MIME タイプ

- オーディオおよびビデオに関するコメント
- オーディオ特性: エンコーディング・タイプ、チャンネル数、サンプリング・レート、サンプル・サイズ、圧縮形式、および再生時間
- イメージ特性: 幅と高さ、イメージのコンテンツ長、イメージのコンテンツ形式、およびイメージの圧縮形式
- ビデオ特性: フレームの幅と高さ、フレームの解像度、フレーム・レート、再生時間、フレーム数、圧縮形式、色数、およびビット・レート

さらに、イメージおよびデータ操作メソッドの最小セットも提供されます。イメージの場合、イメージに合致するイメージ・プロパティの検証、イメージ・フォーマットの変換と圧縮、イメージのスケール変更、クロップ、コピー、および削除がこれに含まれます。

次の方法で、**Oracle8i interMedia** を拡張できます。

- 提供されたマルチメディア・オブジェクト型に基づき、新規オブジェクト型または新規複合オブジェクト型を作成する。詳細は、2.1.15 項および 2.3.13 項の例を参照してください。
- 現在サポートされていないオーディオ、イメージ、およびビデオ・データの他の外部ソースをサポートする特別のプラグインを作成する。詳細は、1.6.1 項を参照してください。
- 現在サポートされていない他のオーディオおよびビデオ・データ・フォーマットをサポートする特別のオーディオおよびビデオ・データ・フォーマット・プラグインを作成する。詳細は、1.6.1 項を参照してください。
- 外部イメージに対応した **setProperties()** メソッドを使用し、他の特定イメージ・フォーマットの認識を可能にする。詳細は、1.6.1 項および第 4 章の「外部イメージの **setProperties()** メソッド」を参照してください。
- オーディオおよびビデオ・データの処理用メソッドを使用して、特定のオーディオまたはビデオ・コマンドやその引数を渡して、オーディオまたはビデオ・データ処理を可能にする。詳細は、1.6.2 項および 1.6.3 項を参照してください。

1.2 オーディオの概念

この項では、デジタル・オーディオの概念、および **interMedia audio** を使用したオーディオ・アプリケーションや特別な **interMedia audio** オブジェクトの作成方法を説明します。

1.2.1 デジタル・オーディオ

オーディオ制作は、オーディオ・レコーダ、マイクロフォンなどのオーディオ・ソース、デジタル・オーディオ、他の特殊オーディオ録音用デバイス、またはプログラム・アルゴリズムを使用して行うことができます。オーディオ録音デバイスは、マイクロフォンや磁気メディアに録音されたサウンドなどのアナログ（継続した）信号を受け取り、特定のオーディオ

オ特性（フォーマット、エンコーディング・タイプ、チャンネル数、サンプリング・レート、サンプル・サイズ、圧縮形式、および再生時間など）を持つデジタル値に変換します。

interMedia audio は、Oracle8i を使用して Oracle データベース内のデジタル・オーディオ・データの格納、取得、および管理機能を統合します。

interMedia audio は、特定のサンプリング・レート、エンコーディング・タイプ、サンプル・サイズ、およびチャンネル数を持つ特定のファイル形式のオーディオ・データを、ユーザーが定義した形式で利用可能なハードウェア機能や処理能力に基づき、再生または処理するアプリケーションをサポートします。*interMedia audio* を拡張することにより、任意の特殊オーディオ特性をサポートできます。

1.2.2 オーディオ・コンポーネント

デジタル・オーディオは、オーディオ・データ（デジタル・ビット）およびオーディオ・データの情報や特性を示す属性で構成されます。オーディオ・アプリケーションは、データベース内のオーディオ・クリップの説明、録音日、制作者およびアーティストなどのアプリケーション固有の情報を、属性（データベース表の列）内に格納することにより、オーディオ・データと関連付ける場合があります。

オーディオ・データは、そのデジタル録音の方法により、データ形式、エンコーディング・タイプ、圧縮形式、チャンネル数、サンプリング・レート、サンプル・サイズ、および再生時間が異なります。オーディオ・データの特性は、オーディオ・データへアクセスする上で重要です。音質はオーディオ・データの特性により決定されます。

圧縮、説明、およびソース情報（ソース・タイプ、ソースの位置、ソース名）のないオーディオ・クリップでは、オーディオ・クリップと共に送信される最小属性には、ファイル形式、MIME タイプ、エンコーディング・タイプ、チャンネル数、サンプリング・レート、サンプル・サイズ、および再生時間が含まれます。これらのデータ属性は、オーディオ・データがデジタル録音デバイスで録音または制作された時点での説明や特性です。

デジタル・オーディオのサイズ（バイト数）は、数値やテキストなどの従来のコンピュータで使用するオブジェクトに比較して大きくなる傾向があります。このため、いくつかのエンコード方法を使用してオーディオ・データを圧縮し、記憶デバイスやネットワークの負荷を減らすことができます。

1.3 イメージの概念

この項では、デジタル・イメージの概念、および *interMedia image* を使用したイメージ・アプリケーションや特殊 *interMedia image* オブジェクトの作成方法を説明します。

1.3.1 デジタル・イメージ

interMedia image は、Oracle8i を使用して Oracle データベース内のデジタル・イメージの格納、取得、および管理機能を統合します。

interMedia image は、実世界のオブジェクトや場面のバイナリ表現である、二次元の静的デジタル・イメージの格納をサポートします。イメージは、ドキュメントや写真のスキャナ、ビデオ・キャプチャ装置に接続されたカメラやビデオデッキなどのビデオ・ソース、他の特殊イメージ・キャプチャ・デバイス、またはプログラム・アルゴリズムを使用して制作できます。キャプチャ・デバイスは、カメラのフィルムに映写した光などのアナログ（連続）信号を受け取り、ピクセルとして知られるデータ・ポイントで構成される二次元グリッド上のデジタル値に変換します。イメージのキャプチャおよび表示を行うデバイスは、アプリケーションから制御します。

1.3.2 イメージ・コンポーネント

デジタル・イメージは、イメージ・データ（デジタル・ビット）およびイメージ・データの情報や特性を示す属性で構成されます。イメージ・アプリケーションは、被写体の名前、イメージの説明、撮影日、カメラマンなどのアプリケーション固有の情報をデータベース表内の属性（列）に格納することにより、イメージ・データに関連付けます。

イメージ・データ（ピクセル）は、イメージのキャプチャ方法により色の深度（各ピクセルのビット数）が異なります。また、イメージ・データはさまざまな方法で構成が可能です。イメージ・データの構成のことを、データ形式と言います。**interMedia image** では、現在のさまざまなデータ形式を使用してイメージ・データの読み込みおよび書き込みが可能です。サポートされるデータ形式の一覧は、付録 B を参照してください。また、外部イメージ（**interMedia image** がネイティブにサポートしない形式）によっては、限られた範囲でサポートされるものもあります。詳細は、付録 D を参照してください。

圧縮およびソース情報（ソース・タイプ、ソースの位置、ソース名）を保持しないイメージでは、イメージの最小属性にはそのサイズ（走査線の高さおよび幅のピクセル値）、およびファイル形式とコンテンツ形式などが含まれます。

デジタル・イメージに必要な記憶領域は、数値やテキストなどの従来の属性データに比較して大きくなる傾向があります。イメージを圧縮してバイト・サイズを減らすことのできるさまざまな圧縮方法を利用できます。このため、記憶デバイスおよびネットワークの負荷を減らすことができます。可逆圧縮を利用すると、展開後のイメージが劣化せず、元のイメージとビット単位で同一になります。非可逆圧縮を利用すると、展開後のイメージは元のイメージとは同一ではなくなります。ただし、変化は人間の目ではほとんど感知できません。

イメージの交換形式では、イメージの構成方法およびその属性、データ、また圧縮方法が巧みに記述される場合が多くあるため、異なるアプリケーションを使用してイメージの作成、交換、および使用ができます。多くの場合、交換形式は、ディスク・ファイル内またはディスク・ファイルとして格納されます。交換形式は、ネットワーク上をシーケンシャルに交換可能であるため、プロトコルとも言われます。デジタル・イメージの世界では、多くのアプリケーション・サブドメイン、およびこれらの内部でデジタル・イメージを作成または利用できる多くのアプリケーションが存在します。**interMedia image** は、多くの交換形式の読み込みおよび書き込みをサポートします。（付録 B を参照してください。）

1.4 ビデオの概念

この項では、デジタル・ビデオの概念、および *interMedia video* を使用したビデオ・アプリケーションや特別な *interMedia video* オブジェクトの作成方法を説明します。

1.4.1 デジタル・ビデオ

ビデオは、ビデオ・レコーダ、ビデオ・カメラ、デジタル・アニメーション・ビデオ、その他の特殊ビデオ録画用デバイス、またはプログラム・アルゴリズムを使用して制作します。ビデオ録画デバイスは、ビデオ・カメラから取得したビデオや磁気メディアに録画されたビデオなどのアナログ（連続した）信号を受け取り、ビデオ・フォーマット、エンコーディング・タイプ、フレーム・レート、フレーム・サイズ、フレームの解像度、ビデオの長さ、圧縮形式、色数、およびビット・レートなどの特定のビデオ特性を持つデジタル値に変換します。

interMedia video を利用すると、Oracle8i を使用した Oracle データベース内のデジタル・ビデオ・データの格納、取得、および管理の統合が容易になります。

interMedia video は、特定のファイル形式のビデオ・データの再生または処理のいずれかを実行するアプリケーションをサポートします。このファイル形式では、利用可能なハードウェア機能やユーザー定義形式の処理能力に基づく、特定のフレーム・レート、フレーム・サイズ、フレームの解像度、圧縮形式、ビデオの長さ、ビット・レート、および色数を保持します。*interMedia video* を拡張することにより、任意の特殊ビデオ特性をサポートできます。

1.4.2 ビデオ・コンポーネント

デジタル・ビデオは、ビデオ・データ（デジタル・ビット）およびビデオ・データの情報や特性を示す属性で構成されます。ビデオ・アプリケーションは、ビデオ・トレーニング・テープの説明、録画日、インストラクタ名、プロデューサ名などのアプリケーション固有情報を、属性（データベース表の列）内に格納することにより、ビデオ・データと関連付ける場合があります。

ビデオ・データは、ビデオ・データのデジタル録画方式により、データ形式、圧縮形式、フレーム・レート、フレーム・サイズ、フレームの解像度、再生時間、色数、およびビット・レートが異なる場合があります。ビデオ・データの特性は、ビデオ・データへアクセスする上で重要です。ビデオ・データの品質は、ビデオ・データの特性により決定されます。

圧縮、説明、およびソース情報（ソース・タイプ、ソースの位置、ソース名）のないビデオ・クリップでは、ビデオ・クリップと共に送信される最小属性には、ファイル形式、MIME タイプ、エンコーディング・タイプ、フレーム・レート、フレーム・サイズ、フレームの解像度、再生時間の合計、総フレーム数、色数、およびビット・レートが含まれます。これらのデータ属性は、ビデオ・データがデジタル録画デバイスを使用して録画または制作された時点での説明です。

デジタル・ビデオのサイズ（バイト数）は、数値やテキストなどの従来のコンピュータで使用するオブジェクトに比較して大きくなる傾向があります。このため、いくつかのエンコー

ド方法を使用してビデオ・データを圧縮し、記憶デバイスやネットワークの負荷を減らすことができます。

1.5 オブジェクト・リレーショナル・テクノロジー

Oracle8i は、オブジェクト・リレーショナル・データベース管理システムです。これは、リレーショナル・データを安全かつ効率的に管理するという従来の役割に加え、オブジェクト関連データおよびオブジェクトに対して実行可能な操作（メソッド）を含む、オブジェクト型の定義もサポートすることを意味します。オブジェクト指向の世界で十分確立されたこの強力なメカニズムには、デジタル・オーディオ、イメージ、ビデオなどの複雑なオブジェクトを Oracle8i データベースに追加するための基盤を提供する BLOB の完全なサポートが含まれます。

Oracle8i *interMedia* では、オーディオ・データ特性は **ORDAudio** として知られるオブジェクト・リレーショナル・タイプであり、イメージ・データ特性は **ORDImage** として知られるオブジェクト・リレーショナル・タイプであり、ビデオ・データ特性は **ORDVideo** として知られるオブジェクト・リレーショナル・タイプです。これら 3 つはすべて、**ORDSource** として知られるオブジェクト・リレーショナル・タイプ内にデータ・ソース情報を格納します。

BLOB と BFILE の使用に関する詳細は、次の情報を参照してください。

- Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト
 - Oracle8i 概要
- オブジェクト・ビューに関する章を参照してください。

1.5.1 マルチメディア・オブジェクト型およびメソッド

Oracle8i *interMedia* が提供する ORDAudio、ORDImage、および ORDVideo オブジェクト型およびメソッドを使用して、次の機能を実行できます。

- updateTime ORDSOURCE 属性の操作
- 説明属性の操作 (*interMedia* オーディオとビデオのみ)
- マルチメディア・データ・ソース属性情報の操作
- Oracle8i *interMedia*、Web サーバー、および他のサーバーからのマルチメディア・データの取得と管理
- ソースに対するファイル操作（オープン、クローズ、切捨て、読み込み、書き込み）の実行 (*interMedia* オーディオおよびビデオのみ)
- コメント属性の操作 (*interMedia* オーディオおよびビデオのみ)
- マルチメディア・データからの属性の抽出
- マルチメディア・データに対する最小操作の実行 (*interMedia* イメージのみ)

- マルチメディア・データ操作コマンド (`processAudioCommand` および `processVideoCommand`) の処理 (*interMedia* オーディオおよびビデオのみ)

1.5.2 ORDSource オブジェクト型およびメソッド

Oracle8i *interMedia* は、マルチメディア・データ・ソース操作の ORDSource オブジェクト型およびメソッドを提供します。この項では、ORDSource オブジェクト型メソッドの概要について説明します。

注意: ORDSource メソッドは直接コールしないでください。そのかわりに、ORDSource メソッドに対応するメディア・オブジェクトのラッパー・メソッドを起動してください。ここで提供する情報は、独自のユーザー定義ソースを記述するユーザーを対象としています。

1.5.2.1 マルチメディア・データの格納

interMedia では、マルチメディア・データを Oracle8i データベース内およびトランザクション制御下で、内部ソースおよび BLOB として格納できます。また、デジタル・マルチメディア・データを、ローカル・ファイル・システムのオペレーティング・システム固有の BFILE 内に格納された外部ソースとして、HTTP サーバー上の URL として、メディア・サーバーに格納されたストリーミング・オーディオまたはビデオとして、また他のサーバー上のユーザー定義ソースとして、外部参照することも可能です。これらの外部記憶メカニズムは、既存のマルチメディア・データ・セットを Oracle8i データベースと統合するための非常に便利な方法ではありますが、マルチメディア・データをトランザクション制御することはできません。

BLOB は、領域を最適化し、効率的なアクセスを可能にする方法でデータベースの表領域に格納されます。BLOB は、他のロー・データとインラインに格納されない場合もあります。BLOB のサイズに基づき、ロケータは行に格納され、実際の BLOB (4 ギガバイトまで) は他の表領域に格納されます。ロケータを、BLOB 値の実際の位置へのポインタとみなすこともできます。BLOB を選択すると、値のかわりにロケータが選択されます。これは透過的に行われます。この設計の利点は、1 つの行に複数の BLOB ロケータを配置可能なことです。たとえば、トレーニング・テープの短いビデオ・クリップ、内容の短い説明を含むオーディオ録音、コースの摘要、インストラクタの写真、各トレーニング・センターの地図と案内を格納します。

BFILE は、データベースのトランザクション制御下にあるため、ユーザーはデータベースを更新しなくても外部ソースを変更可能です。このため、BFILE ロケータの非一貫性という問題が生じます。BFILE、URL、メディア・サーバー上のストリーミング・オーディオまたはビデオ・ソース、また他のサーバー上のユーザー定義ソース用の各ロケータだけが行内に格納されます。BLOB および BFILE の使用法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』および『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。

interMedia は、ローカル属性操作、updateTime 属性操作、ソース属性操作、インポート / エクスポート操作、ソース・コンテンツ操作、ソース・アクセス操作、ソース読み込みおよび書き込み操作、およびコマンド処理操作に対応した ORDSOURCE オブジェクト型およびメソッドを提供します。

1.5.2.2 マルチメディア・データの問合せ

マルチメディア・データが Oracle8i データベース内に格納されると、表の多様な英数字列 (属性) を使用して目的のデータを含む行を検索することにより、マルチメディア・データの問合せおよび取得が可能になります。たとえば、Training 表からコース名が 'Oracle8i Concepts' であるビデオ・クリップを選択できます。

データベース内のマルチメディア・データのコレクションは、関連する内容を説明した属性やキーワードのセットと関連付けることができます。マルチメディア・データの内容説明は、テキスト・コンポーネントおよび日付や ID 番号などの数値属性を使用して行うことができます。Oracle8i では、データ属性はオブジェクト型と同じ表に配置可能です。別の方法として、アプリケーション設計者は複合オブジェクト型を定義して、その中に 1 つのオブジェクト型と他の属性を配置することもできます。

1.5.2.3 マルチメディア・データへのアクセス

アプリケーションは、SQL、PL/SQL、または JAVA を使用し、オブジェクト・リレーショナル・タイプ ORDAudio、ORDImage、および ORDVideo 経由で、マルチメディア・データへのアクセスおよび操作を実行します。Java の使用法の詳細は、『Oracle8i interMedia Audio, Image, Video Java Client ユーザーズ・ガイドおよびリファレンス』を参照してください。複合オブジェクト内で属性へアクセスするためのオブジェクト構文は、次のようにドット表記法を使用します。

```
variable.data_attribute
```

複合オブジェクトの起動メソッドの構文も、次のようにドット表記法を使用します。

```
variable.function(parameter1, parameter2, ...)
```

この構文および他の SQL 構文の詳細は、『Oracle8i 概要』を参照してください。

1.6 Oracle8i interMedia の拡張

interMedia audio、*interMedia* image、および *interMedia* video を拡張して、次の機能をサポートすることができます。

- 現在未サポートのオーディオ、イメージ、およびビデオ・データのその他の外部ソース
- 現在未サポートの、他のオーディオ、イメージ、およびビデオ・データ・フォーマット
- オーディオおよびビデオ・データ処理

次の項では、これらの各トピックおよび詳細情報の参照先について説明します。

1.6.1 他の外部ソースおよび他のオーディオ、イメージ、およびビデオ・データ・フォーマットのサポート

一意な外部オーディオ、イメージ、またはビデオ・データ・ソースごと、またはサポートする一意なオーディオまたはビデオ・データ・フォーマットごとに、次の作業を行う必要があります。

1. 新規データ・ソースまたは新規オーディオやビデオ・データ・フォーマットを設計します。
2. 新規データ・ソースまたは新規オーディオやビデオ・データ・フォーマットをインプリメントします。
3. 新規プラグインを ORDPLUGINS スキーマにインストールします。
4. 新規プラグインに対し PUBLIC への EXECUTE 権限を付与します。

他の外部ソースのサポート

新規データ・ソースをインプリメントするには、ORDPLUGINS スキーマ内の ORDX_
<srcType>_SOURCE パッケージの必要なインタフェースをインプリメントします。
(<srcType> は、新規外部ソース・タイプ名を表します。) 6.3.2 項内のパッケージ本体例をテンプレートとして使用して、パッケージ本体を作成します。その後、setSourceInformation()
) コール内のソース・パラメータを適切なソース値に設定し、オーディオ、イメージ、またはビデオ・オブジェクトに対し、パッケージ ORDPLUGINS.ORDX_
<srcType>_SOURCE がプラグインとして利用可能であることを示します。他の外部オーディオ、イメージ、またはビデオ・ソースにサポートを拡張する場合は、ORDPLUGINS.ORDX_FILE_SOURCE および ORDPLUGINS.ORDX_HTTP_SOURCE パッケージをガイドとして使用します。

サポートされるオーディオ、イメージ、およびビデオ・データの外部ソースの拡張例および詳細は、2.4 項、6.3.1 項、6.3.2 項、および 6.3.4 項を参照してください。

他のオーディオおよびビデオ・データ・フォーマットのサポート

新たなオーディオまたはビデオ・データ・フォーマットをインプリメントする場合は、ORDPLUGINS スキーマ内の ORDPLUGINS.ORDX_
<format>_
<media> パッケージの必要とされるインタフェースをインプリメントします。(<format> は新規オーディオまたはビデオ・データ・フォーマットを、<media> はメディア・フォーマットを表します。) 他のオーディオまたはビデオ・データ・フォーマットへサポートを拡張する場合は、ORDPLUGINS.ORDX_DEFAULT_
<media> パッケージをガイドとして使用します。3.3.6 項および 5.3.6 項のパッケージ本体の例をテンプレートとして使用して、オーディオまたはビデオ・パッケージ本体をそれぞれ作成します。その後、setFormat() コールの新規フォーマット・パラメータを適切なフォーマット値に設定して、オーディオまたはビデオ・オブジェクトに対し、パッケージ ORDPLUGINS.ORDX_
<format>_
<media> がプラグインとして利用可能であることを示します。

独自のフォーマット・プラグインのインストールおよび提供されたサンプル・スクリプトの実行の詳細は、E.1 項および E.3 項を参照してください。

サポートされるオーディオおよびビデオ・データ属性の拡張例および詳細は、2.1.14 項、2.3.12 項、3.3.1 項、および 5.3.1 項を参照してください。

他のイメージ・データ・フォーマットのサポート

Oracle8i *interMedia* イメージは、外部イメージ用の `setProperties()` メソッドを使用して、他の特定イメージ・フォーマットをサポートします。このメソッドを使用すると、`setProperties()` メソッドに外部イメージ用として渡される値を既存の `ORDImage` データ属性に書き込むことにより、他のイメージ・フォーマットの認識が可能になります。詳細は、第 4 章の「外部イメージの `setProperties()` メソッド」を参照してください。

1.6.2 オーディオ・データ処理のサポート

オーディオ・データ処理をサポートする（つまり、オーディオ処理コマンドおよび引数のセットをフォーマット・プラグインに渡して処理を行う）場合は、`processAudioCommand()` メソッドを使用します。このメソッドは、ユーザー定義形式でのみ使用可能です。

詳細は、第 3 章の「`processAudioCommand()` メソッド」および 2.1.14 項を参照してください。

1.6.3 ビデオ・データ処理のサポート

ビデオ・データ処理をサポートする（つまりコマンドおよび引数のセットをフォーマット・プラグインに渡して処理を行う）場合は、`processVideoCommand()` メソッドを使用します。このメソッドは、ユーザー定義形式でのみ使用可能です。

詳細は、第 5 章の「`processVideoCommand()` メソッド」および 2.3.12 項を参照してください。

この章の例は、Oracle8i *interMedia* の一般的な操作方法を示します。まず、オーディオ、イメージ、およびビデオ・データ・グループの例を示し、その後 *interMedia* を拡張して新規データ・ソースをサポートする方法を示します。

2.1 オーディオ・データの例

interMedia audio の例には、次の共通操作が含まれます。

- songObject という名前のソング・オブジェクトの定義
- SongsTable という名前のオブジェクト・テーブルの作成
- ソング・リストを含む songList という名前のリスト・オブジェクトの作成
- songList オブジェクトのインプリメンテーションの定義
- CD オブジェクトおよび CdTable テーブルの作成
- SongsTable テーブルへのソングの挿入
- CD の CdTable テーブルへの挿入
- SongsTable テーブルへのソングのロード
- ソング・オブジェクトへの参照の、CdTable テーブル内のソング・リストへの挿入
- ソングへの CD 参照の追加
- ソングに関するコメントの追加
- CD 内のソングからのオーディオ・データの取得
- CD 内のソングのすべてのコメントの表示
- 新規オーディオ・データ・フォーマットをサポートするための *interMedia* の拡張
- 新規オブジェクト型を使用した *interMedia* オーディオの拡張

■ オブジェクト・ビューでの *interMedia* の使用

この項のオーディオ・データ例では、ソング・テーブルおよび CD テーブルを使用します。ソングごとに、次の情報が格納されます。CDRef (CD テーブル内への REF)、ソング ID、ソング・タイトル、アーティスト、受賞、制作時期、再生時間、clipRef (オーディオ・クリップ・テーブルまたはミュージック・ビデオへの REF)、テキスト内容、および歌詞を含むオーディオ・ソースです。(REF は、グローバルに一意なオブジェクト ID を持つ行オブジェクトへの参照で、行オブジェクト間の参照を獲得します。アクセス速度を向上させるために、行オブジェクトの索引が自動的に作成されます。) CD ごとに、次の情報が格納されます。項目 ID、CD DB ID、CD タイトル、CD アーティスト、CD カテゴリ、著作権、プロデューサー名、受賞、制作時期、評価、再生時間、テキスト内容、カバー写真、および CD のソング・リストです。

これらの例で使用するメソッドに関する参照情報は、第 3 章を参照してください。

2.1.1 ソング・オブジェクトの定義

例 2-1 に、ソング・オブジェクトの定義方法を示します。

例 2-1 ソング・オブジェクトの定義

```
CREATE TYPE songObject as OBJECT (  
  cdRef      REF CdObject,  -- REF into the cd table  
  songId     VARCHAR2 (20),  
  title      VARCHAR2 (4000),  
  artist     VARCHAR2 (4000),  
  awards     VARCHAR2 (4000),  
  timePeriod VARCHAR2 (20),  
  duration   INTEGER,  
  clipRef    REF clipObject, -- REF into the clips table (music video)  
  txtcontent CLOB,  
  audioSource ORDSYS.ORDAUDIO  
);
```

2.1.2 オブジェクト・テーブル SongsTable の作成

例 2-2 に、SongsTable という名前のオブジェクト・テーブルの作成方法を示します。

例 2-2 SongsTable テーブルの作成

```
CREATE TABLE SongsTable of songObject (UNIQUE (songId), songId NOT NULL);
```

2.1.3 ソングへの参照リストを含むリスト・オブジェクトの作成

例 2-3 に、ソングへの参照リストを含むリスト・オブジェクトの作成方法を示します。

例 2-3 ソングへの参照リストを含むリスト・オブジェクトの作成

```
CREATE TYPE songNstType AS TABLE OF REF songObject;  
  
CREATE TYPE songList AS OBJECT (songs songNstType,  
    MEMBER PROCEDURE addSong(s IN REF songObject));
```

2.1.4 songList オブジェクトのインプリメンテーションの定義

例 2-4 に、songList オブジェクトのインプリメンテーションを定義する方法を示します。

例 2-4 songList オブジェクトのインプリメンテーションの定義

```
CREATE TYPE BODY songList AS  
    MEMBER PROCEDURE addSong(s IN REF songObject)  
    IS  
        pos INTEGER := 0;  
    BEGIN  
        IF songs IS NULL THEN  
            songs := songNstType(NULL);  
            pos := 0;  
        ELSE  
            pos := songs.count;  
        END IF;  
        songs.EXTEND;  
        songs(pos+1) := s;  
    END;  
END;
```

2.1.5 CD オブジェクトと CD テーブルの作成

この項では、オーディオ・クリップの CD オブジェクトと CD テーブルの作成方法について説明します。各オーディオ・クリップには、次の情報が含まれます。

- 項目 ID
- CD DB ID
- CD タイトル
- CD アーティスト

- CD カテゴリ
- 著作権
- プロデューサ名
- 受賞
- 制作時期
- 評価
- 再生時間
- テキスト内容
- カバー写真
- ソング

例 2-5 では、CdObject という名前の CD オブジェクト、および CD 情報を格納する CdTable という名前の CD テーブルを作成します。

例 2-5 CD 情報を格納する CD テーブルの作成

```
CREATE TYPE CdObject as OBJECT (  
  itemId      INTEGER,  
  cddbID      INTEGER,  
  title       VARCHAR2(4000),  
  artist      VARCHAR2(4000),  
  category    VARCHAR2(20),  
  copyright   VARCHAR2(4000),  
  producer    VARCHAR2(4000),  
  awards      VARCHAR2(4000),  
  timePeriod  VARCHAR2(20),  
  rating      VARCHAR2(256),  
  duration    INTEGER,  
  txtcontent  CLOB,  
  coverImg    REF ORDSYS.ORDImage,  
  songs       songList);  
  
CREATE TABLE CdTable OF CdObject (UNIQUE(itemId), itemId NOT NULL)  
  NESTED TABLE songs.songs STORE AS song_store_table;
```

2.1.6 SongsTable テーブルへのソングの挿入

例 2-6 に、SongsTable テーブルへのソングの挿入方法を示します。

例 2-6 SongsTable テーブルへのソングの挿入

```
-- insert a song into the songs table
```



```

INSERT INTO SongsTable VALUES (NULL,
                                '00',
                                'Under Pressure',
                                'Queen',
                                'no awards',
                                '80-90',
                                243,
                                NULL,
                                EMPTY_CLOB(),
                                ORDSYS.ORDAudio(NULL,
ORDSYS.ORDSource(EMPTY_BLOB(),NULL,NULL,NULL,NULL,NULL),
NULL, NULL, EMPTY_CLOB(), NULL, NULL, NULL, NULL, NULL, NULL));

-- check songs insertion
SELECT s.title
FROM   SongsTable s
WHERE  songId = '00';

```

2.1.7 CdTable テーブルへの CD の挿入

例 2-7 に、CdTable テーブルへの CD の挿入方法を示します。

例 2-7 CdTable テーブルへの CD の挿入

```

-- insert a cd into the cd table
INSERT INTO CdTable VALUES (1, 23232323,
                              'Queen Classics',
                              'Queen',
                              'rock',
                              'BMV Company',
                              'BMV',
                              'Grammy',
                              '80-90',
                              'no rating',
                              4000,           -- in seconds
                              EMPTY_CLOB(),
                              NULL,
                              songList(NULL));

-- check cd insertion
SELECT cd.title
FROM   Cdtable cd;

```

2.1.8 SongsTable テーブルへのソングのロード

例 2-8 に、SongsTable テーブルにソングをロードする方法を示します。この例では、AUDDIR ディレクトリを定義する必要があります。コード中のコメントを参照してください。

例 2-8 SongsTable テーブルへのソングのロード

```
-- Load a Song into the SongsTable
-- Create your directory specification below
-- CREATE OR REPLACE DIRECTORY AUDDIR AS '/audio/';
DECLARE
    audioObj ORDSYS.ORDAUDIO;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT S.audioSource INTO audioObj
    FROM   SongsTable S
    WHERE  S.songId = '00'
    FOR UPDATE;

    audioObj.setSource('FILE', 'AUDDIR', 'UnderPressure.au');
    audioObj.setMimeType('audio/basic');
    audioObj.import(ctx);
    audioObj.setProperties(ctx);

    UPDATE SongsTable S
    SET    S.audioSource = audioObj
    WHERE  S.songId = '00';
    COMMIT;
END;

-- check song insertion
DECLARE
    audioObj ORDSYS.ORDAUDIO;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT S.audioSource INTO audioObj
    FROM   SongsTable S
    WHERE  S.songId = '00';

    dbms_output.put_line('Content Length: ' ||
                          audioObj.getContentLength(ctx));
    dbms_output.put_line('Content MimeType: ' ||
                          audioObj.getMimeType());
END;
```

2.1.9 ソング・オブジェクトへの参照の、CdTable テーブル内のソング・リストへの挿入

例 2-9 に、ソング・オブジェクトへの参照を、CdTable テーブル内のソング・リストに挿入する方法を示します。

例 2-9 ソング・オブジェクトへの参照の、CdTable テーブル内のソング・リストへの挿入

```
-- Insert a reference to a SongObject into the Songs List in the CdTable Table
DECLARE
    songRef REF SongObject;
    songListInstance songList;
BEGIN
    SELECT REF(S) into songRef
    FROM   SongsTable S
    where  S.songId = '00';

    SELECT C.songs INTO songListInstance
    FROM   CdTable C
    WHERE  C.itemId = 1
    FOR UPDATE;

    songListInstance.addSong(songRef);

    UPDATE CdTable C
    SET    C.songs = songListInstance
    WHERE  C.itemId = 1;

    COMMIT;
END;

-- check insertion of ref
-- this example works for the first entry inserted in the songList
DECLARE
    song          SongObject;
    songRef       REF SongObject;
    songListInstance songList;
    songType      songNstType;
BEGIN
    SELECT C.songs INTO songListInstance
    FROM   CdTable C
    WHERE  C.itemId = 1;

    SELECT songListInstance.songs INTO songType FROM DUAL;
    songRef := songType(1);
    SELECT Deref(songRef) INTO song FROM DUAL;
```

```
        dbms_output.put_line('Song Title: ' ||  
                               song.title);  
END;
```

2.1.10 ソングへの CD 参照の追加

例 2-10 に、CD 参照をソングに追加する方法を示します。

例 2-10 ソングへの CD 参照の追加

```
-- Adding a cd reference to a song  
DECLARE  
    songCdRef  REF CdObject;  
BEGIN  
    SELECT S.cdRef INTO songCdRef  
    FROM   SongsTable S  
    WHERE  S.songId = '00'  
    FOR UPDATE;  
  
    SELECT REF(C) INTO songCdRef  
    FROM   CdTable C  
    WHERE  C.itemId = 1;  
  
    UPDATE SongsTable S  
    SET    S.cdRef = songCdRef  
    WHERE  S.songId = '00';  
  
    COMMIT;  
END;  
  
-- check cd Ref  
DECLARE  
    cdRef REF CdObject;  
    cd    CdObject;  
BEGIN  
    SELECT S.cdRef INTO cdRef  
    FROM   SongsTable S  
    WHERE  S.songId = '00';  
  
    SELECT Deref(cdRef) INTO cd FROM DUAL;  
  
    dbms_output.put_line('Cd Title: ' ||  
                           cd.title);  
END;
```

2.1.11 ソングに関するコメントの追加

例 2-11 に、ソングのコメントを追加する方法を示します。

例 2-11 ソングに関するコメントの追加

```
-- Adding comments about a song
DECLARE
    audioObj ORDSYS.ORDAUDIO;
    comments VARCHAR2(256);
BEGIN
    SELECT S.audioSource into audioObj
    FROM   SongsTable S
    WHERE  S.songId = '00'
    FOR UPDATE;

    comments := 'I like this song when I am Under Pressure';
    audioObj.writeToComments(1, 41, comments);

    UPDATE SongsTable S
    SET     S.audioSource = audioObj
    WHERE  S.songId = '00';

    COMMIT;
END;
```

2.1.12 CD 内のソングからのオーディオ・データの取得

例 2-12 に、CD 内のソングからオーディオ・データを取得する方法を示します。

例 2-12 CD 内のソングからのオーディオ・データの取得

```
FUNCTION retrieveAudio(cdId IN INTEGER,
    songId IN INTEGER) RETURN BLOB IS obj ORDSYS.ORDAudio;
BEGIN
    select S.audioSource into obj from SongsTable S
    where S.songId = songId;
    return obj.getContent;
END;
```

2.1.13 CD 内のソングに関するすべてのコメントの表示

例 2-13 に、CD 内のソングに関するコメントをすべて表示する方法を示します。

例 2-13 CD 内のソングに関するすべてのコメントの表示

```
-- Displaying comments about a song
```

```
DECLARE
    audioObj ORDSYS.ORDAUDIO;
    comments VARCHAR2(256);
BEGIN
    SELECT S.audioSource into audioObj
    FROM   SongsTable S
    WHERE  S.songId = '00';

    comments := audioObj.readFromComments(1, 41);
    dbms_output.put_line('Comments: ' || comments);
END;
```

2.1.14 *interMedia* の拡張による新しいオーディオ・データ・フォーマットのサポート

新しいオーディオ・データ・フォーマットをサポートする場合、ORDPLUGINS スキーマ内の ORDX_<format>_AUDIO パッケージの必要なインタフェースを実装します。（<format>は、新しいオーディオ・データ・フォーマット名を表します。）ORDX_DEFAULT_AUDIO パッケージのインタフェースの詳細は、3.3.1 項を参照してください。3.3.6 項内のパッケージ本体の例をテンプレートとして使用して、オーディオ・パッケージ本体を作成します。その後、setFormat コール内の新規形式パラメータを適切なフォーマット値に設定して、オーディオ・オブジェクトに対し、パッケージ ORDPLUGINS.ORDX_<format>_AUDIO がプラグインとして利用可能であることを示します。

独自のフォーマット・プラグインのインストールおよび提供されるサンプル・スクリプトの実行方法の詳細は、E.1 項を参照してください。\$ORACLE_HOME/ord/aud/demo/ ディレクトリにインストールされる fplugins.sql および fpluginb.sql ファイルを参照してください。これらは、サポートしようとするフォーマット・プラグインを記述する際の指針となるデモ・プラグインです。独自のフォーマット・プラグインのインストール方法については、同じディレクトリ内の auddemo.sql ファイルを参照してください。

2.1.15 新しいオブジェクト型を使用した *interMedia* の拡張

この項では、新しいオブジェクト型を使用して Oracle8i *interMedia* を拡張する方法について説明します。

任意の *interMedia* オブジェクト型を、新たに作成する独自型のベースとして使用できます。

この簡単な例を、例 2-3 および例 2-4 に示します。詳細な例と説明は、例 2-26 を参照してください。

2.1.16 オブジェクト・ビューと共に使用するオーディオ型

この項では、オブジェクト・ビューと共にオーディオ型を使用する方法を説明します。ビューが仮想テーブルであるように、オブジェクト・ビューは仮想的なオブジェクト・テーブルです。

Oracle は、オブジェクト・ビューを基本的なリレーショナル・ビュー・メカニズムとして提供しています。オブジェクト・ビューを使用することにより、データベースのリレーショナルまたはオブジェクト・テーブル列に格納された、ビルトインまたはユーザー定義のいずれかのデータから仮想的なオブジェクト・テーブルを作成できます。

オブジェクト・ビューを使用すると、データベース内のデータおよびオブジェクトに対する、特殊または制限付きのアクセスを設定できます。たとえば、オブジェクト・ビューを使用して、機密性の高いデータや削除メソッドを含む属性を持たない従業員オブジェクト・テーブルのバージョンを提供できます。また、オブジェクト・ビューを使用することにより、恒久的にテーブルを変換することなくオブジェクト指向プログラミングを行えます。オブジェクト・ビューを使用して、リレーショナル表からオブジェクト・リレーショナル表へ段階的かつ透過的に変換できます。

次の非オブジェクト・オーディオ・テーブルについて考えてみましょう。

```
create table flat (
  id                NUMBER,
  description       VARCHAR2(4000),
  localData        BLOB,
  srcType           VARCHAR2(4000),
  srcLocation       VARCHAR2(4000),
  srcName           VARCHAR2(4000),
  upDateTime        DATE,
  local            NUMBER,
  format            VARCHAR2(31),
  mimeType          VARCHAR2(4000),
  comments          CLOB,
  encoding          VARCHAR2(256),
  numberOfChannels  NUMBER,
  samplingRate      NUMBER,
  sampleSize        NUMBER,
  compressionType   VARCHAR2(4000),
  audioDuration     NUMBER,
  audioclip         RAW(2000)
);
```

次のようにして、フラット・テーブル上にオブジェクト・ビューを作成できます。

```
create or replace view object_audio_v as
select
  id,
  ordsys.ORDAudio(
    T.description,
```

```
T.localData,  
T.comments,  
T.format,  
T.encoding,  
T.numberOfChannels,  
T.samplingRate,  
T.sampleSize,  
T.compressionType,  
T.audioDuration,  
T.audioclip) AUDIO  
from flat T;
```

オブジェクト・ビューを使用すると、1つのリレーショナル・データまたはオブジェクト・データを複数の方法で柔軟に表示できます。このため、データベース内のデータ格納方法を変更せずに、メモリー内のオブジェクト表現をさまざまなアプリケーションに合わせることができます。オブジェクト・ビューの定義、使用、および更新の詳細は、『Oracle8i 概要』を参照してください。

2.2 イメージ・データの例

interMedia image の例には、次の共通操作が含まれます。

- 新規または既存のテーブルへの型の追加
- BLOB イメージを使用した行の挿入
- BLOB イメージを使用した行の移入
- BFILE イメージを使用した行の挿入
- BFILE イメージを使用した行の移入
- 行の問合せ
- 外部ファイルからデータベース内へのイメージのインポート
- イメージのコピー
- イメージ・フォーマットの変換
- イメージのコピーおよび変換の一括操作
- 新規オブジェクト型を使用した *interMedia* の拡張
- オブジェクト・ビューでのイメージ型の使用
- 各国語サポート (NLS) 問題への取り組み

2.2.1 既存のテーブルへのイメージ型の追加

'emp' という名前の既存のテーブルに次の列が含まれる場合を考えます。

```
ename      VARCHAR2(50)
salary     NUMBER
job         VARCHAR2(50)
department INTEGER
```

ORDImage 型を使用して 'emp' テーブルに 'photo' という名前の列を追加する場合は、例 2-14 に示す文を発行します。

例 2-14 では、emp テーブルに ORDImage 型の列を新たに追加します。

例 2-14 ORDImage 型の新規列の emp テーブルへの追加

```
ALTER TABLE emp
ADD (photo ORDSYS.ORDImage);
```

2.2.2 新規テーブルへのイメージ型の追加

次の情報を保持する、'emp' という名前の新規テーブルを作成する場合を考えます。

- 従業員名
- 給与
- 職種
- 部署名
- バッジ用写真
- 大判写真

バッジ用写真（大判の人物写真をクロップおよび倍率変更して作成した縮小イメージ）列および 'large_photo' 列は、ORDImage 型を使用します。例 2-15 の文では、テーブルを作成し、作成したテーブルに ORDImage 型を追加します。

例 2-15 ORDImage 型の新規テーブルへの追加

```
CREATE TABLE emp (
  ename VARCHAR2(50),
  salary NUMBER,
  job VARCHAR2(50),
  department INTEGER,
  photo ORDSYS.ORDImage,
  large_photo ORDSYS.ORDImage);
```

2.2.3 BLOB イメージを使用した行の挿入

イメージ内容を格納するテーブル内に、ORDImage 型を使用して行を挿入する場合は、初期化機能を使用して型を移入する必要があります。これは、NULL とは異なることに注意してください。NULL 値を使用して ORDImage 型を使用しようとすると、エラーが発生します。

例 2-16 に、ORDImage 型を使用してテーブル内に行を挿入する方法を示します。テーブル 'emp' に次の列が含まれるとします。

```
ename      VARCHAR2 (50)
salary     NUMBER
job         VARCHAR2 (50)
department INTEGER
photo      ORDImage
```

イメージ・データをデータベースに格納する場合（バイナリ・ラージ・オブジェクト（BLOB）形式で）、ある値で ORDSys.localData 属性を移入し、empty_blob() コンストラクタを使用して localData 属性用の記憶領域を初期化する必要があります。'photo' 列のデータが空白の行をテーブルに挿入するには、例 2-16 に示す文を発行します。

例 2-16 に、ORDImage 型列が空白データである行をテーブルに挿入する方法を示します。

例 2-16 ORDImage 型列のデータが空白である行のテーブルへの挿入

```
INSERT INTO emp VALUES (
  'John Doe', 24000, 'Technical Writer', 123,
  ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob()), NULL, NULL, NULL, SYSDATE, 1),
  NULL, NULL, NULL, NULL, NULL, NULL, NULL);
```

2.2.4 BLOB イメージを使用した行の移入

BLOB 値を更新する前に、BLOB ロケータを含む行をロックする必要があります。これは、通常、SQL および PL/SQL プログラムの SELECT FOR UPDATE 文、または OCI プログラムの Oracle コール・インタフェース（OCI）確保機能やロック機能を使用して行われます。

例 2-17 に、ORDImage BLOB データを使用した行の移入を示します。

例 2-17 ORDImage BLOB データを使用した行の移入

```
DECLARE
  -- applicaiton variables
  Image ORDSYS.ORDImage;
BEGIN
  INSERT INTO emp VALUES (
    'John Doe', 24000, 'Technical Writer', 123,
    ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob()), NULL, NULL, NULL, SYSDATE, 1),
    NULL, NULL, NULL, NULL, NULL, NULL, NULL);
  -- select the newly inserted row for update
```

```

SELECT photo INTO Image FROM emp
      WHERE ename = 'John Doe' for UPDATE;
--BEGIN
-- use the getContent method to get the LOB locator.
-- populate the data with dbms lob calls or write an OCI program to
-- fill in the image BLOB.
--END;
-- set property attributes for the image data
Image.setProperties;
UPDATE emp SET photo = Image WHERE ename = 'John Doe';
-- continue processing
END;
```

UPDATE 文は、プロパティ属性の更新に必要です。setProperties() メソッドおよび UPDATE 文をすぐに使用しない場合は、コミットしてイメージへの変更を BLOB 属性（プロパティではなく）に反映させることができます。BLOB の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

2.2.5 BFILE イメージを使用した行の挿入

外部ファイルのイメージ内容を格納するテーブル内に、ORDImage 型を使用して行を挿入する場合は、初期化機能を使用して型を移入する必要があります。これは、NULL とは異なることに注意してください。NULL 値を使用して ORDImage 型を使用しようとすると、エラーが発生します。

例 2-18 に、ORDImage 型を使用してテーブル内に行を挿入する方法を示します。テーブル 'emp' に次の列が含まれるとします。

```

ename      VARCHAR2(50)
salary     NUMBER
job         VARCHAR2(50)
department INTEGER
large_photo ORDImage
```

ORDImage 型の列を使用する場合は、まず、ある値でその列を移入する必要があります。ORDImage 型列の値を、ファイル内に外部的に格納されたイメージで移入する場合は、その行をファイル・コンストラクタを使用して移入する必要があります。

例 2-18 に、行をテーブルに挿入して、ORDIMGDIR ディレクトリ内の 'jdoe.gif' というイメージをその中に格納する方法を示します。

例 2-18 ORDImage 型列にイメージを含む行の、テーブルへの挿入

```

INSERT INTO emp VALUES (
  'John Doe', 24000, 'Technical Writer', 123,
  ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob(), 'file', 'ORDIMGDIR',
    'jdoe.gif', SYSDATE, 0),
```

```
NULL,NULL,NULL,NULL,NULL,NULL,NULL));
```

注意: このリリースの Oracle8i では、ORDImage BFILE 型の内容は読み専用です。

オブジェクト型への行の挿入方法については、第 4 章および『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

sourceLocation 引数 'ORDIMGDIR' は、ファイル・システム・ディレクトリへの参照を示すディレクトリです。ディレクトリ名は大文字でなければならないことに注意してください。次のコードは、ORDIMGDIR という名前のディレクトリを作成します。

```
connect internal
-- make a directory referring to a file system directory
create directory ORDIMGDIR as '<MYIMAGEDIRECTORY>';
grant read on directory ORDIMGDIR to <user-or-role>;
```

<MYIMAGEDIRECTORY> はシステム・ディレクトリを、<user-or-role> は読み込みアクセス権を与える特定のユーザーを、それぞれ表します。

2.2.6 BFILE イメージを使用した行の移入

例 2-19 に、外部ファイルに格納された ORDImage データを使用して行を移入する方法を示します。

例 2-19 ORDImage 外部ファイル・データを使用した行の移入

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    INSERT INTO emp VALUES ('John Doe', 24000, 'Technical Writer', 123,
        ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob(), 'file', 'ORDIMGDIR',
            'jdoe.gif', SYSDATE, 0),
            NULL, NULL, NULL, NULL, NULL, NULL, NULL));
    -- select the newly inserted row for update
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- set property attributes for the image data
    Image.setProperties;
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
    -- continue processing
END;
```

2.2.7 行の問合せ

例 2-20 および例 2-21 では、次のようなテーブルがあることを前提としています。

```
create table emp (  
  ename VARCHAR2(50),  
  salary NUMBER,  
  job VARCHAR2(50),  
  department INTEGER,  
  photo ORDSYS.ORDImage,  
  large_photo ORDSYS.ORDImage);
```

例 2-20 では、ORDImage データを保持する行の問合せを行います。SELECT 文で型を参照する際に、テーブルの別名（この例では E）を作成する必要があります。

例 2-20 任意のコンテンツ長の ORDImage データを保持する行の問合せ

```
SELECT ename, E.large_photo.getWidth()  
FROM emp E  
WHERE ename = 'John Doe' and  
      E.large_photo.getWidth() > 32;
```

例 2-21 では、ORDImage データを保持する行の問合せを行います。

例 2-21 特定のコンテンツ長の ORDImage データを保持する行の問合せ

```
SELECT ename, E.large_photo.getCompressionFormat()  
FROM emp E  
WHERE ename = 'John Doe' and  
      E.large_photo.getWidth() > 32 and  
      E.large_photo.getContentLength() > 10000;
```

2.2.8 外部ファイルからデータベース内へのイメージのインポート

外部ファイルからデータベース内にイメージをインポートする場合は、ORDImage.import メソッドを使用します。例 2-22 に、外部ファイルからデータベース内へイメージ・データをインポートするプログラムを示します。import() メソッドをコールする前に、ソース・タイプ、ソース位置、およびソース名を設定する必要があります。

例 2-22 外部ファイルからのイメージのインポート

```
DECLARE  
  Image ORDSYS.ORDImage;  
  ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT large_photo  
  INTO Image FROM emp
```

```
        WHERE ename = 'John Doe' FOR UPDATE;
-- import the image into the database
Image.import(ctx);
UPDATE emp SET large_photo = IMAGE
        WHERE ename = 'John Doe';
END;
```

2.2.9 イメージのコピー

イメージをコピーするには、`ORDImage.copy` メソッドを使用します。例 2-23 に、イメージ・データをコピーするプログラムを示します。

例 2-23 イメージのコピー

```
DECLARE
    Image_1 ORDSYS.ORDImage;
    Image_2 ORDSYS.ORDImage;
BEGIN
    SELECT photo INTO Image_1
        FROM emp WHERE ename = 'John Doe';
    SELECT photo INTO Image_2
        FROM emp WHERE ename = 'Also John Doe' FOR UPDATE;
    -- copy the data from Image_1 to Image_2
    Image_1.copy(Image_2);
    -- continue processing
    UPDATE emp SET photo = Image_2
        WHERE ename = 'Also John Doe';
END;
```

2.2.10 イメージ・フォーマットの変換

イメージ・データを異なる形式に変換するには、`process()` メソッドを使用します。

注意: `process()` メソッドは BLOB 内だけを処理対象とするため、イメージ・データはローカルに格納する必要があります。

例 2-24 に、イメージ・データを TIFF ファイル形式に変換するプログラムを示します。

例 2-24 イメージ形式の変換

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
```

```
-- convert the image to TIFF (in place)
Image.process('fileFormat=TIFF');
UPDATE emp SET photo = Image WHERE ename = 'John Doe';
END;
```

2.2.11 コピーおよび変換の一括操作

イメージのコピーと変換を一括して実行する場合は、`processCopy()` メソッドを使用します。

注意: `processCopy()` メソッドは BLOB 内だけを処理対象とするため、宛先イメージをローカルに設定し、ソースの `localData` 属性を初期化する必要があります。

例 2-25 に示すプログラムは、縮小イメージを作成し、イメージ・データを TIFF イメージ・ファイル形式へ変換して BLOB へコピーし、かつ元のイメージは変更しません。

例 2-25 イメージ形式のコピーと変換

```
DECLARE
    Image_1 ORDSYS.ORDImage;
    Image_2 ORDSYS.ORDImage;
BEGIN
    SELECT photo, large_photo
    INTO Image_2, Image_1
    FROM emp
    WHERE ename = 'John Doe' FOR UPDATE;
    -- convert the image to a tiff thumbnail and store the result in Image_2
    Image_1.processCopy('fileFormat=TIFF fixedScale=32 32', Image_2);
    -- continue processing
    UPDATE emp SET photo = Image_2 WHERE ename = 'John Doe';
END;
```

`processCopy()` メソッドを使用して行った変更は、ロール・バック可能です。この方法は、形式を一時的に変換する場合に役立ちます。

2.2.12 新しい型を使用した *interMedia* の拡張

例 2-26 に示すように、`ORDImage` 型を新たに作成する独自型のベースとして使用できます。

例 2-26 新しいオブジェクト型を使用した Oracle8i *interMedia* イメージの拡張

```
CREATE TYPE AnnotatedImage AS OBJECT
( image ORDSYS.ORDImage,
```

イメージ・データの例

```
        description VARCHAR2(2000),
        MEMBER PROCEDURE SetProperties(SELF IN OUT AnnotatedImage),
        MEMBER PROCEDURE Copy(dest IN OUT AnnotatedImage),
        MEMBER PROCEDURE ProcessCopy(command IN VARCHAR2,
                                         dest IN OUT AnnotatedImage)
    );
/

CREATE TYPE BODY AnnotatedImage AS
    MEMBER PROCEDURE SetProperties(SELF IN OUT AnnotatedImage) IS
    BEGIN
        SELF.image.setProperties;
        SELF.description :=
            'This is an example of using Image object as a subtype';
    END SetProperties;
    MEMBER PROCEDURE Copy(dest IN OUT AnnotatedImage) IS
    BEGIN
        SELF.image.copy(dest.image);
        dest.description := SELF.description;
    END Copy;
    MEMBER PROCEDURE ProcessCopy(command IN VARCHAR2,
                                   dest IN OUT AnnotatedImage) IS
    BEGIN
        SELF.Image.processCopy(command, dest.image);
        dest.description := SELF.description;
    END ProcessCopy;
END;
/
```

新たな型を作成した後は、その型を他の任意の型と同様に使用できます。

例：

```
create or replace directory TEST_DIR as 'C:\TESTS';

CREATE TABLE my_example(id NUMBER, an_image AnnotatedImage);
INSERT INTO my_example VALUES (1,
    AnnotatedImage(
        ORDSYS.ORDImage(
            ORDSYS.ORDSource(empty_blob(), 'file', 'ORDIMGDIR',
                'jdoe.gif', SYSDATE, 0),
            NULL, NULL, NULL, NULL, NULL, NULL, NULL),
        NULL));
COMMIT;
DECLARE
    myimage AnnotatedImage;
BEGIN
```



```

SELECT an_image INTO myimage FROM my_example;
myimage.SetProperties;
DBMS_OUTPUT.PUT_LINE('This image has a description of ');
DBMS_OUTPUT.PUT_LINE(myimage.description);
UPDATE my_example SET an_image = myimage;
END;
/

```

2.2.13 オブジェクト・ビューと共に使用するイメージ型

ビューが仮想テーブルであるように、オブジェクト・ビューは仮想的なオブジェクト・テーブルです。

Oracle は、オブジェクト・ビューを基本的なリレーショナル・ビュー・メカニズムとして提供しています。オブジェクト・ビューを使用することにより、データベースのリレーショナルまたはオブジェクト・テーブル列に格納された、ビルトインまたはユーザー定義のいずれかのデータから仮想的なオブジェクト・テーブルを作成できます。

オブジェクト・ビューを使用すると、データベース内のデータおよびオブジェクトに対する、特殊または制限付きのアクセスを設定できます。たとえば、オブジェクト・ビューを使用して、機密性の高いデータや削除メソッドを含む属性を持たない従業員オブジェクト・テーブルのバージョンを提供できます。また、オブジェクト・ビューを使用することにより、恒久的にテーブルを変換することなくオブジェクト指向プログラミングを行えます。オブジェクト・ビューを使用して、リレーショナル表からオブジェクト・リレーショナル表へ段階的かつ透過的に変換できます。

次の非オブジェクト・イメージ・テーブルについて考慮してみましょう。

```

CREATE TABLE flat(
  id          NUMBER,
  localData   BLOB,
  srcType     VARCHAR2(4000),
  srcLocation VARCHAR2(4000),
  srcName     VARCHAR2(4000),
  updateTime  DATE,
  local       NUMBER,
  height      INTEGER,
  width       INTEGER,
  contentLength INTEGER,
  fileFormat  VARCHAR2(4000),
  contentFormat VARCHAR2(4000),
  compressionFormat VARCHAR2(4000),
  mimeType    VARCHAR2(4000)
);

```

例 2-27 に、フラット・テーブル上でオブジェクト・ビューを作成する方法を示します。

例 2-27 フラット・テーブル上でのオブジェクト・ビューの作成

```
CREATE OR REPLACE VIEW object_images_v AS
SELECT
  id,
  ORDSYS.ORDImage(
    ORDSYS.ORDSource(
      T.localData,
      T.srcType,
      T.srcLocation,
      T.srcName,
      T.updateTime,
      T.local),
    T.height,
    T.width,
    T.contentLength,
    T.fileFormat,
    T.contentFormat,
    T.compressionFormat,
    T.mimeType
  ) IMAGE
FROM flat T;
```

オブジェクト・ビューを使用すると、1つのリレーショナル・データまたはオブジェクト・データを複数の方法で柔軟に表示できます。このため、データベース内のデータ格納方法を変更せずに、メモリー内のオブジェクト表現をさまざまなアプリケーションに合わせるすることができます。オブジェクト・ビューの定義、使用、および更新の詳細は、『Oracle8i 概要』を参照してください。

2.2.14 各国語サポート（NLS）問題への取組み

例 2-28 に、カンマを小数点として使用する言語設定で `processCopy()` メソッドを使用する方法を示します。たとえば、地域が **FRANCE** である場合、小数点はカンマであることが予想されます。スケール・ファクタとして指定された "75" に留意してください。次のアプリケーションは、各国語サポートの問題に対処します。

例 2-28 各国語サポート問題への対処

```
ALTER SESSION SET NLS_LANGUAGE = FRENCH;
ALTER SESSION SET NLS_TERRITORY = FRANCE;
DECLARE
  myimage ORDSYS.ORDImage;
  mylargeimage ORDSYS.ORDImage;
BEGIN
  SELECT photo, large_photo INTO myimage, mylargeimage
  FROM emp FOR UPDATE;
```

```

myimage.setProperties;
myimage.ProcessCopy('scale=',75'', mylargeimage);
UPDATE emp SET photo = myimage, large_photo = mylargeimage;
END;
/

```

2.3 ビデオ・データの例

interMedia video の例には、次の共通操作が含まれます。

- clipObject という名前のクリップ・オブジェクトの定義
- clipsTable という名前のオブジェクト・テーブルの作成
- クリップ・リストを含む clipList という名前のリスト・オブジェクトの作成
- clipList オブジェクトのインプリメンテーションの定義
- ビデオ・オブジェクトおよび videoTable テーブルの作成
- clipsTable テーブルへのビデオ・クリップの挿入
- videoTable テーブルへの行の挿入
- clipsTable テーブルへのビデオのロード
- clipObject への参照の、ビデオ・テーブルのクリップ・リストへの挿入
- ビデオ・オブジェクトへの参照の、クリップへの挿入
- videoTable テーブルからのビデオ・クリップの取得
- 新規ビデオ・データ・フォーマットをサポートするための *interMedia* の拡張
- 新規オブジェクト型を使用した *interMedia* の拡張
- オブジェクト・ビューでのビデオ型の使用

この項のビデオ例では、ビデオ・クリップ・テーブルおよびビデオ・テーブルを使用します。ビデオ・クリップごとに、次の情報が格納されます。**videoRef** (ビデオ・テーブルへの REF)、クリップ ID、タイトル、ディレクタ、キャスト、カテゴリ、著作権、プロデューサ、受賞、制作時期、評価、再生時間、**cdRef** (サウンド・トラック用 **CdObject** への REF)、テキスト内容 (CONTEXT による索引)、カバー写真 (イメージ・テーブルへの REF)、およびビデオ・ソースです。ビデオごとに、次の情報が格納されます。項目 ID、再生時間、テキスト内容 (CONTEXT による索引)、カバー写真 (イメージ・テーブルへの REF)、およびビデオ・クリップ・リストです。

これらの例で使用されるメソッドに関する参照情報は、第 5 章および付録 F を参照してください。

2.3.1 Clip オブジェクトの定義

例 2-29 に、クリップ・オブジェクトの定義方法を示します。

例 2-29 クリップ・オブジェクトの定義

```
CREATE TYPE clipObject as OBJECT (  
    videoRef      REF VideoObject,      -- REF into the video table  
    clipId        VARCHAR2(20),         -- Id inside of the clip table  
    title         VARCHAR2(4000),  
    director      VARCHAR2(4000),  
    casting       ORDSYS.ORDAnnotations, -- Pairs of characters/actors  
    category      VARCHAR2(20),  
    copyright     VARCHAR2(4000),  
    producer      VARCHAR2(4000),  
    awards        VARCHAR2(4000),  
    timePeriod    VARCHAR2(20),  
    rating        VARCHAR2(256),  
    duration      INTEGER,  
    cdRef         REF CdObject,         -- REF into a CdObject(soundtrack)  
    txtcontent     CLOB,  
    coverImg      REF ORDSYS.ORDImage,  -- REF into the ImageTable  
    videoSource   ORDSYS.ORDVideo);
```

ORDAnnotations 型およびそのメソッドについては、付録 F を参照してください。

2.3.2 オブジェクト・テーブル clipsTable の作成

例 2-30 に、clipsTable という名前のオブジェクト・テーブルの作成方法を示します。

例 2-30 clipsTable テーブルの作成

```
CREATE TABLE ClipsTable of clipObject (UNIQUE (clipId), clipId NOT NULL)  
    NESTED TABLE casting.annotations STORE AS annot_store_table2;
```

2.3.3 クリップ・リストを含むリスト・オブジェクトの作成

例 2-31 に、クリップ・リストを含むリスト・オブジェクトの作成方法を示します。

例 2-31 クリップ・リストを含むリスト・オブジェクトの作成

```
CREATE TYPE clipNstType AS TABLE of REF clipObject;  
  
CREATE TYPE clipList AS OBJECT (clips clipNstType,  
    MEMBER PROCEDURE addClip(c IN REF clipObject));
```

2.3.4 clipList オブジェクトのインプリメンテーションの定義

例 2-32 に、clipList オブジェクトのインプリメンテーションを定義する方法を示します。

例 2-32 clipList オブジェクトのインプリメンテーションの定義

```
CREATE TYPE BODY clipList AS
  MEMBER PROCEDURE addClip(c IN REF clipObject)
  IS
    pos INTEGER := 0;
  BEGIN
    IF clips IS NULL THEN
      clips := clipNstType(NULL);
      pos := 0;
    ELSE
      pos := clips.count;
    END IF;
    clips.EXTEND;
    clips(pos+1) := c;
  END;
END;
```

2.3.5 ビデオ・オブジェクトとビデオ・テーブルの作成

この項では、ビデオ・クリップのビデオ・オブジェクトとビデオ・テーブルの作成方法について説明します。各ビデオ・クリップには、次の情報が含まれます。

- 項目 ID
- 再生時間
- テキスト内容
- カバー写真
- クリップ

例 2-33 に、videoObject という名前のビデオ・オブジェクト、ビデオ情報を保持する videoTable という名前のビデオ・テーブルの作成方法を示します。

例 2-33 ビデオ情報を格納するビデオ・テーブルの作成

```
CREATE TYPE VideoObject as OBJECT (
  itemId      INTEGER,
  duration    INTEGER,
  txtcontent  CLOB,
  coverImg    REF ORDSYS.ORDImage,
  clips       clipList);
```

```
CREATE TABLE VideoTable OF VideoObject (UNIQUE(itemId),itemId NOT NULL)
    NESTED TABLE clips.clips STORE AS clip_store_table;
```

2.3.6 clipsTable テーブルへのビデオ・クリップの挿入

例 2-34 に、clipsTable テーブルへのビデオ・クリップの挿入方法を示します。

例 2-34 clipsTable テーブルへのビデオ・クリップの挿入

```
-- Insert a Video Clip into the ClipsTable
insert into ClipsTable values (NULL,
    '11',
    'Oracle Commercial',
    'Larry Ellison',
    ORDSYS.ORDAnnotations(NULL),
    'commercial',
    'Oracle Corporation',
    '',
    'no awards',
    '90s',
    'no rating',
    30,
    NULL,
    EMPTY_CLOB(),
    NULL,
    ORDSYS.ORDVIDEO('Oracle Commercial 1 Video Clip',
    ORDSYS.ORDSource(EMPTY_BLOB(),NULL,NULL,NULL,NULL),
    'QuickTime File Format',
    'video/quicktime',
    EMPTY_CLOB(),
    160, 120, 72, 15, 30, 450, 'Cinepak', 256, 15000));
```

ORDAnnotations オブジェクト型およびそのメソッドについては、付録 F を参照してください。

2.3.7 videoTable テーブルへの行の挿入

例 2-35 に、videoTable テーブルへ行を挿入する方法を示します。

例 2-35 videoTable テーブルへの行の挿入

```
-- Insert a row into the VideoTable
insert into VideoTable values (11,
    30,
    NULL,
    NULL,
```

```
clipList(NULL));
```

2.3.8 clipsTable テーブルへのビデオのロード

例 2-36 に、clipsTable テーブルへビデオをロードする方法を示します。この例では、VIDDIR ディレクトリを定義する必要があります。コード中のコメントを参照してください。

例 2-36 clipsTable テーブルへのビデオのロード

```
-- Load a Video into a clip
-- Create your directory specification below
-- CREATE OR REPLACE DIRECTORY VIDDIR AS '/video/';
DECLARE
    videoObj ORDSYS.ORDVIDEO;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT C.videoSource INTO videoObj
    FROM   ClipsTable C
    WHERE  C.clipId = '11'
    FOR UPDATE;

    videoObj.setDescription('Under Pressure Video Clip');
    videoObj.setMimeType('video/quicktime');
    videoObj.setFormat('QuickTime File Format');
    videoObj.setFrameSize(160, 120);
    videoObj.setFrameResolution(72);
    videoObj.setFrameRate(15);
    videoObj.setVideoDuration(30);
    videoObj.setNumberOfFrames(450);
    videoObj.setCompressionType('Cinepak');
    videoObj.setNumberOfColors(256);
    videoObj.setSource('FILE', 'VIDDIR', 'UnderPressure.mov');
    videoObj.import(ctx);

    UPDATE ClipsTable C
    SET    C.videoSource = videoObj
    WHERE  C.clipId = '11';
    COMMIT;
END;

-- check video insertion
DECLARE
    videoObj ORDSYS.ORDVideo;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT C.videoSource INTO videoObj
```

```
FROM   ClipsTable C
WHERE  C.clipId = '11';

dbms_output.put_line('Content Length: ' ||
                      videoObj.getContentLength(ctx));
dbms_output.put_line('Content MimeType: ' ||
                      videoObj.getMimeType());
END;
```

2.3.9 クリップ・オブジェクトへの参照の、videoTable テーブル内のクリップ・リストへの挿入

例 2-37 に、クリップ・オブジェクトへの参照を、videoTable テーブル内のクリップ・リストに挿入する方法を示します。

例 2-37 クリップ・オブジェクトへの参照の、videoTable テーブル内のクリップ・リストへの挿入

```
-- Insert a reference to a ClipObject into the Clips List in the VideoTable
DECLARE
    clipRef          REF ClipObject;
    clipListInstance clipList;
BEGIN
    SELECT REF(C) into clipRef
    FROM   ClipsTable C
    where  C.clipId = '11';

    SELECT V.clips INTO clipListInstance
    FROM   VideoTable V
    WHERE  V.itemId = 11
    FOR UPDATE;

    clipListInstance.addClip(clipRef);

    UPDATE VideoTable V
    SET    V.clips = clipListInstance
    WHERE  V.itemId = 11;

    COMMIT;
END;

-- check insertion of clip ref
DECLARE
    clip          ClipObject;
    clipRef       REF ClipObject;
    clipListInstance clipList;
```



```

        clipType          clipNstType;
BEGIN
    SELECT V.clips INTO clipListInstance
    FROM   VideoTable V
    WHERE  V.itemId = 11;

    SELECT clipListInstance.clips INTO clipType FROM DUAL;
    clipRef := clipType(1);
    SELECT Deref(clipRef) INTO clip FROM DUAL;

    dbms_output.put_line('Clip Title: ' ||
                          clip.title);
END;
```

2.3.10 ビデオ・オブジェクトへの参照の、クリップへの挿入

例 2-38 に、ビデオ・オブジェクトへの参照をクリップに挿入する方法を示します。

例 2-38 ビデオ・オブジェクトへの参照の、クリップへの挿入

```

-- Insert a reference to a video object into the clip
DECLARE
    aVideoRef REF VideoObject;
BEGIN
    -- make a VideoRef an obj to use for update
    SELECT Cp.videoRef INTO aVideoRef
    FROM   ClipsTable Cp
    WHERE  Cp.clipId = '11'
    FOR UPDATE;

    -- change its value
    SELECT REF(V) INTO aVideoRef
    FROM   VideoTable V
    WHERE  V.itemId = 11;

    -- update database
    UPDATE ClipsTable C
    SET    C.videoRef = aVideoRef
    WHERE  C.clipId = '11';

    COMMIT;
END;
```

2.3.11 videoTable テーブルからのビデオ・クリップの取得

例 2-39 に、videoTable テーブルからビデオ・クリップを取得する方法、およびビデオ・クリップを BLOB として返す方法を示します。プログラム・セグメントは、次の操作を実行します。

1. retrieveVideo() メソッドを定義し、clipId を ORDVVideo BLOB としてビデオ・クリップを取得します。
2. 目的のビデオ・クリップを選択し (C.clipId = clipId の部分)、getContent メソッドを使用してそのビデオ・クリップを返します。

例 2-39 ビデオ・クリップの取得

```
FUNCTION retrieveVideo(clipId IN INTEGER)
RETURN BLOB IS
    obj ORDSYS.ORDVideo;

BEGIN
    -- Select the desired video clip from the ClipTable table.
    SELECT C.videoSource INTO obj from ClipTable C
        WHERE C.clipId = clipId;
    return obj.getContent;
END;
```

2.3.12 interMedia の拡張による新しいビデオ・データ・フォーマットのサポート

この項では、Oracle8i interMedia を拡張して新しいビデオ・データ・フォーマットをサポートする方法を説明します。

新しいビデオ・データ・フォーマットをサポートする場合、ORDPLUGINS スキーマ内の ORDX_<format>_VIDEO パッケージの必要なインタフェースを実装します。(<format> は、新しいビデオ・データ・フォーマット名を表します。) ORDX_DEFAULT_VIDEO パッケージのインタフェースの詳細は、5.3.1 項を参照してください。5.3.6 項内のパッケージ本体の例をテンプレートとして使用して、ビデオ・パッケージ本体を作成します。

その後、setFormat コール内の新規形式パラメータを適切なフォーマット値に設定して、ビデオ・オブジェクトに対し、パッケージ ORDPLUGINS.ORDX_<format>_VIDEO がプラグインとして利用可能であることを示します。

独自のフォーマット・プラグインのインストールおよび提供されるサンプル・スクリプトの実行方法の詳細は、E.3 項を参照してください。\$ORACLE_HOME/ord/vid/demo/ ディレクトリにインストールされる fplugins.sql および fpluginb.sql ファイルを参照してください。これらは、サポートしようとするフォーマット・プラグインを記述する際の指針となるデモ・プラグインです。独自のフォーマット・プラグインのインストール方法については、同じディレクトリ内の viddemo.sql ファイルを参照してください。

2.3.13 新しいオブジェクト型を使用した *interMedia* の拡張

この項では、新しいオブジェクト型を使用して Oracle8i *interMedia* を拡張する方法について説明します。

ORDSource、ORDVideo、および ORDAnnotations 型を、新たに作成する独自型のベースとして使用できます。

この簡単な例を例 2-31 および例 2-32 に示します。詳細な例と説明は、例 2-26 を参照してください。

2.3.14 オブジェクト・ビューでのビデオ型の使用

この項では、オブジェクト・ビューでビデオ型を使用する方法を説明します。ビューが仮想テーブルであるように、オブジェクト・ビューは仮想的なオブジェクト・テーブルです。

Oracle は、オブジェクト・ビューを基本的なリレーショナル・ビュー・メカニズムとして提供しています。オブジェクト・ビューを使用することにより、データベースのリレーショナルまたはオブジェクト・テーブル列に格納された、ビルトインまたはユーザー定義のいずれかのデータから仮想的なオブジェクト・テーブルを作成できます。

オブジェクト・ビューを使用すると、データベース内のデータおよびオブジェクトに対する、特殊または制限付きのアクセスを設定できます。たとえば、オブジェクト・ビューを使用して、機密性の高いデータや削除メソッドを含む属性を持たない従業員オブジェクト・テーブルのバージョンを提供できます。また、オブジェクト・ビューを使用することにより、恒久的にテーブルを変換することなくオブジェクト指向プログラミングを行えます。オブジェクト・ビューを使用して、リレーショナル表からオブジェクト・リレーショナル表へ段階的かつ透過的に変換できます。

次の非オブジェクト・ビデオ・テーブルについて考慮してみます。

```
create table flat (
  id          number,
  description  VARCHAR2(4000),
  localData   BLOB,
  srcType     VARCHAR2(4000),
  srcLocation VARCHAR2(4000),
  srcName     VARCHAR2(4000),
  upDateTime  DATE,
  local       NUMBER,
  format      VARCHAR2(31),
  mimeType    VARCHAR2(4000),
  comments    CLOB,
  width       INTEGER,
  height      INTEGER,
  frameResolution INTEGER,
  frameRate   INTEGER,
  videoDuration INTEGER,
```

```
numberOfFrames    INTEGER,  
compressionType   VARCHAR2(4000),  
numberOfColors    INTEGER,  
bitRate           INTEGER,  
videoclip         RAW(2000)  
);
```

次のようにして、フラット・テーブル上にオブジェクト・ビューを作成できます。

```
create or replace view object_video_v as  
select  
    id,  
    ordsys.ORDVideo(  
        T.description,  
        T.localData,  
        T.comments,  
        T.format,  
        T.width,  
        T.height,  
        T.frameResolution,  
        T.frameRate,  
        T.videoDuration,  
        T.numberOfFrames,  
        T.compressionType,  
        T.numberOfColors,  
        T.bitRate,  
        T.videoclip) VIDEO  
from flat T;
```

オブジェクト・ビューを使用すると、1つのリレーショナル・データまたはオブジェクト・データを複数の方法で柔軟に表示できます。このため、データベース内のデータ格納方法を変更せずに、メモリー内のオブジェクト表現をさまざまなアプリケーションに合わせる事ができます。オブジェクト・ビューの定義、使用、および更新の詳細は、『Oracle8i 概要』を参照してください。

2.4 *interMedia* の拡張による新しいデータ・ソースのサポート

この項では、Oracle8i *interMedia* を拡張して新しいデータ・ソースをサポートする方法を説明します。

新規データ・ソースをサポートするには、ORDPLUGINS スキーマ内の ORDX_<srcType>_SOURCE パッケージの必要なインタフェースをインプリメントします。(<srcType> は、新規外部ソース・タイプ名を表します。) ORDX_FILE_SOURCE および ORDX_HTTP_SOURCE パッケージのインタフェースの詳細は、6.3.1 項および 6.3.2 項を参照してください

い。提供されるパッケージ本体リストの修正例は、6.3.4 項を参照してください。その後、**setSourceInformation** コール内のソース型パラメータを適切なソース型に設定して、ビデオ・オブジェクトに対し、パッケージ `ORDPLUGINS.ORDX_<srcType>_SOURCE` がプラグインとして利用可能であることを示します。

ORDAudio 参照情報

Oracle8i *interMedia* には、ORDAudio 型に関する次の情報が含まれます。

- オブジェクト型 -- 3.1 項を参照
- メソッド -- 3.2 項を参照
- パッケージまたは PL/SQL プラグイン -- 3.3 項を参照

この章の例では、テスト・オーディオ・テーブル TAUD が作成され、データが格納されていることを前提にしています。このテーブルは、3.2.1 項で説明した SQL 文を使用して作成されました。

注意: オーディオ・データ自体を操作 (BLOB を直接修正するか、外部ソースを変更して) する場合、オブジェクト属性の同期が維持されていること、および更新時間が修正されていることを確認する必要があります。さもないと、オブジェクト属性はオーディオ・データと合致しなくなります。

ソース・プラグインに渡されて処理される、ORDSource レベルで起動されたメソッドは、最初の引数に **ctx (RAW(4000))** を取ります。これらのメソッドのいずれかを最初にコールする前に、クライアントは **ctx** 構造を割り当て、それを **NULL** に初期化してから、**openSource()** メソッドを起動します。この時点で、ソース・プラグインはこのクライアント用のコンテキストを初期化できます。処理の完了時に、クライアントは **closeSource()** メソッドを起動する必要があります。

ソース・プラグインから起動したメソッドは、最初の引数に **ctx (RAW(4000))** を取ります。

フォーマット・プラグインに渡されて処理される、ORDSource レベルで起動されるメソッドは、最初の引数に **ctx (RAW(4000))** を取ります。これらのメソッドのいずれかを最初にコールする前に、クライアントは **ctx** 構造を割り当て、**NULL** に初期化する必要があります。

注意: 現在のリリースでは、すべてのソースまたはフォーマット・プラグインが `ctx` 引数を使用するわけではありません。ただし、既に表示したコードを使用するのなら、使用するアプリケーションで現在または将来登場するソースまたはフォーマット・プラグインが動作するはずです。

3.1 オブジェクト型

Oracle8i *interMedia* は、オーディオ・データの格納と管理をサポートする `ORDAudio` オブジェクト型を記述します。

ORDAudio オブジェクト型

ORDAudio オブジェクト型は、オーディオ・データの格納と管理をサポートします。オブジェクト型は、次のように定義します。

```
CREATE OR REPLACE TYPE ORDAudio
AS OBJECT
(
    -- ATTRIBUTES
    description          VARCHAR2(4000),
    source               ORDSOURCE,
    format               VARCHAR2(31),
    mimeType             VARCHAR2(4000),
    comments             CLOB,
    -- AUDIO RELATED ATTRIBUTES
    encoding             VARCHAR2(256),
    numberOfChannels     INTEGER,
    samplingRate         INTEGER,
    sampleSize           INTEGER,
    compressionType     VARCHAR2(4000),
    audioDuration        INTEGER,

    -- METHODS
    -- Methods associated with the date attribute
    MEMBER FUNCTION getUpdateTime RETURN DATE,
    PRAGMA RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS),
    MEMBER PROCEDURE setUpdateTime(current_time DATE),
    -- Methods associated with the description attribute
    MEMBER PROCEDURE setDescription(user_description IN VARCHAR2),
    MEMBER FUNCTION getDescription RETURN VARCHAR2,
    PRAGMA RESTRICT_REFERENCES(getDescription, WNDS, WNPS, RNDS, RNPS),

    -- Methods associated with the mimeType attribute
    MEMBER PROCEDURE setMimeType(mime IN VARCHAR2),
    MEMBER FUNCTION getMimeType RETURN VARCHAR2,
    PRAGMA RESTRICT_REFERENCES(getMimeType, WNDS, WNPS, RNDS, RNPS),

    -- Methods associated with the source attribute
    MEMBER FUNCTION processSourceCommand(
                                ctx          IN OUT RAW,
                                cmd         IN VARCHAR2,
                                arguments IN VARCHAR2,
                                result      OUT RAW)
                                RETURN RAW,
```

ORDAudio オブジェクト型

```
MEMBER FUNCTION isLocal RETURN BOOLEAN,
PRAGMA RESTRICT_REFERENCES(isLocal, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE setLocal,
MEMBER PROCEDURE clearLocal,

MEMBER PROCEDURE setSource(
    source_type      IN VARCHAR2,
    source_location  IN VARCHAR2,
    source_name      IN VARCHAR2),
MEMBER FUNCTION getSource RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSource, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getSourceType RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getSourceLocation RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getSourceName RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE import(ctx IN OUT RAW),
MEMBER PROCEDURE importFrom(
    ctx          IN OUT RAW,
    source_type  IN VARCHAR2,
    source_location IN VARCHAR2,
    source_name  IN VARCHAR2),
MEMBER PROCEDURE export(
    ctx          IN OUT RAW,
    source_type  IN VARCHAR2,
    source_location IN VARCHAR2,
    source_name  IN VARCHAR2),
MEMBER FUNCTION getContentLength(ctx IN OUT RAW) RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getContentLength, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getContentInLob(
    ctx          IN OUT RAW,
    dest_lob     IN OUT NOCOPY BLOB,
    mimeType     OUT VARCHAR2,
    format       OUT VARCHAR2),

MEMBER FUNCTION getContent RETURN BLOB,
PRAGMA RESTRICT_REFERENCES(getContent, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE deleteContent,
```

```

-- Methods associated with file operations on the source
MEMBER FUNCTION openSource(userArg IN RAW, ctx OUT RAW) RETURN INTEGER,
MEMBER FUNCTION closeSource(ctx IN OUT RAW) RETURN INTEGER,
MEMBER FUNCTION trimSource(ctx      IN OUT RAW,
                           newlen  IN INTEGER) RETURN INTEGER,
MEMBER PROCEDURE readFromSource(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   OUT RAW),
MEMBER PROCEDURE writeToSource(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   IN RAW),

-- Methods associated with the comments attribute
MEMBER PROCEDURE appendToComments(amount IN BINARY_INTEGER,
    buffer IN VARCHAR2),
MEMBER PROCEDURE writeToComments(offset IN INTEGER,
    amount IN BINARY_INTEGER,
    buffer IN VARCHAR2),
MEMBER FUNCTION readFromComments(offset IN INTEGER,
    amount IN BINARY_INTEGER := 32767)
    RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(readFromComments, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION locateInComments(pattern  IN VARCHAR2,
    offset      IN INTEGER := 1,
    occurrence  IN INTEGER := 1)
    RETURN INTEGER,
MEMBER PROCEDURE trimComments(newlen IN INTEGER),
MEMBER PROCEDURE eraseFromComments(amount IN OUT NOCOPY INTEGER,
    offset IN INTEGER := 1),
MEMBER PROCEDURE deleteComments,
MEMBER PROCEDURE loadCommentsFromFile(fileobj  IN BFILE,
    amount  IN INTEGER,
    from_loc IN INTEGER := 1,
    to_loc  IN INTEGER := 1),
MEMBER PROCEDURE copyCommentsOut(dest      IN OUT NOCOPY CLOB,
    amount  IN INTEGER,
    from_loc IN INTEGER := 1,
    to_loc  IN INTEGER := 1),
MEMBER FUNCTION compareComments(
    compare_with_lob  IN CLOB,
    amount            IN INTEGER := 4294967295,
    starting_pos_in_comment IN INTEGER := 1,

```

```
        starting_pos_in_compare IN INTEGER := 1)
    RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(compareComments, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getCommentLength RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getCommentLength, WNDS, WNPS, RNDS, RNPS),

-- Methods associated with audio attributes accessors
MEMBER PROCEDURE setFormat(knownformat IN VARCHAR2),
MEMBER FUNCTION getFormat RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getFormat(ctx IN OUT RAW) RETURN VARCHAR2,

MEMBER PROCEDURE setEncoding(knownEncoding IN VARCHAR2),
MEMBER FUNCTION getEncoding RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getEncoding, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getEncoding(ctx IN OUT RAW) RETURN VARCHAR2,

MEMBER PROCEDURE setNumberOfChannels(knownNumberOfChannels IN INTEGER),
MEMBER FUNCTION getNumberOfChannels RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getNumberOfChannels, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getNumberOfChannels(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setSamplingRate(knownSamplingRate IN INTEGER),
MEMBER FUNCTION getSamplingRate RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getSamplingRate, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getSamplingRate(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setSampleSize(knownSampleSize IN INTEGER),
MEMBER FUNCTION getSampleSize RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getSampleSize, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getSampleSize(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setCompressionType(knownCompressionType IN VARCHAR2),
MEMBER FUNCTION getCompressionType RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2,

MEMBER PROCEDURE setAudioDuration(knownAudioDuration IN INTEGER),
MEMBER FUNCTION getAudioDuration RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getAudioDuration, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getAudioDuration(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setKnownAttributes(
    knownFormat IN VARCHAR2,
    knownEncoding IN VARCHAR2,
    knownNumberOfChannels IN INTEGER,
```

```

        knownSamplingRate IN INTEGER,
        knownSampleSize IN INTEGER,
        knownCompressionType IN VARCHAR2,
        knownAudioDuration IN INTEGER),

-- Methods associated with setting all the properties
MEMBER PROCEDURE setProperties(ctx IN OUT RAW),
MEMBER FUNCTION checkProperties(ctx IN OUT RAW) RETURN BOOLEAN,

MEMBER FUNCTION getAttribute(
        ctx IN OUT RAW,
        name IN VARCHAR2) RETURN VARCHAR2,

MEMBER PROCEDURE getAllAttributes(
        ctx IN OUT RAW,
        attributes IN OUT NOCOPY CLOB),

-- Methods associated with audio processing
MEMBER FUNCTION processAudioCommand(
        ctx IN OUT RAW,
        cmd IN VARCHAR2,
        arguments IN VARCHAR2,
        result OUT RAW)
        RETURN RAW
);

```

パラメータは次のとおりです。

- **description:** オーディオ・オブジェクトの説明
- **source:** オーディオ・データが格納される **ORDSource**
- **format:** オーディオ・データの格納形式
- **mimeType:** MIME タイプ情報
- **comments:** オーディオ・オブジェクトのコメント情報
- **encoding:** オーディオ・データのエンコーディング・タイプ
- **numberOfChannels:** オーディオ・データのオーディオ・チャンネル数
- **samplingRate:** オーディオ・データ録音時のサンプリング・レート（秒単位）
- **sampleSize:** データ内のオーディオのサンプル幅またはサンプル数
- **compressionType:** オーディオ・データの圧縮タイプ
- **audioDuration:** 格納されたオーディオ・データの総再生時間

3.2 メソッド

この項では、オーディオ・データの操作で使用する **Oracle8i *interMedia*** のメソッドに関する参照情報を提供します。これらのメソッドは、次のようにグループ化されています。

updateTime 属性と関連付けられた ORDAudio メソッド

- **getUpdateTime**: オーディオ・オブジェクトの最終更新時刻を返します。
- **setUpdateTime()**: オーディオ・オブジェクトの更新時刻を設定します。

説明属性と関連付けられた ORDAudio メソッド

- **setDescription()**: オーディオ・データの説明を設定します。
- **getDescription**: オーディオ・データの説明を返します。

mimeType 属性と関連付けられた ORDAudio メソッド

- **setMimeType()**: 格納されるオーディオ・データの MIME タイプを設定します。
- **getMimeType**: 格納されるオーディオ・データの MIME タイプを返します。

ソース属性と関連付けられた ORDAudio メソッド

- **processSourceCommand()**: ソース・プラグインにコマンドおよび関連する引数を送信します。
- **isLocal**: データが BLOB 内でローカルに格納される場合は **TRUE** を、データが外部に格納される場合は **FALSE** を返します。
- **setLocal**: データが BLOB 内でローカルに格納されることを示すフラグを設定します。
- **clearLocal**: フラグを消去して、データが外部に格納されることを示します。
- **setSource()**: ソース情報にオーディオ・データの格納位置を設定します。
- **getSource**: URL として書式化された外部データ・ソースの完全な情報を含む書式化された文字列を返します。
- **getSourceType**: オーディオ・データの外部ソース・タイプを返します。
- **getSourceLocation**: オーディオ・データの外部ソースの位置を返します。
- **getSourceName**: オーディオ・データの外部ソース名を返します。
- **import()**: ローカル属性の値を "1" (ローカルおよびタイムスタンプの更新を意味する) に設定して、データを、外部データ・ソース (**setSourceInformation()** のコールにより指定される) から Oracle データベース内のローカル・ソース (**localData**) に転送します。

- **importFrom():** ローカル属性の値を "1" (ローカルおよびタイムスタンプの更新を意味する) に設定して、データを、指定された外部データ・ソース (ソース・タイプ、位置、名前) から Oracle データベース内のローカル・ソース (localData) に転送します。
- **export():** ローカル属性の値を "0" (外部、およびソース内のソース情報格納を意味する) に設定して、データを、Oracle データベース内のローカル・ソース (localData) から指定された外部データ・ソースに転送します。

注意: *interMedia* がネイティブにサポートするソースは書込み不可であるため、エクスポート・メソッドはサポートされません。ユーザー定義ソースは、エクスポート・メソッドをサポート可能です。

- **getContentLength():** データ・ソースの長さを (バイト数で) 返します。
- **getContentInLob():** 内容をテンポラリ LOB に返します。
- **getContent:** 内容をローカルに格納するために使用する BLOB へのハンドルを返します。
- **deleteContent:** ローカル BLOB の内容を削除します。

ファイル操作と関連付けられた ORDAudio メソッド

- **openSource():** データ・ソースまたは BLOB をオープンします。
- **closeSource():** データ・ソースまたは BLOB をクローズします。
- **trimSource():** データ・ソースまたは BLOB を切り捨てます。
- **readFromSource():** 開始位置から *n* バイトのバッファをソースから読み込みます。
- **writeToSource():** 開始位置から *n* バイトのバッファをソースに書き込みます。

コメント属性と関連付けられた ORDAudio メソッド

- **appendToComments():** 指定されたバッファおよび量のコメント・データをオーディオ・データ・コメントの末尾に追加します。
- **writeToComments():** 指定されたバッファおよび量のコメント・データを、指定されたオフセット開始位置でオーディオ・データ・コメントに追加します。
- **readFromComments():** 指定された量のコメント・データを、オーディオ・データ・コメントの指定されたオフセット開始位置から読み込みます。
- **locateInComments():** 指定された文字パターンをオーディオ・データ・コメント内で検索し、その位置を特定します。
- **trimComments():** オーディオ・データ・コメントを指定された長さに切り捨てます。
- **eraseFromComments():** 指定された量のコメント・データを、オーディオ・データ・コメントの指定されたオフセット開始位置から消去します。

- **deleteComments**: オーディオ・データ・コメントを削除します。
- **loadCommentsFromFile()**: コメントを、指定された **BFILE** からオーディオ・データ・コメント内にロードします。
- **copyCommentsOut()**: オーディオ・データ・コメントを指定された **Character LOB** (**CLOB**) にコピーします。
- **compareComments()**: オーディオ・データ・コメントを、指定された別のオーディオ・データ **CLOB** のコメントと比較します。
- **getCommentLength**: オーディオ・データ・コメントの長さを返します。

オーディオ属性アクセサと関連付けられた **ORDAudio** メソッド

- **setFormat()**: オーディオ・データ・フォーマットのオブジェクト属性値を設定します。
- **getFormat**: オーディオ・データの格納形式のオブジェクト属性値を返します。
- **getFormat()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際の形式を読み込みます。
- **setEncoding()**: オーディオ・データのエンコーディング・タイプのオブジェクト属性値を設定します。
- **getEncoding**: オーディオ・データのエンコーディング・タイプのオブジェクト属性値を返します。
- **getEncoding()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のエンコーディング・タイプを読み込みます。
- **setNumberOfChannels()**: オーディオ・データのオーディオ・チャンネル数のオブジェクト属性値を設定します。
- **getNumberOfChannels**: オーディオ・データのオーディオ・チャンネル数のオブジェクト属性値を返します。
- **getNumberOfChannels()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のチャンネル数を読み込みます。
- **setSamplingRate()**: オーディオ・データのサンプリング・レートのオブジェクト属性値を設定します。
- **getSamplingRate**: オーディオ・データの録音時のサンプリング・レート（秒ごとのサンプル）のオブジェクト属性値を返します。
- **getSamplingRate()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のサンプリング・レートを読み込みます。
- **setSampleSize()**: データ内のオーディオ・サンプル幅またはサンプル数のオブジェクト属性値を設定します。

- **getSampleSize**: データ内のオーディオ・サンプル幅またはサンプル数のオブジェクト属性値を返します。
- **getSampleSize()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のサンプル幅またはサンプル数を読み込みます。
- **setCompressionType()**: オーディオ・データの圧縮タイプのオブジェクト属性値を設定します。
- **getCompressionType**: オーディオ・データの圧縮タイプのオブジェクト属性値を返します。
- **getCompressionType**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際の圧縮タイプを読み込みます。このメソッドは、ユーザー定義フォーマット・プラグインでのみ使用可能です。
- **setAudioDuration()**: オーディオ・データの再生に要する合計時間のオブジェクト属性値を設定します。
- **getAudioDuration**: オーディオ・データの再生に要する合計時間のオブジェクト属性値を返します。
- **getAudioDuration()**: フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際の再生時間を読み込みます。このメソッドは、ユーザー定義フォーマット・プラグインでのみ使用可能です。
- **setKnownAttributes()**: オーディオ・データの形式、エンコーディング・タイプ、チャンネル数、サンプリング・レート、サンプル・サイズ、圧縮タイプ、および再生時間を含む、既知のオーディオ属性を設定します。パラメータは、このコール内で渡されます。
- **setProperties()**: オーディオ・データを読み込んでオブジェクト属性値を取得し、次に取得した属性値をオブジェクト内に格納します。**ORDAudio** が理解する既知の属性に関しては、このメソッドはそのプロパティを設定します。既知の属性には、オーディオ・データの形式、エンコーディング・タイプ、データ型、チャンネル数、サンプリング・レート、およびサンプル・サイズが含まれます。
- **checkProperties()**: フォーマット・プラグインをコールしてプロパティ（オーディオ・データの形式、エンコーディング・タイプ、チャンネル数、サンプリング・レート、およびサンプル・サイズを含む）をチェックし、オブジェクト属性に格納されたプロパティがオーディオ・データに格納されたプロパティと合致する場合はブール値 **TRUE** を返します。
- **getAttribute()**: 要求された属性値を返します。このメソッドは、ユーザー定義フォーマット・プラグインでのみ使用可能です。
- **getAllAttributes()**: クライアント・アクセスを容易にする書式化された文字列を返します。ネイティブにサポートされた形式では、文字列には、カンマ(',') で区切られたオーディオ・データ属性リスト (**FileFormat**、**Encoding**、**NumberOfChannels**、**SamplingRate**、および **SampleSize**) が含まれます。異なるフォーマット・プラグイン

を使用すると、このコール内の任意の形式でデータを返すことができます。ユーザー定義形式の場合、文字列はフォーマット・プラグインにより定義されます。

オーディオ・データ処理と関連付けられた ORDAudio メソッド

- **processAudioCommand()**: コマンドおよび関連する引数をフォーマット・プラグインに送信して処理します。このメソッドは、ユーザー定義フォーマット・プラグインでのみ使用可能です。

オブジェクト型およびメソッドの詳細は、『Oracle8i 概要』を参照してください。

3.2.1 例題テーブル定義

この章で説明するメソッドは、テスト用オーディオ・テーブル **TAUD** に基づく例題で使用します。3.2.2 項から 3.2.9 項までを読む際に、次の **TAUD** テーブル定義を参照してください。

TAUD テーブル定義

```
CREATE TABLE TAUD (n NUMBER, aud ORDSYS.ORDAUDIO)
storage (initial 100K next 100K pctincrease 0);

INSERT INTO TAUD VALUES (1, ORDSYS.ORDAudio(
    NULL,
    ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL),
    NULL,
    NULL,
    EMPTY_CLOB(), NULL, 0, 0, 0, NULL, 0)
);
INSERT INTO TAUD VALUES (2, ORDSYS.ORDAudio(
    NULL,
    ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL),
    NULL,
    NULL,
    EMPTY_CLOB(), NULL, 0, 0, 0, NULL, 0)
);

CREATE TABLE TS (n NUMBER, s ORDSYS.ORDSOURCE);
INSERT INTO TS VALUES (1, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL));
INSERT INTO TS VALUES (2, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL));
INSERT INTO TS VALUES (3, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL));
INSERT INTO TS VALUES (4, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL));
```

3.2.2 updateTime 属性と関連付けられた ORDAudio メソッド

この項では、updateTime 属性と関連付けられた ORDAudio メソッドの参照情報を提供します。

getUpdateTime メソッド

形式

```
getUpdateTime RETURN DATE;
```

説明

オブジェクトの最終更新時を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

オーディオ・データの更新時を取得します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N = 1 ;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getUpdateTime, 'MM-DD-YYYY HH24:MI:SS'));
  EXCEPTION
    WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
```

/

setUpdateTime() メソッド

形式

```
setUpdateTime(current_time DATE);
```

説明

オーディオ・データの最終更新時を設定します。オーディオ・データを変更した場合は、常にこのメソッドを使用します。メソッド `setDescription()`、`setMimeType()`、`setSource()`、`import()`、`importFrom()`、`export()`、`deleteContent`、およびすべての設定オーディオ・アクセサは、このメソッドを暗黙的にコールします。

パラメータ

current_time

格納されるタイムスタンプ。デフォルトは `SYSDATE` です。

使用方法

オーディオ・データを変更した場合は、常にこのメソッドを起動する必要があります。

プラグマ

なし

例外

なし

例

オーディオ・データの更新時を設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N = 1;
  obj.setUpdateTime(SYSDATE);
  UPDATE TAUD SET aud=obj WHERE N = 1;
  COMMIT;
END;
/
```

3.2.3 説明属性と関連付けられた ORDAudio メソッド

この項では、説明属性と関連付けられた ORDAudio メソッドの参照情報を提供します。

setDescription() メソッド

形式

```
setDescription (user_description IN VARCHAR2);
```

説明

オーディオ・データの説明を設定します。

パラメータ

user_description

オーディオ・データの説明

使用方法

各オーディオ・オブジェクトが、クライアント・アプリケーションの機能支援用の説明を必要とする場合があります。たとえば、Web ベースのクライアントでは、オーディオに関する説明をリスト表示し、ユーザーがいずれかを選択してオーディオ・データにアクセス可能にできます。

Web アクセス・コンポーネントおよび *Oracle8i interMedia* と共に提供されるクライアント・コンポーネントは、この説明属性を利用してユーザーにオーディオ・データを提供します。

このメソッドをコールすると、`setUpdateTime()` メソッドが暗黙的にコールされます。

プラグマ

なし

例外

なし

例

オーディオ・データの説明属性を設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('writing title');
  setDescription('-----');
```

setDescription() メソッド

```
obj.setDescription('audio1.wav');  
DBMS_OUTPUT.PUT_LINE(obj.getDescription);  
UPDATE TAUD SET aud=obj WHERE N=1;  
COMMIT;  
END;  
/
```

getDescription メソッド

形式

```
getDescription RETURN VARCHAR2;
```

説明

オーディオ・データの説明を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getDescription, WNDS, WNPS, RNDS, RNPS)

例外

説明が設定されていない状態で `getDescription()` メソッドをコールする場合、`DESCRIPTION_IS_NOT_SET` 例外が呼び出されます。

例

3-17 ページの「`setDescription()` メソッド」の例を参照してください。

3.2.4 mimeType 属性と関連付けられた ORDAudio メソッド

この項では、mimeType 属性と関連付けられた ORDAudio メソッドの参照情報を提供します。

setMimeType() メソッド

形式

```
setMimeType(mime IN VARCHAR2);
```

説明

オーディオ・データの MIME タイプの設定を可能にします。

パラメータ

mime
MIME タイプ

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

ソースがファイルまたは BLOB の場合、**setMimeType()** メソッドをコールして MIME タイプを設定します。

HTTP ソースに対するインポート時に、MIME タイプが HTTP ヘッダーから抽出されます。

プラグマ

なし

例外

setMimeType() メソッドをコールし、かつ **contentType** の値が NULL の場合、**INVALID_MIME_TYPE** 例外が呼び出されます。

例

格納されるオーディオ・データの MIME タイプを設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('writing mimetype');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.setMimeType('audio/basic');
  DBMS_OUTPUT.PUT_LINE(obj.getMimeType);
```

setMimeType() メソッド

```
UPDATE TAUD SET aud=obj WHERE N=1;  
COMMIT;  
END;  
/
```

getMimeType メソッド

形式

```
getMimeType RETURN VARCHAR2;
```

説明

オーディオ・データの MIME タイプを返します。

パラメータ

なし

使用方法

ソースが HTTP サーバーの場合、HTTP ヘッダー情報が MIME タイプ情報から読み込まれます。ソースがファイルまたは BLOB の場合、`setMimeType()` メソッドをコールし、MIME タイプを設定する必要があります。

プラグマ

```
Pragma RESTRICT_REFERENCES(getMimeType, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

3-21 ページの「`setMimeType()` メソッド」の例を参照してください。

3.2.5 ソース属性と関連付けられた ORDAudio メソッド

この項では、ソース属性と関連付けられた ORDAudio メソッドの参照情報を提供します。

processSourceCommand() メソッド

形式

```
processSourceCommand(  
    ctx          IN OUT RAW,  
    cmd          IN VARCHAR2,  
    arguments    IN VARCHAR2,  
    result       OUT RAW)  
RETURN RAW;
```

説明

任意のコマンドおよびその引数のソース・プラグインへの送信を可能にします。このメソッドはユーザー定義のソース・プラグインでのみ利用可能です。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てする必要があります。
`openSource()` メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

cmd

ソース・プラグインにより認識される任意のコマンド

arguments

コマンドの引数

result

ソース・プラグインにより返される、このメソッドへのコール結果

使用方法

このメソッドを使用して、任意のコマンドおよびその引数をソース・プラグインに送信します。このメソッドはコマンドを解釈しません。ただ受け取って処理へ渡すだけです。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが processSourceCommand() メソッドをサポートしない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、例外 ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

いくつかのコマンドを処理します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR(4000);
  argList VARCHAR(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for UPDATE;
  -- assign command
  -- assign argList
  res := obj.processSourceCommand (ctx, command, argList, result);
  UPDATE TAUD SET aud=obj WHERE N=1 ;
  COMMIT;
  EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
      DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
      DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('EXCEPTION caught');
END;
/
```

isLocal メソッド

形式

```
isLocal RETURN BOOLEAN;
```

説明

データが BLOB 内でローカルに格納される場合は TRUE を、データが外部に格納される場合は FALSE を返します。

パラメータ

なし

使用方法

ローカル属性が 1 または NULL に設定されている場合、このメソッドは TRUE を返し、それ以外の場合は FALSE を返します。

プラグマ

```
Pragma RESTRICT_REFERENCES(getLocal, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

データがローカルに格納されているかどうかを判断します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT s INTO obj FROM TAUD WHERE N = 1 ;
  if(obj.isLocal = TRUE) then
    DBMS_OUTPUT.put_line('local is set true');
  else
    DBMS_OUTPUT.put_line('local is set false');
  end if;
END;
/
```

setLocal メソッド

形式

```
setLocal;
```

説明

データが **BLOB** 内部に格納されることを示すローカル属性を設定します。ローカル設定時には、オーディオ・メソッドはオーディオ・データを **source.localData** 属性内で検索します。

パラメータ

なし

使用方法

このメソッドは、ローカル属性を **1**（データが **localData** 内にローカルに格納されることを意味する）に設定します。

プラグマ

なし

例外

なし

例

データがローカルに格納されることを示すフラグを設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT s INTO obj FROM TAUD WHERE N = 1 FOR UPDATE;
  obj.setLocal;
  UPDATE TAUD SET s=obj WHERE N = 1;
  COMMIT;
END;
/
```

clearLocal メソッド

形式

```
clearLocal;
```

説明

ローカル・フラグをリセットして、データが外部に格納されることを示します。ローカル・フラグを消去に設定すると、オーディオ・メソッドは `srcLocation`、`srcName`、および `srcType` 属性を使用してオーディオ・データを検索します。

パラメータ

なし

使用方法

このメソッドは、ローカル属性を **0**（データが外部的に、または *Oracle8i* の外部に格納されることを示す）に設定します。

プラグマ

なし

例外

なし

例

データ用のローカル・フラグの値を消去します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT s INTO obj FROM TAUD WHERE N = 1 FOR UPDATE;
  obj.clearLocal;
  UPDATE TAUD SET s=obj WHERE N = 1;
  COMMIT;
END;
/
```

setSource() メソッド

形式

```
setSource(  
    source_type      IN VARCHAR2,  
    source_location  IN VARCHAR2,  
    source_name      IN VARCHAR2);
```

説明

オーディオ・データの外部ソース情報を設定または変更します。

パラメータ

source_type

外部データのソース・タイプ。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

source_location

外部データのソース位置。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

source_name

外部データのソース名。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

使用方法

ユーザーは、このメソッドを使用してオーディオ・データ・ソースを新規 BFILE または URL に設定できます。

このメソッドを使用する前に、ディレクトリが存在する、または作成されることを確認する必要があります。

このメソッドをコールすると、setUpdateTime() メソッドが暗黙的にコールされます。

プラグマ

なし

例外

なし

例

オーディオ・データのソースを設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('setting and getting source');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.setSource('FILE', 'AUDIODIR', 'audio.au');
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  UPDATE TAUD SET aud=obj WHERE N=1;
  COMMIT;
END;
/
```

getSource メソッド

形式

```
getSource RETURN VARCHAR2;
```

説明

オーディオ・データの外部位置情報を URL 形式で返します。

パラメータ

なし

使用方法

返される可能性のある戻り値は、次のとおりです。

- FILE://<DIR OBJECT NAME>/<FILE NAME> (ファイル・ソースの場合)
- HTTP://<URL> (HTTP ソースの場合)
- ユーザー定義ソース

プラグマ

Pragma RESTRICT_REFERENCES(getSource, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

3-31 ページの「setSource() メソッド」の例を参照してください。

getSourceType メソッド

形式

```
getSourceType RETURN VARCHAR2;
```

説明

外部オーディオ・データ・ソース・タイプを含む文字列を返します。

パラメータ

なし

使用方法

'FILE' などの外部オーディオ・データ・ソース・タイプを含む文字列 VARCHAR2 を返します。

プラグマ

Pragma RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

オーディオ・データ・ソースのソース・タイプ情報を取得します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('setting and getting source');
  DBMS_OUTPUT.PUT_LINE('-----');
  -- set source to a file
  obj.setSource('FILE', 'AUDIODIR', 'testaud.dat');
  -- get source information
  DBMS_OUTPUT.put_line(obj.getSource);
  DBMS_OUTPUT.put_line(obj.getSourceType);
  DBMS_OUTPUT.put_line(obj.getSourceLocation);
  DBMS_OUTPUT.put_line(obj.getSourceName);
  UPDATE TAUD SET aud=obj WHERE N=1;
```

getSourceType メソッド

```
        COMMIT;
    EXCEPTION
        WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
    END;
/
```

getSourceLocation メソッド

形式

```
getSourceLocation RETURN VARCHAR2;
```

説明

外部オーディオ・データ・ソース位置を示す値を含む文字列を返します。

パラメータ

なし

使用方法

このメソッドは、'BFILEDIR' などの外部オーディオ・データ位置を示す値を含む文字列 VARCHAR2 を返します。

プラグマ

```
Pragma RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS)
```

例外

srcLocation の値が NULL の場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_LOCATION 例外が呼び出されます。

例

3-33 ページの「getSourceType メソッド」の例を参照してください。

getSourceName メソッド

形式

```
getSourceName RETURN VARCHAR2;
```

説明

外部オーディオ・データ・ソース名を含む文字列を返します。

パラメータ

なし

使用方法

このメソッドは、'testaud.dat' などの外部データ・ソース名を含む文字列 **VARCHAR2** を返します。

プラグマ

Pragma RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS)

例外

srcName の値が NULL の場合、このメソッドのコールにより **ORDSourceExceptions.INCOMPLETE_SOURCE_NAME** 例外が呼び出されます。

例

3-33 ページの「getSourceType メソッド」の例を参照してください。

import() メソッド

形式

```
import (ctx IN OUT RAW);
```

説明

外部オーディオ・データ・ソースから Oracle データベース内のローカル・ソース (localData) へ、オーディオ・データを転送します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てする必要があります。
openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

import() メソッドをコールする前に、setSource() メソッドを使用して外部ソース・タイプ、位置、および名前を設定します。

このメソッドを使用する前に、ディレクトリが存在する、または作成されることを確認する必要があります。

外部オーディオ・データ・ソースからローカル・ソース (Oracle データベース内) へデータをインポートした後も、ソース情報は変更されません。(つまり、データのインポート元を指したままです。)

このメソッドを起動すると、setUpdateTime() および setLocal メソッドが暗黙的にコールされます。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

dlob の値が NULL の場合、このメソッドのコールにより ORDSourceExceptions.NULL_SOURCE 例外が呼び出されます。

import() メソッド

使用するプラグインが **import()** メソッドをサポートしない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

外部オーディオ・データ・ソースからローカル・ソースへ、オーディオ・データをインポートします。

```
DECLARE
  obj ORDSYS.ORDAudio;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('setting and getting source');
  DBMS_OUTPUT.PUT_LINE('-----');
  -- set source to a file
  obj.setSource('FILE','AUDIODIR','testaud.dat');
  -- get source information
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  -- import data
  obj.import(ctx);
  -- check size
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  DBMS_OUTPUT.PUT_LINE('deleting contents');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.deleteContent;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  UPDATE TAUD SET aud=obj WHERE N=1;
  COMMIT;
END;
/
```

importFrom() メソッド

形式

```
importFrom(ctx          IN OUT RAW,  
            source_type  IN VARCHAR2,  
            source_location IN VARCHAR2,  
            source_name   IN VARCHAR2);
```

説明

指定された外部オーディオ・データ・ソースから **Oracle** データベース内のローカル・ソース（**localData**）へ、オーディオ・データを転送します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てする必要があります。
openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

source_type

オーディオ・データのソース・タイプ

source_location

オーディオ・データのインポート元の位置

source_name

オーディオ・データの名前

使用方法

このメソッドは、ソース情報が（別個に指定されるのではなく）メソッド内で指定される点を除き、**import()** メソッドに類似しています。

このメソッドを使用する前に、ディレクトリが存在する、または作成されることを確認する必要があります。

外部オーディオ・データ・ソースからローカル・ソース（**Oracle** データベース内）へデータをインポートした後も、ソース情報は入力された値に設定されています。（つまり、データのインポート元を指したままです。）

このメソッドを起動すると、**setUpdateTime()** および **setLocal** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

dlob の値が NULL の場合、このメソッドのコールにより `ORDSourceExceptions.NULL_SOURCE` 例外が呼び出されます。

使用するソース・プラグインが `importFrom()` メソッドをサポートしない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、`ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

指定された外部データ・ソースからローカル・ソースへ、オーディオ・データをインポートします。

```
DECLARE
    obj ORDSYS.ORDAudio;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('setting and getting source');
    DBMS_OUTPUT.PUT_LINE('-----');
    -- set source to a file
    -- import data
    obj.importFrom(ctx, 'FILE', 'AUDIODIR', 'testaud.dat');
    -- check size
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
    DBMS_OUTPUT.PUT_LINE(obj.getSource);
    DBMS_OUTPUT.PUT_LINE('deleting contents');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.deleteContent;
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.GETLENGTH(obj.getContent)));
    UPDATE TAUD SET aud=obj WHERE N=1;
    COMMIT;
    EXCEPTION
        WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
```

/

export() メソッド

形式

```
export (
    ctx                IN OUT RAW,
    source_type         IN VARCHAR2,
    source_location     IN VARCHAR2,
    source_name         IN VARCHAR2);
```

説明

Oracle データベース内のローカル・ソース (localData) から外部オーディオ・データ・ソースへ、オーディオ・データを転送します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

source_type

データのエクスポート先のソース・タイプ

source_location

オーディオ・データのエクスポート先の位置

source_name

データのエクスポート先のオーディオ・オブジェクト名

使用方法

オーディオ・データのエクスポート後も、変更されるオーディオ属性はありません。また、srcType、srcLocation、およびsrcName は、入力値で更新されます。export() メソッドのコール後に、deleteContent メソッドをコールしてローカル・データの内容を削除できます。

export() メソッドに関しては、サーバーサイドのネイティブ・サポートはありません。このメソッドは、export() メソッドをサポート可能なユーザー定義ソースでのみ利用可能です。

このメソッドを起動すると、setUpdateTime() メソッドが暗黙的にコールされます。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドのコールにより
ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが export() メソッドをサポートしない場合、
ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、
ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

オーディオ・データをローカル・ソースから外部オーディオ・データ・ソースへエクスポートしてから、内容を削除します。

```
DECLARE
    obj ORDSYS.ORDAudio;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N =1;
    obj.export(ctx, 'FILE', 'AUDIODIR', 'complete.wav');
    EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.put_line('OTHER EXCEPTION caught');
END;
/
```

getLength() メソッド

形式

```
getLength(ctx IN OUT RAW) RETURN INTEGER;
```

説明

ソースに格納されたオーディオ・データ内容の長さを返します。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

使用方法

このメソッドは、すべてのソース・タイプでサポートされているわけではありません。たとえば、"HTTP" タイプ・ソースはこのメソッドをサポートしません。"HTTP" タイプ・ソースに対してこのコールをインプリメントする場合は、修正した独自の "HTTP" ソース・タイプを定義し、そのソース・タイプに対してこのメソッドをインプリメントする必要があります。

プラグマ

Pragma RESTRICT_REFERENCES(getLength, WNDS, WNPS, RNDS, RNPS)

例外

srcType の値が NULL で、データが BLOB 内にローカルに格納されない場合にこのメソッドをコールすると、例外 ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

この例外の詳細は、付録 H を参照してください。

例

3-38 ページの「import() メソッド」の例を参照してください。

getContentInLob() メソッド

形式

```
getContentInLob(  
    ctx          IN OUT RAW,  
    dest_lob     IN OUT NOCOPY BLOB,  
    mimeType     OUT VARCHAR2,  
    format       OUT VARCHAR2)
```

説明

データ・ソースから指定された **BLOB** へ、データを転送します。**BLOB** は、オブジェクト用の **BLOB** ではない、別の **BLOB** とすることが可能です。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

dest_lob
データの受信先 LOB

mimeType
データの MIME タイプ。これは値が返される場合も、そうでない場合もあります。

形式
データ形式。これは値が返される場合も、そうでない場合もあります。

使用方法

なし

プラグマ

なし

例外

srcType の値が **NULL** の場合、このメソッドのコールにより **ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

使用するソース・プラグインが **getContentInLob()** メソッドをサポートしない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

データ・ソースからデータを取得して、指定された BLOB 内へ配置します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  tempBlob BLOB;
  mimeType VARCHAR2(4000);
  format VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N = 1 ;
  if(obj.isLocal) then
    DBMS_OUTPUT.put_line('local is true');
  end if;
  DBMS_LOB.CREATETEMPORARY(tempBlob, true, 10);
  obj.getContentInLob(ctx,tempBlob, mimeType,format);
  -- process tempBlob
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.getLength(tempBlob)));
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDAudioExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

getContent メソッド

形式

```
getContent RETURN BLOB;
```

説明

ローカル BLOB 記憶域（ORDAudio オブジェクト内の BLOB）へのハンドルを返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getContent, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

クライアントがオーディオ・データへアクセスします。

```
DECLARE
    obj ORDSYS.ORDAudio;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=1 ;
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.put_line(TO_CHAR(DBMS_LOB.getLength(obj.getContent)));
    EXCEPTION
        WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

deleteContent メソッド

形式

```
deleteContent;
```

説明

現在のローカル・ソース (localData) からローカル・データを削除します。

パラメータ

なし

使用方法

ローカル・ソースから外部オーディオ・データ・ソースヘデータをエクスポートして、ローカル・ソース内にデータがなくなった後に、このメソッドをコールできます。

オブジェクトを新規オブジェクトを使用して更新する場合に、このメソッドをコールします。

プラグマ

なし

例外

なし

例

3-38 ページの「import() メソッド」の例を参照してください。

3.2.6 ファイル関連操作と関連付けられた ORDAudio メソッド

この項では、データ・ソースに対するファイル関連操作と関連付けられた **ORDAudio** メソッドの参照情報を提供します。次のメソッドを具体的に使用して、オーディオ・データを操作できます。

openSource() メソッド

形式

```
openSource(userArg IN RAW, ctx OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをオープンします。

パラメータ

userArg

ユーザー引数。これは、ユーザー定義ソース・プラグインから使用可能です。

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てる必要があります。
openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

返される INTEGER は、成功時には 0 (ゼロ)、失敗時には >0 (1 など) になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、1 は " ファイルが見つからない " を、2 は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

srcType の値が NULL で、かつデータがローカルではない場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが openSource() メソッドをサポートしない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

外部データ・ソースをオープンします。

```
DECLARE
  obj ORDSYS.ORDAudio;
  res INTEGER;
  ctx RAW(4000) :=NULL;
  userArg RAW(4000);
BEGIN
  select aud into obj from TAUD where N =1 for UPDATE;
  res := obj.openSource(userArg, ctx);
  UPDATE TAUD SET aud=obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DEMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DEMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

closeSource() メソッド

形式

```
closeSource(ctx IN OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをクローズします。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

返される INTEGER は、成功時には 0（ゼロ）、失敗時には >0（1 など）になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、1 は " ファイルが見つからない " を、2 は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

srcType の値が NULL で、かつデータがローカルではない場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが closeSource() メソッドをサポートしない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

外部データ・ソースをクローズします。

closeSource() メソッド

```
DECLARE
    obj ORDSYS.ORDAudio;
    res INTEGER;
    ctx RAW(4000) :=NULL;
BEGIN
    select aud into obj from TAUD where N =2 for UPDATE;
    res := obj.closeSource(ctx);
    UPDATE TAUD SET aud=obj WHERE N=2 ;
    COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

trimSource() メソッド

形式

```
trim(ctx      IN OUT RAW,  
      newlen IN INTEGER) RETURN INTEGER;
```

説明

データ・ソースの切捨てを実行します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てる必要があります。
`openSource()` メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

newlen

切捨て実行後の新しい長さ

使用方法

返される `INTEGER` は、成功時には `0` (ゼロ)、失敗時には `>0` (1 など) になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、`1` は " ファイルが見つからない " を、`2` は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

`srcType` の値が `NULL` で、かつデータがローカルではない場合、このメソッドのコールにより `ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION` 例外が呼び出されます。

使用するソース・プラグインが `trimSource()` メソッドをサポートしない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、`ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

外部データ・ソースの切捨てを実行します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  res INTEGER;
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for UPDATE;
  res := obj.trimSource(ctx,0);
  UPDATE TAUD SET aud=obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

readFromSource() メソッド

形式

```
readFromSource(  
    ctx          IN OUT RAW,  
    startPos IN INTEGER,  
    numBytes IN OUT INTEGER,  
    buffer      OUT RAW);
```

説明

開始位置から **n** バイトのバッファの、ソースからの読み込みを可能にします。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てする必要があります。
openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

startPos

データ・ソース内の開始位置

numBytes

データ・ソースから読み込むバイト数

buffer

データの読み込み先バッファ

使用方法

このメソッドは、HTTP ソースではサポートされません。

"HTTP" ソース・タイプの読み込みを行うには、URL ソース全体の読み込み要求を実行する必要があります。"HTTP" ソース・タイプ用の読み込みメソッドをインプリメントする場合は、HTTP ソース・タイプ用に修正されたソース・プラグイン内でこのメソッドのインプリメンテーションを独自に提供する必要があります。

プラグマ

なし

例外

`srcType` の値が `NULL` で、かつデータがローカルではない場合、このメソッドのコールにより `ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION` 例外が呼び出されます。

データがローカルでも `localData` が `NULL` の場合、このメソッドのコールにより `ORDSourceExceptions.NULL_SOURCE` 例外が呼び出されます。

使用するソース・プラグインが `readFromSource()` メソッドをサポートしない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、`ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

ソースからバッファを読み込みます。

```
DECLARE
  obj ORDSYS.ORDAudio;
  buffer RAW(4000);
  i INTEGER;
  ctx RAW(4000) :=NULL;
BEGIN
  i := 20;
  select aud into obj from TAUD where N =1 ;
  obj.readFromSource(ctx,1,i,buffer);
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
      DEMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
    WHEN OTHERS THEN
      DEMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

writeToSource() メソッド

形式

```
writeToSource(  
    ctx          IN OUT RAW,  
    startPos     IN INTEGER,  
    numBytes     IN OUT INTEGER,  
    buffer       IN RAW);
```

説明

n バイトのバッファの、ソースの開始位置への書き込みを可能にします。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。これは必ず割り当てする必要があります。
openSource() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

startPos

ソース内でバッファのコピーが開始される位置

numBytes

ソースへ書き込まれるバイト数

buffer

書き込まれるデータ・バッファ

使用方法

このメソッドは、書き込み可能ソースに対し、ランダムなバイト位置からバイト数 n の書き込みが可能である、ということを前提にしています。**FILE** および **HTTP** ソース・タイプは、書き込み可能ソースではないため、このメソッドをサポートしません。データがローカル **BLOB** に格納されるか、データへのアクセスがユーザー定義のソース・プラグイン経由で可能な場合に、このメソッドは機能します。

プラグマ

なし

例外

srcType の値が NULL で、かつデータがローカルではない場合、このメソッドのコールにより ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

データがローカルでも localData が NULL の場合、このメソッドのコールにより ORDSourceExceptions.NULL_SOURCE 例外が呼び出されます。

使用するソース・プラグインが writeToSource() メソッドをサポートしない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外呼出し時に、ソース・プラグイン内でこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

バッファをソースに書き込みます。

```
DECLARE
  obj ORDSYS.ORDAudio;
  n INTEGER := 6;
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for update;
  obj.writeToSource(ctx,1,n,UTL_RAW.CAST_TO_RAW('helloP'));
  DBMS_OUTPUT.PUT_LINE (TO_CHAR(obj.getContentLength(ctx)));
  update TAUD set aud = obj where N =1 ;
  COMMIT;
  EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
      DEMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
    WHEN OTHERS THEN
      DEMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```


3.2.7 コメント属性と関連付けられた ORDAudio メソッド

この項では、コメント属性と関連付けられた ORDAudio メソッドの参照情報を提供します。

appendToComments() メソッド

形式

```
appendToComments (amount IN BINARY_INTEGER,  
                  buffer IN VARCHAR2);
```

説明

指定されたバッファおよび量のコメント・データを、オーディオ・オブジェクトのコメント属性の末尾に追加します。

パラメータ

amount

追加するコメント・データの量

buffer

追加するコメント・データのバッファ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

コメント情報を、オーディオ・オブジェクトのコメント属性に追加します。

```
DECLARE  
  obj ORDSYS.ORDAudio;  
  i INTEGER;  
  j INTEGER;  
BEGIN  
  SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
```

```
obj.deleteComments;
obj.writeToComments(1,18,'This is a Comments');
obj.appendToComments(18,'This is a Comments');
-- check comments
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,obj.getCommentLength));
DBMS_OUTPUT.PUT_LINE(obj.locateInComments('Comments',1));
obj.trimComments(18);
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,18));
i := 8;
j := 9;
obj.eraseFromComments(i,j);
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,10));
obj.deleteComments;
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,10));
UPDATE TAUD SET aud=obj WHERE N=1;
COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

writeToComments() メソッド

形式

```
writeToComments(offset IN INTEGER,  
                 amount IN BINARY_INTEGER,  
                 buffer IN VARCHAR2);
```

説明

指定された量のコメント・バッファ・データを、指定されたオフセット開始位置でオーディオ・オブジェクトのコメント属性に書き込みます。

パラメータ

offset

コメント・データが書き込まれる、コメント内の開始オフセット位置

amount

書き込まれるコメント・データの量

buffer

書き込まれるコメント・バッファ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

readFromComments() メソッド

形式

```
readFromComments(offset IN INTEGER,  
                  amount IN BINARY_INTEGER :=32767)  
RETURN VARCHAR2;
```

説明

指定された量のコメント・データを、オーディオ・オブジェクトのコメント属性の、指定されたオフセット開始位置から読み込みます。

パラメータ

offset

コメント・データが読み込まれる、コメント内の開始オフセット位置

amount

読み込まれるコメント・データの量

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(readFromComments, WNDS, WNPS, RNDS, RNPS)

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

locateInComments() メソッド

形式

```
locateInComments(pattern    IN VARCHAR2,  
                  offset    IN INTEGER := 1,  
                  occurrence IN INTEGER := 1)  
RETURN INTEGER;
```

説明

オーディオ・オブジェクトのコメント属性内を指定されたオフセットから検索し、指定された文字データ・パターンと **n** 番目に一致した位置を示します。

パラメータ

pattern

検索するコメント・データのパターン

offset

コメント内の検索開始オフセット位置

occurrence

コメント・データのパターンがコメント内で一致する回数 (**n** 番目)

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

trimComments() メソッド

形式

```
trimComments(newlen IN INTEGER);
```

説明

オーディオ・オブジェクトのコメントを、新たに指定された長さになるように切り捨てます。

パラメータ

newlen
切捨て実行後のコメントの長さ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

eraseFromComments() メソッド

形式

```
eraseFromComments(amount IN OUT NOCOPY INTEGER,  
                  offset IN INTEGER := 1);
```

説明

オーディオ・オブジェクトのコメント属性の指定されたオフセット開始位置から、指定された量のコメント・データを消去します。

パラメータ

amount

消去するコメント・データの量

offset

コメント・データが消去される、コメント内の開始オフセット位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

deleteComments メソッド

形式

```
deleteComments;
```

説明

オーディオ・オブジェクトのコメント属性を削除します。

パラメータ

なし

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-60 ページの「appendToComments() メソッド」の例を参照してください。

loadCommentsFromFile() メソッド

形式

```
loadCommentsFromFile(fileobj IN BFILE,  
                     amount  IN INTEGER,  
                     from_loc IN INTEGER := 1,  
                     to_loc  IN INTEGER := 1);
```

説明

BFILE からオーディオ・オブジェクトのコメント属性内の指定されたオフセット開始位置へ、指定された量のコメント・データをロードします。

パラメータ

fileobj

ロード元のファイル・オブジェクト

amount

BFILE からロードするコメント・データの量

from_loc

BFILE からコメントをロードする際の、ロード元の位置

to_loc

コメントのロード先の位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

コメント情報を **BFILE** からオーディオ・データのコメント内へロードします。

```
DECLARE
    file_handle BFILE;
    obj ORDSYS.ORDAudio;
    isopen BINARY_INTEGER;
    amount INTEGER;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=1 FOR UPDATE;
    file_handle := BFILENAME(obj.getSourceLocation, obj.getSourceName);
    isopen := DBMS_LOB.FILEISOPEN(file_handle);
    IF isopen = 0 THEN
        dbms_output.put_line('File Not Open');
        DBMS_LOB.FILEOPEN(file_handle, DBMS_LOB.FILE_READONLY);
    END IF;
    dbms_output.put_line('File is now Open');
    isopen := DBMS_LOB.FILEISOPEN(file_handle);
    IF isopen <> 0 THEN
        amount := DBMS_LOB.GETLENGTH(file_handle);
    END IF;
    obj.deleteComments;
    obj.loadCommentsFromFile(file_handle, 18, 1, 18);
    DBMS_OUTPUT.put_line(TO_CHAR(obj.getCommentLength));
    UPDATE TAUD SET aud=obj WHERE N=1;
    COMMIT;
END;
/
```

copyCommentsOut() メソッド

形式

```
copyCommentsOut (dest      IN OUT NOCOPY CLOB,  
                  amount    IN INTEGER,  
                  from_loc  IN INTEGER := 1,  
                  to_loc    IN INTEGER := 1);
```

説明

指定された量のオーディオ・オブジェクトのコメント属性を、指定された CLOB にコピーします。

パラメータ

dest
コメントのコピー先

amount
コピーするコメントの量

from_loc
コメントのコピー元の位置

to_loc
コメントのコピー先の位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

オーディオ・データのコメントを指定された CLOB へコピーします。

```
DECLARE
    obj ORDSYS.ORDAudio;
    obj1 ORDSYS.ORDAudio;
BEGIN
    SELECT aud INTO obj1 FROM TAUD WHERE N=2 FOR UPDATE;
    SELECT aud INTO obj FROM TAUD WHERE N=1;
    obj.copyCommentsOut(obj1.comments,obj1.getCommentLength,1,10);
    DBMS_OUTPUT.put_line(obj1.getCommentLength);
    DBMS_OUTPUT.put_line(obj.getCommentLength);
    UPDATE TAUD SET aud=obj1 WHERE N=2;
    COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

compareComments() メソッド

形式

```
compareComments(compare_with_lob      IN CLOB,  
                 amount                IN INTEGER := 4294967295,  
                 starting_pos_in_comment IN INTEGER := 1,  
                 starting_pos_in_compare IN INTEGER := 1)  
RETURN INTEGER;
```

説明

指定された量のオーディオ・データのコメントを、提供された他の **CLOB** のコメントと比較します。

パラメータ

compare_with_lob

比較コメント

amount

比較コメントとの比較対象のオーディオ・データのコメントの量

starting_pos_in_comment

オーディオ・オブジェクトのコメント属性内の開始位置

starting_pos_in_compare

比較コメント内の開始位置

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(compareComments, WNDS, WNPS, RNDS, RNPS)

例外

このメソッドで例外が呼び出される動作は、**DBMS_LOB** ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。**DBMS_LOB** ファンクションおよびプロシージャで呼び出される例外のリストは、『**Oracle8i** パッケージ・プロシージャ リファレンス』の「**DBMS_LOB**」のパッケージの説明を参照してください。

例

オーディオ・データのコメントを別の CLOB のコメントと比較します。

```
DECLARE
    file_handle BFILE;
    obj ORDSYS.ORDAudio;
    obj1 ORDSYS.ORDAudio;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=2 ;
    SELECT aud INTO obj1 FROM TAUD WHERE N=1;
    DBMS_OUTPUT.put_line('comparison output');
    DBMS_OUTPUT.put_line(obj.compareComments(obj1.comments,obj.getCommentLength,1,18));
    EXCEPTION
        WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

getCommentLength() メソッド

形式

```
getCommentLength RETURN INTEGER;
```

説明

オーディオ・オブジェクトのコメント属性の長さを返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getCommentLength, WNDS, WNPS, RNDS, RNPS)

例外

このメソッドで例外が呼び出される動作は、DBMS_LOB ファンクションおよびプロシージャで例外が呼び出される動作に類似しています。DBMS_LOB ファンクションおよびプロシージャで呼び出される例外のリストは、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

3-73 ページの「compareComments() メソッド」の例を参照してください。

3.2.8 オーディオ属性アクセサと関連付けられた ORDAudio メソッド

この項では、オーディオ属性アクセサと関連付けられた ORDAudio メソッドの参照情報を提供します。

setFormat() メソッド

形式

```
setFormat(knownFormat IN VARCHAR2);
```

説明

オーディオ・オブジェクトのフォーマット属性を設定します。

パラメータ

knownFormat

オーディオ・オブジェクト内で設定される、既知のオーディオ・データ形式

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownFormat パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

オーディオ・データの形式を設定します。

```
DECLARE
    obj ORDSYS.ORDAudio;
BEGIN
    select aud into obj from TAUD where N =1 for update;
    obj.setFormat('AUFF');
    obj.setEncoding('MULAW');
    obj.setNumberOfChannels(1);
    obj.setSamplingRate(8);
    obj.setSampleSize(8);
    obj.setCompressionType('8BITMONOAUDIO');
    obj.setAudioDuration(16);
    DBMS_OUTPUT.put_line('format: ' || obj.getformat);
    DBMS_OUTPUT.put_line('encoding: ' || obj.getEncoding);
```

```
DBMS_OUTPUT.put_line('numberOfChannels: ' || TO_CHAR(obj.getNumberOfChannels));
DBMS_OUTPUT.put_line('samplingRate: ' || TO_CHAR(obj.getSamplingRate));
DBMS_OUTPUT.put_line('sampleSize: ' || TO_CHAR(obj.getSampleSize));
DBMS_OUTPUT.put_line('compressionType : ' || obj.getCompressionType);
DBMS_OUTPUT.put_line('audioDuration: ' || TO_CHAR(obj.getAudioDuration));
COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

getFormat メソッド

形式

```
getFormat RETURN VARCHAR2;
```

説明

オーディオ・オブジェクトのフォーマット属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS)

例外

フォーマットの値が **NULL** の場合、このメソッドのコールにより **AUDIO_FORMAT_IS_NULL** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getFormat() メソッド

形式

```
getFormat (ctx IN OUT RAW) RETURN VARCHAR2;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた形式を読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オブジェクト内で検索されたフォーマットが NULL の場合、**getFormat()** メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、フォーマットを決定します。それ以外の場合は、フォーマットで指定されたプラグインが使用されます。**AUFF**、**AIFF**、**AIFC**、および **WAVE** プラグインも提供されているため、ユーザーはこれらのプラグインも使用できます。

オーディオ・ファイル形式情報は、オーディオ・データ自体から抽出可能です。**ORDPLUGINS.ORDX_<format>_AUDIO** パッケージをインプリメントすることにより、**ORDAudio** オブジェクトが認識しないファイル形式をサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、**getFormat()** メソッドは **AUDIO_PLUGIN_EXCEPTION** 例外を呼び出します。

例

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際の形式を読み込みます。

```
DECLARE  
  obj ORDSYS.ORDAudio;
```

getFormat() メソッド

```
ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1;
  DBMS_OUTPUT.PUT_LINE('getting audio file format');
  DBMS_OUTPUT.PUT_LINE('-----');
  DBMS_OUTPUT.PUT_LINE(obj.getFormat(ctx));
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.PUT_LINE('AUDIO_PLUGIN_EXCEPTION caught');
END;
/
```

setEncoding() メソッド

形式

```
setEncoding (knownEncoding IN VARCHAR2);
```

説明

オーディオ・オブジェクトのエンコーディング属性値を設定します。

パラメータ

knownEncoding

既知のエンコーディング・タイプ

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownEncoding パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getEncoding メソッド

形式

```
getEncoding RETURN VARCHAR2;
```

説明

オーディオ・オブジェクトのエンコーディング属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getEncoding, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

3-76 ページの「setFormat() メソッド」の例を参照してください。

getEncoding() メソッド

形式

```
getEncoding(ctx IN OUT RAW) RETURN VARCHAR2;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれたエンコーディングを読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オブジェクト内で検索されたフォーマットが NULL の場合、**getEncoding()** メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、エンコーディングを決定します。それ以外の場合は、フォーマットで指定されたプラグインが使用されます。

オーディオ・エンコーディング情報は、オーディオ・データ自体から抽出可能です。**ORDPLUGIN.ORDX_<format>_AUDIO** パッケージをインプリメントすることにより、**ORDAudio** オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

この関数は、エンコーディング・タイプを判別できない場合、値 **UNKNOWN** を返します。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、**AUDIO_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のエンコーディングを読み込みます。

```
DECLARE
```

getEncoding() メソッド

```
obj ORDSYS.ORDAudio;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT aud INTO obj FROM TAUD WHERE N=1;  
  DBMS_OUTPUT.PUT_LINE('getting audio encoding');  
  DBMS_OUTPUT.PUT_LINE('-----');  
  DBMS_OUTPUT.PUT_LINE(obj.getEncoding(ctx));  
  EXCEPTION  
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.PUT_LINE('AUDIO_PLUGIN_EXCEPTION caught');  
END;  
/
```

setNumberOfChannels() メソッド

形式

```
setNumberOfChannels (knownNumberOfChannels IN INTEGER) ;
```

説明

オーディオ・オブジェクトの **numberOfChannels** 属性値を設定します。

パラメータ

knownNumberOfChannels
既知のチャンネル数

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownNumberOfChannels パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getNumberOfChannels メソッド

形式

```
getNumberOfChannels RETURN INTEGER;
```

説明

オーディオ・オブジェクトの **numberOfChannels** 属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

```
Pragma RESTRICT_REFERENCES(getNumberOfChannels, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getNumberOfChannels() メソッド

形式

```
getNumberOfChannels(ctx IN OUT RAW) RETURN INTEGER;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれたオーディオ・チャンネル数を読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オーディオ・チャンネル情報の数は、書式化されたオーディオ・データのヘッダーから利用可能です。

オブジェクト内で検索されたフォーマットが NULL の場合、getNumberOfChannels() メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、チャンネル数を決定します。それ以外の場合は、フォーマットで指定されたプラグインが使用されます。

オーディオ・チャンネル情報番号は、オーディオ・データ自体から抽出可能です。ORDPLUGINS.ORDX_<format>_AUDIO パッケージをインプリメントすることにより、ORDAudio オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

例

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた実際のオーディオ・チャンネル数を読み込みます。

getNumberOfChannels() メソッド

```
DECLARE
    obj ORDSYS.ORDAudio;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=1;
    DBMS_OUTPUT.PUT_LINE('getting audio channels');
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getNumberOfChannels(ctx)));
EXCEPTION
    WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.PUT_LINE('AUDIO_PLUGIN_EXCEPTION caught');
END;
/
```

setSamplingRate() メソッド

形式

```
setSamplingRate(knownSamplingRate IN INTEGER);
```

説明

オーディオ・オブジェクトの **samplingRate** 属性値を設定します。

パラメータ

knownSamplingRate
既知のサンプリング・レート

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownSamplingRate パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getSamplingRate メソッド

形式

```
getSamplingRate IN INTEGER;
```

説明

オーディオ・オブジェクトの **samplingRate** 属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getSamplingRate, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getSamplingRate() メソッド

形式

```
getSamplingRate(ctx IN OUT INTEGER);
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれたサンプリング・レートを読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オーディオ・サンプリング・レート情報は、書式化されたオーディオ・データのヘッダーから利用可能です。

オブジェクト内で検索されたフォーマットが NULL の場合、getSamplingRate() メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、サンプリング・レートを決定します。それ以外の場合は、フォーマットで指定されたプラグインが使用されます。

オーディオ・サンプリング・レート情報は、そのオーディオ・データが認識するサンプリング・レートに設定可能です。ORDPLUGINS.ORDX_<format>_AUDIO パッケージをインポートすることにより、ORDAudio オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

例

データベース内に格納されたオーディオ・データのサンプリング・レートを返します。

```
DECLARE
```

getSamplingRate() メソッド

```
obj ORDSYS.ORDAudio;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT aud INTO obj FROM TAUD WHERE N=1;  
  DBMS_OUTPUT.PUT_LINE('getting sampling rate');  
  DBMS_OUTPUT.PUT_LINE('-----');  
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getSamplingRate(ctx)) || ' KHz');  
EXCEPTION  
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.PUT_LINE('AUDIO_PLUGIN_EXCEPTION caught');  
END;  
/
```

setSampleSize() メソッド

形式

```
setSampleSize(knownSampleSize IN INTEGER);
```

説明

オーディオ・オブジェクトの **sampleSize** 属性値を設定します。

パラメータ

knownSampleSize
既知のサンプル・サイズ

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownSampleSize パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getSampleSize メソッド

形式

```
getSampleSize RETURN INTEGER;
```

説明

オーディオ・オブジェクトの **sampleSize** 属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

```
Pragma RESTRICT_REFERENCES(getSampleSize, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getSampleSize() メソッド

形式

```
getSampleSize(ctx IN OUT RAW) RETURN INTEGER;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれたサンプル・サイズを読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オーディオ・サンプル・サイズ情報は、書式化されたオーディオ・データのヘッダーから利用可能です。

オブジェクト内で検索されたフォーマットが NULL の場合、`getSampleSize()` メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、サンプル・サイズ・フォーマットを決定します。それ以外の場合は、フォーマットで指定されたプラグインが使用されます。

オーディオ・サンプル・サイズ情報は、オーディオ・データ自体から抽出可能です。`ORDPLUGINS.ORDX_<format>_AUDIO` パッケージをインプリメントすることにより、`ORDAudio` オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、`AUDIO_PLUGIN_EXCEPTION` 例外が呼び出されます。

例

データベース内に格納されたオーディオ・データのサンプル・サイズを返します。

```
DECLARE
```

getSampleSize() メソッド

```
obj ORDSYS.ORDAudio;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT aud INTO obj FROM TAUD WHERE N=1;  
  DBMS_OUTPUT.PUT_LINE('getting sampling size');  
  DBMS_OUTPUT.PUT_LINE('-----');  
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getSampleSize(ctx)) || ' bits');  
EXCEPTION  
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.PUT_LINE('AUDIO_PLUGIN_EXCEPTION caught');  
END;  
/
```

setCompressionType() メソッド

形式

```
setCompressionType(knownCompressionType IN VARCHAR2);
```

説明

オーディオ・オブジェクトの `compressionType` 属性値を設定します。

パラメータ

knownCompressionType
既知の圧縮タイプ

使用方法

このメソッドをコールすると、`setUpdateTime()` メソッドが暗黙的にコールされます。

プラグマ

なし

例外

`knownCompressionType` パラメータの値が `NULL` の場合にこのメソッドをコールすると、`NULL_INPUT_VALUE` 例外が呼び出されます。

例

3-76 ページの「`setFormat()` メソッド」の例を参照してください。

getCompressionType メソッド

形式

```
getCompressionType RETURN VARCHAR2;
```

説明

オーディオ・オブジェクトの `compressionType` 属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

3-76 ページの「`setFormat()` メソッド」の例を参照してください。

getCompressionType() メソッド

形式

```
getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれた圧縮タイプを読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オーディオ圧縮タイプ情報は、書式化されたオーディオ・データのヘッダーから入手可能です。

オブジェクト内で検索されたフォーマットが NULL の場合、`getCompressionType()` メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、圧縮タイプを決定します。それ以外の場合は、ユーザー定義のフォーマット・プラグインが使用されます。

オーディオ圧縮タイプ情報は、オーディオ・データ自体から抽出可能です。
`ORDPLUGINS.ORDX_<format>_AUDIO` パッケージをインプリメントすることにより、`ORDAudio` オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、`AUDIO_PLUGIN_EXCEPTION` 例外が呼び出されます。

例

データベース内に格納された、オーディオ・データ用の圧縮タイプを返します。

```
DECLARE
```

getCompressionType() メソッド

```
obj ORDSYS.ORDAudio;
ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1;
  DBMS_OUTPUT.PUT_LINE('getting compression type ');
  DBMS_OUTPUT.PUT_LINE('-----');
  DBMS_OUTPUT.PUT_LINE(obj.getCompressionType(ctx)|| ' ');
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('EXCEPTION caught');
END;
/
```

setAudioDuration() メソッド

形式

```
setAudioDuration(knownAudioDuration IN INTEGER);
```

説明

オーディオ・オブジェクトの **audioDuration** 属性値を設定します。

パラメータ

knownAudioDuration
既知のオーディオ再生時間

使用方法

このメソッドをコールすると、**setUpdateTime()** メソッドが暗黙的にコールされます。

プラグマ

なし

例外

knownAudioDuration パラメータの値が **NULL** の場合にこのメソッドをコールすると、**NULL_INPUT_VALUE** 例外が呼び出されます。

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getAudioDuration メソッド

形式

```
getAudioDuration RETURN INTEGER;
```

説明

オーディオ・オブジェクトの **audioDuration** 属性値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAudioDuration, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

3-76 ページの「**setFormat()** メソッド」の例を参照してください。

getAudioDuration() メソッド

形式

```
getAudioDuration(ctx IN OUT RAW) RETURN INTEGER;
```

説明

フォーマット・プラグインをコールして、格納されたオーディオ・データに埋め込まれたオーディオ再生時間を読み込みます。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

使用方法

オーディオ再生時間情報は、書式化されたオーディオ・データのヘッダーから利用可能です。

オブジェクト内で検索されたフォーマットが NULL の場合、`getAudioDuration()` メソッドはデフォルトのフォーマット・プラグインを使用してオーディオ・データを読み込み、オーディオ再生時間を決定します。それ以外の場合は、ユーザー定義のフォーマット・プラグインが使用されます。

オーディオ再生情報は、オーディオ・データ自体から抽出可能です。ORDPLUGINS.ORDX_<format>_AUDIO パッケージをインプリメントすることにより、ORDAudio オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

例

データベース内に格納されたオーディオ・データの再生時間を返します。

```
DECLARE
```

getAudioDuration() メソッド

```
obj ORDSYS.ORDAudio;
ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1 for update;
  DBMS_OUTPUT.PUT_LINE('getting audio duration');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.setFormat('WAVE');
  DBMS_OUTPUT.PUT_LINE(obj.getAudioDuration(ctx));
  update taud set aud = obj where n =1;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('EXCEPTION caught ');
END;
/
```

setKnownAttributes() メソッド

形式

```
setKnownAttributes(  
    knownFormat          IN VARCHAR2,  
    knownEncoding         IN VARCHAR2,  
    knownNumberOfChannels IN INTEGER,  
    knownSamplingRate     IN INTEGER,  
    knownSampleSize       IN INTEGER,  
    knownCompressionType  IN VARCHAR2,  
    knownAudioDuration    IN INTEGER);
```

説明

オーディオ・オブジェクト用の既知のオーディオ属性を設定します。

パラメータ

knownFormat

既知の形式

knownEncoding

既知のエンコーディング・タイプ

knownNumberOfChannels

既知のチャネル数

knownSamplingRate

既知のサンプリング・レート

knownSampleSize

既知のサンプル・サイズ

knownCompressionType

既知の圧縮タイプ

knownAudioDuration

既知のオーディオ再生時間

使用方法

このメソッドをコールすると、setUpdateTime() メソッドが暗黙的にコールされます。

プラグマ

なし

例外

なし

例

オーディオ・データ用の既知の属性を設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
BEGIN
  select aud into obj from TAUD where N =1 for update;
  obj.setKnownAttributes('AUFF','MULAW', 1, 8, 8, '8BITMONOAUDIO',16);
  DBMS_OUTPUT.put_line('format: ' || obj.getformat);
  DBMS_OUTPUT.put_line('encoding: ' || obj.getEncoding);
  DBMS_OUTPUT.put_line('numberOfChannels: ' || TO_CHAR(obj.getNumberOfChannels));
  DBMS_OUTPUT.put_line('samplingRate: ' || TO_CHAR(obj.getSamplingRate));
  DBMS_OUTPUT.put_line('sampleSize: ' || TO_CHAR(obj.getSampleSize));
  DBMS_OUTPUT.put_line('compressionType : ' || obj.getCompressionType);
  DBMS_OUTPUT.put_line('audioDuration: ' || TO_CHAR(obj.getAudioDuration));
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDAudioExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

setProperties() メソッド

形式

```
setProperties(ctx IN OUT RAW);
```

説明

オーディオ・データを読み込んでオブジェクト属性値を取得し、取得した属性値をオブジェクト属性内に格納します。このメソッドは、フォーマット、エンコーディング・タイプ、チャンネル数、サンプリング・レート、およびサンプル・サイズといったオーディオ・データ属性のプロパティを設定します。

パラメータ

ctx
フォーマット・プラグインのコンテキスト情報

使用方法

フォーマットが **NULL** に設定されている場合、**setProperties()** メソッドはデフォルトのフォーマット・プラグインを使用します。それ以外の場合は、フォーマットで指定されたプラグインを使用します。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、**AUDIO_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

既知のオーディオ属性のプロパティ情報を設定します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for update;
  obj.setProperties(ctx);
  --DBMS_OUTPUT.put_line('format: ' || obj.getformat);
  DBMS_OUTPUT.put_line('encoding: ' || obj.getEncoding);
```

setProperty() メソッド

```
DBMS_OUTPUT.put_line('numberOfChannels: ' || TO_CHAR(obj.getNumberOfChannels));
DBMS_OUTPUT.put_line('samplingRate: ' || TO_CHAR(obj.getSamplingRate));
DBMS_OUTPUT.put_line('sampleSize: ' || TO_CHAR(obj.getSampleSize));
update TAUD set aud = obj where N =1 ;
COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDAudioExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

checkProperties() メソッド

形式

```
checkProperties(ctx IN OUT RAW) RETURN BOOLEAN;
```

説明

格納されたオーディオ・データのプロパティをチェックします。サンプル・サイズ、サンプル・レート、チャンネル数、フォーマット、およびエンコーディング・タイプといったオーディオ属性を含みます。

パラメータ

ctx
フォーマット・プラグインのコンテキスト情報

使用方法

フォーマットが **NULL** に設定されている場合、**checkProperties()** メソッドはデフォルトのフォーマット・プラグインを使用します。それ以外の場合は、フォーマットで指定されたプラグインを使用します。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、**AUDIO_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

既知のオーディオ属性のプロパティ情報をチェックします。

```
DECLARE
  obj ORDSYS.ORDAudio;
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for update;
  if( obj.checkProperties(ctx) = TRUE ) then
    DBMS_OUTPUT.put_line('true');
  else
```

checkProperties() メソッド

```
DEMS_OUTPUT.put_line('false');
end if;
EXCEPTION
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DEMS_OUTPUT.put_line('ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION caught');
  WHEN OTHERS THEN
    DEMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

getAttribute() メソッド

形式

```
getAttribute(  
    ctx  IN OUT RAW,  
    name IN VARCHAR2)  
RETURN VARCHAR2;
```

説明

ユーザー定義形式のみに対応したオーディオ・データから、要求された属性値を返します。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

name

属性の名前

使用方法

オーディオ・データ属性は、書式化されたオーディオ・データのヘッダーから利用可能です。

フォーマットが **NULL** に設定されている場合、**getAttribute()** メソッドはデフォルトのフォーマット・プラグインを使用します。それ以外の場合は、独自のユーザー定義フォーマット・プラグインを使用します。

オーディオ・データ属性情報は、オーディオ・データ自体から抽出可能です。**ORDPLUGINS.ORDX_<format>_AUDIO** パッケージをインプリメントすることにより、**ORDAudio** オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、**AUDIO_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

データベースに格納されたオーディオ・データ用の、指定されたオーディオ属性情報を返します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  res VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT aud INTO obj FROM TAUD WHERE N=1;
  DBMS_OUTPUT.PUT_LINE('getting audio sample size');
  DBMS_OUTPUT.PUT_LINE('-----');
  res := obj.getAttribute(ctx, 'sample_size');
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('EXCEPTION caught');
END;
/
```

getAllAttributes() メソッド

形式

```
getAllAttributes(  
    ctx          IN OUT RAW,  
    attributes IN OUT NOCOPY CLOB);
```

説明

クライアント・アクセスを容易にする書式化された文字列を返します。ネイティブにサポートされた形式では、文字列には、カンマ (',') で区切られたオーディオ・データ属性リスト (FileFormat、Encoding、NumberOfChannels、SamplingRate、および SampleSize) が含まれます。ユーザー定義形式の場合、文字列はフォーマット・プラグインにより定義されます。

パラメータ

ctx
フォーマット・プラグインのコンテキスト情報

attributes
属性

使用方法

これらのオーディオ・データ属性は、書式化されたオーディオ・データのヘッダーから利用可能です。

フォーマットが NULL に設定されている場合、getAllAttributes() メソッドはデフォルトのフォーマット・プラグインを使用します。それ以外の場合は、フォーマットにより指定されたプラグインを使用します。

オーディオ・データ属性情報は、オーディオ・データ自体から抽出可能です。
ORDPLUGINS.ORDX_<format>_AUDIO パッケージをインプリメントすることにより、ORDAudio オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

例

データベース内に格納されたオーディオ・データのすべてのオーディオ属性を返します。

```
DECLARE
    obj ORDSYS.ORDAudio;
    tempLob CLOB;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT aud INTO obj FROM TAUD WHERE N=1;
    DBMS_OUTPUT.PUT_LINE('getting comma separated list of all attribs');
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_LOB.CREATETEMPORARY(tempLob, FALSE, DBMS_LOB.CALL);
    obj.getAllAttributes(ctx,tempLob);
    DBMS_OUTPUT.put_line(DBMS_LOB.substr(tempLob, DBMS_LOB.getLength(tempLob) , 1));
    EXCEPTION
    WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.PUT_LINE('ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION caught');
END;
/
```


3.2.9 オーディオ・データ処理と関連付けられた ORDAudio メソッド

この項では、オーディオ・データ処理と関連付けられた ORDAudio メソッドの参照情報を提供します。

processAudioCommand() メソッド

形式

```
processAudioCommand(  
    ctx          IN OUT RAW,  
    cmd          IN VARCHAR2,  
    arguments    IN VARCHAR2,  
    result       OUT RAW)  
  
RETURN RAW;
```

説明

コマンドおよび関連する引数をフォーマット・プラグインに送信して処理することを可能にします。このメソッドは、ユーザー定義フォーマット・プラグインでのみサポートされています。

パラメータ

ctx

フォーマット・プラグインのコンテキスト情報

cmd

フォーマット・プラグインにより認識される任意のコマンド

arguments

コマンドの引数

result

フォーマット・プラグインにより返される、この関数へのコール結果

使用方法

このメソッドを使用して、任意のオーディオ・コマンドおよびその引数をフォーマット・プラグインに送信します。このメソッドはコマンドを解釈せず、ただ受け取ってフォーマット・プラグインに渡すだけで、処理はフォーマット・プラグインで行われます。

フォーマットが **NULL** に設定されている場合、**processAudioCommand()** メソッドはデフォルトのフォーマット・プラグインを使用します。それ以外の場合は、ユーザー定義のフォーマット・プラグインを使用します。

ORDPLUGINS.ORDX_<format>_AUDIO パッケージを用意することにより、**ORDAudio** オブジェクトが認識しないフォーマットをサポートすることもできます。詳細は、[2.3.13 項](#)を参照してください。

プラグマ

なし

例外

このメソッドのコール時にオーディオ・プラグインが例外を呼び出す場合、AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

例

コマンド・セットを処理します。

```
DECLARE
  obj ORDSYS.ORDAudio;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR(4000);
  argList VARCHAR(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  select aud into obj from TAUD where N =1 for UPDATE;
  -- assign command
  -- assign argList
  res := obj.processAudioCommand (ctx, command, argList, result);
  UPDATE TAUD SET aud=obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('AUDIO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDAudioExceptions.AUDIO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('AUDIO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

3.3 パッケージまたは PL/SQL プラグイン

この項では、提供されるパッケージまたは PL/SQL プラグインの参照情報を提供します。

3.3.1 ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージ

独自の ORDPLUGINS.ORDX_<format>_AUDIO オーディオ・フォーマット・パッケージを開発する際のガイドとして、次の、提供された ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージを使用してください。

```
CREATE OR REPLACE PACKAGE ORDX_DEFAULT_AUDIO
authid current_user
AS
--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT ORDSYS.ORDAudio) RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    name IN VARCHAR2) RETURN VARCHAR2;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    attributes IN OUT NOCOPY CLOB);
--AUDIO PROCESSING METHODS
FUNCTION processCommand(ctx          IN OUT RAW,
    obj          IN OUT NOCOPY ORDSYS.ORDAudio,
    cmd          IN VARCHAR2,
    arguments    IN VARCHAR2,
    result       OUT RAW)
    RETURN RAW;

PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getEncoding, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfChannels, WNDS, WNPS, RNDS, RNPS);
```

```

PRAGMA RESTRICT_REFERENCES (getSamplingRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getSampleSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAudioDuration, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 3-1 に、ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に呼び出される例外を示します。

表 3-1 ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージでサポートされるメソッド

メソッド名	サポート・レベル
getFormat	サポートされます。ソースがローカルの場合、属性を取得してファイル形式を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getEncoding	サポートされます。ソースがローカルの場合、属性を取得してエンコーディングを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getNumberOfChannels	サポートされます。ソースがローカルの場合、属性を取得してチャンネル数を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSamplingRate	サポートされます。ソースがローカルの場合、属性を取得してサンプリング・レートを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSampleSize	サポートされます。ソースがローカルの場合、属性を取得してサンプル・サイズを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。

表 3-1 ORDPLUGINS.ORDX_DEFAULT_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
getCompressionType	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAudioDuration	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
setProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティを設定します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティを設定します。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティを設定します。
checkProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティをチェックします。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティをチェックします。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティをチェックします。
getAttribute	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAllAttributes	サポートされます。ソースがローカルの場合、属性を取得して返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
processCommand	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

3.3.2 ORDPLUGINS.ORDX_AUFF_AUDIO パッケージ

提供される ORDPLUGINS.ORDX_AUFF_AUDIO パッケージが、AUFF オーディオ・フォーマット仕様で使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_AUFF_AUDIO
authid current_user
AS
--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
```

```

FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT ORDSYS.ORDAudio)
    RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    name IN VARCHAR2) RETURN VARCHAR2;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    attributes IN OUT NOCOPY CLOB);

--AUDIO PROCESSING METHODS
FUNCTION processCommand(ctx          IN OUT RAW,
    obj          IN OUT NOCOPY ORDSYS.ORDAudio,
    cmd          IN VARCHAR2,
    arguments IN VARCHAR2,
    result      OUT RAW)
    RETURN RAW;

PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getEncoding, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfChannels, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getSamplingRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getSampleSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAudioDuration, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 3-2 に、ORDPLUGINS.ORDX_AUFF_AUDIO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に呼び出される例外を示します。

表 3-2 ORDPLUGINS.ORDX_AUFF_AUDIO パッケージでサポートされるメソッド

メソッド名	サポート・レベル
getFormat	サポートされます。ソースがローカルの場合、属性を取得してファイル形式を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getEncoding	サポートされます。ソースがローカルの場合、属性を取得してエンコーディングを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getNumberOfChannels	サポートされます。ソースがローカルの場合、属性を取得してチャンネル数を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSamplingRate	サポートされます。ソースがローカルの場合、属性を取得してサンプリング・レートを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSampleSize	サポートされます。ソースがローカルの場合、属性を取得してサンプル・サイズを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getCompressionType	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAudioDuration	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
setProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティを設定します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティを設定します。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティを設定します。

表 3-2 ORDPLUGINS.ORDX_AUFF_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
checkProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティをチェックします。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティをチェックします。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティをチェックします。
getAttribute	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAllAttributes	サポートされます。ソースがローカルの場合、属性を取得して返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
processCommand	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

3.3.3 ORDPLUGINS.ORDX_AIFF_AUDIO パッケージ

提供される ORDPLUGINS.ORDX_AIFF_AUDIO パッケージが、AIFF オーディオ・フォーマット仕様で使用されます。

```

CREATE OR REPLACE PACKAGE ORDX_AIFF_AUDIO
authid current_user
AS
--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT ORDSYS.ORDAudio) RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
obj IN ORDSYS.ORDAudio,
```

```

        name IN VARCHAR2) RETURN VARCHAR2;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDAudio,
                           attributes IN OUT NOCOPY CLOB);
--AUDIO PROCESSING METHODS
FUNCTION processCommand(ctx          IN OUT RAW,
                        obj          IN OUT NOCOPY ORDSYS.ORDAudio,
                        cmd          IN VARCHAR2,
                        arguments IN VARCHAR2,
                        result       OUT RAW)
    RETURN RAW;

PRAGMA RESTRICT_REFERENCES (getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getEncoding, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getNumberOfChannels, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getSamplingRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getSampleSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAudioDuration, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 3-3 に、ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に呼び出される例外を示します。

表 3-3 ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド

メソッド名	サポート・レベル
getFormat	サポートされます。ソースがローカルの場合、属性を取得してファイル形式を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getEncoding	サポートされます。ソースがローカルの場合、属性を取得してエンコーディングを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。

表 3-3 ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
getNumberOfChannels	サポートされます。ソースがローカルの場合、属性を取得してチャンネル数を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSamplingRate	サポートされます。ソースがローカルの場合、属性を取得してサンプリング・レートを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSampleSize	サポートされます。ソースがローカルの場合、属性を取得してサンプル・サイズを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getCompressionType	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAudioDuration	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
setProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティを設定します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティを設定します。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティを設定します。
checkProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティをチェックします。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティをチェックします。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティをチェックします。
getAttribute	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAllAttributes	サポートされます。ソースがローカルの場合、属性を取得して返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。

表 3-3 ORDPLUGINS.ORDX_AIFF_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
processCommand	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

3.3.4 ORDPLUGINS.ORDX_AIFC_AUDIO パッケージ

提供される ORDPLUGINS.ORDX_AIFC_AUDIO パッケージが、AIFC オーディオ・フォーマット仕様で使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_AIFC_AUDIO
authid current_user
AS
--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT ORDSYS.ORDAudio)
RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDAudio,
                      name IN VARCHAR2) RETURN VARCHAR2;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                          obj IN ORDSYS.ORDAudio,
                          attributes IN OUT NOCOPY CLOB);

--AUDIO PROCESSING METHODS
FUNCTION processCommand(ctx          IN OUT RAW,
                      obj            IN OUT NOCOPY ORDSYS.ORDAudio,
                      cmd            IN VARCHAR2,
                      arguments      IN VARCHAR2,
                      result         OUT RAW)
RETURN RAW;

PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RND, RNPS);
```

```

PRAGMA RESTRICT_REFERENCES (getEncoding, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getNumberOfChannels, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getSamplingRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getSampleSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES (getAudioDuration, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 3-4 に、ORDPLUGINS.ORDX_AIFC_AUDIO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に呼び出される例外を示します。

表 3-4 ORDPLUGINS.ORDX_AIFC_AUDIO パッケージでサポートされるメソッド

メソッド名	サポート・レベル
getFormat	サポートされます。ソースがローカルの場合、属性を取得してファイル形式を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getEncoding	サポートされます。ソースがローカルの場合、属性を取得してエンコーディングを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getNumberOfChannels	サポートされます。ソースがローカルの場合、属性を取得してチャンネル数を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSamplingRate	サポートされます。ソースがローカルの場合、属性を取得してサンプリング・レートを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。

表 3-4 ORDPLUGINS.ORDX_AIFC_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
getSampleSize	サポートされます。ソースがローカルの場合、属性を取得してサンプル・サイズを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getCompressionType	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAudioDuration	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
setProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティを設定します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティを設定します。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティを設定します。
checkProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティをチェックします。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティをチェックします。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティをチェックします。
getAttribute	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAllAttributes	サポートされます。ソースがローカルの場合、属性を取得して返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
processCommand	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

3.3.5 ORDPLUGINS.ORDX_WAVE_AUDIO パッケージ

提供される ORDPLUGINS.ORDX_WAVE_AUDIO パッケージが、WAVE オーディオ・フォーマット仕様で使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_WAVE_AUDIO
authid current_user
AS
--AUDIO ATTRIBUTES ACCESSORS
```

```

FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
    RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT ORDSYS.ORDAudio)
    RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    name IN VARCHAR2) RETURN VARCHAR2;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    attributes IN OUT NOCOPY CLOB);

--AUDIO PROCESSING METHODS
FUNCTION processCommand(ctx          IN OUT RAW,
    obj          IN OUT NOCOPY ORDSYS.ORDAudio,
    cmd          IN VARCHAR2,
    arguments IN VARCHAR2,
    result      OUT RAW)
    RETURN RAW;

PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getEncoding, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfChannels, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getSamplingRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getSampleSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAudioDuration, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 3-5 に、ORDPLUGINS.ORDX_WAVE_AUDIO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に呼び出される例外を示します。

表 3-5 ORDPLUGINS.ORDX_WAVE_AUDIO パッケージでサポートされるメソッド

メソッド名	サポート・レベル
getFormat	サポートされます。ソースがローカルの場合、属性を取得してファイル形式を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getEncoding	サポートされます。ソースがローカルの場合、属性を取得してエンコーディングを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getNumberOfChannels	サポートされます。ソースがローカルの場合、属性を取得してチャンネル数を返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSamplingRate	サポートされます。ソースがローカルの場合、属性を取得してサンプリング・レートを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getSampleSize	サポートされます。ソースがローカルの場合、属性を取得してサンプル・サイズを返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
getCompressionType	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAudioDuration	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
setProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティを設定します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティを設定します。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティを設定します。

表 3-5 ORDPLUGINS.ORDX_WAVE_AUDIO パッケージでサポートされるメソッド (続き)

メソッド名	サポート・レベル
checkProperties	サポートされます。ソースがローカルの場合、ローカル・データを処理してプロパティをチェックします。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。ソースが BFILE の場合、BFILE を処理してプロパティをチェックします。ソースが local でも BFILE でもない場合、メディア内容をテンポラリ LOB 内に取得してデータを処理し、プロパティをチェックします。
getAttribute	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。
getAllAttributes	サポートされます。ソースがローカルの場合、属性を取得して返します。ソースが NULL の場合、ORDSYS.ORDSourceExceptions.EMPTY_SOURCE 例外が呼び出されます。それ以外の場合は、ソースが外部にあると、ORDSYS.ORDAudioExceptions.UNSUPPORTED_DATA_SOURCE 例外が呼び出されます。
processCommand	サポートされません。METHOD_NOT_SUPPORTED および AUDIO_PLUGIN_EXCEPTION 例外が呼び出されます。

3.3.6 *interMedia* の拡張による新しいオーディオ・データ・フォーマットのサポート

interMedia を拡張して、新規オーディオ・データ・フォーマットをサポートします。次の 4 つの手順で実行します。

1. 新規オーディオ・データ・フォーマットを設計します。
2. 新規オーディオ・データ・フォーマットをインプリメントして、名前 (ORDX_MY_AUDIO.SQL など) を付けます。
3. 新規 ORDX_MY_AUDIO.SQL プラグインを ORDPLUGINS スキーマにインストールします。
4. 新規プラグインに EXECUTE 権限を付与します。たとえば、ORDX_MY_AUDIO.SQL プラグインに PUBLIC への権限を付与します。

2.1.14 項で、新規オーディオ・データ・フォーマットをサポートし、インタフェースを記述するために、*interMedia* を拡張する方法を簡潔に説明します。この操作方法については、例 3-1 のパッケージ本体リストを参照してください。"--Your variables go here" と示された箇所に変数を追加し、"--Your code goes here" と示された箇所にコードを追加します。

独自のオーディオ・フォーマット・プラグインのインストールおよび提供されるサンプル・スクリプトの実行方法の詳細は、E.1 項を参照してください。

例 3-1 新規オーディオ・データ・フォーマットへのサポートを拡張する、パッケージ本体リスト

```
CREATE OR REPLACE PACKAGE BODY ORDX_MY_AUDIO
```

```
AS
--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getAudioDuration(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDAudio)
RETURN INTEGER
```

```

IS
--Your variables go here
BEGIN
--Your code goes here
END;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio)
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDAudio)
RETURN NUMBER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDAudio,
                      name IN VARCHAR2)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDAudio,
                           attributes IN OUT NOCOPY CLOB)
IS
--Your variables go here
BEGIN
--Your code goes here
END;
-- AUDIO PROCESSING METHODS
FUNCTION processCommand(
                                ctx          IN OUT RAW,
                                obj          IN OUT NOCOPY ORDSYS.ORDAudio,
                                cmd          IN VARCHAR2,
                                arguments    IN VARCHAR2,
                                result       OUT RAW)
RETURN RAW
IS
--Your variables go here
BEGIN
--Your code goes here

```

パッケージまたは PL/SQL プラグイン

```
END;  
END;  
/  
show errors;
```

ORDImage 参照情報

Oracle8i *interMedia* には **ORDImage** 型についての次の情報が格納されています。

- オブジェクト型 -- 4.1 項を参照してください。
- メソッド -- 4.2 項を参照してください。

この章の例では、テスト・イメージ表 **EMP** および **OLD_IMAGE** が作成され、データが入力済みであることを想定しています。これらの表は 4.2.1 項で説明する SQL 文によって作成されています。

注意: イメージ・データ自体を操作する (BLOB を直接変更するか外部ソースを変更する) 場合、オブジェクト属性が同期化されたまま更新時間が変更されることを確認する必要があります。そうしないと、オブジェクト属性がイメージ・データと合致しないことがあります。

処理のためにソース・プラグインに渡された **ORDSource** レベルで起動されるメソッドには、最初の引数として **ctx (RAW(4000))** があります。最初にこれらのメソッドのいずれかをコールする前に、クライアントで **ctx** 構造を割り当て、**NULL** に初期化して、**source.open()** メソッドを起動する必要があります。この時点で、このクライアントに対するソース・プラグインのコンテキストを初期化できます。処理が完了したら、クライアントから **source.close()** メソッドが起動されます。

ソース・プラグイン・コールから起動されるメソッドの最初の引数は **ctx (RAW(4000))** になります。

注意: 現行のリリースでは、すべてのソース・プラグインで **ctx** 引数が使われるわけではありませんが、上記のようにコードすることで、アプリケーションがそれ以降に追加されたすべてのソース・プラグインとともに動作するようにできます。

4.1 オブジェクト型

Oracle8i *interMedia* では、イメージ・データの格納、管理および操作をサポートする ORDIImage オブジェクト型が記述されています。

ORDImage オブジェクト型

ORDImage オブジェクト型では、イメージ・データの格納および管理がサポートされます。このオブジェクト型の定義は次のとおりです。

```
CREATE OR REPLACE TYPE ORDImage
AS OBJECT
(
  -----
  -- TYPE ATTRIBUTES
  -----
  source          ORDSource,
  height          INTEGER,
  width           INTEGER,
  contentLength   INTEGER,
  fileFormat      VARCHAR2(4000),
  contentFormat   VARCHAR2(4000),
  compressionFormat VARCHAR2(4000),
  mimeType        VARCHAR2(4000),

  -----
  -- METHOD DECLARATION
  -----

  -- Methods associated with copy operations
  MEMBER PROCEDURE copy(dest IN OUT ORDImage),

  -- Methods associated with image process operations
  MEMBER PROCEDURE process(command IN VARCHAR2),

  MEMBER PROCEDURE processCopy(command IN   VARCHAR2,
                                dest    IN OUT ORDImage),

  -- Methods associated with image property set and check operations
  MEMBER PROCEDURE setProperties,

  MEMBER PROCEDURE setProperties(description IN VARCHAR2),

  MEMBER FUNCTION checkProperties RETURN BOOLEAN,

  -- Methods associated with image attributes accessors
  MEMBER FUNCTION getHeight RETURN INTEGER,
  PRAGMA RESTRICT_REFERENCES(getHeight, WNDS, WNPS, RNDS, RNPS),

  MEMBER FUNCTION getWidth RETURN INTEGER,
```

ORDImage オブジェクト型

```
PRAGMA RESTRICT_REFERENCES (getWidth, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getContenLength RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES (getContenLength, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getFileFormat RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getFileFormat, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getContentFormat RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getContentFormat, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getCompressionFormat RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getCompressionFormat, WNDS, WNPS, RNDS, RNPS),

-- Methods associated with the local attribute
MEMBER PROCEDURE setLocal,
MEMBER PROCEDURE clearLocal,
MEMBER FUNCTION isLocal RETURN BOOLEAN,
PRAGMA RESTRICT_REFERENCES (isLocal, WNDS, WNPS, RNDS, RNPS),

-- Methods associated with the date attribute
MEMBER FUNCTION getUpdateTime RETURN DATE,
PRAGMA RESTRICT_REFERENCES (getUpdateTime, WNDS, WNPS, RNDS, RNPS),
MEMBER PROCEDURE setUpdateTime(current_time DATE),

-- Methods associated with the mimeType attribute
MEMBER FUNCTION getMimeType RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getMimeType, WNDS, WNPS, RNDS, RNPS),
MEMBER PROCEDURE setMimeType(mime IN VARCHAR2),

-- Methods associated with the source attribute
MEMBER FUNCTION getContent RETURN BLOB,
PRAGMA RESTRICT_REFERENCES (getContent, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getBFILE RETURN BFILE,
PRAGMA RESTRICT_REFERENCES (getBFILE, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE deleteContent,

MEMBER PROCEDURE setSource(source_type      IN VARCHAR2,
                           source_location IN VARCHAR2,
                           source_name     IN VARCHAR2),
MEMBER FUNCTION getSource RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getSource, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getSourceType RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (getSourceType, WNDS, WNPS, RNDS, RNPS),
```



```
MEMBER FUNCTION getSourceLocation RETURN VARCHAR2,  
PRAGMA RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS),  
  
MEMBER FUNCTION getSourceName RETURN VARCHAR2,  
PRAGMA RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS),  
  
MEMBER PROCEDURE import(ctx IN OUT RAW),  
MEMBER PROCEDURE importFrom(ctx          IN OUT RAW,  
                             source_type   IN VARCHAR2,  
                             source_location IN VARCHAR2,  
                             source_name    IN VARCHAR2),  
MEMBER PROCEDURE export(ctx          IN OUT RAW,  
                        source_type   IN VARCHAR2,  
                        source_location IN VARCHAR2,  
                        source_name    IN VARCHAR2),  
  
-- Methods associated with image migration  
MEMBER PROCEDURE migrateFromORDImgB(old_object ORDImgB),  
MEMBER PROCEDURE migrateFromORDImgF(old_object ORDImgF)  
);
```

パラメータは次のとおりです。

- **source** : 格納したイメージ・データのソース
- **height** : イメージのピクセル単位の高さ
- **width** : イメージのピクセル単位の幅
- **contentLength** : ディスク上のイメージ・ファイルのバイト単位のサイズ
- **fileFormat** : イメージ・データを格納するファイル・タイプまたは形式 (TIFF、JIFF など)
- **contentFormat** : イメージの種類 (モノクロ、8 ビット・グレースケールなど)
- **compressionFormat** : イメージ・データに使われる圧縮アルゴリズム
- **contentType** : MIME タイプ情報

4.2 メソッド

この項では、イメージ・データ操作に使われる Oracle8i *interMedia* メソッドの参照情報を提供します。これらのメソッドを次のように分類して説明します。

copy 操作に関連する ORDImage メソッド

- `copy()` : 別の `ORDImage` にイメージのコピーを作成します。

process 操作に関連する ORDImage メソッド

- `process()` : BLOB に格納されたイメージのインプレース・イメージ処理を実行します。
- `processCopy()` : イメージを別の `ORDImage` BLOB データ型にコピー中にイメージ処理を実行します。

プロパティの set および check 操作に関連する ORDImage メソッド

- `setProperties` : システム固有のイメージ・フォーマットのイメージの属性フィールドにデータを入力します。
- `setProperties()` : イメージの属性フィールドにデータを入力し、外部イメージ・フォーマットの説明パラメータを挿入します。外部イメージについての説明は、「外部イメージの `setProperties()` メソッド」を参照してください。
- `checkProperties` : 格納されたイメージ属性が実際のイメージと一致することを確認します。

イメージ属性に関連する ORDImage メソッド

- `getHeight` : イメージの高さをピクセル単位で戻します。
- `getWidth` : イメージの幅をピクセル単位で戻します。
- `getContentLength` : イメージのサイズをバイト単位で戻します。
- `getFormat` : イメージのファイル・タイプを戻します。
- `getContentFormat` : イメージの形式を戻します。
- `getCompressionFormat` : イメージに使われた圧縮の種類を戻します。

ローカル属性に関連する ORDImage メソッド

- `setLocal` : データが BLOB にローカルに格納されていることを示すフラグを設定します。
- `clearLocal` : フラグを消去して、データが外部的に格納されていることを示します。
- `isLocal` : データが BLOB にローカルに格納されている場合は `TRUE` を、外部的にデータが格納されている場合は `FALSE` を戻します。

日付属性に関連する ORDIImage メソッド

- `getUpdateTime` : イメージ・オブジェクトを最後に更新した時刻を返します。
- `setUpdateTime()` : イメージ・オブジェクトの更新時刻を設定します。

contentType 属性に関連する ORDIImage メソッド

- `getMimeType` : 格納したイメージ・データの MIME タイプを返します。
- `setMimeType()` : 格納したイメージ・データに MIME タイプを設定します。

ソース属性に関連する ORDIImage メソッド

- `getContent` : ローカル・データのコンテンツを返します。
- `getBFILE` : 外部コンテンツを BFILE として返します。
- `deleteContent` : ローカル・データのコンテンツを削除します。
- `setSource()` : 外部イメージ・データの格納されている場所にソース情報を設定します。
- `getSource` : URL 形式の外部データ・ソースに関する完全な情報を含む文字列を返します。
- `getSourceType` : イメージ・データの外部ソース・タイプを返します。
- `getSourceLocation` : イメージ・データの外部ソースの場所を返します。
- `getSourceName` : イメージ・データの外部ソース名を返します。
- `import()` : (`setSourceInformation()` をコールして指定される) 外部データ・ソースから Oracle データベース内のローカル・ソース (`localData`) にデータを転送し、ローカル属性の値にローカルであることを示す "1" を設定して、タイムスタンプおよびイメージ属性を更新します。
- `importFrom()` : 指定した外部データ・ソースのデータ (ソース・タイプ、位置、名前) を Oracle データベース内のローカル・ソース (`localData`) に転送し、ローカル属性の値にローカルであることを示す "1" を設定して、タイムスタンプおよびイメージ属性を更新します。
- `export()` : Oracle データベース内のローカル・ソース (`localData`) から指定した外部データ・ソースにデータを転送し、ローカル属性の値に外部であることを示す "0" を設定します。属性はすべて変更しないでそのままにしておきます。

注意 : *interMedia* によってサポートされるシステム固有のソースは書込み不可のため、`export` メソッドはサポートされません。ユーザー定義ソースでは、`export` メソッドがサポートされます。

イメージ移行に関連する ORDIImage メソッド

- `migrateFromORDImgB` : 古い `ORDImgB` イメージを `ORDImage` オブジェクトにコピーします。
- `migrateFromORDImgF` : 古い `ORDImgF` イメージを `ORDImage` オブジェクトにコピーします。

4.2.1 表定義の例

この章で説明するメソッドでは、テスト・イメージ表 `EMP` を基にした例を示します。4.2.2 項から 4.2.9 項の例を読むときは、次の `EMP` 表定義を参考にしてください。

EMP 表定義

```
CREATE TABLE emp (  
    ename VARCHAR2(50),  
    salary NUMBER,  
    job VARCHAR2(50),  
    department INTEGER,  
    photo ORDSYS.ORDImage,  
    large_photo ORDSYS.ORDImage);  
DECLARE  
    Image ORDSYS.ORDImage;  
BEGIN  
    INSERT INTO emp VALUES ('John Doe', 24000, 'Technical Writer', 123,  
        ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob(), 'file', 'ORDIMGDIR',  
            'jdoe.gif', SYSDATE, 0),  
            NULL, NULL, NULL, NULL, NULL, NULL, NULL),  
        ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob(), 'file', 'ORDIMGDIR',  
            'jdoe.gif', SYSDATE, 0),  
            NULL, NULL, NULL, NULL, NULL, NULL, NULL));  
    SELECT large_photo INTO Image FROM emp  
        WHERE ename = 'John Doe' FOR UPDATE;  
    Image.setProperties;  
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';  
    COMMIT;  
END;  
/
```

4.2.10 項の例を読むときは、次の `EMP` および `OLD_IMAGES` の表定義を参考にしてください。

EMP および OLD_IMAGES 表定義

```
CREATE TABLE emp (  
    ename VARCHAR2(50),  
    salary NUMBER,
```

```
job VARCHAR2(50),
department INTEGER,
large_photo ORDSYS.ORDImage);
CREATE TABLE old_images (
  id NUMBER,
  imageb ORDSYS.ORDIMGB,
  imagef ORDSYS.ORDIMGF);
DECLARE
  blobimage ORDSYS.ORDIMGB;
  bfileimage ORDSYS.ORDIMGF;
BEGIN
  INSERT INTO old_images values
    (1, ORDSYS.ORDIMGB(empty_blob()), NULL, NULL, NULL, NULL, NULL, NULL),
    (ORDSYS.ORDIMGF(bfilename('ORDIMGDIR', 'jdoe.gif'),
      NULL, NULL, NULL, NULL, NULL, NULL));
  SELECT imageb, imagef INTO blobimage, bfileimage
    FROM old_images WHERE id = 1 FOR UPDATE;
  bfileimage.copyContent(blobimage.content);
  blobimage.setProperties;
  bfileimage.setProperties;
  UPDATE old_images SET imageb=blobimage, imagef=bfileimage WHERE id = 1;
  INSERT INTO emp values
    ('John Doe', 24000, 'Technical Writer', 123,
      ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob()), NULL, NULL, NULL,
        SYSDATE, 1),
      NULL, NULL, NULL, NULL, NULL, NULL, NULL);
  COMMIT;
end;
/
```

4.2.2 Copy 操作に関連する ORDImage メソッド

この項では、copy 操作に関連する ORDImage メソッドの参照情報を提供します。

copy() メソッド

形式

```
copy(dest IN OUT ORDImage);
```

説明

イメージを変更しないでコピーします。

パラメータ

dest
新しいイメージのコピー先

使用方法

このメソッドはソースおよびイメージ属性もすべてを含め、指定したローカルのコピー先イメージにイメージ・データをそのままコピーします。

データがソースにローカルに格納されている場合、このメソッドをコールすると宛先の **source.localData** 属性に **BLOB** がコピーされます。

このメソッドをコールすると、ソース・データがローカルに格納されているかどうかにかかわらず、外部ソース情報が新しいイメージの外部ソース情報にコピーされます。

このメソッドをコールすると、暗黙的に **setUpdateTime** メソッドが宛先オブジェクトにコールされてタイムスタンプ情報が更新されます。

プラグマ

なし

例外

宛先の **source.localData** 属性が初期化されていない場合、このメソッドをコールすると **NULL_LOCAL_DATA** 例外が呼び出されます。

source.isLocal 属性値が 1 で **source.localData** 属性値の値が **NULL** の場合、このメソッドをコールすると **NULL_LOCAL_DATA** 例外が呼び出されます。

例

イメージのコピーを作成します。

```
DECLARE
```

copy() メソッド

```
Image_1 ORDSYS.ORDImage;
Image_2 ORDSYS.ORDImage;
BEGIN
  SELECT photo, large_photo
    INTO Image_2, Image_1
    FROM emp
    WHERE ename = 'John Doe' FOR UPDATE;
  -- copy the data from Image_1 to Image_2
  Image_1.copy(Image_2);
  UPDATE emp SET photo = Image_2
    WHERE ename = 'John Doe';
END;
/
```


4.2.3 Process 操作に関連する ORDImage メソッド

この項では、Process 操作に関連する ORDImage メソッドの参照情報を提供します。

process() メソッド

形式

```
process(command IN VARCHAR2);
```

説明

イメージを上書きして BLOB に対する 1 つまたは複数のイメージ処理操作を実行します。

パラメータ

command
イメージに対して実行するイメージ処理操作のリスト

使用方法

表 4-1 に示す 1 つ以上のイメージ属性を変更できます。表 4-2 はロー・ピクセルおよび外部イメージに対してのみ追加で変更できるものを示しています。

表 4-1 イメージ処理演算子

演算子名	使用方法	値
compressionFormat	圧縮タイプ / 形式	JPEG、SUNRL、BMPRL、TARGARL、LZW、LZWHDIFF、FAX3、FAX4、HUFFMAN3、Packbits、GIFLZW
compressionQuality	圧縮品質	MAXCOMPRATIO、MAXINTEGRITY、LOWCOMP、MEDCOMP、HIGHCOMP
contentFormat	イメージの種類 / ピクセル / データ形式	MONOCHROME、8BITGRAYSCALE、8BITGREYSCALE、8BITLUT、24BITRGB
cut	カットまたはクロップするウィンドウ (起点 .x 起点 .y 幅 高さ)	(INTEGER INTEGER INTEGER INTEGER) 最大値は 65535
fileFormat	イメージのファイル形式	BMPF、CALS、GIFF、JFIF、PICT、RASL、RPIX、TGAF、TIFF
fixedScale	ピクセル単位で特定のサイズに変更 (幅、高さ)	(INTEGER INTEGER)
maxScale	アスペクト比を維持したままピクセル単位で特定のサイズに変更 (maxWidth、maxHeight)	(INTEGER INTEGER)

表 4-1 イメージ処理演算子 (続き)

演算子名	使用方法	値
scale	スケール要素 (例、0.5 または 2.0)	<FLOAT> 正の数
xScale	X 軸スケール要素 (デフォルトは 1)	<FLOAT> 正の数
yScale	Y 軸スケール要素 (デフォルトは 1)	<FLOAT> 正の数

表 4-2 ロー・ピクセルおよび外部イメージの追加イメージ処理演算子

演算子名	使用方法	値
channelOrder	イメージ内の赤、緑および青チャンネル (バンド) の相対的な位置を示します。	RGB (デフォルト)、RBG、GRB、GBR、BRG、BGR
inputChannels	マルチバンド・イメージでは、1 つの整数 (グレースケール) または赤 (第 1)、緑 (第 2) および青 (第 3) を示す 3 つの整数を指定します。このパラメータはコピー先ではなくソース・イメージに影響することに注意してください。	INTEGER または INTEGER INTEGER INTEGER
interleave	イメージ内のバンドのレイアウトをピクセル単位のバンドインターリーブ、行単位のバンド・インターリーブ、バンド順序で制御します。	BIP (デフォルト)、BIL、BSQ
pixelOrder	NORMAL の場合、イメージの左端のピクセルが先頭になります。	NORMAL (デフォルト)、REVERSE
scanlineOrder	NORMAL の場合、イメージの上端の走査線が先頭になります。	NORMAL (デフォルト)、INVERSE

注意: 浮動小数点を含む値を指定する場合、値の前後に二重引用符 ("") を使う必要があります。二重引用符を使わないと、値が正しく渡されず、間違った結果が取得されます。

このメソッドをコールしても `import()` または `importFrom()` の暗黙的なコールは実行されません。データが外部にある場合、処理の前にまず `import()` または `importFrom()` をコールしてデータをローカルに移す必要があります。

`process()` メソッドをコールすると、`setProperties()`、`setUpdateTime()` および `setMimeType()` メソッドが暗黙的に実行されます。

`process()` メソッド演算子の詳細は、付録 C を参照してください。

プラグマ

なし

例外

データがローカルにないか、**source.localData** 属性が初期化されていない場合、このメソッドをコールすると **DATA_NOT_LOCAL** 例外が呼び出されます。

例

image1 のファイル形式を **GIF** に変更します。

```
image1.process('fileFormat=GIF');
```

image1 をより低い品質の **JPEG** 圧縮に変更し、イメージの長さを **X** 軸にそって倍にします。

```
image1.process('compressionFormat=JPEG, compressionQuality=MAXCOMPRATIO,  
xScale="2.0");
```

一方の軸の長さを変更しても (**xScale=2.0** など) 他方の軸の長さは影響を受けず、イメージがゆがむ結果になることに注意してください。また、1 回の操作では、**xScale** と **yScale** パラメータのみを組み合わせることができます。それ以外のスケール演算子の組み合わせではエラーが発生します。

maxScale および **fixedScale** 演算子は、さまざまなサイズのオリジナルからサムネイル・イメージを作成する場合に特に便利です。次の行は元のアスペクト比を保って **32 × 32** ピクセルのサムネイル・イメージを作成します。

```
image1.process(maxScale="32 32");
```

イメージを **TIFF** に変換します。

```
DECLARE  
    Image ORDSYS.ORDImage;  
BEGIN  
    SELECT photo INTO Image FROM emp  
        WHERE ename = 'John Doe' FOR UPDATE;  
        Image.process('fileFormat=TIFF');  
    UPDATE emp SET photo = Image WHERE ename = 'John Doe';  
END;  
/
```

processCopy() メソッド

形式

```
processCopy (command IN      VARCHAR2,  
             dest      IN OUT ORDDImage);
```

説明

内部的または外部的に格納したイメージを、BLOB に内部的に格納した別のイメージにコピーします。

パラメータ

command

新しいコピーのイメージに対して行うイメージ処理変更のリスト

dest

新しいイメージのコピー先

使用方法

表 4-1 の「イメージ処理演算子」および表 4-2 の「ロー・ピクセルおよび外部イメージの追加イメージ処理演算子」を参照してください。

同じ BLOB をソースおよびコピー先の両方に指定することはできません。

このメソッドをコールすると、ソース（ローカルまたは外部）からイメージがコピー先 BLOB にコピーされます。

processCopy() メソッドをコールすると、setProperty()、setUpdateTime() および setMimeType() メソッドが暗黙的に実行されます。

processCopy 演算子の詳細は、付録 C を参照してください。

プラグマ

なし

例外

dest の値が NULL の場合、このメソッドをコールすると NULL_DESTINATION 例外が呼び出されます。

dest.source.isLocal 属性値が FALSE の場合（宛先イメージはローカルに存在する必要があります）、このメソッドをコールすると DATA_NOT_LOCAL 例外が呼び出されます。

processCopy() メソッド

`dest.source.localData` 属性値が `NULL` の場合 (宛先イメージは初期化されている必要があります)、このメソッドをコールすると `NULL_LOCAL_DATA` 例外が呼び出されます。

`source.isLocal` 属性値が `1` で `source.localData` 属性値が `NULL` の場合、このメソッドをコールすると `NULL_LOCAL_DATA` 例外が呼び出されます。

例

コピー先イメージでファイル形式、圧縮形式およびデータ形式を変更してイメージをコピーします。

```
DECLARE
    Image_1 ORDSYS.ORDImage;
    Image_2 ORDSYS.ORDImage;
    mycommand VARCHAR2(400);
BEGIN
    SELECT photo, large_photo
        INTO Image_2, Image_1
        FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    mycommand := 'fileFormat=tiff compressionFormat=packbits
                  contentFormat = 8bitlut';
    Image_1.processCopy(mycommand, Image_2);
    UPDATE emp SET photo = Image_2 WHERE ename = 'John Doe';
END;
/
```

4.2.4 プロパティの設定および check 操作に関連する ORDImage メソッド

この項では、プロパティの set および check 操作に関連する ORDImage メソッドの参照情報を提供します。

setProperty メソッド

形式

```
setProperty;
```

説明

イメージ・データを読み込んでオブジェクト属性の値を取得し、適切な属性フィールドに格納します。イメージ・データは、データベースの **BLOB** に格納するか、外部的に **BFILE** または **URL** に格納できます。データが外部で **BFILE** 以外に格納される場合、データはテンポラリ **BLOB** に読み込まれてイメージ特性が判断されます。

このメソッドは外部イメージに対してはコールできません。外部イメージに対しては **setProperty(description)** メソッドを使います。

パラメータ

なし

使用方法

システム固有形式のイメージのコピー、格納または処理を行った後で、**setProperty** メソッドをコールして、新しいコンテンツの現行の特性を設定します。このメソッドが暗黙的にコールされる場合はその必要はありません。

このメソッドを使用して、イメージに関する次の情報を設定します。

- ピクセル単位の高さ
- ピクセル単位の幅
- ディスク上のイメージのバイト単位のデータ・サイズ
- ファイル・タイプ (**TIFF**、**JFIF** など)
- イメージの種類 (モノクロ、8 ビット・グレースケールなど)
- 圧縮タイプ (**JPEG**、**LZW** など)
- **MIME** タイプ (ファイル形式に基づいて生成)

このメソッドをコールすると、暗黙的に **setUpdateTime()** および **setMimeType()** メソッドがコールされます。

プラグマ

なし

例外

`source.isLocal` 属性値が 1 で `source.localData` 属性値の値が NULL の場合、このメソッドをコールすると NULL_LOCAL_DATA 例外が呼び出されます。

例

イメージを選択してから、`setProperties` メソッドで属性を設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    INSERT INTO emp VALUES ('John Doe', 24000, 'Technical Writer', 123,
        ORDSYS.ORDImage(ORDSYS.ORDSource(empty_blob(),'file','ORDIMGDIR',
            'jdoe.gif',SYSDATE,0),
            NULL,NULL,NULL,NULL,NULL,NULL,NULL));
    -- select the newly inserted row for update
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- set property attributes for the image data
    Image.setProperties;
    DBMS_OUTPUT.PUT_LINE('image width = ' || image.getWidth);
    DBMS_OUTPUT.PUT_LINE('image height = ' || image.getHeight);
    DBMS_OUTPUT.PUT_LINE('image size = ' || image.getContentLength);
    DBMS_OUTPUT.PUT_LINE('image file type = ' || image.getFileFormat);
    DBMS_OUTPUT.PUT_LINE('image type = ' || image.getContentType);
    DBMS_OUTPUT.PUT_LINE('image compression = ' || image.getCompressionFormat);
    DBMS_OUTPUT.PUT_LINE('image mime type = ' || image.getMimeType);
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

出力例

```
image width = 360
image height = 490
image size = 66318
image file type = JFIF
image type = 24BITRGB
image compression = JPEG
image mime type = image/jpeg
```

外部イメージの setProperties() メソッド

形式

```
setProperties(description IN VARCHAR2);
```

説明

外部イメージの特性を適切な属性フィールドに書き込むことができます。

注意: このメソッドでは、MIME タイプ属性は設定されません。外部イメージに MIME タイプを設定する `setMimeType()` メソッドを起動する必要があります。

パラメータ

description

外部イメージに設定するイメージ特性を指定します。

使用方法

注意: いったん外部イメージのプロパティを設定したら、プロパティが一貫したものとなるようにユーザーが管理する必要があります。*interMedia image* で不明なファイル形式が検出されると、暗黙的にプロパティが設定されます。

外部イメージのコピー、格納または処理を行った後で、このメソッドをコールして、新しいイメージ・コンテンツの特性を設定します。付録 D で説明しているシステム固有のイメージの種類とは異なり、外部（ヘッダーなし）イメージではファイルのビットを解釈する方法についての情報が含まれていないか、*interMedia image* で情報を解読することができません。この場合、情報は明示的に設定する必要があります。

表 4-3 に示すとおり、外部イメージには次のイメージ特性を設定できます。

表 4-3 外部（ヘッダーなし）ファイルのイメージ特性

フィールド	データ型	説明
CompressionFormat	STRING	値は CCITG3、CCITG4 または NONE（デフォルト）にする必要があります。
DataOffset	INTEGER	オフセットでは、イメージに <i>interMedia image</i> では解釈されないヘッダーを設定できます。オフセットにはヘッダーになる可能性のあるものを避けて設定します。値は LOB 長よりも小さい正の整数でなければなりません。デフォルトは 0（ゼロ）です。
DefaultChannelSelection	INTEGER	マルチバンド・イメージでは、1 つの整数（グレースケール）または赤（第 1）、緑（第 2）および青（第 3）を示す 3 つの整数を指定します。
Height	INTEGER	ピクセル単位のイメージの高さ。値は正の整数でなければなりません。デフォルトはないため、値は必ず指定する必要があります。
Interleave	STRING	イメージ内のバンド・レイアウト。有効なスタイルは次のとおりです。 <ul style="list-style-type: none"> ■ BIP（デフォルト）ピクセル単位のバンド・インターリーブ ■ BIL 行単位のバンド・インターリーブ ■ BSQ バンド順序
NumberOfBands	INTEGER	イメージのカラー帯域バンドを指定する値は 255 よりも小さい正の整数でなければなりません。デフォルトは 3 です。
PixelOrder	STRING	NORMAL（デフォルト）の場合、左端のピクセルがファイルの先頭になります。REVERSE の場合、右端のピクセルが先頭になります。
ScanlineOrder	STRING	NORMAL（デフォルト）の場合、上端の走査線がファイルの先頭になります。INVERSE の場合は、下端に走査線が先頭になります。
UserString	STRING	4 文字で記述する文字列。使用する場合、文字列は fileFormat フィールドに格納され、ファイル形式 ("OTHER:") に追加されます。デフォルトはブランクで fileFormat は "OTHER" に設定されます。
Width	INTEGER	ピクセル単位のイメージの幅。値は正の整数でなければなりません。デフォルトはないため、値は必ず指定する必要があります。

setProperties() に指定した値が既存の ORDImage データ属性に書き込まれます。fileFormat は "OTHER" に設定され、指定する場合ユーザー文字列が含まれます。例えば、'OTHER : LANDSAT' のようになります。

プラグマ

なし

例外

説明属性値が NULL の場合、このメソッドをコールすると、NULL_PROPERTIES_DESCRIPTION 例外が呼び出されます。

例

外部イメージを選択し、イメージのプロパティを設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- set property attributes for the image data
    Image.setProperties('width=123 height=321 compressionFormat=NONE' ||
        ' userString=DJM dataOffset=128' ||
        ' scanlineOrder=INVERSE pixelOrder=REVERSE' ||
        ' interleaving=BIL numberOfBands=1' ||
        ' defaultChannelSelection=1');
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

checkProperties メソッド

形式

```
checkProperties RETURN BOOLEAN;
```

説明

イメージ・オブジェクトの属性に格納されたプロパティがイメージのプロパティと一致することを検証します。このメソッドは外部イメージには使用しません。

パラメータ

なし

使用方法

このメソッドを使用してイメージ属性が実際のイメージと一致することを検証します。

プラグマ

なし

例外

なし

例

イメージ属性をチェックします。

```
DECLARE
    Image ORDSYS.ORDImage;
    properties_match BOOLEAN;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- check that properties match the image
    properties_match := Image.checkProperties;
    IF properties_match THEN
        DBMS_OUTPUT.PUT_LINE('Check Properties succeeded');
    END IF;
END;
```

/

4.2.5 イメージ属性に関連する ORDImage メソッド

この項では、イメージ属性に関連する ORDImage メソッドの参照情報を提供します。

getHeight メソッド

形式

```
getHeight RETURN INTEGER;
```

説明

ピクセル単位のイメージの高さを返します。このメソッドは高さ属性の値を返す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

高さ属性に直接アクセスすると **ORDImage** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getHeight, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージの高さを取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    Height INTEGER;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image height
    Height := Image.getHeight;
END;
/
```

getWidth メソッド

形式

```
getWidth RETURN INTEGER;
```

説明

イメージの幅をピクセル単位で返します。このメソッドは幅属性の値を返す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

幅属性に直接アクセスすると **ORDImage** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getWidth, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージの幅を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    Width INTEGER;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image width
    Width := Image.getWidth;
END;
/
```

getContentLength メソッド

形式

```
getContentLength RETURN INTEGER;
```

説明

ソースに格納されたイメージ・データ・コンテンツの長さを戻します。このメソッドは、コンテンツ長属性の値を戻す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

`contentLength` 属性に直接アクセスすると `ORDImage` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getContentLength, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

ソースに格納されたイメージ・データ・コンテンツの長さを取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    ContentLength INTEGER;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image size
    ContentLength := Image.getContentLength;
END;
```

getFileFormat メソッド

形式

```
getFileFormat RETURN VARCHAR2;
```

説明

イメージのファイル・タイプ（TIFF または JFIF など）を戻します。このメソッドは **fileFormat** 属性の値を戻す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

fileFormat 属性に直接アクセスすると **ORDImage** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getFileFormat, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージのファイル・タイプを取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    FileFormat VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image file format
    FileFormat := Image.getFileFormat;
END;
```

getContentTypeFormat メソッド

形式

```
getContentTypeFormat RETURN VARCHAR2;
```

説明

イメージのコンテンツ・タイプ（モノクロまたは 8 ビット・グレースケールなど）を返します。このメソッドは `contentTypeFormat` 属性の値を返す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

`contentTypeFormat` 属性に直接アクセスすると `ORDImage` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getContentTypeFormat, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージの種類を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    ContentFormat VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
    WHERE ename = 'John Doe';
    -- get the image content format
    ContentFormat := Image.getContentTypeFormat;
END;
```

getCompressionFormat メソッド

形式

```
getCompressionFormat RETURN VARCHAR2;
```

説明

イメージの圧縮タイプを戻します。このメソッドは `compressionFormat` 属性の値を戻す単純なアクセサ・メソッドで、実際にはイメージを読み込みません。

パラメータ

なし

使用方法

`compressionFormat` 属性に直接アクセスすると `ORDImage` オブジェクトの内部表現を変更してしまう可能性があるので、それを避けるためにこのメソッドを使います。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getCompressionFormat, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージの圧縮タイプを取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    CompressionFormat VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image compression format
    CompressionFormat := Image.getCompressionFormat;
END;
```

4.2.6 ローカル属性に関連する ORDImage メソッド

この項では、ローカル属性に関連する ORDImage メソッドの参照情報を提供します。

setLocal メソッド

形式

```
setLocal;
```

説明

データが **BLOB** に内部的に格納されたことを示すローカル属性を設定します。**local** が設定されている場合、**image** メソッドは **source.localData** 属性のイメージ・データを探します。

パラメータ

なし

使用方法

ローカル属性を 1 に設定します。これは、データが **localData** 属性にローカルに格納されていることを示します。

プラグマ

なし

例外

source.localData 属性値の値が **NULL** の場合、このメソッドをコールすると **NULL_LOCAL_DATA** 例外が呼び出されます。

例

データのフラグをローカルに設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp WHERE ename = 'John Doe' FOR UPDATE;
    -- set local so we look for the image in the database
    Image.setLocal;
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

clearLocal メソッド

形式

```
clearLocal;
```

説明

ローカル・フラグをリセットして、データが外部に格納されていることを示します。ローカル・フラグの設定が消去されると、**image** メソッドは **srcLocation**、**srcName** および **srcType** 属性を使用してイメージ・データを検索します。

パラメータ

なし

使用方法

このメソッドはローカル属性を **0** に設定し、データが外部的に、つまり **Oracle8i** の外に格納されていることを示します。

プラグマ

なし

例外

なし

例

データのローカル・フラグの値を消去します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp WHERE ename = 'John Doe' FOR UPDATE;
    -- clear local so we look for image externally
    Image.clearLocal;
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

isLocal メソッド

形式

```
isLocal RETURN BOOLEAN;
```

説明

データが **BLOB** にローカルに格納している場合は **TRUE** を返し、データが外部的に格納されている場合は **FALSE** を返します。

パラメータ

なし

使用方法

ローカル属性が **1** または **NULL** に設定されている場合、このメソッドにより **TRUE** が返されます。それ以外の場合、このメソッドにより **FALSE** が返されます。

プラグマ

Pragma RESTRICT_REFERENCES(isLocal, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

データがローカルにあるかどうかを判断します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp WHERE ename = 'John Doe';
    -- check to see if image is stored locally
    IF Image.isLocal THEN
        DBMS_OUTPUT.PUT_LINE('Image is stored locally');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Image is stored externally');
    END IF;
END;
/
```


4.2.7 日付属性に関連する ORDImage メソッド

この項では、日付属性に関連する ORDImage メソッドの参照情報を提供します。

getUpdateTime メソッド

形式

```
getUpdateTime RETURN DATE;
```

説明

オブジェクトが最後に更新された時間を返します。

パラメータ

なし

使用方法

なし

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージ・オブジェクトの一部が更新された時間を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    UpdateTime DATE;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image update time
    UpdateTime := Image.getUpdateTime;
END;
/
```

setUpdateTime() メソッド

形式

```
setUpdateTime(current_time DATE);
```

説明

イメージ・データが最後に更新された時間を設定します。イメージ・データを変更したときにこのメソッドを使います。`copy()`、`process()`、`processCopy()`、`setProperties`、`setMimeType()` および `export()` メソッドをコールすると、このメソッドが暗黙的にコールされます。

パラメータ

current_time

格納するタイムスタンプ。デフォルトは `SYSDATE` です。

使用方法

イメージ・データを自分で変更したときにこのメソッドを起動する必要があります。

プラグマ

なし

例外

なし

例

イメージ・データの一部分が更新された時間を設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- set the image update time
    Image.setUpdateTime(SYSDATE);
END;
/
```

4.2.8 mimeType 属性に関連する ORDImage メソッド

この項では mimeType 属性に関連する ORDImage メソッドの参照情報を提供します。

getMimeType メソッド

形式

```
getMimeType RETURN VARCHAR2;
```

説明

イメージ・データの MIME タイプを返します。これは **mimeType** 属性の値を返す単純なアクセサ・メソッドです。

パラメータ

なし

使用方法

mimeType 属性に直接アクセスすると **ORDImage** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。ソースがファイルまたは **BLOB** の場合、MIME タイプ情報が生成されます。

認識できないファイル形式の場合、ユーザーは **setMimeType()** メソッドをコールして MIME タイプを指定する必要があります。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getMimeType, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

MIME タイプを返します。

```
DECLARE
    Image ORDSYS.ORDImage;
    MimeType VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image mime type
    MimeType := Image.getMimeType;
END;
/
```

setMimeType() メソッド

形式

```
setMimeType(mime IN VARCHAR2);
```

説明

イメージ・データの **MIME** タイプを設定できます。

パラメータ

mime
MIME タイプ

使用方法

このメソッドをコールすると、MIME 情報の自動設定を指定した **MIME** 値で上書きできます。

外部イメージには **MIME** タイプを設定する **setMimeType()** メソッドを起動する必要があります。

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドが呼び出されます。

setProperties、**process()** および **processCopy()** メソッドではこのメソッドが暗黙的にコールされます。

HTTP ソースのインポート時に HTTP ヘッダーから **MIME** タイプが抽出されます。

プラグマ

なし

例外

なし

例

格納された一部のイメージ・データに **MIME** タイプを設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp
```

```
WHERE ename = 'John Doe';
-- set the image mime type
Image.setMimeType('image/bmp');
END;
```

4.2.9 ソース属性に関連する ORDIImage メソッド

この項では、ソース属性に関連する ORDIImage メソッドの参照情報を提供します。

getContent メソッド

形式

```
getContent RETURN BLOB;
```

説明

ハンドルをローカル LOB 記憶域に戻します。これは、ORDImage オブジェクト内の BLOB になります。

パラメータ

なし

使用方法

なし

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getContent, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

クライアントがイメージ・データにアクセスします。

```
DECLARE
    Image ORDSYS.ORDImage;
    localData BLOB;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image BLOB
    localData := Image.getContent;
END;
/
```

getBFILE メソッド

形式

```
getBFILE RETURN BFILE;
```

説明

イメージを含む **BFILE** の **LOB** ロケータを戻します。

パラメータ

なし

使用方法

このメソッドにより、格納されている **source.srcLocation** および **source.srcName** 属性の情報を使用して **BFILE** を構築し、戻します。**source.srcLocation** 属性には定義されたディレクトリ・オブジェクトが含まれていなければなりません。**source.srcName** 属性は有効なファイル名にする必要があります。

プラグマ

PRAGMA RESTRICT_REFERENCES(getBFILE, WNDS, WNPS, RNDS, RNPS)

例外

source.srcType 属性値が **NULL** の場合、このメソッドをコールすると **INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

srcType の値が **FILE** 以外の場合、このメソッドをコールすると **INVALID_SOURCE_TYPE** 例外が呼び出されます。

例

```
DECLARE
    Image ORDSYS.ORDImage;
    imagebfile BFILE;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image BFILE
    imagebfile := Image.getBFILE;
END;
```

deleteContent メソッド

形式

```
deleteContent;
```

説明

現行のローカル・ソース (localData) からローカル・データを削除します。

パラメータ

なし

使用方法

このメソッドは、ローカル・ソースから外部イメージ・データ・ソースにデータをエクスポートした後、このデータがローカル・ソースで必要なくなったときにコールします。

オブジェクトを新しいオブジェクトで更新する場合にこのメソッドをコールします。

プラグマ

なし

例外

なし

例

現行のローカル・ソースからローカル・データを削除します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- delete the local content of the image
    Image.deleteContent;
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

setSource() メソッド

形式

```
setSource(source_type      IN VARCHAR2,  
          source_location  IN VARCHAR2,  
          source_name      IN VARCHAR2);
```

説明

イメージ・データの外部ソースについての情報を設定または変更します。

パラメータ

source_type

外部データのソース・タイプ。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

source_location

外部データのソースの場所。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

source_name

外部データのソース名。詳細は、第 6 章の「ORDSource オブジェクト型」の定義を参照してください。

使用方法

ユーザーはこのメソッドを使用してイメージ・データ・ソースを新しい BFILE または URL に設定できます。このメソッドをコールすると、暗黙的に setUpdateTime() メソッドがコールされます。

プラグマ

なし

例外

なし

例

イメージ・データのソースを設定します。

```
DECLARE
    Image ORDSYS.ORDImage;
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- set source information for the image
    Image.setSource('file',
        'ORDIMGDIR',
        'jdoe.gif');
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

getSource メソッド

形式

```
getSource RETURN VARCHAR2;
```

説明

URL 形式のイメージ・データの外部位置についての情報を戻します。

パラメータ

なし

使用方法

考えられる戻り値は次のとおりです。

- ファイル・ソースに対しては **FILE : //<DIR OBJECT NAME>/<FILE NAME>**
- HTTP ソースに対しては **HTTP : //<URL>**
- ユーザー定義ソース

プラグマ

PRAGMA RESTRICT_REFERENCES(getSource, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

イメージ・データのソースを取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    SourceInfo VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image source information
    SourceInfo := Image.getSource;
END;
```

getSourceType メソッド

形式

```
getSourceType RETURN VARCHAR2;
```

説明

外部イメージ・データ・ソースのタイプを含む文字列を戻します。

パラメータ

なし

使用方法

このメソッドは外部イメージ・データ・ソースのタイプ ('FILE' など) を含む VARCHAR2 文字列を戻します。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

イメージ・データ・ソースについてのソース・タイプ情報を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    SourceType VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image source type
    SourceType := Image.getSourceType;
END;
/
```

getSourceLocation メソッド

形式

```
getSourceLocation RETURN VARCHAR2;
```

説明

外部イメージ・データ・ソースの場所の値を含む文字列を返します。

パラメータ

なし

使用方法

このメソッドは外部イメージ・データ位置の値 ('BFILEDIR' など) を含む **VARCHAR2** 文字列を返します。

このメソッドを使う前に、ディレクトリが存在するか作成されたことを確認する必要があります。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS)
```

例外

srcLocation の値が **NULL** の場合、このメソッドをコールすると **ORDSourceExceptions.INCOMPLETE_SOURCE_LOCATION** 例外が呼び出されます。

例

イメージ・データ・ソースについてのソースの場所情報を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    SourceLocation VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image source location
    SourceLocation := Image.getSourceLocation;
END;
```

getSourceName メソッド

形式

```
getSourceName RETURN VARCHAR2;
```

説明

外部イメージ・データ・ソースの名前を含む文字列を返します。

パラメータ

なし

使用方法

外部データ・ソースの名前 ('testing.dat' など) を含む VARCHAR2 文字列を返します。

プラグマ

```
PRAGMA RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS)
```

例外

srcName の値が NULL の場合、このメソッドをコールすると
ORDSourceExceptions.INCOMPLETE_SOURCE_NAME 例外が呼び出されます。

例

イメージ・データ・ソースについてのソース名情報を取得します。

```
DECLARE
    Image ORDSYS.ORDImage;
    SourceName VARCHAR2(4000);
BEGIN
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe';
    -- get the image source name
    SourceName := Image.getSourceName;
END;
/
```

import() メソッド

形式

```
MEMBER PROCEDURE import (ctx IN OUT RAW);
```

説明

外部イメージ・データ・ソースから Oracle データベース内のローカル・ソース (localData) にイメージ・データを転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。これは必ず割り当てなければなりません。
source.open() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

import() メソッドをコールする前に、**setSource()** メソッドを使用して外部ソース・タイプ、場所および名前を設定します。

このメソッドを使う前に、ディレクトリが存在するか作成されたことを確認する必要があります。

外部イメージ・データ・ソースから (Oracle データベース内の) ローカル・ソースにデータをインポートしても、(データがインポートされたソースを示す) ソース情報は変更されずにそのまま残ります。

このメソッドを起動すると、暗黙的に **setUpdateTime()** メソッドおよび **setLocal** メソッドがコールされます。

インポートされたイメージのファイル形式がシステム固有の場合、**setProperties()** メソッドもコールされます。

プラグマ

なし

例外

srcType の値が **NULL** の場合、このメソッドをコールすると **ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

`dlob` の値が `NULL` の場合、このメソッドをコールすると `ORDSourceExceptions.NULL_SOURCE` 例外が呼び出されます。

使用しているソース・プラグインで `import()` メソッドがサポートされない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

外部イメージ・データ・ソースからローカル・ソースにイメージ・データをインポートします。

```
DECLARE
    Image ORDSYS.ORDImage;
    ctx RAW(4000) :=NULL;
BEGIN
    -- select the image to be imported
    SELECT large_photo INTO Image FROM emp
        WHERE ename = 'John Doe' FOR UPDATE;
    -- import the image into the database
    Image.import(ctx);
    -- update the image object
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

importFrom() メソッド

形式

```
importFrom(ctx          IN OUT RAW,  
            source_type  IN VARCHAR2,  
            source_location IN VARCHAR2,  
            source_name   IN VARCHAR2);
```

説明

指定した外部イメージ・データ・ソースから Oracle データベース内のローカル・ソース (localData) にイメージ・データを転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。これは必ず割り当てなければなりません。
source.open() メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

source_type

イメージ・データのソース・タイプ

source_location

イメージ・データのインポート元の位置

source_name

イメージ・データの名前

使用方法

このメソッドは **import()** メソッドに似ていますが、ソース情報を別々にではなくメソッド内で指定する点異なります。

このメソッドを使う前に、ディレクトリが存在するか作成されたことを確認する必要があります。

外部イメージ・データ・ソースから (Oracle データベース内の) ローカル・ソースにデータをインポートすると、(データをインポートしたソースを示す) ソース情報が入力値に設定されます。

このメソッドを起動すると、暗黙的に **setUpdateTime()** メソッドおよび **setLocal** メソッドが呼び出されます。

インポートするイメージのファイル形式がシステム固有の場合、`setProperties()` メソッドもコールされます。

プラグマ

なし

例外

`dlob` の値が `NULL` の場合、このメソッドをコールすると `ORDSourceExceptions.NULL_SOURCE` 例外が呼び出されます。

使用しているソース・プラグインで `importFrom()` メソッドがサポートされていない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が呼び出されます。

他の例外が呼び出されているときにソース・プラグイン内でこのメソッドをコールすると `ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

指定した外部データ・ソースからローカル・ソースにイメージ・データをインポートします。

```
DECLARE
  Image ORDSYS.ORDImage;
  ctx RAW(4000) :=NULL;
BEGIN
  -- select the image to be imported
  SELECT large_photo INTO Image FROM emp
    WHERE ename = 'John Doe' FOR UPDATE;
  -- import the image into the database
  Image.importFrom(ctx,
    'file',
    'ORDIMGDIR',
    'jdoe.gif');
  -- update the image object
  UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

export() メソッド

形式

```
export (ctx          IN OUT RAW,  
       source_type   IN VARCHAR2,  
       source_location IN VARCHAR2,  
       source_name    IN VARCHAR2);
```

説明

Oracle データベース内のローカル・ソース (localData) から外部イメージ・データ・ソースにイメージ・データを転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報

source_type

データがエクスポートされる場所のソース・タイプ

source_location

イメージ・データがエクスポートされる場所

source_name

データがエクスポートされるイメージ・オブジェクトの名前

使用方法

イメージ・データをエクスポートした後もイメージ属性はすべて変更されません。ソース属性 (ソース・タイプ、ソースの場所およびソース名) の情報が更新されます。**export** メソッドをコールした後、**deleteContent** メソッドをコールしてローカル・データのコンテンツを削除できます。

export メソッドのサポートはサーバー側に固有のものではありません。このメソッドは **export** メソッドをサポートするユーザー定義ソースで使用できます。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドをコールすると
ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用しているソース・プラグインで export() メソッドがサポートされていない場合、
ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の例外が呼び出されているときにこのメソッドをソース・プラグイン内でコールすると、
ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

これらの例外の詳細は、付録 H を参照してください。

例

なし

4.2.10 イメージ移行に関連する ORDImage メソッド

この項では、古い ORDImgB イメージおよび ORDImgF イメージを新しい ORDImage イメージに変換するイメージ移行に関連する ORDImage メソッドの参照情報を提供します。

migrateFromORDImgB() メソッド

形式

```
migrateFromORDImgB(old_object ORDImgB);
```

説明

古い ORDImgB イメージを新しい ORDImage オブジェクトに移行できるようにします。

パラメータ

old_object

古い ORDImgB イメージ

使用方法

このメソッドはソース BLOB からコピー先 BLOB にコピーし、古いオブジェクトから新しいオブジェクトにすべてのイメージ属性をコピーして、更新時間およびローカル属性を設定します。

プラグマ

なし

例外

src (old_object) の値が NULL の場合、このメソッドをコールすると NULL_SOURCE 例外が呼び出されます。

dest の値が NULL (ORDImage) の場合、このメソッドをコールすると NULL_DESTINATION 例外が呼び出されます。

src.content の値が NULL (old.object コンテンツ属性) の場合、このメソッドをコールすると NULL_CONTENT 例外が呼び出されます。

dest.source.localData 値が NULL (dest ORDImage source.localData) の場合、このメソッドをコールすると NULL_LOCAL_DATA 例外が呼び出されます。

例

古い ORDImgB イメージを新しい ORDImage イメージに移行します。

```
DECLARE  
  Image ORDSYS.ORDImage;  
  old_image ORDSYS.ORDIMGB;
```

migrateFromORDImgB() メソッド

```
BEGIN
  -- Select the old image
  SELECT imageb INTO old_image FROM old_images WHERE id = 1;
  -- Select the new image
  SELECT large_photo INTO Image FROM emp WHERE ename = 'John Doe' FOR UPDATE;
  -- Migrate from the old to the new
  Image.migrateFromORDImgB(old_image);
  UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
END;
/
```

migrateFromORDImgF() メソッド

形式

```
migrateFromORDImgF(old_object ORDImgF);
```

説明

古い ORDImgF イメージを新しい ORDImage オブジェクトに移行できるようにします。

パラメータ

old_object

古い ORDImgF イメージ

使用方法

このメソッドはソースからディレクトリ名およびファイル名を抽出し、コピー先の **srcLocation** および **srcName** 属性にそれらをコピーします。さらに、古いイメージ・オブジェクトから新しいイメージ・オブジェクトにイメージ属性をすべてコピーし、**updateTime** 属性を設定して、ローカル属性を消去します。

プラグマ

なし

例外

なし

例

古い ORDImgF イメージを新しい ORDImage イメージに移行します。

```
DECLARE
    Image ORDSYS.ORDImage;
    old_image ORDImgF;
BEGIN
    -- Select the old image
    SELECT imagef INTO old_image FROM old_images WHERE id = 1;
    -- Select the new image
    SELECT large_photo INTO Image FROM emp WHERE ename = 'John Doe' FOR UPDATE;
    -- Migrate from the old to the new
    Image.migrateFromORDImgf(old_image);
    UPDATE emp SET large_photo = Image WHERE ename = 'John Doe';
```

migrateFromORDImgF() メソッド

```
END;  
/
```

ORDVideo 参照情報

Oracle8i *interMedia* には、ORDVideo 型に関する次の情報が含まれています。

- オブジェクト型 -- 5.1 項を参照してください。
- メソッド -- 5.2 項を参照してください。
- パッケージまたは PL/SQL プラグイン -- 5.3 項を参照してください。

この章の例では、テスト・ビデオ表 TVID の作成およびデータの挿入が行われていることを前提としています。この表は、5.2.1 項で説明する SQL 文を使用して作成します。

注意: ビデオ・データを操作した場合 (BLOB を直接修正するか外部ソースを変更することにより)、オブジェクト属性の同期がとれていて、その更新時間が変更されていなければなりません。それ以外の場合は、オブジェクト属性はビデオ・データと一致しません。

メソッドは、処理のためにソース・プラグインに渡された **ORDSource** レベルでコールされます。このメソッドには、最初の引数として **ctx (RAW(4000))** が指定されています。これらのメソッドのいずれかをクライアントから初めてコールする場合、**ctx** 構造を割り当てて **NULL** に初期化してから、**openSource()** メソッドをコールする必要があります。このとき、ソース・プラグインによりこのクライアントのコンテキストを初期化できます。処理が完了したら、クライアントから **closeSource()** メソッドをコールする必要があります。

ソース・プラグインのコールにより起動されたメソッドには、最初の引数として **ctx (RAW(4000))** が指定されています。

メソッドは、処理のために形式プラグインに渡された **ORDVideo** レベルでコールされます。このメソッドには、最初の引数として **ctx (RAW(4000))** が指定されています。これらのメソッドのいずれかをクライアントから初めてコールする場合、**ctx** 構造を割り当て、**NULL** に初期化する必要があります。

注意: 現行のリリースでは、すべてのソースまたは形式プラグインで `ctx` 引数が使われるわけではありませんが、すでに説明した方法でコーディングした場合、アプリケーションはそれ以降に追加されたすべてのソース・プラグインまたは形式プラグインとともに使う必要があります。

5.1 オブジェクト型

Oracle8i *interMedia* には、ORDVideo オブジェクト型が記述されています。ORDVideo オブジェクト型によって、ビデオ・データの格納および管理がサポートされます。

ORDVideo オブジェクト型

ORDVideo オブジェクト型では、ビデオ・データの格納および管理がサポートされます。このオブジェクト型の定義は次のとおりです。

```
CREATE OR REPLACE TYPE ORDVideo
AS OBJECT
(
    -- ATTRIBUTES
    description          VARCHAR2(4000),
    source               ORDSOURCE,
    format               VARCHAR2(31),
    mimeType             VARCHAR2(4000),
    comments             CLOB,
    -- VIDEO RELATED ATTRIBUTES
    width               INTEGER,
    height              INTEGER,
    frameResolution     INTEGER,
    frameRate           INTEGER,
    videoDuration       INTEGER,
    numberOfFrames      INTEGER,
    compressionType     VARCHAR2(4000),
    numberOfColors      INTEGER,
    bitRate             INTEGER,

    -- METHODS
    -- Methods associated with the date attribute
    MEMBER FUNCTION getUpdateTime RETURN DATE,
    PRAGMA RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS),
    MEMBER PROCEDURE setUpdateTime(current_time DATE),
    -- Methods associated with the description attribute
    MEMBER PROCEDURE setDescription(user_description IN VARCHAR2),
    MEMBER FUNCTION getDescription RETURN VARCHAR2,
    PRAGMA RESTRICT_REFERENCES(getDescription, WNDS, WNPS, RNDS, RNPS),

    -- Methods associated with the mimeType attribute
    MEMBER PROCEDURE setMimeType(mime IN VARCHAR2),
    MEMBER FUNCTION getMimeType RETURN VARCHAR2,
    PRAGMA RESTRICT_REFERENCES(getMimeType, WNDS, WNPS, RNDS, RNPS),

    -- Methods associated with the source attribute
    MEMBER FUNCTION processSourceCommand(
                                ctx          IN OUT RAW,
                                cmd         IN VARCHAR2,
                                arguments IN VARCHAR2,
```

ORDVideo オブジェクト型

```
                                result    OUT RAW)
RETURN RAW,

MEMBER FUNCTION  isLocal RETURN BOOLEAN,
PRAGMA RESTRICT_REFERENCES(isLocal, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE  setLocal,
MEMBER PROCEDURE  clearLocal,

MEMBER PROCEDURE  setSource(
                                source_type    IN VARCHAR2,
                                source_location IN VARCHAR2,
                                source_name     IN VARCHAR2),
MEMBER FUNCTION  getSource RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSource, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION  getSourceType RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION  getSourceLocation RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION  getSourceName RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE  import(ctx IN OUT RAW),
MEMBER PROCEDURE  importFrom(
                                ctx                IN OUT RAW,
                                source_type        IN VARCHAR2,
                                source_location    IN VARCHAR2,
                                source_name        IN VARCHAR2),
MEMBER PROCEDURE  export(
                                ctx                IN OUT RAW,
                                source_type        IN VARCHAR2,
                                source_location    IN VARCHAR2,
                                source_name        IN VARCHAR2),

MEMBER FUNCTION  getLength(ctx IN OUT RAW) RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getLength, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION  getContentInLob(
                                ctx                IN OUT RAW,
                                dest_lob          IN OUT NOCOPY BLOB,
                                mimeType          OUT VARCHAR2,
                                format            OUT VARCHAR2),

MEMBER FUNCTION  getContent RETURN BLOB,
```



```

PRAGMA RESTRICT_REFERENCES (getContent, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE deleteContent,

-- Methods associated with file operations on the source
MEMBER FUNCTION openSource(userArg IN RAW, ctx OUT RAW) RETURN INTEGER,
MEMBER FUNCTION closeSource(ctx IN OUT RAW) RETURN INTEGER,
MEMBER FUNCTION trimSource(ctx      IN OUT RAW,
                           newlen  IN INTEGER) RETURN INTEGER,
MEMBER PROCEDURE readFromSource(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   OUT RAW),
MEMBER PROCEDURE writeToSource(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   IN RAW),

-- Methods associated with the comments attribute
MEMBER PROCEDURE appendToComments (amount IN BINARY_INTEGER,
    buffer IN VARCHAR2),
MEMBER PROCEDURE writeToComments (offset IN INTEGER,
    amount IN BINARY_INTEGER,
    buffer IN VARCHAR2),
MEMBER FUNCTION readFromComments (offset IN INTEGER,
    amount IN BINARY_INTEGER := 32767)
    RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES (readFromComments, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION locateInComments (pattern  IN VARCHAR2,
    offset  IN INTEGER := 1,
    occurrence IN INTEGER := 1)
    RETURN INTEGER,
MEMBER PROCEDURE trimComments (newlen IN INTEGER),
MEMBER PROCEDURE eraseFromComments (amount IN OUT NOCOPY INTEGER,
    offset IN INTEGER := 1),
MEMBER PROCEDURE deleteComments,
MEMBER PROCEDURE loadCommentsFromFile (fileobj  IN BFILE,
    amount  IN INTEGER,
    from_loc IN INTEGER := 1,
    to_loc  IN INTEGER := 1),
MEMBER PROCEDURE copyCommentsOut (dest      IN OUT NOCOPY CLOB,
    amount  IN INTEGER,
    from_loc IN INTEGER := 1,
    to_loc  IN INTEGER := 1),

```

ORDVideo オブジェクト型

```
MEMBER FUNCTION compareComments(  
    compare_with_lob          IN CLOB,  
    amount                    IN INTEGER := 4294967295,  
    starting_pos_in_comment IN INTEGER := 1,  
    starting_pos_in_compare IN INTEGER := 1)  
    RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(compareComments, WNDS, WNPS, RNDS, RNPS),  
  
MEMBER FUNCTION getCommentLength RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(getCommentLength, WNDS, WNPS, RNDS, RNPS),  
  
-- Methods associated with the video attributes accessors  
MEMBER PROCEDURE setFormat(knownformat IN VARCHAR2),  
MEMBER FUNCTION getFormat RETURN VARCHAR2,  
PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS),  
MEMBER FUNCTION getFormat(ctx IN OUT RAW) RETURN VARCHAR2,  
  
MEMBER PROCEDURE setFrameSize(knownWidth IN INTEGER, knownHeight IN INTEGER),  
MEMBER PROCEDURE setFrameSize(retWidth OUT INTEGER, retHeight OUT INTEGER),  
PRAGMA RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS),  
MEMBER PROCEDURE setFrameSize(  
    ctx IN OUT RAW,  
    retWidth OUT INTEGER,  
    retHeight OUT INTEGER),  
  
MEMBER PROCEDURE setFrameResolution(knownFrameResolution IN INTEGER),  
MEMBER FUNCTION getFrameResolution RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS),  
MEMBER FUNCTION getFrameResolution(ctx IN OUT RAW) RETURN INTEGER,  
  
MEMBER PROCEDURE setFrameRate(knownFrameRate IN INTEGER),  
MEMBER FUNCTION getFrameRate RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS),  
MEMBER FUNCTION getFrameRate(ctx IN OUT RAW) RETURN INTEGER,  
  
MEMBER PROCEDURE setVideoDuration(knownVideoDuration IN INTEGER),  
MEMBER FUNCTION getVideoDuration RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS),  
MEMBER FUNCTION getVideoDuration(ctx IN OUT RAW) RETURN INTEGER,  
  
MEMBER PROCEDURE setNumberOfFrames(knownNumberOfFrames IN INTEGER),  
MEMBER FUNCTION getNumberOfFrames RETURN INTEGER,  
PRAGMA RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS),  
MEMBER FUNCTION getNumberOfFrames(ctx IN OUT RAW) RETURN INTEGER,  
  
MEMBER PROCEDURE setCompressionType(knownCompressionType IN VARCHAR2),  
MEMBER FUNCTION getCompressionType RETURN VARCHAR2,
```

```
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2,

MEMBER PROCEDURE setNumberOfColors(knownNumberOfColors IN INTEGER),
MEMBER FUNCTION getNumberOfColors RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getNumberOfColors(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setBitRate(knownBitRate IN INTEGER),
MEMBER FUNCTION getBitRate RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getBitRate, WNDS, WNPS, RNDS, RNPS),
MEMBER FUNCTION getBitRate(ctx IN OUT RAW) RETURN INTEGER,

MEMBER PROCEDURE setKnownAttributes(
    knownFormat IN VARCHAR2,
    knownWidth IN INTEGER,
    knownHeight IN INTEGER,
    knownFrameResolution IN INTEGER,
    knownFrameRate IN INTEGER,
    knownVideoDuration IN INTEGER,
    knownNumberOfFrames IN INTEGER,
    knownCompressionType IN VARCHAR2,
    knownNumberOfColors IN INTEGER,
    knownBitRate IN INTEGER),

-- Methods associated with setting all the properties
MEMBER PROCEDURE setProperties(ctx IN OUT RAW),
MEMBER FUNCTION checkProperties(ctx IN OUT RAW) RETURN BOOLEAN,

MEMBER FUNCTION getAttribute(
    ctx IN OUT RAW,
    name IN VARCHAR2) RETURN VARCHAR2,

MEMBER PROCEDURE getAllAttributes(
    ctx IN OUT RAW,
    attributes IN OUT NOCOPY CLOB),

-- Methods associated with video processing
MEMBER FUNCTION processVideoCommand(
    ctx IN OUT RAW,
    cmd IN VARCHAR2,
    arguments IN VARCHAR2,
    result OUT RAW)
    RETURN RAW
);
```

パラメータは次のとおりです。

- **description:** ビデオ・オブジェクトの説明
- **source:** ビデオ・データが検出される **ORDSource**
- **format:** ビデオ・データを保存するフォーマット
- **mimeType:** MIME タイプについての情報
- **comments:** ビデオ・オブジェクトのコメント情報
- **width:** ビデオ・データの各フレームの幅
- **height:** ビデオ・データの各フレームの高さ
- **frameResolution:** ビデオ・データのフレームの解像度
- **frameRate:** ビデオ・データのフレーム・レート
- **videoDuration:** 格納されたビデオ・データの総再生時間
- **numberOfFrames:** ビデオ・データのフレーム数
- **compressionType:** ビデオ・データの圧縮タイプ
- **numberOfColors:** ビデオ・データの色数
- **bitRate:** ビデオ・データのビット・レート

5.2 メソッド

この項では、ビデオ・データ操作で使われる *Oracle8i interMedia* メソッドの参照情報について説明します。これらのメソッドを次のように分類して説明します。

updateTime 属性と関連付けられた ORDVideo メソッド

- `getUpdateTime()`: ビデオ・オブジェクトが最後に更新された時刻を返します。
- `setUpdateTime()`: ビデオ・オブジェクトの更新時刻を設定します。

title 属性と関連付けられた ORDVideo メソッド

- `setDescription()`: ビデオ・データの定義を設定します。
- `getDescription()`: ビデオ・データの定義を返します。

mimeType 属性と関連付けられた ORDVideo メソッド

- `setMimeType()`: 保存されたビデオ・データの MIME タイプを設定します。
- `getMimeType()`: ビデオ・データの MIME タイプを返します。

source 属性と関連付けられた ORDVideo メソッド

- `processSourceCommand()`: ソース・プラグインにコマンドおよび関連した引数を送信します。
- `isLocal`: データが BLOB にローカルに格納されている場合は真 (TRUE)、外部に格納されている場合は偽 (FALSE) を返します。
- `setLocal`: データが BLOB にローカル格納されていることを示すフラグを設定します。
- `clearLocal`: データが外部に格納されていることを示すフラグを消去します。
- `setSource()`: ビデオ・データの検出場所にソース情報を設定します。
- `getSource`: URL としてフォーマットされた外部データ・ソースについての完全な情報が含まれるフォーマット文字列を返します。
- `getSourceType`: ビデオ・データの外部ソース・タイプを返します。
- `getSourceLocation`: ビデオ・データの外部ソースの位置を返します。
- `getSourceName`: ビデオ・データの外部ソース名を返します。
- `import()`: `setSourceInformation()` をコールして指定した外部データ・ソースのデータを、Oracle データベース内のローカル・ソース (`localData`) に転送し、ローカル属性の値を "1" に設定します。この値は、ローカル・データであること、およびタイムスタンプを更新することを意味します。

- **importFrom():** 指定した外部データ・ソース（ソース・タイプ、位置、名前）のデータを、Oracle データベース内のローカル・ソース (**localData**) に転送し、ローカル属性の値を "1" に設定します。この値は、ローカル・データであること、およびタイムスタンプを更新することを意味します。
- **export():** Oracle データベース内のローカル・ソース (**localData**) のデータを、指定した外部データ・ソースに転送し、ローカル属性の値を "0" に設定します。この値は、外部データであること、およびソース情報をソースに格納することを意味します。

注意: *interMedia* によってネイティブにサポートされているソースには書き込むことができないため、エクスポート方法はサポートされていません。ユーザー定義のソースでは、エクスポート方法がサポートされていることがあります。

- **getContentLength():** データ・ソースの長さ（バイト数）を戻します。
- **getContentInLob():** コンテンツをテンポラリ LOB に戻します。
- **getContent:** コンテンツをローカルに格納するときに使われる BLOB に、ハンドルを戻します。
- **deleteContent:** ローカル BLOB のコンテンツを削除します。

ファイル操作に関連した ORDAudio メソッド

- **openSource():** データ・ソースをオープンします。
- **closeSource():** データ・ソースをクローズします。
- **trimSource():** データ・ソースを切り捨てます。
- **readFromSource():** ソースの開始位置から、n バイトのソースをバッファに読み込みます。
- **writeToSource():** ソースの開始位置から、n バイトのソースをバッファに書き込みます。

コメント属性に関連付けられた ORDVideo メソッド

- **appendToComments():** 指定した量のバッファおよびコメント・データを、ビデオ・データ・コメントの最後に追加します。
- **writeToComments():** 指定した量のバッファおよびコメント・データを、指定したオフセットで始まるビデオ・データ・コメントに書き込みます。
- **readFromComments():** 指定した量のコメント・データを、指定したオフセットで始まるビデオ・データのコメントから読み込みます。
- **locateInComments():** ビデオ・データのコメント内で、指定したパターンの文字列からオカレンスとの一致およびそこへの位置付けを行います。

- **trimComments():** ビデオ・データ・コメントを、指定した長さに調節します。
- **eraseFromComments():** 指定したオフセットで始まるビデオ・データ・コメントから、指定した量のコメント・データを消去します。
- **deleteComments:** ビデオ・データ・コメントを削除します。
- **loadCommentsFromFile():** 指定した BFILE のコメントを、ビデオ・データ・コメントにロードします。
- **copyCommentsOut():** ビデオ・データ・コメントを、指定した CLOB (Character LOB) にコピーします。
- **compareComments():** ビデオ・データ・コメントを、ビデオ・データの別の CLOB のコメントと比較します。
- **getCommentLength:** ビデオ・データの長さを戻します。

ビデオ属性アクセサと関連付けられた ORDVideo メソッド

次のメソッドは、ユーザー定義の形式プラグインによってのみサポートされます。

- **setFormat():** ビデオ・データのフォーマット (オブジェクト属性値) を設定します。
- **getFormat:** ビデオ・データが格納されるフォーマット (オブジェクト属性値) を戻します。
- **getFormat():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フォーマットを読み込みます。
- **setFrameSize():** ビデオ・データの各フレームの幅と高さ (オブジェクト属性値) をピクセル単位で設定します。
- **getFrameSize:** ビデオ・データの各フレームの幅と高さ (オブジェクト属性値) をピクセル単位で戻します。
- **getFrameSize():** 形式プラグインをコールして、格納済みのオーディオ・データに埋め込まれた各フレームの幅と高さの実値 (ピクセル単位) を読み込みます。
- **setFrameResolution():** ビデオ・データ・フレームのインチ当たりのピクセル数 (オブジェクト属性値) を設定します。
- **getFrameResolution:** ビデオ・データ・フレームのインチ当たりのピクセル数 (オブジェクト属性値) を戻します。
- **getFrameResolution():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレームの解像度を読み込みます。
- **setFrameRate():** ビデオ・データが録音された、1 秒当たりのフレーム・レート (オブジェクト属性値) を設定します。
- **getFrameRate:** ビデオ・データが録音された、1 秒当たりのフレーム・レート (オブジェクト属性値) を戻します。

- **getFrameRate():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレーム・レートを読み込みます。
- **setVideoDuration():** ビデオ・データ全体の再生に必要な合計時間（オブジェクト属性値）を設定します。
- **getVideoDuration:** ビデオ・データ全体の再生に必要な合計時間（オブジェクト属性値）を戻します。
- **getVideoDuration():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実ビデオ再生時間を読み込みます。
- **setNumberOfFrames():** ビデオ・データの合計フレーム数（オブジェクト属性値）を設定します。
- **getNumberOfFrames:** ビデオ・データの合計フレーム数（オブジェクト属性値）を戻します。
- **getNumberOfFrames():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレーム数を読み込みます。
- **setNumberOfColors():** ビデオ・データの色数（オブジェクト属性値）を設定します。
- **getNumberOfColors:** ビデオ・データの色数（オブジェクト属性値）を戻します。
- **getNumberOfColors():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実色数を読み込みます。
- **setBitRate():** ビデオ・データのビット・レート（オブジェクト属性値）を設定します。
- **getBitRate:** ビデオ・データのビット・レート（オブジェクト属性値）を戻します。
- **getBitRate():** 形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実ビット・レートを読み込みます。
- **setKnownAttributes():** ビデオ・データのフォーマット、フレーム・サイズ、フレームの解像度、フレーム・レート、ビデオ再生時間、フレーム数、圧縮タイプ、色数、およびビット・レートなどのビデオ属性を設定します。このメソッドをコールするとパラメータが渡されます。
- **setProperties():** ビデオ・データを読み込んでオブジェクト属性の値を取得し、その値をオブジェクトに格納します。ORDVideo によって認識される既知の属性の場合は、その属性のプロパティを設定します。既知の属性には、ビデオ・データのフォーマット、フレーム・サイズ、フレームの解像度、フレーム・レート、ビデオ再生時間、フレーム数、圧縮タイプ、色数、およびビット・レートがあります。
- **checkProperties():** 形式プラグインをコールして、ビデオ・データのフォーマット、フレーム・サイズ、フレームの解像度、フレーム・レート、ビデオ再生時間、フレーム数、圧縮タイプ、色数、およびビット・レートなどのプロパティをチェックします。オブジェクト属性に格納されたプロパティがビデオ・データのプロパティと一致する場合、真（TRUE）がブール値で戻されます。

- **getAttribute():** 要求された属性値を返します。このメソッドは、ユーザー定義の形式プラグインでのみ使うことができます。
- **getAllAttributes():** 使用するクライアント・アクセス用のフォーマット文字列を返します。ネイティブでサポートされるフォーマット文字列には、次のビデオ・データ属性がカンマ (,) で区切られて含まれます。format、frameSize、frameResolution、frameRate、videoDuration、numberOfFrames、compressionType、numberOfColors、および bitRate です。文字列は、ユーザー定義の形式プラグインにより定義されます。

ビデオ・データの処理に関連付けられた ORDVide オメソッド

- **processVideoCommand():** コマンドおよび関連した引数を、処理のために形式プラグインに送信します。

オブジェクト型およびメソッドの詳細は、『Oracle8i 概要』を参照してください。

5.2.1 表定義の例

この章の参照で使われるメソッドは、テスト・ビデオ表 TVID に基づいた例です。5.2.2 項から 5.2.9 項で紹介される例を学習する場合、次の TVID 表定義を参照してください。

TVID 表定義

```
CREATE TABLE TVID(n NUMBER, vid ORDSYS.ORDVIDEO)
storage (initial 100K next 100K pctincrease 0);

INSERT INTO TVID VALUES(1, ORDSYS.ORDVideo(
    NULL,
    ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE, NULL),
    NULL,
    NULL,
    EMPTY_CLOB(), 0,0,0,0,0,0,0,0,0,0)
);
INSERT INTO TVID VALUES(2, ORDSYS.ORDVideo(
    NULL,
    ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL,
SYSDATE,NULL),
    NULL,
    NULL,
    EMPTY_CLOB(), 0,0,0,0,0,0,0,0,0,0)
);
```

5.2.2 updateTime 属性と関連付けられた ORDVideo メソッド

この項では、updateTime 属性と関連付けられた ORDVideo メソッドの参照情報について説明します。

getUpdateTime メソッド

形式

```
getUpdateTime RETURN DATE;
```

説明

オブジェクトが最後に更新された時刻を返します。

パラメータ

なし

使用方法

なし

プラグマ

```
Pragma RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

ビデオ・データから更新時刻を取得します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  SELECT TVID INTO obj FROM TVID WHERE N = 1 ;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getUpdateTime, 'MM-DD-YYYY HH24:MI:SS'));
END;
/
```

setUpdateTime() メソッド

形式

```
setUpdateTime(current_time DATE);
```

説明

ビデオ・データが最後に更新された時刻を設定します。ビデオ・データを変更したときにこのメソッドを使います。**setDescription()** メソッド、**setMimeType()** メソッド、**setSource()** メソッド、**import()** メソッド、**importFrom()** メソッド、**export()** メソッド、**deleteContent** メソッド、およびすべてのビデオ・アクセサ設定では、暗黙的にこのメソッドがコールされます。

パラメータ

current_time

格納されるタイムスタンプ。デフォルト値は **SYSDATE** です。

使用方法

ビデオ・データを変更したときは、このメソッドをコールする必要があります。

プラグマ

なし

例外

なし

例

5-15 ページの「**getUpdateTime** メソッド」の例も参照してください。

ビデオ・データの更新時刻を設定します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N = 1;
  obj.setUpdateTime(SYSDATE);
  UPDATE TVID SET vid=obj WHERE N = 1;
  COMMIT;
END;
```

5.2.3 説明属性に関連付けられた ORDVideo メソッド

この項では、説明属性に関連する ORDVideo メソッドの参照情報を提供します。

setDescription() メソッド

形式

```
setDescription (user_description IN VARCHAR2);
```

説明

ビデオ・データの説明を設定します。

パラメータ

user_description
ビデオ・データの説明

使用方法

各ビデオ・オブジェクトには、クライアント・アプリケーションの理解に役立つ説明を指定する必要があります。たとえば、**Web** ベース・クライアントにビデオの説明をいくつか指定し、ユーザーはその説明からアクセスするビデオ・データを選択します。

Oracle8i *interMedia* に付属する **Web** アクセス・コンポーネントおよびその他のクライアント・コンポーネントでは、ビデオ・データを提供するときにこの説明属性を利用できます。

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

なし

例

ビデオ・データに説明属性を設定します。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('writing description');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.setDescription('video1');
```

```
DBMS_OUTPUT.PUT_LINE(obj.getDescription);  
UPDATE TVID SET vid=obj WHERE N=1;  
COMMIT;  
END;  
/
```

getDescription メソッド

形式

```
getDescription RETURN VARCHAR2;
```

説明

ビデオ・データの説明を戻します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getTitle, WNDS, WNPS, RNDS, RNPS)

例外

setDescription メソッドをコールしたときに説明が設定されていない場合は、DESCRIPTION_IS_NOT_SET 例外が発生します。

例

5-18 ページの「setDescription() メソッド」の例を参照してください。

5.2.4 mimeType 属性と関連付けられた ORDVideo メソッド

この項では、mimeType 属性と関連付けられた ORDVideo メソッドの参照情報について説明します。

setMimeType() メソッド

形式

```
setMimeType(mime IN VARCHAR2);
```

説明

ビデオ・データの **MIME** タイプを設定できます。

パラメータ

mime
MIME タイプ

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

ソースがファイルまたは **BLOB** の場合は、**setMimeType()** メソッドをコールして **MIME** タイプを設定します。

MIME タイプは、**HTTP** ソースのインポート時に **HTTP** ヘッダーから抽出されます。

プラグマ

なし

例外

setMimeType() メソッドをコールしたときに **MIME** タイプの値が **NULL** の場合は、**INVALID_MIME_TYPE** 例外が発生します。

例

格納済みのビデオ・データに対して **MIME** タイプを設定します。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('writing mimetype');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.setMimeType('video/avi');
    DBMS_OUTPUT.PUT_LINE(obj.getMimeType);
```

```
UPDATE TVID SET vid=obj WHERE N=1;  
COMMIT;  
END;  
/
```

getMimeType メソッド

形式

```
getMimeType RETURN VARCHAR2;
```

説明

ビデオ・データの **MIME** タイプを戻します。

パラメータ

なし

使用方法

ソースが **HTTP** サーバーの場合、**HTTP** ヘッダー情報から **MIME** タイプ情報が読み込まれます。ソースがファイルまたは **BLOB** の場合、**MIME** タイプを設定するには、**setMimeType()** メソッドをコールする必要があります。

プラグマ

```
Pragma RESTRICT_REFERENCES(getMimeType, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

5-22 ページの「**setMimeType()** メソッド」の例を参照してください。

5.2.5 ソース属性と関連付けられた ORDVideo メソッド

この項では、ソース属性と関連付けられた ORDVideo メソッドの参照情報について説明します。

processSourceCommand() メソッド

形式

```
processSourceCommand(  
    ctx          IN OUT RAW,  
    cmd          IN  VARCHAR2,  
    arguments    IN  VARCHAR2,  
    result       OUT RAW)  
  
RETURN RAW;
```

説明

コマンドとその引数をソース・プラグインに送信できます。このメソッドは、ユーザー定義のソース・プラグインでだけ利用できます。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。openSource() メソッドもコールする必要があります。詳細は、この章の初めの部分を参照してください。

cmd

ソース・プラグインによって認識されるすべてのコマンド

arguments

コマンドの引数

result

ソース・プラグインから戻されたこの関数のコールの結果

使用方法

このメソッドを使用して、コマンドとその引数をソース・プラグインに送信します。コマンドを受け取ると、解釈せずに処理のために渡します。

プラグマ

なし

例外

srcType の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が発生します。

processSourceCommand() メソッドが使用中のソース・プラグインによってサポートされていない場合は、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が発生します。

他の例外の発生時に、ソース・プラグインでこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

コマンドを処理します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR(4000);
  argList VARCHAR(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  select vid into obj from TVID where N =1 for UPDATE;
  -- assign command
  -- assign argList
  res := obj.processSourceCommand (ctx, command,
                                   argList, result);
UPDATE TVID SET vid=obj WHERE N=1 ;
COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('EXCEPTION caught');
END;
/
```

isLocal メソッド

形式

```
isLocal RETURN BOOLEAN;
```

説明

データが **BLOB** にローカルに格納されている場合は真 (**TRUE**)、外部に格納されている場合は偽 (**FALSE**) を返します。

パラメータ

なし

使用方法

ローカル属性値が **1** または **NULL** に設定されている場合は、このメソッドでは真 (**TRUE**) を返し、それ以外の場合は偽 (**FALSE**) を返します。

プラグマ

Pragma **RESTRICT_REFERENCES**(getLocal, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

データがローカルかどうかを判断します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N = 1 FOR UPDATE;
  if(obj.isLocal) then
    DBMS_OUTPUT.put_line('local is true');
  else
    DBMS_OUTPUT.put_line('local is false');
  endif;
END;
/
```

setLocal メソッド

形式

```
setLocal;
```

説明

データが BLOB の内部に格納されていることを示す、ローカル属性を設定します。ローカル・フラグが設定されているときは、ビデオ・メソッドでは `source.localData` 属性のビデオ・データを検索します。

パラメータ

なし

使用方法

このメソッドでは、ローカル属性に **1** を設定します。これは、そのデータがローカルに格納され、`localData` 属性であることを意味しています。

プラグマ

なし

例外

なし

例

データのフラグに、ローカルを設定します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  SELECT s INTO obj FROM TVID WHERE N = 1 FOR UPDATE;
  obj.setLocal;
  UPDATE TVID SET s=obj WHERE N = 1;
  COMMIT;
END;
/
```

clearLocal メソッド

形式

```
clearLocal;
```

説明

データが外部に格納されていることを示すために、ローカル・フラグをリセットします。ローカル・フラグが消去に設定されると、ビデオ・メソッドでは、**srcLocation**、**srcName**、および **srcType** 属性を使用してビデオ・データを検索します。

パラメータ

なし

使用方法

このメソッドでは、ローカル属性に **0** を設定します。これは、データが外部または **Oracle8i** 以外に格納されていることを意味します。

プラグマ

なし

例外

なし

例

データのローカル・フラグの値を消去します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  SELECT s INTO obj FROM TVID WHERE N = 1 FOR UPDATE;
  obj.clearLocal;
  UPDATE TVID SET s=obj WHERE N = 1;
  COMMIT;
END;
/
```

setSource() メソッド

形式

```
setSource(  
    source_type      IN VARCHAR2,  
    source_location  IN VARCHAR2,  
    source_name      IN VARCHAR2);
```

説明

ビデオ・データの外部ソースについての情報を設定または変更します。

パラメータ

source_type

外部データのソース・タイプ。詳細は、第 6 章の「ORDSource オブジェクト型」の説明を参照してください。

source_location

外部データのソースの位置。詳細は、第 6 章の「ORDSource オブジェクト型」の説明を参照してください。

source_name

外部データのソース名。詳細は、第 6 章の「ORDSource オブジェクト型」の説明を参照してください。

使用方法

ビデオ・データ・ソースを新しい BFILE または URL に設定できます。

このメソッドを使う前に、ディレクトリが存在することを確認してください。ディレクトリが存在しない場合は作成する必要があります。

このメソッドをコールすると、暗黙的に setUpdateTime() メソッドがコールされます。

プラグマ

なし

例外

なし

例

ビデオ・データをエクスポートする前に、ソースをエクスポート・ファイルに変更します。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('setting and getting source');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.setSource('LOCAL','VIDEODIR','video.dat');
    DBMS_OUTPUT.PUT_LINE(obj.getSource);
    UPDATE TVID SET vid=obj WHERE N=1;
    COMMIT;
END;
/
```

getSource メソッド

形式

```
getSource RETURN VARCHAR2;
```

説明

外部のビデオ・データの位置についての情報を、URL フォーマットで返します。

パラメータ

なし

使用方法

戻り値は、次のとおりです。

- FILE://<DIR OBJECT NAME>/<FILE NAME> (ソース・ファイルの場合)
- HTTP://<URL> (HTTP ソースの場合)
- ユーザー定義のソース

プラグマ

Pragma RESTRICT_REFERENCES(getSource, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

5-32 ページの「setSource() メソッド」の例を参照してください。

getSourceType メソッド

形式

```
getSourceType RETURN VARCHAR2;
```

説明

外部ビデオ・データ・ソースのタイプを含む文字列を返します。

パラメータ

なし

使用方法

このメソッドでは、外部ビデオ・データ・ソースのタイプを含む **VARCHAR2** 文字列（たとえば、**'FILE'** など）が返されます。

プラグマ

Pragma RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・データ・ソースについてのソース・タイプ情報を取得します。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('setting and getting source');
    DBMS_OUTPUT.PUT_LINE('-----');
    -- set source to a file
    obj.setSource('FILE', 'VIDEODIR', 'MV1.AVI');
    -- get source information
    DBMS_OUTPUT.put_line(obj.getSource);
    DBMS_OUTPUT.put_line(obj.getSourceType);
    DBMS_OUTPUT.put_line(obj.getSourceLocation);
    DBMS_OUTPUT.put_line(obj.getSourceName);
    UPDATE TVID SET vid=obj WHERE N=1;
    COMMIT;
```

```
EXCEPTION
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

getSourceLocation メソッド

形式

```
getSourceLocation RETURN VARCHAR2;
```

説明

外部ビデオ・データ・ソースの位置の値を含む文字列を返します。

パラメータ

なし

使用方法

このメソッドでは、外部ビデオ・データの位置の値を含む **VARCHAR2** 文字列（たとえば、'BFILEDIR' など）が返されます。

プラグマ

Pragma RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS)

例外

srcLocation の値が **NULL** の場合は、このメソッドをコールすると、**ORDSourceExceptions.INCOMPLETE_SOURCE_LOCATION** 例外が発生します。

例

5-34 ページの「getSourceType メソッド」の例を参照してください。

getSourceName メソッド

形式

```
getSourceName RETURN VARCHAR2;
```

説明

外部ビデオ・データ・ソースの名前を含む文字列を返します。

パラメータ

なし

使用方法

このメソッドでは、外部ビデオ・データ・ソースの名前を含む VARCHAR2 文字列（たとえば、'testvid.dat' など）が返されます。

プラグマ

```
Pragma RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS)
```

例外

srcName の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.INCOMPLETE_SOURCE_NAME 例外が発生します。

例

5-34 ページの「getSourceType メソッド」の例を参照してください。

import() メソッド

形式

```
import (ctx IN OUT RAW);
```

説明

外部ビデオ・データ・ソースのビデオ・データを、Oracle データベース内のローカル・ソース (localData) に転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。openSource() メソッドもコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

import メソッドをコールする前に、setSource() メソッドを使用して、外部ソース・タイプ、位置、および名前を設定します。

このメソッドを使う前にディレクトリが存在することを確認してください。ディレクトリが存在しない場合は作成する必要があります。

外部ビデオ・データ・ソースのデータをローカル・ソース (Oracle データベース内) にインポートしても、ソース情報は変更されません。つまり、データのインポート元のソースを指しています。

このメソッドをコールすると、暗黙的に setUpdateTime() および setLocal メソッドがコールされます。

プラグマ

なし

例外

srcType の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が発生します。

dlob の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.NULL_SOURCE 例外が発生します。

import() メソッドが使用中のソース・プラグインによってサポートされていない場合は、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、
ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が発生します。
 これらの例外の詳細は、付録 H を参照してください。

例

ビデオ・データをインポートするには、最初にソースを設定します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('setting and getting source');
  DBMS_OUTPUT.PUT_LINE('-----');
  -- set source to a file
  obj.setSource('FILE','VIDEODIR','testvid.dat');
  -- get source information
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  -- import data
  obj.import(ctx);
  -- check size
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  DBMS_OUTPUT.PUT_LINE('deleting contents');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.deleteContent;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  UPDATE TVID SET vid=obj WHERE N=1;
  COMMIT;
END;
/
```

importFrom() メソッド

形式

```
importFrom(ctx IN OUT RAW,  
           source_type      IN VARCHAR2,  
           source_location  IN VARCHAR2,  
           source_name      IN VARCHAR2);
```

説明

指定した外部ビデオ・データ・ソースのビデオ・データを、Oracle データベース内のローカル・ソース（localData）に転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。openSource() メソッドもコールする必要があります。詳細は、この章の初めの部分を参照してください。

source_type

ビデオ・データのソース・タイプ

source_location

ビデオ・データのインポート元の位置

source_name

ビデオ・データの名前

使用方法

このメソッドは、ソース情報を個別に指定せずにメソッド内で指定すること以外は、import() メソッドと同じです。

このメソッドを使う前にディレクトリが存在することを確認してください。ディレクトリが存在しない場合は作成する必要があります。

外部ビデオ・データ・ソースのデータをローカル・ソース（Oracle データベース内）にインポートすると、入力値にソース情報（データのインポート元のソースを指しています）が設定されます。

このメソッドをコールすると、暗黙的に setUpdateTime() および setLocal メソッドがコールされます。

プラグマ

なし

例外

dlob の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.NULL_SOURCE 例外が発生します。

importFrom() メソッドが、使用中のソース・プラグインによってサポートされていない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

指定した外部データ・ソースのビデオ・データを、ローカル・ソースにインポートします。

```
DECLARE
  obj ORDSYS.ORDVideo;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE('setting and getting source');
  DBMS_OUTPUT.PUT_LINE('-----');
  -- import data
  obj.importFrom(ctx, 'FILE', 'VIDEODIR', 'MV1.AVI');
  -- check size
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContenLength(ctx)));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.GETLENGTH(obj.getConten)));
  DBMS_OUTPUT.PUT_LINE(obj.getSource);
  DBMS_OUTPUT.PUT_LINE('deleting contents');
  DBMS_OUTPUT.PUT_LINE('-----');
  obj.deleteContent;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContenLength(ctx)));
  UPDATE TVID SET vid=obj WHERE N=1;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
```

importFrom() メソッド

```
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('EXCEPTION Caught');
    END;
/
```

export() メソッド

形式

```
export (
    ctx                IN OUT RAW,
    source_type        IN VARCHAR2,
    source_location    IN VARCHAR2,
    source_name        IN VARCHAR2);
```

説明

ローカル・ソース (**localData**) のビデオ・データを、Oracle データベース内の外部ビデオ・データ・ソースに転送します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報

source_type

データのエクスポート先のソース・タイプ

source_location

ビデオ・データのエクスポート先の位置

source_name

データのエクスポート先のビデオ・オブジェクト名

使用方法

ビデオ・データをエクスポートしても、すべてのビデオ属性は変更されません。また **srcType**、**srcLocation**、および **srcName** は入力値によって更新されます。**export** メソッドをコールした後で、ローカル・データのコンテンツを削除するには、**deleteContent** メソッドをコールします。

サーバー側では、**export** メソッドはネイティブでサポートされていません。このメソッドは、**export** メソッドをサポートしているユーザー定義のソースの場合に使うことができます。

プラグマ

なし

例外

srcType の値が NULL の場合は、このメソッドをコールすると、ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が発生します。

export() メソッドが使用中のソース・プラグインによってサポートされていない場合、ORDSourceExceptions.METHOD_NOT_SUPPORTED 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

ローカル・ソースのビデオ・データを、外部ビデオ・データ・ソースにエクスポートします。

```
DECLARE
    obj ORDSYS.ORDVideo;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N =1;
    obj.export(ctx, 'FILE', 'VIDEODIR', 'x.mov');
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.put_line('OTHER EXCEPTION caught');
END;
/
```

getContenLength() メソッド

形式

```
getContenLength (ctx IN OUT RAW) RETURN INTEGER;
```

説明

ソースに格納されているビデオ・データ・コンテンツの長さを戻します。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報

使用方法

このメソッドをサポートしていないソース・タイプもあります。たとえば、"HTTP" タイプ・ソースでは、このメソッドはサポートされません。"HTTP" タイプ・ソースに対してこのメソッドをインプリメントするには、独自に修正した "HTTP" タイプ・ソースを定義し、そのソースに対してこのメソッドをインプリメントする必要があります。

プラグマ

Pragma RESTRICT_REFERENCES(getContenLength, WNDS, WNPS, RNDS, RNPS)

例外

srcType の値が NULL で、データが BLOB にローカル格納されていない場合は、このメソッドをコールすると、ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION 例外が発生します。

この例外の詳細は、付録 H を参照してください。

例

5-39 ページの「import() メソッド」の例を参照してください。

getContentInLob() メソッド

形式

```
getContentInLob(  
    ctx          IN OUT RAW,  
    dest_lob     IN OUT NOCOPY BLOB,  
    mimeType     OUT VARCHAR2,  
    format       OUT VARCHAR2)
```

説明

データ・ソースのデータを、指定した **BLOB** に転送します。オブジェクトの **BLOB** 以外の **BLOB** を指定することもできます。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報

dest_lob

データを受け取る LOB

mimeType

データの MIME タイプ。戻されないこともあります。

format

データの形式。戻されないこともあります。

使用方法

なし

プラグマ

なし

例外

srcType の値が **NULL** の場合は、このメソッドをコールすると、**ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が発生します。

getContentInLob() メソッドが使用中のソース・プラグインによってサポートされていない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が発生します。
これらの例外の詳細は、付録 H を参照してください。

例

データ・ソースのデータを取得し、ローカル・ソースの指定した BLOB に格納します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  tempBLob BLOB;
  mimeType VARCHAR2(4000);
  format VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N = 1 ;
  if(obj.isLocal) then
    DBMS_OUTPUT.put_line('local is true');
  end if;
  DBMS_LOB.CREATETEMPORARY(tempBLob, true, 10);
  obj.getContentInLob(ctx,tempBLob, mimeType,format);
  -- process tempBLob
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.getLength(tempBLob)));
EXCEPTION
WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
  DBMS_OUTPUT.put_line('ORDVideoExceptions.METHOD_NOT_SUPPORTED caught');
WHEN OTHERS THEN
  DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

getContent メソッド

形式

```
getContent RETURN BLOB;
```

説明

ローカル BLOB 記憶領域 (ORDVideo オブジェクト内の BLOB) にハンドルを戻します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getContent, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

Web ベースのプレーヤーに配置するビデオ・データに、クライアントからアクセスします。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('setting and getting source');
    DBMS_OUTPUT.PUT_LINE('-----');
    -- import data
    obj.importFrom(ctx, 'FILE', 'VIDEODIR', 'MV1.AVI');
    -- check size
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.GETLENGTH(obj.getContent)));
    DBMS_OUTPUT.PUT_LINE(obj.getSource);
    DBMS_OUTPUT.PUT_LINE('deleting contents');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.deleteContent;
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
```

```
UPDATE TVID SET vid=obj WHERE N=1;
COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('EXCEPTION Caught');
END;
/
```

deleteContent メソッド

形式

```
deleteContent;
```

説明

現在のローカル・ソース (**localData**) のローカル・データを削除します。

パラメータ

なし

使用方法

ローカル・ソースのデータを外部ビデオ・データ・ソースにエクスポートし、このデータがローカル・ソースで必要なくなったときに、このメソッドをコールできます。

そのオブジェクトを新しいオブジェクトで更新する場合、このメソッドをコールします。

プラグマ

なし

例外

なし

例

5-39 ページの「**import()** メソッド」の例を参照してください。

5.2.6 ファイル関係の操作に関連付けられた ORDVideo メソッド

この項では、データ・ソース上で行うファイル関係の操作に関連付けられた **ORDVideo** メソッドの参照情報について説明します。この項で説明するメソッドは、特にビデオ・データを操作する場合に使うことができます。

openSource() メソッド

形式

```
openSource(userArg IN RAW, ctx OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをオープンします。

パラメータ

userArg

ユーザー引数。ユーザー定義のソース・プラグインで使うことができます。

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。**openSource()** メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

成功した場合は **INTEGER** に **0** (ゼロ)、失敗した場合は **>0** (たとえば、**1**) が戻されます。正確な数値とその意味は、プラグインにより定義されます。たとえば、ファイル・プラグインの場合、**1** は " ファイルが見つかりません "、**2** は " 指定されたディレクトリが見つかりません " になります。

プラグマ

なし

例外

srcType の値が **NULL** でデータがローカルに存在しない場合は、このメソッドをコールすると、**ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が発生します。

openSource() メソッドが使用中のソース・プラグインによってサポートされていない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

ローカル・データ・ソースを開きます。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
  ctx RAW(4000) :=NULL;
  userArg RAW(4000);
BEGIN
  select vid into obj from TVID where N =1 for UPDATE;
  res := obj.openSource(userArg, ctx);
  UPDATE TVID SET vid =obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

closeSource() メソッド

形式

```
closeSource(ctx IN OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをクローズします。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。**openSource()** メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

使用方法

成功した場合は **INTEGER** に **0** (ゼロ)、失敗した場合は **>0** (たとえば **1**) が戻されます。正確な数字とその意味は、プラグインにより定義されます。たとえば、ファイル・プラグインの場合、**1** は " ファイルが見つかりません "、**2** は " 指定されたディレクトリが見つかりません " になります。

プラグマ

なし

例外

srcType の値が **NULL** でデータがローカルに存在しない場合は、このメソッドをコールすると、**ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が発生します。

closeSource() メソッドが使用中のソース・プラグインによってサポートされていない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

外部 **BFILE** データ・ソースをクローズします。

```
DECLARE  
  obj ORDSYS.ORDVideo;
```

```
res INTEGER;  
ctx RAW(4000) :=NULL;  
BEGIN  
  select vid into obj from TVID where N =2 for UPDATE;  
  obj.source.clearLocal;  
  res := obj.closeSource(ctx);  
  UPDATE TVID SET vid=obj WHERE N=2 ;  
  COMMIT;  
  EXCEPTION  
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN  
      DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');  
    WHEN OTHERS THEN  
      DBMS_OUTPUT.put_line('EXCEPTION caught');  
END;  
/
```

trimSource() メソッド

形式

```
trim(ctx IN OUT RAW,  
      newlen IN INTEGER) RETURN RAW;
```

説明

データ・ソースを切り捨てます。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。**openSource()** メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

newlen

切り捨て後の新しい長さ

使用方法

成功した場合は **INTEGER** に **0** (ゼロ)、失敗した場合は **>0** (たとえば **1**) が戻されます。正確な数字とその意味は、プラグインにより定義されます。たとえば、ファイル・プラグインの場合、**1** は " ファイルが見つかりません "、**2** は " 指定されたディレクトリが見つかりません " になります。

プラグマ

なし

例外

srcType の値が **NULL** でデータがローカルに存在しない場合は、このメソッドをコールすると、**ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION** 例外が発生します。

trimSource() メソッドが使用中のソース・プラグインによってサポートされていない場合、**ORDSourceExceptions.METHOD_NOT_SUPPORTED** 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、**ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION** 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

ローカル・データ・ソースを切り捨てます。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
  ctx RAW(4000) :=NULL;
BEGIN
  select vid into obj from TVID where N =1 for UPDATE;
  res := obj.trimSource(ctx,0);
  UPDATE TVID SET vid=obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DEMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DEMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

readFromSource() メソッド

形式

```
readFromSource(  
    ctx          IN OUT RAW,  
    startPos     IN INTEGER,  
    numBytes     IN OUT INTEGER,  
    buffer       OUT RAW);
```

説明

ソースの開始位置から、**n** バイトのソースをバッファに読み込むことができます。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。**openSource()** メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

startPos

データ・ソースの開始位置

numBytes

データ・ソースから読み込むバイト数

buffer

データの読み込み先バッファ

使用方法

このメソッドでは、**HTTP** ソースをサポートしていません。

"HTTP" ソース・タイプを正しく読み込むには、すべての **URL** ソースを読み込む必要があります。"HTTP" ソース・タイプに対して **read** メソッドをインプリメントする場合は、**HTTP** ソース・タイプ用に変更したソース・プラグイン内で、このメソッドを独自にインプリメントする必要があります。

プラグマ

なし

例外

`srcType` の値が `NULL` でデータがローカルに存在しない場合は、このメソッドをコールすると、`ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION` 例外が発生します。

データはローカルに存在するが `srcType` の値が `NULL` である場合は、このメソッドをコールすると、`ORDSourceExceptions.NULL_SOURCE` 例外が発生します。

`readFromSource()` メソッドが使用中のソース・プラグインによってサポートされていない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、`ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

ソースからバッファに読み込みます。

```
DECLARE
  obj ORDSYS.ORDVideo;
  buffer RAW(4000);
  i INTEGER;
  ctx RAW(4000) :=NULL;
BEGIN
  i := 20;
  select vid into obj from TVID where N =1 ;
  obj.readFromSource(ctx,1,i,buffer);
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
  DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
  DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
```

writeToSource() メソッド

形式

```
writeToSource(  
    ctx          IN OUT RAW,  
    startPos     IN INTEGER,  
    numBytes     IN OUT INTEGER,  
    buffer       IN RAW);
```

説明

ソースの開始位置から、**n** バイトのソースをバッファに書き込むことができます。

パラメータ

ctx

ソース・プラグイン・コンテキスト情報。必ず割り当てる必要があります。**openSource()** メソッドをコールする必要があります。詳細は、この章の初めの部分を参照してください。

startPos

バッファのコピー先のソースの開始位置

numBytes

ソースに書き込まれるバイト数

buffer

データの書き込み先のバッファ

使用方法

このメソッドでは、書込み可能なソースに、任意のバイト位置から **n** バイトを書き込めることを前提としています。**FILE** および **HTTP** ソース・タイプは書き込み可能ソースではないため、このメソッドをサポートしていません。データがローカル **BLOB** に格納されている場合、またはユーザー定義ソース・プラグインを使用してデータにアクセスできる場合に、このメソッドを使うことができます。

プラグマ

なし

例外

`srcType` の値が `NULL` でデータがローカルに存在しない場合は、このメソッドをコールすると、`ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION` 例外が発生します。

データはローカルに存在するが `srcType` の値が `NULL` である場合は、このメソッドをコールすると、`ORDSourceExceptions.NULL_SOURCE` 例外が発生します。

`writeToSource()` メソッドが使用中のソース・プラグインによってサポートされていない場合、`ORDSourceExceptions.METHOD_NOT_SUPPORTED` 例外が発生します。

他の例外の発生時にソース・プラグインでこのメソッドをコールすると、`ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION` 例外が発生します。

これらの例外の詳細は、付録 H を参照してください。

例

バッファからソースに書き込みます。

```
DECLARE
  obj ORDSYS.ORDVideo;
  n INTEGER := 6;
  ctx RAW(4000) :=NULL;
BEGIN
  select vid into obj from TVID where N =1 for update;
  obj.writeToSource(ctx,1,n,UTL_RAW.CAST_TO_RAW('helloP'));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(obj.getContentLength(ctx)));
  update TVID set vid = obj where N = 1;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('ORDSourceExceptions.METHOD_NOT_SUPPORTED caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
```

5.2.7 コメント属性に関連する ORDVide オメソッド

この項では、コメント属性に関連する ORDVide オメソッドの参照情報を提供します。

appendToComments() メソッド

形式

```
appendToComments (amount IN BINARY_INTEGER,  
                  buffer IN VARCHAR2);
```

説明

指定した量のバッファおよびコメント・データを、ビデオ・オブジェクトのコメント属性の最後に追加します。

パラメータ

amount

追加されるコメント・データの量

buffer

追加されるコメント・データのバッファ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

コメント情報を、ビデオ・オブジェクトのコメント属性に追加します。

```
DECLARE  
  obj ORDSYS.ORDVideo;  
  i INTEGER;  
  j INTEGER;  
BEGIN
```

appendToComments() メソッド

```
SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
obj.writeToComments(1,18,'This is a Comments');
obj.appendToComments(18,'This is a Comments');
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,obj.getCommentLength));
DBMS_OUTPUT.PUT_LINE(obj.locateInComments('Comments',1));
obj.trimComments(18);
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,18));
i := 8;
j := 9;
obj.eraseFromComments(i,j);
DBMS_OUTPUT.PUT_LINE(obj.readFromComments(1,10));
obj.deleteComments;
UPDATE TVID SET vid=obj WHERE N=1;
COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

writeToComments() メソッド

形式

```
writeToComments(offset IN INTEGER,  
                 amount IN BINARY_INTEGER,  
                 buffer IN VARCHAR2);
```

説明

指定した量のコメント・バッファ・データを、指定したオフセットで始まるビデオ・オブジェクトのコメント属性に書き込みます。

パラメータ

offset

コメント・データが書き込まれる、コメント内のオフセット開始位置

amount

書き込むコメント・データの量

buffer

コメント・データの書き込み先バッファ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

readFromComments() メソッド

形式

```
readFromComments(offset IN INTEGER,  
                  amount IN BINARY_INTEGER :=32767)  
RETURN VARCHAR2;
```

説明

指定した量のコメント・データを、指定したオフセットで始まるビデオ・オブジェクトのコメント属性から読み込みます。

パラメータ

offset

コメント・データを読み込む、コメント内のオフセット開始位置

amount

読み込むコメント・データの量

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(readFromComments, WNDS, WNPS, RNDS, RNPS)

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

locateInComments() メソッド

形式

```
locateInComments (pattern      IN VARCHAR2,  
                  offset       IN INTEGER := 1,  
                  occurrence    IN INTEGER := 1)  
RETURN INTEGER;
```

説明

指定したオフセットで始まるビデオ・オブジェクトのコメント属性内で、指定したパターンの文字列データから、**n** 番目のオカレンスとの一致およびそこへの位置付けを行います。

パラメータ

pattern

検索するコメント・データのパターン

offset

一致の検索を開始するコメント内の、オフセット開始位置

occurrence

コメント・データのパターンが検出されたコメント内の、**n** 番目のオカレンス

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

trimComments() メソッド

形式

```
trimComments(newlen IN INTEGER);
```

説明

ビデオ・オブジェクトのコメントの長さを、指定した新しい長さに切り捨てます。

パラメータ

newlen
切捨て後の、コメントの新しい長さ

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

eraseFromComments() メソッド

形式

```
eraseFromComments(amount IN OUT NOCOPY INTEGER,  
                   offset IN INTEGER := 1);
```

説明

指定したオフセットで始まるビデオ・オブジェクトのコメント属性から、指定した量のコメント・データを消去します。

パラメータ

amount

消去するコメント・データの量

offset

コメント・データを消去するコメントの、オフセット開始位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

deleteComments() メソッド

形式

```
deleteComments;
```

説明

ビデオ・オブジェクトのコメント属性を削除します。

パラメータ

なし

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-63 ページの「appendToComments() メソッド」の例を参照してください。

loadCommentsFromFile() メソッド

形式

```
loadCommentsFromFile(fileobj IN BFILE,  
                     amount  IN INTEGER,  
                     from_loc IN INTEGER := 1,  
                     to_loc  IN INTEGER := 1);
```

説明

指定した量の **BFILE** のコメント・データを、指定したオフセットで始まるビデオ・オブジェクトのコメント属性にロードします。

パラメータ

fileobj

ロードするファイル・オブジェクト

amount

BFILE からロードするコメント・データの量

from_loc

コメントのロード元の **BFILE** の位置

to_loc

コメントのロード先の位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、**DBMS_LOB** ファンクションおよびプロシージャの動作と類似しています。**DBMS_LOB** ファンクションおよびプロシージャで発生する例外の一覧は、『**Oracle8i** パッケージ・プロシージャ リファレンス』の「**DBMS_LOB**」のパッケージの説明を参照してください。

例

BFILE のコメント情報を、ビデオ・データのコメントにロードします。

```
DECLARE
    file_handle BFILE;
    obj ORDSYS.ORDVideo;
    isopen BINARY_INTEGER;
    amount INTEGER;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    --file_handle := BFILENAME(obj.getSourceLocation, obj.getSourceName);
    file_handle := BFILENAME('VIDEODIR', 'testvid.dat');
    isopen := DBMS_LOB.FILEISOPEN(file_handle);
    IF isopen = 0 THEN
        --dbms_output.put_line('File Not Open');
        DBMS_LOB.FILEOPEN(file_handle, DBMS_LOB.FILE_READONLY);
    END IF;
    --dbms_output.put_line('File is now Open');
    isopen := DBMS_LOB.FILEISOPEN(file_handle);
    IF isopen <> 0 THEN
        amount := DBMS_LOB.GETLENGTH(file_handle);
    END IF;
    obj.loadCommentsFromFile(file_handle, 18, 1, 18);
    dbms_output.put_line(obj.getCommentLength);
    UPDATE TVID SET vid=obj WHERE N=1;
    COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

copyCommentsOut() メソッド

形式

```
copyCommentsOut (dest      IN OUT NOCOPY CLOB,  
                  amount    IN  INTEGER,  
                  from_loc  IN  INTEGER := 1,  
                  to_loc    IN  INTEGER := 1);
```

説明

指定した量のビデオ・オブジェクトのコメント属性を、指定された **CLOB** にコピーします。

パラメータ

dest

コメントのコピー先

amount

コピーするコメント・データの量

from_loc

コメントのコピー元の位置

to_loc

コメントのコピー先の位置

使用方法

なし

プラグマ

なし

例外

このメソッドで例外が発生する動作は、**DBMS_LOB** ファンクションおよびプロシージャの動作と類似しています。**DBMS_LOB** ファンクションおよびプロシージャで発生する例外の一覧は、『*Oracle8i* パッケージ・プロシージャ リファレンス』の「**DBMS_LOB**」のパッケージの説明を参照してください。

例

ビデオ・データを、指定された CLOB にコピーします。

```
DECLARE
    file_handle BFILE;
    obj ORDSYS.ORDVideo;
    obj1 ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj1 FROM TVID WHERE N=2 FOR UPDATE;
    SELECT vid INTO obj FROM TVID WHERE N=1;
    obj.copyCommentsOut(obj1.comments,obj.getCommentLength,1,10);
    DBMS_OUTPUT.put_line(obj1.getCommentLength);
    DBMS_OUTPUT.put_line(obj.getCommentLength);
    UPDATE TVID SET vid=obj1 WHERE N=2;
    COMMIT;
    EXCEPTION
        WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

compareComments() メソッド

形式

```
compareComments(compare_with_lob      IN CLOB,  
                amount                IN INTEGER := 4294967295,  
                starting_pos_in_comment IN INTEGER := 1,  
                starting_pos_in_compare IN INTEGER := 1)  
RETURN INTEGER;
```

説明

指定した量のビデオ・データのコメントを、指定した他のコメントと比較します。

パラメータ

compare_with_lob

比較コメント

amount

比較コメントと比較するビデオ・データ・コメントの量

starting_pos_in_comment

ビデオ・オブジェクトのコメント属性内の開始位置

starting_pos_in_compare

比較コメント内の開始位置

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(compareComments, WNDS, WNPS, RNDS, RNPS)

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

ビデオ・データのコメントを、他の CLOB のコメントと比較します。

```
DECLARE
    file_handle BFILE;
    obj ORDSYS.ORDVideo;
    obj1 ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=2 ;
    SELECT vid INTO obj1 FROM TVID WHERE N=1;
    DBMS_OUTPUT.put_line('comparison output');
    DBMS_OUTPUT.put_line(obj.compareComments(obj1.comments,obj.getCommentLength,
    1,18));
    EXCEPTION
        WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

getCommentLength() メソッド

形式

```
getCommentLength RETURN INTEGER;
```

説明

ビデオ・オブジェクトのコメント属性の長さを戻します。

パラメータ

なし

使用方法

なし

プラグマ

```
Pragma RESTRICT_REFERENCES(getCommentLength, WNDS, WNPS, RNDS, RNPS)
```

例外

このメソッドで例外が発生する動作は、DBMS_LOB ファンクションおよびプロシージャの動作と類似しています。DBMS_LOB ファンクションおよびプロシージャで発生する例外の一覧は、『Oracle8i パッケージ・プロシージャ リファレンス』の「DBMS_LOB」のパッケージの説明を参照してください。

例

5-76 ページの「compareComments() メソッド」の例を参照してください。

5.2.8 ビデオ属性アクセサと関連付けられた ORDVideo メソッド

この項では、ビデオ属性アクセサと関連付けられた ORDVideo メソッドの参照情報について説明します。

setFormat() メソッド

形式

```
setFormat (knownFormat IN VARCHAR2);
```

説明

ビデオ・オブジェクトのフォーマット属性を設定します。

パラメータ

knownFormat

オーディオ・オブジェクトに設定するビデオ・データの既知のフォーマット

使用方法

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

`knownFormat` パラメータの値が `NULL` の場合は、このメソッドをコールすると、`NULL_INPUT_VALUE` 例外が発生します。

例

格納済みのビデオ・データのフォーマットを設定します。

```
DECLARE
    obj ORDSYS.ORDVideo;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    DBMS_OUTPUT.PUT_LINE('writing format');
    DBMS_OUTPUT.PUT_LINE('-----');
    obj.setFormat('avi');
    DBMS_OUTPUT.PUT_LINE(obj.getFormat);
    UPDATE TVID SET vid=obj WHERE N=1;
    COMMIT;
END;
/
```

getFormat メソッド

形式

```
getFormat RETURN VARCHAR2;
```

説明

ビデオ・オブジェクトのフォーマット属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getStoredFormat, WNDS, WNPS, RNDS, RNPS)

例外

フォーマットの値が **NULL** の場合は、このメソッドをコールすると、**VIDEO_FORMAT_IS_NULL** 例外が発生します。

例

5-79 ページの「**setFormat()** メソッド」の例を参照してください。

getFormat() メソッド

形式

```
getFormat (ctx IN OUT RAW) RETURN VARCHAR2;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたフォーマットを読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getFormat()** メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、フォーマットを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使います。

ビデオ・ファイル・フォーマット情報は、書式化されたビデオ・データから抽出できます。**ORDVideo** オブジェクトで認識されないファイル形式に対してサポートを拡張するには、そのファイル形式がサポートされている **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときにビデオ・プラグインで例外が発生した場合、**getFormat()** メソッドでは **VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

データベースに格納されたビデオ・データのファイル形式を戻します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
```

getFormat() メソッド

```
SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
res := obj.getFormat(ctx);
DBMS_OUTPUT.put_line(res );
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE_PLUGIN_EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO_PLUGIN_EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setFrameSize() メソッド

形式

```
setFrameSize(  
    knownWidth  IN INTEGER,  
    knownHeight IN INTEGER);
```

説明

ビデオ・オブジェクトの高さおよび幅属性の値を設定します。

パラメータ

knownWidth
フレームの幅（ピクセル）

knownHeight
フレームの高さ（ピクセル）

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

プラグマ

なし

例外

knownWidth または **knownHeight** パラメータ値のいずれかが **NULL** の場合は、このメソッドをコールすると、**NULL_INPUT_VALUE** 例外が発生します。

例

ビデオ・データのフレーム・サイズを設定します。

```
DECLARE  
    obj ORDSYS.ORDVideo;  
BEGIN  
    select vid into obj from TVID where N =1 for update;  
    obj.setFrameSize(1,2);  
    obj.setFrameResolution(4);  
    obj.setFrameRate(5);  
    obj.setVideoDuration(20);
```

setFrameSize() メソッド

```
obj.setNumberOfFrames(8);  
obj.setCompressionType('Cinepak');  
obj.setBitRate(1500);  
obj.setNumberOfColors(256);  
update TVID set vid = obj where N = 1;  
COMMIT;  
END;  
/
```

getFrameSize() メソッド

形式

```
getFrameSize(  
    retWidth OUT INTEGER,  
    retHeight OUT INTEGER);
```

説明

ビデオ・オブジェクトの高さおよび幅属性の値を返します。

パラメータ

retWidth
フレームの幅（ピクセル）

retHeight
フレームの高さ（ピクセル）

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・データのフレーム・サイズを返します。

```
DECLARE  
    obj ORDSYS.ORDVideo;  
    width INTEGER;  
    height INTEGER;  
BEGIN  
    SELECT vid INTO obj FROM TVID WHERE N=1 ;  
    obj.getFrameSize(width, height);  
    DBMS_OUTPUT.put_line('width :' || width);  
    DBMS_OUTPUT.put_line('height :' || height);
```

getFrameSize() メソッド

```
END;  
/
```

getFrameSize() メソッド

形式

```
getFrameSize(  
    ctx      IN OUT RAW,  
    width OUT INTEGER,  
    height OUT INTEGER);
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたフレーム・サイズを読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

width
フレームの幅（ピクセル）

height
フレームの高さ（ピクセル）

使用方法

ビデオ・フレーム・サイズ情報は、書式化されたビデオ・データのヘッダーに表示されます。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getFrameSize()** メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、フレーム・サイズを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使います。

ビデオ・フレーム・サイズ情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットに対してサポートを拡張するには、そのフォーマットがサポートされている **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレーム・サイズを読み込みます。

```
DECLARE
    obj ORDSYS.ORDVideo;
    width VARCHAR2(4000);
    height VARCHAR2(4000);
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    obj.getFrameSize(ctx,width, height);
    EXCEPTION
        WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
        WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
            DBMS_OUTPUT.put_line('SOURCE_PLUGIN_EXCEPTION caught');
        WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
            DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
        WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
            DBMS_OUTPUT.put_line('VIDEO_PLUGIN_EXCEPTION caught');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setFrameResolution() メソッド

形式

```
setFrameResolution(knownFrameResolution IN INTEGER);
```

説明

ビデオ・オブジェクトの **frameResolution** 属性の値を設定します。

パラメータ

knownFrameResolution

1 インチ当たりのピクセル単位で表された既知のフレームの解像度

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

プラグマ

なし

例外

knownFrameResolution パラメータの値が **NULL** の場合は、このメソッドをコールすると、**NULL_INPUT_VALUE** 例外が発生します。

例

5-83 ページの「**setFrameSize()** メソッド」の例を参照してください。

getFrameResolution メソッド

形式

```
getFrameResolution RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **frameResolution** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・データのフレームの解像度の値を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getFrameResolution;
  DBMS_OUTPUT.put_line('resolution : ' || res);
END;
/
```

getFrameResolution() メソッド

形式

```
getFrameResolution(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたフレームの解像度を読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

ビデオのフレームの解像度情報は、書式化されたビデオ・データのヘッダーに表示されます。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getFrameResolution()** メソッドはデフォルトの形式プラグインを使用してビデオ・データを読み込み、フレーム・リゾリューションを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオのフレームの解像度情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされている **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレームの解像度を読み込みます。

getFrameResolution() メソッド

```
DECLARE
    obj ORDSYS.ORDVideo;
    res INTEGER;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
    res := obj.getFrameResolution(ctx);
    DBMS_OUTPUT.put_line(res );
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```


setFrameRate() メソッド

形式

```
setFrameRate (knownFrameRate IN INTEGER);
```

説明

ビデオ・オブジェクトの **frameRate** 属性の値を設定します。

パラメータ

knownFrameRate
フレーム・レート

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

プラグマ

なし

例外

knownFrameRate パラメータの値が **NULL** の場合は、このメソッドをコールすると、**NULL_INPUT_VALUE** 例外が発生します。

例

5-83 ページの「**setFrameSize()** メソッド」の例を参照してください。

getFrameRate メソッド

形式

```
getFrameRate RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **frameRate** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

データベースに格納されたビデオ・データのフレーム・レート（オブジェクト属性値）を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getFrameRate;
  DBMS_OUTPUT.put_line('frame rate : ' || res);
END;
/
```

getFrameRate() メソッド

形式

```
getFrameRate(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたフレーム・レートを読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

ビデオ・フレーム・レート情報は、書式化されたビデオ・データのヘッダーに表示されます。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getFrameRate()** メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、フレーム・レートを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオ・フレーム・レート情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのオーディオ・データに埋め込まれた実フレーム・レートを読み込みます。

```
DECLARE
```

getFrameRate() メソッド

```
obj ORDSYS.ORDVideo;
res INTEGER;
ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
  res := obj.getFrameRate(ctx);
  DBMS_OUTPUT.put_line(res );
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setVideoDuration() メソッド

形式

```
setVideoDuration(knownVideoDuration RETURN INTEGER);
```

説明

ビデオ・オブジェクトの **videoDuration** 属性の値を設定します。

パラメータ

knownVideoDuration
既知のビデオ再生時間

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

プラグマ

なし

例外

knownVideoDuration パラメータの値が **NULL** の場合は、このメソッドをコールすると、**NULL_INPUT_VALUE** 例外が発生します。

例

5-83 ページの「**setFrameSize()** メソッド」の例を参照してください。

getVideoDuration メソッド

形式

```
getVideoDuration RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **videoDuration** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・データの合計再生時間を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getVideoDuration;
  DBMS_OUTPUT.put_line('video duration : ' || res);
END;
/
```

getVideoDuration() メソッド

形式

```
getVideoDuration(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたビデオ再生時間を読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

ビデオ再生時間情報は、書式化されたビデオ・データのヘッダーに表示されます。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getVideoDuration()** メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、ビデオ再生時間を決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使います。

ビデオ再生時間情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実ビデオ再生時間を読み込みます。

```
DECLARE  
obj ORDSYS.ORDVideo;
```

getVideoDuration() メソッド

```
res INTEGER;
ctx RAW(4000) :=NULL;
BEGIN
SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
res := obj.getVideoDuration(ctx);
--DBMS_OUTPUT.put_line(res );
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setNumberOfFrames() メソッド

形式

```
setNumberOfFrames (knownNumberOfFrames RETURN INTEGER) ;
```

説明

ビデオ・オブジェクトの `numberOfFrames` 属性の値を設定します。

パラメータ

knownNumberOfFrames
既知のフレーム数

使用方法

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

`knownNumberOfFrames` パラメータの値が `NULL` の場合は、このメソッドをコールすると、`NULL_INPUT_VALUE` 例外が発生します。

例

5-83 ページの「`setFrameSize()` メソッド」の例を参照してください。

getNumberOfFrames メソッド

形式

```
getNumberOfFrames RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **numberOfFrames** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・データの合計フレーム数（オブジェクト属性値）を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getNumberOfFrames;
  DBMS_OUTPUT.put_line('number of frames : ' || res);
END;
/
```

getNumberOfFrames() メソッド

形式

```
getNumberOfFrames(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたフレーム数を読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

合計フレーム数の情報は、書式化されたビデオ・データのヘッダーから取得します。

オブジェクトで検出されたフォーマットが NULL の場合は、`getNumberOfFrames()` メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、フレーム数を決定します。NULL でない場合は、ユーザー定義の形式プラグインを使用します。

合計フレーム数についての情報は、ビデオ・データから抽出できます。ORDVideo オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる `ORDPLUGINS.ORDX_<format>_VIDEO` パッケージを準備します。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、`VIDEO_PLUGIN_EXCEPTION` 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実フレーム数を読み込みます。

```
DECLARE  
  obj ORDSYS.ORDVideo;
```

getNumberOfFrames() メソッド

```
res INTEGER;
ctx RAW(4000) :=NULL;
BEGIN
SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
res := obj.getNumberOfFrames(ctx);
DBMS_OUTPUT.put_line(res );
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setCompressionType() メソッド

形式

```
setCompressionType(knownCompressionType IN VARCHAR2);
```

説明

ビデオ・オブジェクトの `compressionType` 属性の値を設定します。

パラメータ

knownCompressionType
既知の圧縮タイプ

使用方法

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

`knownCompressionType` パラメータの値が `NULL` の場合は、このメソッドをコールすると、`NULL_INPUT_VALUE` 例外が発生します。

例

5-83 ページの「`setFrameSize()` メソッド」の例を参照してください。

getCompressionType メソッド

形式

```
getCompressionType RETURN VARCHAR2;
```

説明

ビデオ・オブジェクトの `compressionType` 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・オブジェクトの `compressionType` 属性（オブジェクト属性値）を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res VARCHAR2(4000);
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getCompressionType;
  DBMS_OUTPUT.put_line('compression type: ' || res);
END;
/
```

getCompressionType() メソッド

形式

```
getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた圧縮タイプを読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

ビデオ圧縮タイプ情報は、書式化されたビデオ・データのヘッダーから取得します。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getCompressionType()** メソッドはデフォルトの形式プラグインを使用してビデオ・データを読み込み、圧縮タイプを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオ圧縮タイプ情報は、オーディオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージを準備します。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときにオーディオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実圧縮タイプを読み込みます。

```
DECLARE  
  obj ORDSYS.ORDVideo;
```

getCompressionType() メソッド

```
res VARCHAR2(4000);
ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;
  res := obj.getCompressionType(ctx);
  DBMS_OUTPUT.put_line(res );
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('method not supported');
END;
/
```

setNumberOfColors() メソッド

形式

```
setNumberOfColors (knownNumberOfColors RETURN INTEGER) ;
```

説明

ビデオ・オブジェクトの `numberOfColors` 属性の値を設定します。

パラメータ

knownNumberOfColors
既知の色数

使用方法

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

`knownNumberOfColors` パラメータの値が `NULL` の場合は、このメソッドをコールすると、`NULL_INPUT_VALUE` 例外が発生します。

例

5-83 ページの「`setFrameSize()` メソッド」の例を参照してください。

getNumberOfColors メソッド

形式

```
getNumberOfColors RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **numberOfColors** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・オブジェクトの **numberOfColors** 属性（オブジェクト属性値）を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getNumberOfColors;
  DBMS_OUTPUT.put_line('number of colors: ' || res);
END;
/
```

getNumberOfColors() メソッド

形式

```
getNumberOfColors(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた色数を読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

合計色数の情報は、書式化されたビデオ・データのヘッダーから取得します。

オブジェクトで検出されたフォーマットが NULL の場合は、getNumberOfColors() メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、色数を決定します。NULL でない場合は、ユーザー定義の形式プラグインを使用します。

合計色数の情報は、ビデオ・データから抽出できます。ORDVideo オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる ORDPLUGINS.ORDX_<format>_VIDEO パッケージをインプリメントします。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、VIDEO_PLUGIN_EXCEPTION 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実色数を読み込みます。

```
DECLARE  
obj ORDSYS.ORDVideo;
```

getNumberOfColors() メソッド

```
res INTEGER;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;  
  res := obj.getNumberOfColors(ctx);  
  DBMS_OUTPUT.put_line(res );  
EXCEPTION  
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN  
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');  
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');  
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN  
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');  
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('method not supported');  
END;  
/
```

setBitRate() メソッド

形式

```
setBitRate(knownBitRate IN INTEGER);
```

説明

ビデオ・オブジェクトの **bitRate** 属性の値を設定します。

パラメータ

knownBitRate
ビット・レート

使用方法

このメソッドをコールすると、暗黙的に **setUpdateTime()** メソッドがコールされます。

プラグマ

なし

例外

knownBitRate パラメータの値が **NULL** の場合は、このメソッドをコールすると、**NULL_INPUT_VALUE** 例外が発生します。

例

5-83 ページの「**setFrameSize()** メソッド」の例を参照してください。

getBitRate メソッド

形式

```
getBitRate RETURN INTEGER;
```

説明

ビデオ・オブジェクトの **bitRate** 属性の値を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getBitRate, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

ビデオ・オブジェクトの **bitRate** 属性（オブジェクト属性値）を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res INTEGER;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1 ;
  res := obj.getBitRate;
  DBMS_OUTPUT.put_line('bit rate : ' || res );
END;
/
```

getBitRate() メソッド

形式

```
getBitRate(ctx IN OUT RAW) RETURN INTEGER;
```

説明

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれたビット・レートを読み込みます。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

ビデオ・ビット・レート情報は、書式化されたビデオ・データのヘッダーから取得します。

オブジェクトで検出されたフォーマットが **NULL** の場合は、**getBitRate()** メソッドでは、デフォルトの形式プラグインを使用してビデオ・データを読み込み、ビット・レートを決定します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオ・ビット・レート情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージを準備します。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

形式プラグインをコールして、格納済みのビデオ・データに埋め込まれた実ビット・レートを読み込みます。

```
DECLARE  
  obj ORDSYS.ORDVideo;
```

getBitRate() メソッド

```
res INTEGER;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT vid INTO obj FROM TVID WHERE N=1 FOR UPDATE;  
  res := obj.getBitRate(ctx);  
  DBMS_OUTPUT.put_line(res );  
EXCEPTION  
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN  
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');  
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');  
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN  
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');  
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN  
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('method not supported');  
END;  
/
```

setKnownAttributes() メソッド

形式

```
setKnownAttributes(  
    knownFormat          IN VARCHAR2,  
    knownWidth           IN INTEGER,  
    knownHeight          IN INTEGER,  
    knownFrameResolution IN INTEGER,  
    knownFrameRate       IN INTEGER,  
    knownVideoDuration   IN INTEGER,  
    knownNumberOfFrames  IN INTEGER,  
    knownCompressionType IN VARCHAR2,  
    knownNumberOfColors  IN INTEGER,  
    knownBitRate         IN INTEGER);
```

説明

ビデオ・データの既知のビデオ属性を設定します。

パラメータ

knownFormat

既知のフォーマット

knownWidth

既知の幅

knownHeight

既知の高さ

knownFrameResolution

既知のフレームの解像度

knownFrameRate

既知のフレーム・レート

knownVideoDuration

既知のビデオ再生時間

knownNumberOfFrames

既知のフレーム数

knownCompressionType

既知の圧縮タイプ

knownNumberOfColors

既知の色数

knownBitRate

既知のビット・レート

使用方法

このメソッドをコールすると、暗黙的に `setUpdateTime()` メソッドがコールされます。

プラグマ

なし

例外

なし

例

ビデオ・データの既知のすべての属性に対して、プロパティ情報を設定します。

```
DECLARE
  obj ORDSYS.ORDVideo;
BEGIN
  select vid into obj from TVID where N =1 for update;
  obj.setKnownAttributes('MOOV',1,2,4,5,20,8,'Cinepak', 256, 1500);
  DBMS_OUTPUT.put_line('width: ' || TO_CHAR(obj.width));
  DBMS_OUTPUT.put_line('height: ' || TO_CHAR(obj.height));
  DBMS_OUTPUT.put_line('format: ' || obj.getFormat);
  DBMS_OUTPUT.put_line('frame resolution: ' || TO_CHAR(obj.getFrameResolution));
  DBMS_OUTPUT.put_line('frame rate: ' || TO_CHAR(obj.getFrameRate));
  DBMS_OUTPUT.put_line('video duration: ' || TO_CHAR(obj.getVideoDuration));
  DBMS_OUTPUT.put_line('number of frames: ' || TO_CHAR(obj.getNumberOfFrames));
  DBMS_OUTPUT.put_line('compression type: ' || obj.getCompressionType);
  DBMS_OUTPUT.put_line('bite rate: ' || TO_CHAR(obj.getBitRate));
  DBMS_OUTPUT.put_line('number of colors: ' || TO_CHAR(obj.getNumberOfColors));
  update TVID set vid = obj where N = 1;
  COMMIT;
END;
/
```

setProperties() メソッド

形式

```
setProperties(ctx IN OUT RAW);
```

説明

ビデオ・データを読み込んでオブジェクト属性の値を取得し、オブジェクトに格納します。ORDVideo で認識される既知の属性に対して、プロパティを設定します。既知の属性は次のとおりです。フォーマット、フレーム・サイズ、フレームの解像度、フレーム・レート、ビデオ再生時間、フレーム数、圧縮タイプ、色数、およびビット・レートです。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

フォーマットが NULL の場合は、setProperties() メソッドでは、デフォルトの形式プラグインを使用します。NULL でない場合は、ユーザー定義の形式プラグインを使用します。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、VIDEO_PLUGIN_EXCEPTION 例外が発生します。

例

既知のビデオ属性に対して、プロパティ情報を設定します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  ctx RAW(4000) :=NULL;
BEGIN
  select vid into obj from TVID where N =1 for update;
  obj.setProperties(ctx);
  update TVID set vid = obj where N = 1;
  COMMIT;
```

setProperty() メソッド

```
        EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.put_line('exception raised');
    END;
/
```

checkProperties() メソッド

形式

```
checkProperties(ctx IN OUT RAW) RETURN BOOLEAN;
```

説明

格納済みのビデオ・データのプロパティをすべてチェックします。次のビデオ属性をチェックします。フォーマット、フレーム・サイズ、フレームの解像度、フレーム・レート、ビデオ再生時間、フレーム数、圧縮タイプ、色数、およびビット・レートです。

パラメータ

ctx
形式プラグインのコンテキスト情報

使用方法

フォーマットが NULL の場合は、**checkProperties()** メソッドでは、デフォルトの形式プラグインを使用します。NULL でない場合は、ユーザー定義の形式プラグインを使用します。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

既知のビデオ属性に対して、プロパティ情報をチェックします。

```
DECLARE
  obj ORDSYS.ORDVideo;
  ctx RAW(4000) :=NULL;
BEGIN
  select vid into obj from TVID where N =1 ;
  if (obj.checkProperties(ctx)) then
    DBMS_OUTPUT.put_line('check Properties returned true');
  else
    DBMS_OUTPUT.put_line('check Properties returned false');
```

checkProperties() メソッド

```
        end if;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.put_line('exception raised');
    END;
/
```

getAttribute() メソッド

形式

```
getAttribute(  
    ctx  IN OUT RAW,  
    name IN VARCHAR2)  
RETURN VARCHAR2;
```

説明

ユーザー定義のフォーマットに対してだけ、要求されたビデオ・データの属性値を返します。

パラメータ

ctx
形式プラグインのコンテキスト情報

name
属性の名前

使用方法

ビデオ・データ属性は、書式化されたビデオ・データのヘッダーから取得します。

フォーマットが **NULL** の場合は、**getAttribute()** メソッドでは、デフォルトの形式プラグインを使用します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオ・データ属性情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないビデオ・フォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

データベースに格納されたビデオ・データの、特定のビデオ属性についての情報を返します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT vid INTO obj FROM TVID WHERE N=1;
  DBMS_OUTPUT.PUT_LINE('getting video duration');
  DBMS_OUTPUT.PUT_LINE('-----');
  res := obj.getAttribute(ctx, 'video_duration');
  EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
      DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
    WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
      DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('EXCEPTION caught');
END;
/
```

getAllAttributes() メソッド

形式

```
getAllAttributes(  
    ctx          IN OUT RAW,  
    attributes IN OUT NOCOPY CLOB);
```

説明

使用するクライアント・アクセス用のフォーマット文字列を戻します。ネイティブでサポートされるフォーマット文字列には、次のビデオ・データ属性がカンマ (,) で区切られて含まれます。**Format**、**FrameSize**、**FrameResolution**、**FrameRate**、**VideoDuration**、および **NumberOfFrames** です。ユーザー定義のフォーマットの場合は、形式プラグインにより文字列が定義されます。

パラメータ

ctx
形式プラグインのコンテキスト情報

attributes
属性

使用方法

これらのビデオ・データ属性は、書式化されたビデオ・データのヘッダーから取得します。

フォーマットが **NULL** の場合は、**getAllAttributes()** メソッドでは、デフォルトの形式プラグインを使用します。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ビデオ・データ属性情報は、ビデオ・データから抽出できます。**ORDVideo** オブジェクトにより認識されないビデオ・フォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージをインプリメントします。詳細は、2.3.13 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、**VIDEO_PLUGIN_EXCEPTION** 例外または **VIDEO_PLUGIN_EXCEPTION** 例外が発生します。

例

データベースに格納されたビデオ・データのすべてのビデオ属性を戻します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  tempLob CLOB;
  ctx RAW(4000) :=NULL;
BEGIN

  SELECT vid INTO obj FROM TVID WHERE N=1;
  DBMS_OUTPUT.PUT_LINE('getting comma separated list of all attributes');
  DBMS_OUTPUT.PUT_LINE('-----');

  DBMS_LOB.CREATETEMPORARY(tempLob, FALSE, DBMS_LOB.CALL);
  obj.getAllAttributes(ctx,tempLob);
  DBMS_OUTPUT.put_line(DBMS_LOB.substr(tempLob, DBMS_LOB.getLength(tempLob), 1));
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION CAUGHT');
END;
/
```

5.2.9 ビデオ・データの処理に関連付けられた ORDVideo メソッド

この項では、ビデオ・データの処理に関連付けられた ORDVideo メソッドの参照情報を提供します。

processVideoCommand() メソッド

形式

```
processVideoCommand(  
    ctx          IN OUT RAW,  
    cmd          IN VARCHAR2,  
    arguments IN VARCHAR2,  
    result       OUT RAW)  
  
RETURN RAW;
```

説明

コマンドおよびその引数を、処理のために形式プラグインに送信します。このメソッドは、ユーザー定義の形式プラグインでだけサポートされます。

パラメータ

ctx

形式プラグインのコンテキスト情報

cmd

形式プラグインによって認識されるすべてのコマンド

arguments

コマンドの引数

result

形式プラグインから戻される、この関数のコールの結果

使用方法

このメソッドを使用して、ビデオ・コマンドとその引数を形式プラグインに送信します。コマンドを受け取ると、解釈せずに処理のために形式プラグインに渡します。

フォーマットが **NULL** の場合は、**processVideoCommand()** メソッドでは、デフォルトの形式プラグインを使います。**NULL** でない場合は、ユーザー定義の形式プラグインを使用します。

ORDVideo オブジェクトにより認識されないフォーマットをサポートするには、そのフォーマットがサポートされる **ORDPLUGINS.ORDX_<format>_VIDEO** パッケージを準備します。詳細は、**2.3.13** 項を参照してください。

プラグマ

なし

例外

このメソッドをコールしたときに、ビデオ・プラグインで例外が発生した場合は、VIDEO_PLUGIN_EXCEPTION 例外または VIDEO_PLUGIN_EXCEPTION 例外が発生します。

例

コマンド・セットを処理します。

```
DECLARE
  obj ORDSYS.ORDVideo;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR(4000);
  argList VARCHAR(4000);
  ctx RAW(4000) :=NULL;
BEGIN
select vid into obj from TVID where N =1 for UPDATE;
-- assign command
-- assign argList
res := obj.processVideoCommand (ctx, command, argList, result);
UPDATE TVID SET vid=obj WHERE N=1 ;
COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('VIDEO METHOD_NOT_SUPPORTED EXCEPTION caught');
  WHEN ORDSYS.ORDVideoExceptions.VIDEO_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('VIDEO PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('EXCEPTION caught');
END;
/
```

5.3 パッケージまたは PL/SQL プラグイン

この項では、付属のパッケージまたは PL/SQL プラグインの参照情報について説明します。

5.3.1 ビデオ・フォーマット・パッケージ

独自のビデオ・フォーマット・パッケージを開発するときは、次のパッケージをビデオ・フォーマット・テンプレートとして使います。これらのパッケージは単なるテンプレートなので、各パッケージで説明しているメソッドを使うには、パッケージをインプリメントする必要があります。

- ORDPLUGINS.ORDX_DEFAULT_VIDEO
- ORDPLUGINS.ORDX_AVI_VIDEO
- ORDPLUGINS.ORDX_QT_VIDEO (ORDPLUGINS.ORDX_MOOV_VIDEO を参照)
- ORDPLUGINS.ORDX_MOOV_VIDEO
- ORDPLUGINS.ORDX_MPEG_VIDEO

5.3.2 ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージ

独自の ORDPLUGINS.ORDX_<format>_VIDEO ビデオ・フォーマット・パッケージを開発するときは、次の付属の ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージを参考にしてください。

```
CREATE OR REPLACE PACKAGE ORDX_DEFAULT_VIDEO
authid current_user
AS
--VIDEO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      name IN VARCHAR2)
RETURN VARCHAR2;
PROCEDURE getFrameSize(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      width OUT INTEGER,
                      height OUT INTEGER);
FUNCTION getFrameResolution(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getFrameRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getVideoDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getNumberOfFrames(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
```

```

        RETURN INTEGER;
FUNCTION  getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION  getNumberOfColors(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION  getBitRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDVideo);
FUNCTION  checkProperties(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo) RETURN NUMBER;

-- must return name=value; name=value; ... pairs
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDVideo,
                           attributes IN OUT NOCOPY CLOB);

-- VIDEO PROCESSING METHODS
FUNCTION  processCommand(
                           ctx          IN OUT RAW,
                           obj          IN OUT NOCOPY ORDSYS.ORDVideo,
                           cmd          IN VARCHAR2,
                           arguments IN VARCHAR2,
                           result      OUT RAW)
RETURN RAW;
PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getBitRate, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 5-1 は、ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージでサポートされているメソッド、およびサポートされていないメソッドをコールした場合に発生する例外です。

表 5-1 ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージでサポートされているメソッド

メソッド名	サポート・レベル
getFormat	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

表 5-1 ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージでサポートされているメソッド (続き)

メソッド名	サポート・レベル
getAttribute	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameSize	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameResolution	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getVideoDuration	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfFrames	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getCompressionType	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfColors	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getBitRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
setProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
checkProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAllAttributes	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
processCommand	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

5.3.3 ORDPLUGINS.ORDX_AVI_VIDEO パッケージ

AVI ビデオ・フォーマット仕様の場合は、ORDPLUGINS.ORDX_AVI_VIDEO パッケージが付属しており、このパッケージが使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_AVI_VIDEO
authid current_user
AS
--VIDEO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
```



```

        RETURN VARCHAR2;
FUNCTION  getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      name IN VARCHAR2)
        RETURN VARCHAR2;
PROCEDURE setFrameSize(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      width OUT INTEGER,
                      height OUT INTEGER);
FUNCTION  getFrameResolution(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
FUNCTION  getFrameRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
FUNCTION  getVideoDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
FUNCTION  getNumberOfFrames(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
FUNCTION  getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN VARCHAR2;
FUNCTION  getNumberOfColors(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
FUNCTION  getBitRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
        RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDVideo);
FUNCTION  checkProperties(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo) RETURN NUMBER;

-- must return name=value; name=value; ... pairs
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                          obj IN ORDSYS.ORDVideo,
                          attributes IN OUT NOCOPY CLOB);

-- VIDEO PROCESSING METHODS
FUNCTION  processCommand(
                      ctx          IN OUT RAW,
                      obj          IN OUT NOCOPY ORDSYS.ORDVideo,
                      cmd          IN VARCHAR2,
                      arguments IN VARCHAR2,
                      result       OUT RAW)

        RETURN RAW;

PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS);

```

```
PRAGMA RESTRICT_REFERENCES (getBitRate, WNDS, WNPS, RNDS, RNPS);

END;
/
```

表 5-2 は、ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に発生する例外です。

表 5-2 ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド

メソッド名	サポート・レベル
getFormat	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAttribute	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameSize	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameResolution	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getVideoDuration	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfFrames	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getCompressionType	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfColors	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getBitRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
setProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
checkProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAllAttributes	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

表 5-2 ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド (続き)

メソッド名	サポート・レベル
processCommand	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

5.3.4 ORDPLUGINS.ORDX_MOOV_VIDEO パッケージ

MOOV ビデオ・フォーマット仕様の場合は、ORDPLUGINS.ORDX_MOOV_VIDEO パッケージが付属しており、このパッケージが使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_MOOV_VIDEO
authid current_user
AS
--VIDEO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      name IN VARCHAR2)
RETURN VARCHAR2;
PROCEDURE setFrameSize(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      width OUT INTEGER,
                      height OUT INTEGER);
FUNCTION getFrameResolution(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getFrameRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getVideoDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getNumberOfFrames(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION getNumberOfColors(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getBitRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDVideo);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo) RETURN NUMBER;

-- must return name=value; name=value; ... pairs
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                          obj IN ORDSYS.ORDVideo,
                          attributes IN OUT NOCOPY CLOB);
-- VIDEO PROCESSING METHODS
```

```

FUNCTION processCommand(
    ctx          IN OUT RAW,
    obj          IN OUT NOCOPY ORDSYS.ORDVideo,
    cmd          IN VARCHAR2,
    arguments    IN VARCHAR2,
    result       OUT RAW)
    RETURN RAW;
PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getBitRate, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 5-3 は、ORDPLUGINS.ORDX_MOOV_VIDEO パッケージでサポートされるメソッド、およびサポートされていないメソッドをコールした場合に発生する例外です。

表 5-3 ORDPLUGINS.ORDX_MOOV_VIDEO パッケージでサポートされているメソッド

メソッド名	サポート・レベル
getFormat	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAttribute	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameSize	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameResolution	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getVideoDuration	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfFrames	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

表 5-3 ORDPLUGINS.ORDX_MOOV_VIDEO パッケージでサポートされているメソッド (続き)

メソッド名	サポート・レベル
getCompressionType	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfColors	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getBitRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
setProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
checkProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAllAttributes	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
processCommand	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

5.3.5 ORDPLUGINS.ORDX_MPEG_VIDEO パッケージ

MPEG ビデオ・フォーマット仕様の場合は、ORDPLUGINS.ORDX_MPEG_VIDEO パッケージが付属しており、このパッケージが使用されます。

```
CREATE OR REPLACE PACKAGE ORDX_MPEG_VIDEO
authid current_user
AS
--VIDEO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION getAttribute(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      name IN VARCHAR2)
RETURN VARCHAR2;
PROCEDURE setFrameSize(ctx IN OUT RAW,
                      obj IN ORDSYS.ORDVideo,
                      width OUT INTEGER,
                      height OUT INTEGER);
FUNCTION getFrameResolution(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getFrameRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getVideoDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
```

```

FUNCTION getNumberOfFrames(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2;
FUNCTION getNumberOfColors(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
FUNCTION getBitRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDVideo);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo) RETURN NUMBER;

-- must return name=value; name=value; ... pairs
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDVideo,
                           attributes IN OUT NOCOPY CLOB);

-- VIDEO PROCESSING METHODS
FUNCTION processCommand(
                           ctx          IN OUT RAW,
                           obj          IN OUT NOCOPY ORDSYS.ORDVideo,
                           cmd          IN VARCHAR2,
                           arguments IN VARCHAR2,
                           result      OUT RAW)
RETURN RAW;
PRAGMA RESTRICT_REFERENCES(getFormat, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getAttribute, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameSize, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameResolution, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getFrameRate, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getVideoDuration, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfFrames, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getCompressionType, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getNumberOfColors, WNDS, WNPS, RNDS, RNPS);
PRAGMA RESTRICT_REFERENCES(getBitRate, WNDS, WNPS, RNDS, RNPS);

END;
/

```

表 5-4 は、ORDPLUGINS.ORDX_MPEG_VIDEO パッケージでサポートされているメソッド、およびサポートされていないメソッドをコールした場合に発生する例外です。

表 5-4 ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド

メソッド名	サポート・レベル
getFormat	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

表 5-4 ORDPLUGINS.ORDX_AVI_VIDEO パッケージでサポートされているメソッド (続き)

メソッド名	サポート・レベル
getAttribute	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameSize	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameResolution	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getFrameRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getVideoDuration	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfFrames	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getCompressionType	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getNumberOfColors	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getBitRate	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
setProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
checkProperties	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
getAllAttributes	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION
processCommand	サポートされていません。次の例外が発生します。METHOD_NOT_SUPPORTED および VIDEO_PLUGIN_EXCEPTION

5.3.6 *interMedia* を拡張して新しいビデオ・データ・フォーマットをサポートする

interMedia を拡張して新しいビデオ・データ・フォーマットをサポートするには、次の 3 つのステップで実行します。

1. 新しいビデオ・データ・フォーマットを設計します。

2. 新しいビデオ・データ・フォーマットをインプリメントし、ORDX_MY_VIDEO.SQL などの名前を付けます。
3. 新しい ORDX_MY_VIDEO.SQL プラグインを、ORDPLUGINS スキーマにインストールします。
4. 新しいプラグインに対して EXECUTE 権限を割り当てます。たとえば、ORDX_MY_VIDEO.SQL プラグインに対して PUBLIC を割り当てます。

2.3.12 項では、*interMedia* を拡張して新しいビデオ・データ・フォーマットをサポートする方法と、そのインタフェースについて簡単に説明しています。例 5-1 は、パッケージ本体のリストです。この操作を行う場合に参考にしてください。独自の変数を "--Your variables go here" と記述された場所に、独自のコードを "--Your code goes here" と記述された場所に追加します。

独自のビデオ・フォーマット・プラグインのインストール方法および付属のサンプル・スクリプトの実行方法については、E.3 項を参照してください。

例 5-1 新しいビデオ・データ・フォーマットを拡張サポートするためのパッケージ本体のリスト

```
CREATE OR REPLACE PACKAGE BODY ORDX_MY_VIDEO
AS
    --VIDEO ATTRIBUTES ACCESSORS
    FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
    RETURN VARCHAR2
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END;
    FUNCTION getAttribute(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDVideo,
                           name IN VARCHAR2)
    RETURN VARCHAR2
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END;
    PROCEDURE getFrameSize(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDVideo,
                           width OUT INTEGER,
                           height OUT INTEGER)
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END;
```



```

FUNCTION getFrameResolution(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getFrameRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getVideoDuration(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getNumberOfFrames(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getNumberOfColors(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION getBitRate(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo)
RETURN INTEGER
IS
--Your variables go here
BEGIN

```

```
--Your code goes here
END;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY ORDSYS.ORDVideo)
IS
--Your variables go here
BEGIN
--Your code goes here
END;
FUNCTION checkProperties(ctx IN OUT RAW, obj IN ORDSYS.ORDVideo) RETURN NUMBER
IS
IS
--Your variables go here
BEGIN
--Your code goes here
END;
PROCEDURE getAllAttributes(ctx IN OUT RAW,
                           obj IN ORDSYS.ORDVideo,
                           attributes IN OUT NOCOPY CLOB)
IS
--Your variables go here
BEGIN
--Your code goes here
END;
-- VIDEO PROCESSING METHODS
FUNCTION processCommand(
                           ctx          IN OUT RAW,
                           obj          IN OUT NOCOPY ORDSYS.ORDVideo,
                           cmd          IN VARCHAR2,
                           arguments IN VARCHAR2,
                           result OUT RAW)
RETURN RAW
IS
--Your variables go here
BEGIN
--Your code goes here
END;
END;
/
show errors;
```

ORDSource 参照情報

Oracle8i *interMedia* には、ORDSource タイプに関する次の情報が含まれます。

- オブジェクト型 -- 6.1 項を参照してください。
- メソッド -- 6.2 項を参照してください。
- パッケージまたは PL/SQL プラグイン -- 6.3 項を参照してください。

このオブジェクトは、他の Oracle8i *interMedia* オブジェクトだけに使用されます。これを埋込みオブジェクトとして使用して、独自オブジェクトのソース・メカニズムをインプリメントできます。

この章の例では、テスト・ソース・テーブル TS が作成され、データが格納されていることを前提にしています。このテーブルは、6.2.1 項で説明した SQL 文を使用して作成されました。

ソース・プラグインに渡されて処理される、ORDSource レベルで起動されたメソッドは、最初の引数に **ctx (RAW(4000))** を取ります。これらのメソッドのいずれかを最初にコールする前に、クライアントは **ctx** 構造を割り当て、それを **NULL** に初期化してから、**open()** メソッドを起動する必要があります。この時点で、ソース・プラグインはこのクライアント用のコンテキストを初期化できます。処理の完了時に、クライアントは **close()** メソッドを起動する必要があります。

ソース・プラグインから起動したメソッドは、最初の引数に **obj (ORDSource)** を取り、2 番目の引数に **ctx (RAW(4000))** を取ります。

注意: 現在のリリースでは、すべてのソース・プラグインが **ctx** 引数を使用するわけではありません。ただし、既に示したコードを使用するなら、使用するアプリケーションで現在または将来登場するソース・プラグインが動作するはずです。

ORDSource オブジェクトは、一貫性の維持（たとえば、ローカルと **upDateTime** 属性の）を試みません。一貫性の維持は、ユーザーの操作に依存しています。ORDAudio、

ORDImage、および ORDVideo オブジェクトはすべて、これらのオブジェクトが含まれる ORDSource オブジェクトとの一貫性を維持します。

6.1 オブジェクト型

Oracle8i *interMedia* は、ORDSource オブジェクト型を提供します。このオブジェクト型は、多様なマルチメディア・ソースへのアクセスをサポートします。

ORDSource オブジェクト型

ORDSource オブジェクト型は、Oracle データベースの BLOB 内のデータ・ソースへのローカルなアクセスをサポートします。また、ローカル・ファイル・システムの BFILE から、HTTP サーバー上の URL（ファイアウォール内）から、また別のサーバー上のユーザー定義ソースからの、データ・ソースへの外部的なアクセスをサポートします。オブジェクト型は、次のように定義します。

```
CREATE OR REPLACE TYPE ORDsource
AS OBJECT
(
  -- ATTRIBUTES
  localData          BLOB,
  srcType             VARCHAR2(4000),
  srcLocation         VARCHAR2(4000),
  srcName             VARCHAR2(4000),
  updateTime          DATE,
  local              NUMBER,
  -- METHODS
  -- Methods associated with the local attribute
  MEMBER PROCEDURE setLocal,
  MEMBER PROCEDURE clearLocal,
  MEMBER FUNCTION isLocal RETURN BOOLEAN,
  PRAGMA RESTRICT_REFERENCES(isLocal, WNDS, WNPS, RNDS, RNPS),
  -- Methods associated with the updateTime attribute
  MEMBER FUNCTION getUpdateTime RETURN DATE,
  PRAGMA RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS),
  MEMBER PROCEDURE setUpdateTime(current_time DATE),
  -- Methods associated with the source information
  MEMBER PROCEDURE setSourceInformation(
                                source_type      IN VARCHAR2,
                                source_location  IN VARCHAR2,
                                source_name      IN VARCHAR2),
  MEMBER FUNCTION getSourceInformation RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getSourceInformation, WNDS, WNPS, RNDS, RNPS),

  MEMBER FUNCTION getSourceType RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS),

  MEMBER FUNCTION getSourceLocation RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS),

  MEMBER FUNCTION getSourceName RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS),
```

```

MEMBER FUNCTION getBFile RETURN BFILE,
PRAGMA RESTRICT_REFERENCES(getBFile, WNDS, WNPS, RNDS, RNPS),

-- Methods associated with source import/export operations
MEMBER PROCEDURE import(
    ctx      IN OUT RAW,
    dlob     IN OUT NOCOPY BLOB,
    mimetype OUT VARCHAR2,
    format   OUT VARCHAR2),
MEMBER PROCEDURE importFrom(
    ctx      IN OUT RAW,
    dlob     IN OUT NOCOPY BLOB,
    mimetype OUT VARCHAR2,
    format   OUT VARCHAR2,
    source_type IN VARCHAR2,
    source_location IN VARCHAR2,
    source_name IN VARCHAR2),
MEMBER PROCEDURE export(
    ctx      IN OUT RAW,
    source_type IN VARCHAR2,
    source_location IN VARCHAR2,
    source_name IN VARCHAR2),

-- Methods associated with source content-related operations
MEMBER FUNCTION getContentLength(ctx IN OUT RAW) RETURN INTEGER,
PRAGMA RESTRICT_REFERENCES(getContentLength, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getSourceAddress(ctx IN OUT RAW,
                                userData IN VARCHAR2)
    RETURN VARCHAR2,
PRAGMA RESTRICT_REFERENCES(getSourceAddress, WNDS, WNPS, RNDS, RNPS),

MEMBER FUNCTION getLocalContent RETURN BLOB,
PRAGMA RESTRICT_REFERENCES(getLocalContent, WNDS, WNPS, RNDS, RNPS),

MEMBER PROCEDURE getContentInTempLob(
    ctx      IN OUT RAW,
    tempLob  IN OUT NOCOPY BLOB,
    mimetype OUT VARCHAR2,
    format   OUT VARCHAR2,
    duration IN PLS_INTEGER := 10,
    cache    IN BOOLEAN := TRUE),
MEMBER PROCEDURE deleteLocalContent,

-- Methods associated with source access methods
MEMBER FUNCTION open(userArg IN RAW, ctx OUT RAW) RETURN INTEGER,
MEMBER FUNCTION close(ctx IN OUT RAW) RETURN INTEGER,
MEMBER FUNCTION trim(ctx IN OUT RAW,

```

```
newlen IN INTEGER) RETURN INTEGER,

-- Methods associated with content read/write operations
MEMBER PROCEDURE read(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   OUT RAW),
MEMBER PROCEDURE write(
    ctx      IN OUT RAW,
    startPos IN INTEGER,
    numBytes IN OUT INTEGER,
    buffer   IN RAW),
-- Methods associated with any commands to be sent to the external source
MEMBER FUNCTION processCommand(
    ctx      IN OUT RAW,
    command  IN VARCHAR2,
    arglist  IN VARCHAR2,
    result   OUT RAW)
    RETURN RAW
);
```

パラメータは次のとおりです。

- **localData**: オブジェクト内に **BLOB** としてローカルに格納されたマルチメディア・データが含まれます。4 ギガバイトまでのデータを、**Oracle** データベース内で **BLOB** として格納できます。また、格納されたデータは、**Oracle** のセキュリティおよびトランザクション環境で保護されます。
- **srcType**: データ・ソース・タイプを識別します。次の値ソース・タイプをサポートします。

srcType	ソース・タイプ
"FILE"	ローカル・ファイル・システム上の BFILE
"HTTP"	HTTP サーバー
"<name>"	ユーザー定義

注意: プラグインのキーワード 'FILE' は、Oracle Corporation が提供する BFILE ソース用の予約語です。独自のファイル・プラグインをインプリメントする場合は、'MYFILE' などの異なる名前を選択してください。

- **srcLocation:** srcType 値に基づいてデータが配置された場所を識別します。srcType 値に対応した有効な srcLocation 値は、次のとおりです。

srcType	ロケーション値
"FILE"	<DIR> またはディレクトリ・オブジェクト名
"HTTP"	<SourceBase> または基本ディレクトリの検索に必要な URL
"<name>"	<iden> またはユーザー定義ソースへのアクセスに必要な識別子文字列

- **srcName:** データ・オブジェクト名を識別します。srcType 値に対応した有効な srcName 値は、次のとおりです。

srcType	ネーム値
"FILE"	<FILE> またはファイル名
"HTTP"	<Source> またはオブジェクト名
"<name>"	<object name> またはオブジェクト名

- **updateTime:** データが最後に更新された時間です。
- **local:** データがローカルに格納されているかどうかを判断するためのフラグです。
 - 1 は、データが BLOB にあることを意味します。
 - 0 は、データが外部ソースにあることを意味します。NULL（最初の空白行挿入時のデフォルト状態）は、データがローカルにあることを意味するとみなされます。

6.2 メソッド

この項では、ソース・データの操作用に提供された **Oracle8i interMedia** のメソッドに関する **ORDSource** 参照情報を提供します。これらのメソッドは、次のようにグループ化されています。

ローカル属性と関連付けられた **ORDSource** メソッド

- **setLocal**: ローカル属性のフラグ値を "1" に設定します。これは、データ・ソースがローカルにあることを意味します。
- **clearLocal**: ローカル属性のフラグ値を "0" にリセットします。これは、データ・ソースが外部にあることを意味します。
- **isLocal**: TRUE を返す場合は、データ・ソースがローカルまたは BLOB 内にあることを意味します。FALSE を返す場合は、データが外部ソース内にあることを意味します。ローカル属性の値は、戻り値の判断に使用されます。

updateTime 属性と関連付けられた **ORDSource** メソッド

- **getUpdateTime**: updateTime 属性の値を返します。
- **setUpdateTime**: updateTime 属性の値を、引数で指定された時間に設定します。

srcType、**srcLocation**、および **srcName** 属性と関連付けられた **ORDSource** メソッド

- **setSourceInformation()**: データ・ソースに関する情報を設定または変更します。
- **getSourceInformation**: URL として書式化されたデータ・ソースの完全な情報を含む書式化された文字列を返します。
- **getSourceType**: データの外部ソース・タイプを返します。
- **getSourceLocation**: データの外部ソース位置を返します。
- **getSourceName**: データの外部ソース名を返します。
- **getBFile**: srcType が FILE タイプの場合、外部コンテンツを BFILE として返します。

インポートおよびエクスポート操作と関連付けられた **ORDSource** メソッド

- **import()**: 外部データ・ソース (setSourceInformation() をコールして指定された) から Oracle データベース内のローカル・ソース (localData) へ、データを転送します。
- **importFrom()**: 指定された外部データ・ソース (ソース、位置、名前) から Oracle データベース内のローカル・ソース (localData) へ、データを転送します。
- **export()**: Oracle データベース内のローカル・ソース (localData) から、指定された外部データ・ソースへ、データを転送します。このメソッドは、ユーザー定義ソースでのみサポートされています。

localData 属性と関連付けられた ORDSOURCE メソッド

- `getContentLength()`: データ・ソースの長さを（バイト数で）返します。
- `getSourceAddress()`: データ・ソースのアドレスを返します。
- `getLocalContent`: 内容をローカルに格納するために使用する BLOB へのハンドルを返します。
- `getContentInTempLob()`: 内容をテンポラリ LOB に返します。
- `deleteLocalContent`: ローカル BLOB の内容を削除します。

アクセス操作と関連付けられた ORDSOURCE メソッド

- `open()`: データ・ソースをオープンします。
- `close()`: データ・ソースをクローズします。
- `trim()`: データ・ソースの切捨てを実行します。

ソースの読み込み / 書き込み操作と関連付けられた ORDSOURCE メソッド

- `read()`: 開始位置から `n` バイトのバッファをソースから読み込みます。
- `write()`: 開始位置から `n` バイトのバッファをソースに書き込みます。

外部ソースへの処理コマンドと関連付けられた ORDSOURCE メソッド

- `processCommand()`: 外部ソースに対し、任意のコマンドとして処理します。このメソッドは、ユーザー定義ソースでのみサポートされています。

オブジェクト型およびメソッドの詳細は、『Oracle8i 概要』を参照してください。

6.2.1 例題テーブル定義

この章で説明するメソッドは、テスト用ソース・テーブル TS に基づく例題で使います。6.2.2 項から 6.2.9 項までを読む際に、次の TS テーブル定義を参照してください。

TS テーブル定義

```
CREATE TABLE TS (n NUMBER, s ORDSYS.ORDSOURCE);

INSERT INTO TS VALUES (1, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE,
NULL));
INSERT INTO TS VALUES (2, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE,
NULL));
INSERT INTO TS VALUES (3, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE,
NULL));
INSERT INTO TS VALUES (4, ORDSYS.ORDSOURCE(EMPTY_BLOB(), NULL, NULL, NULL, SYSDATE,
NULL));
```

6.2.2 ローカル属性と関連付けられた ORDSOURCE メソッド

この項では、ローカル属性と関連付けられた ORDSOURCE メソッドの参照情報を提供します。

setLocal メソッド

形式

```
setLocal;
```

説明

データが Oracle8i の BLOB 内部に格納されることを示すローカル属性を設定します。

パラメータ

なし

使用方法

このメソッドは、ローカル属性を 1（データが `localData` 属性内にローカルに格納されることを意味する）に設定します。

プラグマ

なし

例外

なし

例

データがローカルに格納されることを示すフラグを設定します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.setLocal;
  UPDATE TS SET S=SRC WHERE N = 1;
  COMMIT;
END;
/
```

clearLocal メソッド

形式

```
clearLocal;
```

説明

フラグの値をローカル（データ・ソースが **Oracle8i** の **BLOB** 内にローカルに格納されることを意味する）から非ローカル（データ・ソースが外部に格納されることを意味する）にリセットします。

パラメータ

なし

使用方法

このメソッドは、ローカル属性を **0**（データが外部的に、または **Oracle8i** の外部に格納されることを示す）に設定します。

プラグマ

なし

例外

なし

例

データ用のローカル・フラグの値を消去します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.clearLocal;
  UPDATE TS SET S=SRC WHERE N = 1;
  COMMIT;
END;
/
```

isLocal メソッド

形式

```
isLocal RETURN BOOLEAN;
```

説明

データが **Oracle8i** の **BLOB** 内でローカルに格納される場合は **TRUE** を、データが外部に格納される場合は **FALSE** を返します。

パラメータ

なし

使用方法

ローカル属性が **1** または **NULL** に設定されている場合、このメソッドは **TRUE** を返し、それ以外の場合は **FALSE** を返します。

プラグマ

Pragma RESTRICT_REFERENCES(isLocal, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

データがローカルに格納されているかどうかを判断します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 ;
  if (SRC.isLocal = TRUE) then
    DBMS_OUTPUT.put_line('local is set true');
  else
    DBMS_OUTPUT.put_line('local is set false');
  end if;
END;
/
```

6.2.3 updateTime 属性と関連付けられた ORDSOURCE メソッド

この項では、updateTime 属性と関連付けられた ORDSOURCE メソッドの参照情報を提供します。

getUpdateTime メソッド

形式

```
getUpdateTime RETURN DATE;
```

説明

ORDSource オブジェクトの **updateTime** 属性値を返します。この値は、オブジェクトが最後に変更されたタイムスタンプ、またはユーザーが **setUpdateTime()** メソッドを明示的にコールして設定したタイムスタンプです。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getUpdateTime, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

データに設定された、現在の **updateTime** 属性値を取得します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.setUpdateTime(SYSDATE);
  UPDATE TS SET S=SRC WHERE N = 1;
  COMMIT;
  SELECT S INTO SRC FROM TS WHERE N = 1 ;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SRC.getUpdateTime, 'MM-DD-YYYY HH24:MI:SS'));
END;
/
```

setUpdateTime() メソッド

形式

```
setUpdateTime(current_time DATE);
```

説明

updateTime 属性の値を指定した時刻に設定します。

パラメータ

current_time
更新時間

使用方法

current_time が NULL の場合、updateTime は SYSDATE（現在の時刻）に設定されます。

プラグマ

なし

例外

なし

例

6-14 ページの「getUpdateTime メソッド」の例を参照してください。

6.2.4 srcType、srcLocation、および srcName 属性と関連付けられた ORDSource メソッド

この項では、srcType、srcLocation、および srcName 属性と関連付けられた ORDSource メソッドの参照情報を提供します。

setSourceInformation() メソッド

形式

```
setSourceInformation(  
    source_type      IN VARCHAR2,  
    source_location  IN VARCHAR2,  
    source_name      IN VARCHAR2);
```

説明

外部データ・ソースを記述する **srcType**、**srcLocation**、および **srcName** に関する、提供されたサブコンポーネント情報を設定します。

パラメータ

source_type

外部データのソース・タイプ。詳細は、この章の「ORDSource オブジェクト型」の定義を参照してください。

source_location

外部データのソース位置。詳細は、この章の「ORDSource オブジェクト型」の定義を参照してください。

source_name

外部データのソース名。詳細は、この章の「ORDSource オブジェクト型」の定義を参照してください。

使用方法

import() メソッドをコールする前に、**setSourceInformation()** メソッドをコールして **srcType**、**srcLocation**、および **srcName** 属性情報を設定し、データ・ソースの位置を記述する必要があります。**importFrom()** または **export()** メソッドをコールする場合、これらの属性は **importFrom()** または **export()** コール成功後に設定されます。

このメソッドを使用する前に、ディレクトリが存在する、または作成されることを確認する必要があります。

プラグマ

なし

例外

`source_type` の値が `NULL` の場合、このメソッドは `INCOMPLETE_SOURCE_INFORMATION` 例外を呼び出します。

例

あるファイルをポイントするようにソースを設定します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.setSourceInformation('FILE', 'AUDIODIR', 'testaud.dat');
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceInformation);
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceType);
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceLocation);
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceName);
  UPDATE TS SET S=SRC WHERE N = 1;
  COMMIT;
END;
/
```

getSourceInformation メソッド

形式

```
getSourceInformation RETURN VARCHAR2;
```

説明

外部データ・ソースの完全な情報を含む、URL として書式化された文字列を返します。

パラメータ

なし

使用方法

このメソッドは、<srcType>://<srcLocation>/<srcName> として書式化された VARCHAR2 文字列を返します。srcType、srcLocation、および srcName は、ORDSource 属性値です。

プラグマ

```
Pragma RESTRICT_REFERENCES(getSourceInformation, WNDS, WNPS, RNDS, RNPS)
```

例外

なし

例

6-18 ページの「setSourceInformation() メソッド」の例を参照してください。

getSourceType メソッド

形式

```
getSourceType RETURN VARCHAR2;
```

説明

外部データ・ソース・タイプを返します。

パラメータ

なし

使用方法

このメソッドは、**srcType** 属性の現在値 ('FILE' など) を返します。

プラグマ

Pragma RESTRICT_REFERENCES(getSourceType, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

6-18 ページの「setSourceInformation() メソッド」の例を参照してください。

getSourceLocation メソッド

形式

```
getSourceLocation RETURN VARCHAR2;
```

説明

外部データ・ソースの位置を返します。

パラメータ

なし

使用方法

このメソッドは、`srcLocation` 属性の現在値 ('BFILEDIR' など) を返します。

プラグマ

```
Pragma RESTRICT_REFERENCES(getSourceLocation, WNDS, WNPS, RNDS, RNPS)
```

例外

`srcLocation` の値が `NULL` の場合、このメソッドをコールすると `INCOMPLETE_SOURCE_LOCATION` 例外が呼び出されます。

例

6-18 ページの「`setSourceInformation()` メソッド」の例を参照してください。

getSourceName メソッド

形式

```
getSourceName RETURN VARCHAR2;
```

説明

外部データ・ソース名を返します。

パラメータ

なし

使用方法

このメソッドは、srcName 属性の現在値 ('testaud.dat' など) を返します。

プラグマ

```
Pragma RESTRICT_REFERENCES(getSourceName, WNDS, WNPS, RNDS, RNPS)
```

例外

srcName の値が NULL の場合、このメソッドをコールすると INCOMPLETE_SOURCE_NAME 例外が呼び出されます。

例

6-18 ページの「setSourceInformation() メソッド」の例を参照してください。

getBFile メソッド

形式

```
getBFile RETURN BFILE;
```

説明

srcType が 'FILE' の場合、BFILE ハンドルを返します。

パラメータ

なし

使用方法

このメソッドは、'FILE' または BFILE ソースの srcType でのみ使用可能です。

プラグマ

Pragma RESTRICT_REFERENCES(getBFile, WNDS, WNPS, RNDS, RNPS)

例外

srcType の値が NULL の場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

srcType の値が FILE 以外の場合、このメソッドをコールすると INVALID_SOURCE_TYPE 例外が呼び出されます。

例

BFILE を取得します。

```
DECLARE
  SRC ORDSYS.ORDSource;
  file_handle BFILE;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 ;
  src.setSourceInformation('FILE', 'BFILEDIR', 'testaud.dat');
  file_handle := SRC.getBFile;
  DBMS_OUTPUT.put_line(DBMS_LOB.GETLENGTH(file_handle));
END;
/
```

6.2.5 インポートおよびエクスポート操作と関連付けられた ORDSource メソッド

この項では、インポートおよびエクスポート操作と関連付けられた ORDSource メソッドの参照情報を提供します。

import() メソッド

形式

```
import (
    ctx          IN OUT RAW,
    dlob         IN OUT NOCOPY BLOB,
    mimetype     OUT VARCHAR2,
    format       OUT VARCHAR2);
```

説明

外部データ・ソース（最初に `setSourceInformation()` をコールして指定された）から Oracle データベース内のローカル・ソースへ、データを転送します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。この情報は、`import()` コールを処理するソース・プラグインへ未解釈のまま渡されます。

dlob

宛先ラージ・オブジェクトまたはデータ・オブジェクト

mimetype

データの MIME タイプ（存在する場合）を受け取る出力パラメータ（'audio/basic' など）

format

データのフォーマット（存在する場合）を受け取る出力パラメータ（'AUFF' など）

使用方法

`setSourceInformation()` をコールして `srcType`、`srcLocation`、および `srcName` 属性情報を設定し、`import()` メソッドをコールする前のデータ・ソース位置を記述します。

このメソッドを使用する前に、ディレクトリが存在すること、または作成されることを確認する必要があります。

このメソッドのコールでは、`ORDPLUGINS.ORDX_<srcType>_SOURCE` プラグイン・パッケージが使用されます。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

dlob の値が NULL の場合、このメソッドをコールすると NULL_SOURCE 例外が呼び出されます。

使用するソース・プラグインが import() メソッドをサポートしない場合、METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

例

外部データ・ソースからローカル・ソースにデータをインポートし、例外をチェックします。

```
DECLARE
    SRC ORDSYS.ORDSource;
    mType VARCHAR2(4000);
    format VARCHAR2(4000);
    dblob BLOB;
    ctx RAW(4000) :=NULL;
BEGIN
    SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
    SRC.setSourceInformation('FILE','BFILEDIR','testaud.dat');
    SRC.import(ctx,dblob,mType,format);
    COMMIT;
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
        DBMS_OUTPUT.put_line('SOURCE_PLUGIN_EXCEPTION caught');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('OTHER EXCEPTION caught');
END;
/
```

importFrom() メソッド

形式

```
importFrom(  
    ctx                IN OUT RAW,  
    dlob               IN OUT NOCOPY BLOB,  
    mimetype           OUT VARCHAR2,  
    format             OUT VARCHAR2  
    source_type        IN VARCHAR2,  
    source_location    IN VARCHAR2,  
    source_name        IN VARCHAR2);
```

説明

指定された外部データ・ソース（タイプ、位置、名前）から **Oracle** データベース内のローカル・ソースヘデータを転送し、ソース属性とタイムスタンプをリセットします。

パラメータ

ctx

ソース・プラグインのコンテキスト情報。この情報は、**importFrom()** コールを処理するソース・プラグインへ未解釈のまま渡されます。

dlob

宛先ラージ・オブジェクトまたはデータ・オブジェクト

mimetype

データの MIME タイプ（存在する場合）を受け取る出力パラメータ（'audio/basic' など）

format

データのフォーマット（存在する場合）を受け取る出力パラメータ（'AUFF' など）

source_type

データのインポート元のソース・タイプ。これにより、**srcType** 属性も設定されます。

source_location

データのインポート元のソース位置。これにより、**srcLocation** 属性も設定されます。

source_name

インポートするソース名。これにより、**srcName** 属性も設定されます。

使用方法

このメソッドは、タイプ、位置、および名前パラメータの値を指定することにより、データ・ソースの位置を記述します。これらの各パラメータは、**importFrom** 操作の成功後に、**srcType**、**srcLocation**、および **srcName** 属性値をそれぞれ設定します。

このメソッドを使用する前に、ディレクトリが存在する、または作成されることを確認する必要があります。

このメソッドは、**setSourceInformation()** コールおよびそれに続く **import()** コールの組合せです。

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

プラグマ

なし

例外

dlob の値が NULL の場合、このメソッドをコールすると **NULL_SOURCE** 例外が呼び出されます。

使用するソース・プラグインが **importFrom()** メソッドをサポートしない場合、**METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

指定されたデータ・ソースから BLOB"l" 内へ、データをインポートします。

```
DECLARE
  SRC ORDSYS.ORDSource;
  mType VARCHAR2(4000);
  format VARCHAR2(4000);
  l BLOB;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.importFrom(ctx, l, mType, format, 'FILE', 'AUDIODIR', 'testaud.dat');
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SRC.getContentLength(ctx)));
  UPDATE TS SET S=SRC WHERE N=1;
  COMMIT;
END;
```

export() メソッド

形式

```
export (
    ctx                IN OUT RAW,
    source_type         IN VARCHAR2,
    source_location     IN VARCHAR2,
    source_name         IN VARCHAR2);
```

説明

Oracle データベース内のローカル・ソース (localData) から、外部データ・ソースへ、データを転送します。

注意: このメソッドは、外部ソースでのみサポートされています。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

source_type
データのエクスポート先のソース・タイプ

source_location
データのエクスポート先の位置

source_name
データのエクスポート先のオブジェクト名

使用方法

このメソッドは、localData から別のソースへデータをエクスポートします。

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

データをエクスポート後に、srcType、srcLocation、および srcName 属性は入力パラメータ値で更新されます。エクスポート・メソッドのコール後に、localData 属性の内容を削除する場合は、deleteLocalContent メソッドをコールしてローカル・データの内容を削除してください。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが export() メソッドをサポートしない場合、METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

例

ローカル・ソースから外部データ・ソースへ、データをエクスポートします。

```
DECLARE
  obj ORDSYS.ORDSource;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT S INTO obj FROM TS WHERE N = 1;
  obj.export(ctx, 'FILE', 'AUDIODIR', 'testaud.dat');
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('OTHER EXCEPTION caught');
END;
/
```


6.2.6 localData 属性と関連付けられた ORDSOURCE メソッド

この項では、localData 属性と関連付けられた ORDSOURCE メソッドの参照情報を提供します。

getContenLength() メソッド

形式

```
getContenLength(ctx IN OUT RAW) RETURN INTEGER;
```

説明

ソースに格納されたデータ・コンテンツの長さを返します。**FILE** ソースおよびローカル **BLOB** データ・ソース内のデータの場合、長さはバイト数で返されます。戻り値のユニット・タイプは、このメソッドをインプリメントするプラグインにより定義されます。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

使用方法

このメソッドは、すべてのソース・タイプでサポートされているわけではありません。たとえば、**HTTP** タイプ・ソースはこのメソッドをサポートしません。**HTTP** タイプ・ソースに対してこのコールをインプリメントする場合は、修正した独自の **HTTP** ソース・プラグインを定義してから、このメソッドをインプリメントする必要があります。

このメソッドのコールでは、**ORDPLUGINS.ORDX_<srcType>_SOURCE** プラグイン・パッケージが使用されます。

プラグマ

Pragma RESTRICT_REFERENCES(getContenLength, WNDS, WNPS, RNDS, RNPS)

例外

srcType の値が **NULL** で、データが **BLOB** 内にローカルに格納されない場合、このメソッドをコールすると **INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

ソースに格納されたデータ・コンテンツの長さを取得します。

```
DECLARE  
SRC ORDSYS.ORDSource;
```

```
ctx RAW(4000) :=NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1;
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceInformation);
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SRC.getLength(ctx)));
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.INCOMPLETE_SOURCE_INFORMATION THEN
    DBMS_OUTPUT.PUT_LINE('Source not specified');
END;
/
```

getSourceAddress() メソッド

形式

```
getSourceAddress(ctx IN OUT RAW,  
                userData IN VARCHAR2) RETURN VARCHAR2;
```

説明

外部データ・ソース内にあるデータのソース・アドレスを返します。このメソッドは、ユーザー定義ソースでのみインプリメントされます。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

userData

目的のソース・アドレスを取得するために、ソースが必要とするユーザーによる入力情報

使用方法

ソースがこの情報を独自の方法でフォーマットする必要がある場合、このメソッドを使用して外部データ・ソースのアドレスを返します。たとえば、`getSourceAddress()` メソッドをコールして、**Real Networks** サーバー・ソースのアドレスまたは **Oracle Application Server** 上のデータ・ソースを含む URL を取得します。

このメソッドのコールでは、`ORDPLUGINS.ORDX_<srcType>_SOURCE` プラグイン・パッケージが使用されます。

プラグマ

`Pragma RESTRICT_REFERENCES(getSourceAddress, WNDS, WNPS, RNDS, RNPS)`

例外

`srcType` の値が `NULL` の場合、このメソッドをコールすると `INCOMPLETE_SOURCE_INFORMATION` 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、`SOURCE_PLUGIN_EXCEPTION` 例外が呼び出されます。

例

外部ソースのソース・アドレスを取得します。

```
DECLARE
    SRC ORDSYS.ORDSource;
    ctx RAW(4000) :=NULL;
    userData VARCHAR2(4000);
BEGIN
    SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
    -- process tempBlob
    SRC.setSourceInformation('FILE','AUDIODIR','testaud.dat');
    userData :=NULL;
    DBMS_OUTPUT.PUT_LINE(SRC.getSourceAddress(ctx,userData));
    COMMIT;
END;
/
```

getLocalContent メソッド

形式

```
getLocalContent RETURN BLOB;
```

説明

ローカル・データの内容または BLOB ハンドルを返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getLocalContent, WNDS, WNPS, RNDS, RNPS)

例外

なし

例

インポート操作に先立ち、ローカル・ソースのローカル・コンテンツを取得します。

```
DECLARE
  SRC ORDSYS.ORDSource;
  l BLOB;
  mimeType VARCHAR2(4000);
  format VARCHAR2(4000);
  ctx RAW(4000) := NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  l := SRC.getLocalContent;
  SRC.importFrom(ctx, l, mimeType, format, 'FILE', 'AUDIODIR', 'testaud.dat');
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.GETLENGTH(l)));
  UPDATE TS SET S=SRC WHERE N=1;
  COMMIT;
END;
```

getContentInTempLob() メソッド

形式

```
getContentInTempLob(  
    ctx          IN OUT RAW,  
    tempLob      IN OUT NOCOPY BLOB,  
    mimetype     OUT VARCHAR2,  
    format       OUT VARCHAR2,  
    duration     IN PLS_INTEGER := 10,  
    cache        IN BOOLEAN := TRUE);
```

説明

現行のデータ・ソースからテンポラリ LOB ヘデータを転送します。テンポラリ LOB は、このコールの一部として割当ておよび初期化が行われます。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

tempLob

このコールで割当てが行われる、未初期化の BLOB ロケータ

mimetype

データの MIME タイプを受け取る出力パラメータ ('audio/basic' など)

format

データのフォーマットを受け取る出力パラメータ ('AUFF' など)

duration

割り当てられるテンポラリ LOB の存続期間。テンポラリ LOB の存続期間は、コール、トランザクション、またはセッションの持続時間です。デフォルトは、DBMS_LOB.SESSION です。各持続状態で有効な値は、次のとおりです。

DBMS_LOB.CALL

DBMS_LOB.TRANSACTION

DBMS_LOB.SESSION

cache

データをキャッシュするかどうかを指定します。値は、TRUE または FALSE です。デフォルトは TRUE です。

使用方法

なし

プラグマ

なし

例外

テンポラリ **LOB** での作業中に、ループ処理の設定された読み込み操作が **LOB** の終端に達して、**LOB** から読むバイトがそれ以上存在しない場合に、**NO_DATA_FOUND** 例外が呼び出される場合があります。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

外部データ・ソースからローカル・ソース上のテンポラリ **LOB** に、データを取得します。

```
DECLARE
  SRC ORDSYS.ORDSource;
  userData VARCHAR2(4000);
  l BLOB;
  tempBlob BLOB;
  mimeType VARCHAR2(4000);
  format VARCHAR2(4000);
  ctx RAW(4000) := NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.importFrom(ctx, src.localData, mimeType, format, 'FILE', 'AUDIODIR', 'testaud.dat');
  SRC.getContentInTempLob(ctx, tempBlob,
                          mimeType, format, 0, FALSE);

  -- process tempBlob

  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SRC.getContentLength(ctx)));
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceAddress(ctx, userData));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(DBMS_LOB.GETLENGTH(tempBlob)));
  UPDATE TS SET S=SRC WHERE N=1;
  COMMIT;
END;
/
```

deleteLocalContent メソッド

形式

```
deleteLocalContent;
```

説明

現行のローカル・ソース (localData) からローカル・データを削除します。

パラメータ

なし

使用方法

ローカル・ソースから外部データ・ソースヘデータをエクスポートして、ローカル・ソース内にデータが必要なくなった後に、このメソッドをコールできます。

プラグマ

なし

例外

なし

例

現行のローカル・ソースからローカル・データを削除します。

```
DECLARE
  SRC ORDSYS.ORDSource;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  SRC.deleteLocalContent;
  UPDATE TS SET S=SRC WHERE N=1;
  COMMIT;
END;
/
```

6.2.7 ファイル操作と関連付けられた ORDSource メソッド

この項では、外部データ・ソースへのアクセスと関連付けられた ORDSource メソッドの参照情報を提供します。

open() メソッド

形式

```
open(userArg IN RAW, ctx OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをオープンします。**ctx** パラメータを受け取る他の任意のメソッドをコールする前に、このメソッドをコールすることをお勧めします。

パラメータ

userArg

ユーザー引数

ctx

ソース・プラグインのコンテキスト情報

使用方法

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

返される **INTEGER** は、成功時には **0** (ゼロ)、失敗時には **>0** (1 など) になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、**1** は " ファイルが見つからない " を、**2** は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

srcType の値が **NULL** で、データがローカルではない場合、このメソッドをコールすると **INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

使用するソース・プラグインが **open()** メソッドをサポートしない場合、**METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

外部データ・ソースをオープンします。

```
DECLARE
    SRC ORDSYS.ORDSource;
    res INTEGER;
    ctx RAW(4000) :=NULL;
    userArg RAW(4000);
BEGIN
    SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
    res := SRC.open(userArg, ctx);
    -- manipulate the source
    res := SRC.close(ctx);
EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
        DEMS_OUTPUT.PUT_LINE('Source not specified');
    WHEN OTHERS THEN
        DEMS_OUTPUT.PUT_LINE('Exception caught');
END;
/
```

close() メソッド

形式

```
close(ctx IN OUT RAW) RETURN INTEGER;
```

説明

データ・ソースをクローズします。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

使用方法

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

返される **INTEGER** は、成功時には **0** (ゼロ)、失敗時には **>0** (1 など) になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、**1** は " ファイルが見つからない " を、**2** は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

srcType の値が **NULL** で、データがローカルではない場合、このメソッドをコールすると **INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

使用するソース・プラグインが **close()** メソッドをサポートしない場合、**METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

外部データ・ソースをクローズします。

```
DECLARE
```

close() メソッド

```
SRC ORDSYS.ORDSource;  
res INTEGER;  
ctx RAW(4000) :=NULL;  
BEGIN  
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;  
  res := SRC.close(ctx);  
  UPDATE TS SET S=SRC WHERE N=1;  
  COMMIT;  
  EXCEPTION  
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN  
      DBMS_OUTPUT.PUT_LINE('Source not specified');  
    WHEN OTHERS THEN  
      DBMS_OUTPUT.PUT_LINE('Exception caught');  
END;  
/
```

trim() メソッド

形式

```
trim(ctx IN OUT RAW,  
      newlen IN INTEGER) RETURN INTEGER;
```

説明

データ・ソースの切捨てを実行します。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

newlen

切捨て実行後の新しい長さ

使用方法

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

返される INTEGER は、成功時には 0（ゼロ）、失敗時には >0（1 など）になります。その数値の正確な値と意味は、プラグインごとに定義されています。たとえば、ファイル・プラグインの場合、1 は " ファイルが見つからない " を、2 は " そのようなディレクトリは存在しない " を意味する、といった具合です。

プラグマ

なし

例外

srcType の値が NULL で、データがローカルではない場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが trim() メソッドをサポートしない場合、METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

例

外部データ・ソースの切捨てを実行します。

```
DECLARE
  SRC ORDSYS.ORDSource;
  res INTEGER;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  res := SRC.trim(ctx,0);
  UPDATE TS SET S=SRC WHERE N=1;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.PUT_LINE('Source not specified');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Exception caught');
END;
/
```


6.2.8 読み込み / 書き込み操作と関連付けられた ORDSOURCE メソッド

この項では、読み込み / 書き込み操作と関連付けられた ORDSOURCE メソッドの参照情報を提供します。

read() メソッド

形式

```
read(  
    ctx      IN OUT RAW,  
    startPos IN INTEGER,  
    numBytes IN OUT INTEGER,  
    buffer   OUT RAW);
```

説明

開始位置（startPos）から numBytes のバッファの、ソースからの読み込みを可能にします。

パラメータ

ctx
ソース・プラグインのコンテキスト情報

startPos
データ・ソース内の開始位置

numBytes
データ・ソースから読み込むバイト数

buffer
データの読み込み先バッファ

使用方法

このメソッドは、HTTP ソースではサポートされません。

HTTP ソース・タイプの読み込みを確実に行うには、URL ソース全体の読み込み要求を実行する必要があります。HTTP ソース・タイプ用の読み込みメソッドをインプリメントする場合は、HTTP ソース・タイプ用に修正されたソース・プラグイン内でこのメソッドのインプリメンテーションを独自に提供する必要があります。

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

プラグマ

なし

例外

データがローカルに格納され、**localData** が **NULL** の場合、このメソッドをコールすると **NULL_SOURCE** 例外が呼び出されます。

srcType の値が **NULL** で、データがローカルではない場合、このメソッドをコールすると **INCOMPLETE_SOURCE_INFORMATION** 例外が呼び出されます。

使用するソース・プラグインが **read()** メソッドをサポートしない場合、**METHOD_NOT_SUPPORTED** 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、**SOURCE_PLUGIN_EXCEPTION** 例外が呼び出されます。

例

ソースからバッファを読み込みます。

```
DECLARE
  SRC ORDSYS.ORDSource;
  i INTEGER;
  buffer RAW(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  i := 20;
  SELECT S INTO SRC FROM TS WHERE N = 1 ;
  SRC.read(ctx, 1, i, buffer);
END;
/
```

write() メソッド

形式

```
write(  
    ctx          IN OUT RAW,  
    startPos IN  INTEGER,  
    numBytes IN OUT INTEGER,  
    buffer   IN  RAW);
```

説明

numBytes のバッファの、ソース上の開始位置 (startPos) への書き込みを可能にします。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

startPos

ソース内でバッファのコピーが開始される位置

numBytes

ソースへ書き込まれるバイト数

buffer

書き込まれるデータ・バッファ

使用方法

このメソッドは、書き込み可能ソースに対し、ランダムなバイト位置から numBytes の書き込みが可能である、ということを前提にしています。たとえば、FILE および HTTP ソース・タイプは、書き込み可能ソースではないため、このメソッドをサポートしません。

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

プラグマ

なし

例外

ローカルの値が 1 または NULL で、localData が NULL の場合、このメソッドをコールすると NULL_SOURCE 例外が呼び出されます。

srcType の値が NULL で、データがローカルではない場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが write() メソッドをサポートしない場合、METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

例

バッファをソースに書き込みます。

```
DECLARE
  SRC ORDSYS.ORDSource;
  n INTEGER := 6;
  ctx RAW(4000) :=NULL;
BEGIN
  SELECT S INTO SRC FROM TS WHERE N = 1 FOR UPDATE;
  DBMS_OUTPUT.PUT_LINE(SRC.getSourceInformation);
  SRC.write(ctx, 1, n, UTL_RAW.CAST_TO_RAW('helloP'));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SRC.getContenLength(ctx)));
  COMMIT;
END;
/
```

6.2.9 外部ソースへの処理コマンドと関連付けられた ORDSOURCE メソッド

この項では、外部ソースへの処理コマンドと関連付けられた ORDSOURCE メソッドの参照情報を提供します。

processCommand() メソッド

形式

```
processCommand(  
    ctx      IN OUT RAW,  
    command  IN VARCHAR2,  
    arglist  IN VARCHAR2,  
    result   OUT RAW)  
RETURN RAW;
```

説明

コマンドおよび関連する引数のソース・プラグインへの送信を可能にします。このメソッドはユーザー定義ソースでのみサポートされています。

パラメータ

ctx

ソース・プラグインのコンテキスト情報

command

ソース・プラグインにより認識される任意のコマンド

arglist

コマンドの引数

result

プラグインにより返される、このメソッドへのコール結果

使用方法

このメソッドを使用して、任意のコマンドおよびその引数をプラグインに送信します。このメソッドはコマンドを解釈しません。受け取って処理へ渡すだけです。

このメソッドのコールでは、ORDPLUGINS.ORDX_<srcType>_SOURCE プラグイン・パッケージが使用されます。

プラグマ

なし

例外

srcType の値が NULL の場合、このメソッドをコールすると INCOMPLETE_SOURCE_INFORMATION 例外が呼び出されます。

使用するソース・プラグインが processCommand() メソッドをサポートしない場合、METHOD_NOT_SUPPORTED 例外が呼び出されます。

他の任意の例外の呼出し時にソース・プラグイン内でこのメソッドをコールすると、SOURCE_PLUGIN_EXCEPTION 例外が呼び出されます。

例

いくつかのコマンドを処理します。

```
DECLARE
  obj ORDSYS.ORDSource ;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR2(4000);
  argList VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  select s into obj from TS where N =1 for UPDATE;
  command := 'xxx ';
  argList := 'yyy ';
  res := obj.processCommand(ctx, command, argList, result);
  UPDATE TS SET s=obj WHERE N=1 ;
  COMMIT;
EXCEPTION
  WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
    DBMS_OUTPUT.put_line('Source METHOD_NOT_SUPPORTED caught');
  WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN_EXCEPTION THEN
    DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
  WHEN OTHERS THEN
    DBMS_OUTPUT.put_line('OTHER EXCEPTION caught');
END;
/
```


6.3 パッケージまたは PL/SQL プラグイン

この項では、提供されるパッケージまたは PL/SQL プラグインの参照情報を提供します。

ソース・プラグインのコールから起動した任意のメソッドは、最初の引数に **obj** (ORDSource) を取り、2 番目の引数に **ctx** (RAW) を取ります。

プラグインの名前は、ORDX_<name>_<module_name> の形式で指定する必要があります。<module_name> には、ORDSource の SOURCE が当てはまります。たとえば、6.3.1 項に示した FILE プラグインは、ORDX_FILE_SOURCE と命名されます。<name> はソース・タイプです。

例外は、ORD_<module_name>Exceptions として命名されたパッケージから呼び出し、また記録する必要があります。たとえば、ORDSource 例外は、ORDSourceExceptions という名前のパッケージ内で呼び出され、また記録されます。(付録 H を参照してください。)

6.3.1 ORDPLUGINS.ORDX_FILE_SOURCE パッケージ

ORDPLUGINS.ORDX_FILE_SOURCE パッケージまたは PL/SQL プラグインが提供されます。

```
CREATE OR REPLACE PACKAGE ORDX_FILE_SOURCE AS
  -- functions/procedures
  FUNCTION processCommand(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                           ctx      IN OUT RAW,
                           cmd      IN VARCHAR2,
                           arglist  IN VARCHAR2,
                           result   OUT RAW)
    RETURN RAW;
  PROCEDURE import(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                   ctx      IN OUT RAW,
                   dlob      IN OUT NOCOPY BLOB,
                   mimetype  OUT VARCHAR2,
                   format    OUT VARCHAR2);
  PROCEDURE importFrom(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                       ctx      IN OUT RAW,
                       dlob      IN OUT NOCOPY BLOB,
                       mimetype  OUT VARCHAR2,
                       format    OUT VARCHAR2,
                       loc       IN VARCHAR2,
                       name      IN VARCHAR2);
  PROCEDURE export(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                   ctx      IN OUT RAW,
                   dlob      IN OUT NOCOPY BLOB,
                   loc       IN VARCHAR2,
                   name      IN VARCHAR2);
  FUNCTION getLength(obj      IN ORDSYS.ORDSource,
                     ctx      IN OUT RAW),
```

```

        RETURN INTEGER;
PRAGMA RESTRICT_REFERENCES (getContentLength, WNDS, WNPS, RNDS, RNPS);
FUNCTION getSourceAddress (obj IN ORDSYS.ORDSource,
                           ctx IN OUT RAW,
                           userData IN VARCHAR2)
    RETURN VARCHAR2;
PRAGMA RESTRICT_REFERENCES (getSourceAddress, WNDS, WNPS, RNDS, RNPS);
FUNCTION open (obj IN OUT NOCOPY ORDSYS.ORDSource, userArg IN RAW,
              ctx OUT RAW) RETURN INTEGER;
FUNCTION close (obj IN OUT NOCOPY ORDSYS.ORDSource, ctx IN OUT RAW)
    RETURN INTEGER;
FUNCTION trim (obj IN OUT NOCOPY ORDSYS.ORDSource,
              ctx IN OUT RAW,
              newlen IN INTEGER) RETURN INTEGER;
PROCEDURE read (obj IN OUT NOCOPY ORDSYS.ORDSource,
               ctx IN OUT RAW,
               startPos IN INTEGER,
               numBytes IN OUT INTEGER,
               buffer OUT RAW);
PROCEDURE write (obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                startPos IN INTEGER,
                numBytes IN OUT INTEGER,
                buffer OUT RAW);
END ORDX_FILE_SOURCE;
/

```

表 6-1 に、ORDX_FILE_SOURCE パッケージがサポートするメソッド、およびサポートしないメソッドをコールした場合に呼び出される例外を示します。

表 6-1 ORDPLUGINS.ORDX_FILE_SOURCE パッケージがサポートするメソッド

メソッド名	サポート・レベル
processCommand	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
import	サポートします。
importFrom	サポートします。
export	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
getContentLength	サポートします。
getSourceAddress	サポートします。
open	サポートします。
close	サポートします。

表 6-1 ORDPLUGINS.ORDX_FILE_SOURCE パッケージがサポートするメソッド (続き)

メソッド名	サポート・レベル
trim	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
read	サポートします。
write	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。

6.3.2 ORDPLUGINS.ORDX_HTTP_SOURCE パッケージ

ORDPLUGINS.ORDX_HTTP_SOURCE パッケージまたは PL/SQL プラグインが提供されます。

```

CREATE OR REPLACE PACKAGE ORDX_HTTP_SOURCE AS
-- functions/procedures
FUNCTION processCommand(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                        ctx      IN OUT RAW,
                        cmd      IN VARCHAR2,
                        arglist  IN VARCHAR2,
                        result   OUT RAW)
    RETURN RAW;
PROCEDURE import(obj      IN OUT NOCOPY ORDSYS.ORDSource,
                 ctx      IN OUT RAW,
                 dlob     IN OUT NOCOPY BLOB,
                 mimetype  OUT VARCHAR2,
                 format    OUT VARCHAR2);
PROCEDURE importFrom(obj   IN OUT NOCOPY ORDSYS.ORDSource,
                    ctx    IN OUT RAW,
                    dlob   IN OUT NOCOPY BLOB,
                    mimetype OUT VARCHAR2,
                    format  OUT VARCHAR2,
                    loc     IN VARCHAR2,
                    name    IN VARCHAR2);
PROCEDURE export(obj   IN OUT NOCOPY ORDSYS.ORDSource,
                 ctx    IN OUT RAW,
                 dlob   IN OUT NOCOPY BLOB,
                 loc    IN VARCHAR2,
                 name    IN VARCHAR2);
FUNCTION getLength(obj   IN ORDSYS.ORDSource,
                 ctx    IN OUT RAW)
    RETURN INTEGER;
PRAGMA RESTRICT_REFERENCES(getLength, WNDS, WNPS, RNDS, RNPS);
FUNCTION getAddress(obj   IN ORDSYS.ORDSource,
                  ctx    IN OUT RAW,

```

```

                                userData IN VARCHAR2)
RETURN VARCHAR2;
PRAGMA RESTRICT_REFERENCES (getSourceAddress, WNDS, WNPS, RNDS, RNPS);
FUNCTION open(obj IN OUT NOCOPY ORDSYS.ORDSource, userArg IN RAW,
              ctx OUT RAW) RETURN INTEGER;
FUNCTION close(obj IN OUT NOCOPY ORDSYS.ORDSource, ctx IN OUT RAW)
RETURN INTEGER;
FUNCTION trim(obj IN OUT NOCOPY ORDSYS.ORDSource,
              ctx IN OUT RAW,
              newlen IN INTEGER) RETURN INTEGER;
PROCEDURE read(obj          IN OUT NOCOPY ORDSYS.ORDSource,
               ctx          IN OUT RAW,
               startPos IN INTEGER,
               numBytes IN OUT INTEGER,
               buffer OUT RAW);
PROCEDURE write(obj          IN OUT NOCOPY ORDSYS.ORDSource,
                ctx          IN OUT RAW,
                startPos IN INTEGER,
                numBytes IN OUT INTEGER,
                buffer OUT RAW);
END ORDX_HTTP_SOURCE;
/

```

表 6-2 に、ORDX_HTTP_SOURCE パッケージがサポートするメソッド、およびサポートしないメソッドをコールした場合に呼び出される例外を示します。

表 6-2 ORDPUGINS.ORDX_HTTP_SOURCE パッケージがサポートするメソッド

メソッド名	サポート・レベル
processCommand	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
import	サポートします。
importFrom	サポートします。
export	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
getContentLength	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
getSourceAddress	サポートします。
open	サポートします。
close	サポートします。
trim	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。

表 6-2 ORDPLUGINS.ORDX_HTTP_SOURCE パッケージがサポートするメソッド (続き)

メソッド名	サポート・レベル
read	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。
write	サポートしません。METHOD_NOT_SUPPORTED 例外が呼び出されます。

6.3.3 ORDPLUGINS.ORDX_<srcType>_SOURCE パッケージ

ORDPLUGINS.ORDX_<srcType>_SOURCE パッケージまたは PL/SQL プラグインを、独自のソース・タイプを作成する際のテンプレートとして使用します。ORDPLUGINS.ORDX_FILE_SOURCE および ORDPLUGINS.ORDX_HTTP_SOURCE パッケージを、新規ソース・タイプ・パッケージを開発する際のガイドとして使用します。

6.3.4 *interMedia* の拡張による新しいデータ・ソースのサポート

interMedia を拡張して、新規データ・ソースをサポートします。次の 4 つの手順で実行します。

1. 新規データ・ソースを設計します。
2. 新規データ・ソースをインプリメントして、名前 (ORDX_MY_SOURCE.SQL など) を付けます。
3. 新規 ORDX_MY_SOURCE.SQL プラグインを ORDPLUGINS スキーマにインストールします。
4. 新規プラグインに EXECUTE 権限を付与します。たとえば、ORDX_MY_SOURCE.SQL プラグインに PUBLIC への権限を付与します。

2.4 項に、*interMedia* を拡張して、オーディオおよびビデオ・データ用の新規データ・ソースをサポートし、インタフェースを記述する方法が簡潔に説明されています。この操作方法については、例 6-1 のパッケージ本体リストを参照してください。"--Your variables go here" と示された箇所に変数を追加し、"--Your code goes here" と示された箇所にコードを追加します。

例 6-1 新規データ・ソースへのサポートを拡張する、パッケージ本体リスト

```
CREATE OR REPLACE PACKAGE BODY ORDX_MY_SOURCE
AS
  -- functions/procedures
  FUNCTION processCommand(
    obj   IN OUT NOCOPY ORDSYS.ORDSource,
    ctx   IN OUT RAW,
    cmd   IN VARCHAR2,
    arglist IN VARCHAR2,
```

```

                                result OUT RAW)
RETURN RAW
IS
--Your variables go here
BEGIN
--Your code goes here
END processCommand;
PROCEDURE import( obj  IN OUT NOCOPY ORDSYS.ORDSource,
                  ctx  IN OUT RAW,
                  dlob IN OUT NOCOPY BLOB,
                  mimetype OUT VARCHAR2,
                  format  OUT VARCHAR2)

IS
--Your variables go here
BEGIN
--Your code goes here
END import;
PROCEDURE importFrom( obj      IN OUT NOCOPY ORDSYS.ORDSource,
                     ctx      IN OUT RAW,
                     dlob     IN OUT NOCOPY BLOB,
                     mimetype OUT VARCHAR2,
                     format   OUT VARCHAR2,
                     loc      IN VARCHAR2,
                     name     IN VARCHAR2)

IS
--Your variables go here
BEGIN
--Your code goes here
END importFrom;
PROCEDURE export( obj  IN OUT NOCOPY ORDSYS.ORDSource,
                 ctx  IN OUT RAW,
                 dlob IN OUT NOCOPY BLOB,
                 loc  IN VARCHAR2,
                 name IN VARCHAR2)

IS
--Your variables go here
BEGIN
--Your code goes here
END export;

FUNCTION getContentLength( obj  IN ORDSYS.ORDSource,
                          ctx  IN OUT RAW)

RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
```

```

END getContentLength;
FUNCTION getSourceAddress(obj IN ORDSYS.ORDSource,
                           ctx IN OUT RAW,
                           userData IN VARCHAR2)
RETURN VARCHAR2
IS
--Your variables go here
BEGIN
--Your code goes here
END getSourceAddress;
FUNCTION open(obj IN OUT NOCOPY ORDSYS.ORDSource, userArg IN RAW, ctx OUT RAW)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END open;
FUNCTION close(obj IN OUT NOCOPY ORDSYS.ORDSource, ctx IN OUT RAW)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END close;
FUNCTION trim(obj          IN OUT NOCOPY ORDSYS.ORDSource,
               ctx          IN OUT RAW,
               newlen IN INTEGER)
RETURN INTEGER
IS
--Your variables go here
BEGIN
--Your code goes here
END trim;
PROCEDURE read(obj          IN OUT NOCOPY ORDSYS.ORDSource,
                ctx          IN OUT RAW,
                startPos IN INTEGER,
                numBytes IN OUT INTEGER,
                buffer OUT RAW)
IS
--Your variables go here
BEGIN
--Your code goes here
END read;
PROCEDURE write(obj          IN OUT NOCOPY ORDSYS.ORDSource,
                 ctx          IN OUT RAW,
                 startPos IN INTEGER,
                 numBytes IN OUT INTEGER,

```

```
                buffer OUT RAW)
IS
  --Your variables go here
BEGIN
  --Your code goes here
  END write;
END ORDX_MY_SOURCE;
/
show errors;
```

DBA のためのチューニング・ヒント

この章では、Oracle DBA が Oracle8i *interMedia* を使うときにマルチメディア・データをデータベースに効率的に格納するためのチューニング・ヒントを提供します。マルチメディア・データは、文字セット、イメージ、オーディオ・クリップ、ビデオ・クリップ、線画などさまざまなメディア・タイプから構成されます。通常、これらのメディア・タイプはすべて、内部データベース表領域内に格納されている内部 LOB か、データベース表領域外のオペレーティング・システム・ファイル内、つまり BFILE 内の外部 LOB のどちらかの LOB に格納されています。

内部 LOB には CLOB、NCLOB および BLOB があり、これらのサイズは最大 4GB にすることができます。BFILE は、最大 4GB までの、オペレーティング・システムによって許可されるサイズにすることができます。

Oracle8i *interMedia* では、さまざまな LOB 型を管理します。次の一般的なトピックは、*interMedia* LOB データのより効果的な管理に役立ちます。

- *interMedia* LOB データの INSERT パフォーマンスの向上
- 他の大量ロード方法の使用
- 最大パフォーマンス実現のためのユーザー・ガイドライン
- *interMedia* LOB データの検索および更新パフォーマンスの向上
- オープン BFILE の最大数の設定
- パーティション表内での *interMedia* LOB の使用

詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』、『Oracle8i 概要』および『Oracle8i チューニング』の各マニュアルの LOB パーティション化、LOB チューニングおよび LOB バッファリングの説明を参照してください。

LOB を使うときに考慮しなければならない制限事項の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

Oracle8i の DIRECTORY 機能を使用して、DBA がサーバー・ファイル・システムのラージ・ファイルへのアクセスを管理できる、単純で柔軟な、割込みのない、安全なメカニズム

を実現するためのガイドラインの詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

7.1 マルチメディア LOB データの INSERT パフォーマンスの向上

LOB を含む表を作成するときの考慮点

次の説明では、LOB 列を含む表を作成するときには考慮しなければならない方針を示します。これらのトピックの詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』で例を挙げて説明しています。次の説明は第 2 章の全体または一部からの抜粋で、トピックの概要を簡単に提示しています。最新情報は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

- 内部 LOB を NULL または EMPTY に初期化します。

NULL に設定された LOB 値にはロケータがありません。いっぽう、表に格納された EMPTY LOB は、ロケータを持つ長さ 0（ゼロ）の LOB です。したがって、EMPTY LOB 列または属性から SELECT するとロケータが返され、そのロケータを使用して、OCI または DBMS_LOB ルーチンでその LOB にデータを格納できます。

- LOB を NULL に設定します。

INSERT 操作時に LOB データがないか、後で SELECT 文を発行したい場合、またはその両方の場合に、行挿入時に内部 LOB 値を NULL に設定できます。

ただし、このアプローチの欠点は、次に SQL UPDATE 文を発行して、NULL LOB 列を EMPTY_BLOB() または EMPTY_CLOB()、あるいは内部 LOB の既知の値または外部 LOB のファイル名にリセットしなければならないことです。重要な点は、NULL である LOB に対して、OCI または PL/SQL DBMS_LOB 関数をコールすることはできないということです。これらの関数はロケータに対してのみ使うことができますが、LOB 列が NULL の場合は、行内にロケータはありません。

- 内部 LOB を EMPTY に設定します。

内部 LOB 列を NULL に設定したくない場合は、INSERT 文で関数 EMPTY_BLOB() または EMPTY_CLOB() を使用して、LOB 値を EMPTY に設定できます。

よりよい方法としては、RETURNING 句を使います。（これによって、後続の SELECT 文に必要なラウンドトリップを削減できます。）その直後に OCI または PL/SQL DBMS_LOB 関数をコールして、LOB にデータを格納します。例は『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

- 内部 LOB の表領域および記憶特性を規定します。
 - 表領域

多くの場合、LOB の最大パフォーマンスは、LOB 列を含む表に使われる表領域とは異なる表領域に、LOB の記憶域を指定することで実現できます。また、多くの異なる LOB が頻繁にアクセスされる場合も、デバイスの競合を減らすために各 LOB 列または属性に個別の表領域を指定すると効果的です。BLOB データ挿入時の割当てを避けるために、表領域を必要な割当てサイズに事前割当てします。例は『Oracle8i SQL リファレンス』を参照してください。具体的には CREATE TABLE 文および LOB 列の例を参照してください。例 7-1 を参照してください。

- LOB 索引

LOB 索引は、LOB 記憶域に深く関連する内部構造体です。このため、ユーザーは LOB 索引を削除して再作成することはできません。LOB 索引は変更できないことに注意してください。LOB データおよび LOB 索引用に使われる表領域は、LOB 記憶域句内のユーザーの設定に応じて、システムによって次のように決定されます。

- * LOB データの表領域を指定しない場合は、表の表領域が LOB データおよび LOB 索引用に使われます。
- * LOB データの表領域を指定した場合は、指定した表領域が LOB データおよび LOB 索引の両方によって使われます。

Oracle リリース 8.1 で表を作成するときに、非パーティション表に対して LOB 索引の表領域を指定した場合は、指定した表領域は無視され、LOB 索引および LOB データによって同一の表領域が使われます。パーティション LOB には LOB 索引構文が含まれません。

各 LOB 記憶域セグメントに個別の表領域を指定すると、表の表領域への競合が減少します。

- PCTVERSION オプション

LOB が修正されたときは、前バージョンの LOB 値の一貫した読取りをサポートするために、新しいバージョンの LOB ページが作成されます。

PCTVERSION は、古いバージョンの LOB データ・ページによって使用可能なすべての使用済み LOB データ領域のパーセントです。古いバージョンの LOB データ・ページによって使用される領域が、使用済み LOB 領域の PCTVERSION 量を超えると、Oracle はただちに古いバージョンを回収し、再使用します。つまり、PCTVERSION は古い LOB データのバージョン管理のために使用可能な、使用済み LOB データ・ブロックのパーセントです。

PCTVERSION を概算する 1 つの方法は、 $PCTVERSION = (\text{所定の時点で更新される LOB の \%}) \times (\text{LOB の更新のたびに更新される各 LOB の \%}) \times (\text{所定の時点で読み取られる LOB の \%})$ を設定することです。基本的にこの方法は、更新されているデータの一貫した読取りを読者によって行うことができるように、LOB 記憶領域のパーセントを古いバージョンの LOB ページとして使うことを可能にするためのものです。

PCTVERSION をデフォルトの 2 倍に設定すると、より多くの空きページを古いバージョンのデータ・ページ用に使うことができます。大規模な問合せには LOB

の一貫した読取りが必要になる可能性があるため、古いバージョンの LOB ページをより多く活用できることは役立ちます。当然、Oracle が空きページを積極的に再使用するわけではないので、LOB 記憶域は増加することがあります。

LOB 更新の頻度と量が減少すると、LOB データの古いコピー用に確保する必要がある領域は減少します。既存の LOB が読取り専用であることがわかっている場合は、古いバージョンのデータ用のページが必要にならないため、PCTVERSION を安全に 0% に設定できます。

- CACHE または NOCACHE オプション

同一の LOB データが頻繁にアクセスされる場合は、LOB に対して CACHE オプションを使います。LOB データが 1 回しか読み取られない場合や、読取り頻度が低い場合は、NOCACHE オプション（デフォルト）を使います。

大量の LOB データの読取りおよび書き込みを行うときは、I/O 操作回数を減らすために NOCACHE オプションを使います。

例 7-1 を参照してください。

- LOGGING または NOLOGGING オプション

NOLOGGING が役立つ例は、データの大量ロードまたは挿入を行う場合です。例 7-1 を参照してください。たとえば、LOB にデータをロードするときに REDO ロギングを考慮せず、ロードが失敗しても再開できる場合は、LOB のデータ・セグメント記憶特性を NOCACHE NOLOGGING に設定します。この設定は、データの初期ロードのパフォーマンスを向上します。データのロードを完了した後は、ALTER TABLE 文を使用して、LOB データ・セグメントの LOB 記憶特性を標準の LOB 操作の希望の特性に変更できます。たとえば、LOB のデータ記憶特性を CACHE LOGGING または NOCACHE LOGGING に変更します。

追加のガイドラインは次のとおりです。大量のデータを LOB に書き込むときは、NOLOGGING オプションを使うと、新しいバージョンの LOB データが作成されたときに REDO ログに書き込まれません。この方法では、生成される REDO が非常に少なくなるため、パフォーマンスが向上します。ただし、データを確実に復元できるようにするために、NOLOGGING モードの INSERT 操作の後に LOB データのバックアップを行う必要があります。

- CHUNK オプション

CHUNK オプションは、1 回でアクセスされる LOB データのブロック数に設定します。たとえば、コール OCILobRead()、OCILobWrite()、DBMS_LOB.READ() または DBMS_LOB.WRITE() による LOB 値への 1 回のアクセス中に読取りまたは書き込みされるブロック数に設定します。CHUNK オプションのデフォルト値は 1Oracle ブロックで、どのシステムでも同一であることに注意してください。たとえば、1 回に LOB データの 1 ブロックのみがアクセスされる場合は、CHUNK オプションを 1 ブロックのサイズに設定します。たとえば、データベース・ブロック・サイズが 2K の場合は、CHUNK オプションを 2K に設定します。

CHUNKSIZE オプションは、挿入されるオーディオ、イメージまたはビデオ・データのサイズより少し大きい値に設定します。少し大きい **CHUNKSIZE** を指定すると、マルチメディア・データの実際のサイズのバリエーションに対応できるので役立ちます。例 7-1 を参照してください。

- **INITIAL** および **NEXT** パラメータ

LOB の記憶特性を明示的に指定する場合は、LOB データ・セグメント記憶域の **INITIAL** および **NEXT** パラメータが **CHUNK** サイズより大きいサイズに設定されていることを確認してください。たとえば、データベース・ブロック・サイズが 2K で、8K の **CHUNK** 値を指定した場合は、**INITIAL** および **NEXT** パラメータが 8K より大きいことを確認してください。8K よりも十分に大きなサイズを設定することをお勧めします（たとえば 16K 以上）。また、**INITIAL** および **NEXT** パラメータは通常同一の値に設定する必要があります。

- **DATABLOCKSIZE**

Oracle では、データベースのデータ・ファイル内の記憶領域を、データ・ブロックと呼ばれる単位で管理します。データ・ブロックは、データベースによって使われる I/O 操作の最小単位です。データベースの作成時に、各 Oracle データベースのデータ・ブロック・サイズを設定します。不必要な I/O 操作を避けるために、データ・ブロック・サイズは、オペレーティング・システムのブロック・サイズの倍数で、ポート固有の最大制限値内の値に設定する必要があります。

LOB データの読取りおよび書込みのためには、**DATABLOCKSIZE** パラメータを、大量ロードを行う LOB データのサイズまたはその等価分数に設定します。たとえば、20KB のイメージの LOB 記憶域の最適化を試行している場合は、各 I/O 操作でより多くの LOB データを移動するために、**DATABLOCKSIZE** パラメータを 10KB または 20KB に設定します。

- **ENABLE | DISABLE STORAGE IN ROW**

LOB をインライン（つまり行内）に格納するか、行外に格納するかを指示するには、**ENABLE | DISABLE STORAGE IN ROW** 句を使います。この仕様は、一度作成すると変更できません。**ENABLE STORAGE IN ROW** に設定した場合は、**DISABLE STORAGE IN ROW** に変更することはできません。また、その逆もできません。デフォルトは **ENABLE STORAGE IN ROW** です。

行内に格納される LOB データの最大量は、最大 **VARCHAR** サイズ（4000）です。これには LOB 値の他に制御情報が含まれることに注意してください。ユーザーが LOB を行内に格納するように指示した場合は、LOB 値および制御情報が 4000 バイトを超えると、LOB 値は自動的に行外に移動されます。

このため、次のガイドラインがあります。LOB が小さい（つまり 4000 バイト未満）の場合は、LOB データを行外に格納するとパフォーマンスが低下します。ただし、LOB を行内に格納すると行サイズが増加します。全表走査や複数行アクセス（範囲走査）などの多くの実表処理、または LOB 列以外の列に対する多くの **UPDATE** 文や **SELECT** 文を実行している場合、これによりパフォーマンスに悪影響が出ます。

LOB データが 4000 バイト未満であると予想されない場合、つまり、すべての LOB が大きい場合は、次の理由でデフォルトの設定が最適です。

(a) LOB データは 4000 バイトを超えると、自動的に行外に移動されます。

(b) LOB データを行外に移動した後も、一部の制御情報を行内に格納できるため、パフォーマンスは多少よくなります。

例 7-1 では、適切な権限を持つユーザーとして CONNECT 文をすでに発行したと仮定しています。この例では、イメージ列の LOB データを格納するのに使われる、MONTANA という名前の個別の表領域を作成します。理想的に、この表領域は、競合を減らすために独自の高速記憶デバイス上に配置されるものとします。他のイメージ属性および imageID 列は、デフォルトの表領域内に格納されます。初期割当てでは、100MB の記憶領域を許可します。挿入されるイメージのサイズは約 20KB です。挿入のパフォーマンスを向上するために、NOCACHE および NOLOGGING オプションを 21KB の CHUNK サイズとともに指定します。

例 7-1 LOB データを格納する個別の表領域の作成

```
SVRMGR> CREATE TABLESPACE MONTANA DATAFILE 'montana.tbs' SIZE 400M;
Statement processed.
SVRMGR> CREATE TABLE images (image ORDSYS.ORDImage, imageID INTEGER)
LOB (image.source.localData) STORE AS
(
  tablespace MONTANA
  STORAGE (
    INITIAL 100M
    NEXT 100M
  )
  CHUNK 21K
  NOCACHE NOLOGGING
);
```

LOB バッファリング

クライアント上の LOB の特定の領域に少量の LOB データの読取りまたは書込みを繰り返す必要がある場合は、LOB バッファリングを使います。通常、Oracle8i オプション、Web サーバーおよびその他のクライアントベースのアプリケーションでは、クライアントのアドレス領域内の 1 つ以上の LOB の内容をバッファに格納しなければならない場合があります。LOB バッファリングを使うと、最大 512KB のバッファ・アクセスを使うことができます。LOB バッファリングの利点は次のとおりです。

- サーバーへの遅延書込み操作が可能になります。LOB のバッファ内の複数の書込み操作をサーバーにフラッシュすることによって、クライアント・アプリケーションからサーバーへのネットワーク・ラウンドトリップ回数が減少するため、LOB 更新パフォーマンス全体が向上します。

- サーバー側での LOB 更新操作の回数全体が減少すると、LOB バージョン数およびロギング量が減少し、これによって LOB パフォーマンスおよびディスク領域使用量全体が向上します。

LOB バッファリングの考慮点および使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

パラレル化

パラレル化は、複数プロセスを使用した単一 SQL 文のパラレル実行です。パラレル実行では、意思決定支援アプリケーション、またはマルチメディア・データベースのような非常に大規模なデータベース環境に関連する、データ集約型操作のパフォーマンスを劇的に向上できます。パラレル実行は大量のデータにアクセスする操作に役立ち、INSERT、UPDATE および DELETE の一括操作の処理や、LOB などのオブジェクトおよびデータ型の処理を向上できます。Oracle データベース・サーバーでは、SQL からコールされた PL/SQL 関数、SPLIT PARTITION、UPDATE、DELETE、INSERT...SELECT などの、多くの操作のためにパラレル実行を使うことができます。パラレル化が役立つ SQL 操作の完全なリストは、『Oracle8i 概要』を参照してください。例は『Oracle8i SQL リファレンス』を参照してください。

パラレル化機能は、LOB データを大量にロードする際に使うことができます。詳細は、『Oracle8i チューニング』マニュアルを参照してください。

7.2 他の大量ロード方法

LOB への FILE データのロードに使用できる他の大量ロード方法は多数あります。次の方法があります。

- SQL*Loader ユーティリティの使用
- OCILobLoadFromFile() リレーショナル関数の使用
- DBMS_LOB パッケージ内の DBMS_LOB.LOADFROMFILE() プロシージャの使用

SQL Loader ユーティリティの使用

Oracle8i では、SQL*Loader を使用してオブジェクト、コレクションおよび LOB の大量ロードを行うことができます。このリリースの SQL*Loader では、次の 2 つのオブジェクト型のロードをサポートします。

- 列オブジェクト
- 行オブジェクト

サポートしているコレクション型は次のとおりです。

- ネストした表
- VARRAY

サポートしている LOB 型は次の 4 つです。

- BLOB: 非構造化バイナリ・データを含む LOB
- CLOB: シングルバイト文字データを含む LOB
- NCLOB: 各国キャラクタ・セットの固定サイズ文字を含む LOB
- BFILE: データベース表領域外のサーバー側オペレーティング・システム・ファイル内に格納されている BLOB

SQL*Loader の詳細は、『Oracle8i ユーティリティ・ガイド』を参照してください。LOB の詳細は、『Oracle8i アプリケーション開発者ガイド 基礎編』を参照してください。

OCILobLoadFromFile() リレーショナル関数の使用

Oracle コール・インタフェース (OCI) は、C などのホスト・プログラミング言語を使用した Oracle データベース内のデータおよびスキーマ操作を可能にする、アプリケーション・プログラミング・インタフェース (API) です。

OCI リレーショナル関数 **OCILobLoadFromFile()** は、FILE の全部または一部を、指定どおり内部 LOB にロードまたはコピーします。データは、ソース FILE から宛先内部 LOB (BLOB/CLOB) にコピーされます。FILE データが CLOB/NCLOB にコピーされるときは、キャラクタ・セット変換は実行されません。また、バイナリ・データが BLOB にロードされるときにも、キャラクタ・セット変換は実行されません。したがって、FILE データはあらかじめデータベース内の LOB と同一のキャラクタ・セットでなければなりません。これを検証するためのエラー・チェックは実行されません。

詳細は、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。

DBMS_LOB パッケージ内の DBMS_LOB.LOADFROMFILE() プロシージャの使用

DBMS_LOB パッケージは、BLOB、CLOB、NCLOB、BFILE およびテンポラリ LOB を操作するためのサブプログラムを提供します。DBMS_LOB パッケージは、完全な LOB や LOB の特定部分に対するアクセスおよび操作のために使うことができます。DBMS_LOB では、BLOB、CLOB および NCLOB の読取りおよび修正や、BFILE に対する読取り専用操作を行うことができます。LOB 操作の主要な大部分が、このパッケージで提供されています。

DBMS_LOB.LOADFROMFILE() プロシージャは、ソース外部 LOB (BFILE) の全部または一部を宛先内部 LOB にコピーします。

ソース LOB および宛先 LOB のオフセット、およびソース BFILE からコピーするバイト数を指定できます。コピーするデータ量および **src_offset** は、BFILE に関連するためバイト数で指定し、宛先オフセットは BLOB の場合はバイト数、CLOB の場合は文字数で指定します。

入力 BFILE は、このプロシージャを使う前にオープンしておく必要があります。バイナリ BFILE データが CLOB にロードされるときは、キャラクタ・セット変換は暗黙的に実行され

ません。BFILE データはあらかじめデータベース内の CLOB と同一のキャラクタ・セットでなければなりません。これを検証するためのエラー・チェックは実行されません。

詳細は、『Oracle8i パッケージ・プロシージャリファレンス』を参照してください。

7.3 最大パフォーマンス実現のためのユーザー・ガイドライン

次のガイドラインは、LOB を操作するとき最大パフォーマンスを実現するのに役立ちます。

- LOB は大きいので、1 回に大量の LOB 値の読取りおよび書込みを行うことによって、最大パフォーマンスが得られます。これは次の点で役立ちます。
 - クライアント側から LOB にアクセスするときに、クライアントがサーバーと異なるノードに存在する場合は、大量の読取り / 書込み操作を行うと、ネットワーク・オーバーヘッドが減少します。
 - NOCACHE オプションを使う場合は、少量の読取り / 書込み操作を行うたびに I/O に影響を与えます。大量のデータの読取り / 書込みを行うと、I/O への影響が減少します。
 - LOB に書込みを行うと、新しいバージョンの LOB チャンクが生成されます。したがって、1 回に少量のデータを書き込むと、少量の書込み操作のたびに新しいバージョンのコストが発生します。ロギングが有効になっている場合は、このチャンクも REDO ログに格納されます。
- クライアント側で少量の LOB データの読取り / 書込みを行う必要がある場合は、LOB バッファリングを使います。OCILobEnableBuffering()、OCILobDisableBuffering()、OCILobFlushBuffer()、OCILobWrite()、OCILobRead() の説明を参照してください。基本的に、少量の LOB データの読取り / 書込みを行う前には LOB バッファリングを有効にします。
- データが LOB へまたは LOB から流れるようにするため、OCILobWrite() および OCILobRead() をコールバックとともに使います。入力値の 'amount' パラメータには、書込み操作全体の長さを必ず設定してください。可能なときは常に、LOB チャンク・サイズの倍数のサイズの読取りおよび書込みを行います。
- LOB 用のチェックアウト / チェックイン・モデルを使います。LOB は次のことができるように最適化されています。
 - LOB データを更新します。SQL UPDATE 操作によって、LOB 値全体が置換されます。
 - LOB データ全体をクライアントにコピーし、クライアント側の LOB データを修正して、LOB データ全体をデータベースに再コピーします。これは、OCILobRead() および OCILobWrite() をストリーミングとともに使うことによって実行できます。

詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

7.4 マルチメディア LOB データの検索および更新パフォーマンスの向上

LOB データがデータベースに格納された後は、LOB データの検索および更新のパフォーマンスを向上するために、7.1 項で説明した挿入の方針を修正して使う必要があります。次のガイドラインを考慮する必要があります。

- 同一の LOB データが他のユーザーによって頻繁にアクセスされる場合は、LOB に対して CACHE オプションを使います。
- CACHE オプションを使う場合は、バッファ数を増加します。
- オブジェクトを保持するのに十分な数のバッファを使います。ラージ・オブジェクト用に少数のバッファを使うのは不適當です。db_block_buffers パラメータを、オブジェクトを保持することがわかっているバッファ数の値に設定します。
- REDO ログ・ファイルが十分に大きいことを確認します。REDO ログ・ファイルが小さいと、特に LOB データに多数の更新を行っている場合には、ログ・スイッチ待機に長時間かかる可能性があります。
- 特にデータベース内のデータの大部分が LOB データである場合には、より大きいページ・サイズ (db_block_size) を使用してください。

7.5 オープン BFILE の最大数の設定

1 セッションに同時にオープンできる BFILE の数には制限があります。デフォルト値は 10 ファイルです。より多くの BFILE を同時にオープンする必要がある場合は、init.ora ファイル内の初期化パラメータ SESSION_MAX_OPEN_FILES の設定値を大きくします。

7.6 表パーティション内での LOB の使用

LOB 列を含む表はパーティション化できるため、LOB セグメントを複数の表領域間に分散して、次の処理を行うことができます。

- I/O ロードの均衡を調整します。
- バックアップおよび回復操作を管理しやすくします。
- LOB のメンテナンスを容易にします。

LOB データをパーティション化すると、I/O の問題を改善し、LOB データを含む表領域のデータ・ファイル間での I/O ロードの均衡をよりよく保持することができます。ストライプ化という方法で複数のデバイスにデータ記憶域を割り当て、パフォーマンスをさらに向上できます。これによって、ディスクの競合なしに、複数プロセスから表の異なる部分に同時にアクセスできるようになります。

LOB データをパーティション化すると、データベースのバックアップおよび回復操作を調整し、これらの操作の実行に必要なリソースをより効率的に使うことができます。たとえば、

パーティション化された表領域が 2 つ以上あると、必要に応じて特定のデータ・ファイルに対してデータベースの部分バックアップおよび部分回復操作を実行できます。

同様に、LOB が含まれた表領域をパーティション化すると、LOB データを日付ごと、サブジェクトごと、カテゴリごとなどのより小さいパーティションに論理的にグループ化することによって、LOB データを簡単にメンテナンスできます。これにより、アプリケーションに基づいて、パーティションの追加、マージ、分割または削除を必要に応じてより簡単に行えるようになります。

これらの各トピックの例および詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。例は『Oracle8i SQL リファレンス』を参照してください。具体的には CREATE TABLE 文および LOB 列を含むパーティション表の例を参照してください。

オーディオ・ファイル形式および圧縮形式

A.1 サポートしているオーディオ・ファイル形式および圧縮形式

次の各表に、*interMedia audio* でサポートしているファイル形式、圧縮形式およびその他のオーディオ機能を示します。

これらの表を使うには、希望のデータ形式を参照して、サポートしている形式を確認します。たとえば、表 A-1 では、*interMedia audio* は AIFF 形式をシングル・チャンネル、ステレオ、8 ビットおよび 16 ビット・サンプル、線形 PCM コード化、非圧縮形式でサポートしていることを示しています。

表 A-1 AIFF データ形式

形式	オーディオ機能
AIFF フォーマット ID 'AIFF' ファイル形式: 'AIFF' ファイル拡張子: .aff	シングル・チャンネル
	ステレオ
	8 ビット・サンプル
	16 ビット・サンプル
	線形 PCM コード化
	圧縮形式
	非圧縮

サポートしているオーディオ・ファイル形式および圧縮形式

表 A-2 AIFF-C データ形式

形式	オーディオ機能
AIFF-C フォーマット ID 'AIFC' ファイル形式: 'AIFC' ファイル拡張子: .afc これらの圧縮形式のいずれかを選択します。	シングル・チャンネル ステレオ 8 ビット・サンプル 16 ビット・サンプル
	圧縮形式
	非圧縮 ADPCM (2:1) 圧縮

表 A-3 AU データ形式

形式	オーディオ機能
AU フォーマット ID 'AUFF' ファイル形式: 'AUFF' ファイル拡張子: .au	シングル・チャンネル ステレオ 8 ビット・サンプル 16 ビット・サンプル Mu-law コード化 線形 PCM コード化
	圧縮形式
	非圧縮

表 A-4 WAV データ形式

形式	オーディオ機能
WAV フォーマット ID 'WAVE' ファイル形式: 'WAVE' ファイル拡張子: .wav	シングル・チャンネル ステレオ
	8 ビット・サンプル 16 ビット・サンプル
	線形 PCM コード化
これらの圧縮形式のいずれかを選択します。	圧縮形式
	非圧縮 Microsoft ADPCM

サポートしているオーディオ・ファイル形式および圧縮形式

B

イメージ・ファイル形式および圧縮形式

B.1 サポートしているイメージ・ファイル形式および圧縮形式

次の表に、Oracle8i *interMedia* でサポートしているイメージ・ファイル形式および圧縮形式を示します。

これらの表を使うには、希望のデータ形式を参照して、サポートしている形式を確認します。たとえば、表 B-1 では、*interMedia image* はモノクロの BMP 形式を読み取り / 書込みアクセスで、32 ビット RGB の BMP 形式を読み取りアクセスでサポートしていることを示しています。

表 B-1 BMP データ形式

形式	ピクセル形式	サポート
BMP ファイル形式: 'BMPF' ファイル拡張子: .bmp MIME: image/bmp	モノクロ	読取り / 書込み
	4 ビット LUT	読取り
	8 ビット LUT	読取り / 書込み
	16 ビット RGB	読取り
	24 ビット RGB	読取り / 書込み
	32 ビット RGB	読取り
	圧縮形式	サポート
これらの圧縮形式のいずれかを選択します。	非圧縮	読取り / 書込み
	BMPRL (8 ビット LUT の場合)	読取り / 書込み
	データ記述	サポート
これらのコンテンツ形式のいずれか（複数可）を選択：	逆 DIB	読取り
	OS/2 形式	読取り

サポートしているイメージ・ファイル形式および圧縮形式

表 B-2 CALS ラスター・データ形式

形式	ピクセル形式	サポート
CALS ラスター ファイル形式: 'CALS' ファイル拡張子: .cal MIME: image/x-ora-cals	モノクロ	読取り / 書込み
	圧縮形式	サポート
	FAX4 (CCITT G4)	読取り / 書込み
	データ記述	サポート
	なし	なし

表 B-3 GIF データ形式

形式	ピクセル形式	サポート
GIF ファイル形式: 'GIF' ファイル拡張子: .gif MIME: image/gif 注意: <i>interMedia image</i> での動画 GIF イメージのサポートには制限があります。 <code>setProperties()</code> はサポートしていますが、 <code>process()</code> メソッドおよび <code>processCopy()</code> (または <code>Analyze</code>) メソッドを使用した処理はサポートしていません。	モノクロ 8 ビット LUT	読取り 読取り / 書込み
	圧縮形式	サポート
	GIFLZW (LZW)	読取り / 書込み
	データ記述	サポート
	なし	なし

表 B-4 JFIF データ形式

形式	ピクセル形式	サポート
JFIF ファイル形式: 'JFIF' ファイル拡張子: .jpg MIME: image/jpeg	8 ビット・グレースケール	読取り / 書込み
	24 ビット RGB	読取り / 書込み
	圧縮	サポート
	JPEG	読取り / 書込み
	データ記述	サポート
	なし	なし

表 B-5 PCX データ形式

形式	ピクセル形式	サポート
PCX v 5 ファイル形式: 'PCXF' ファイル拡張子: .pcx MIME: image/x-ora-pcxf	モノクロ	読取り
	2 ビット LUT	読取り
	4 ビット LUT	読取り
	8 ビット LUT	読取り
	1 ビット RGB	読取り
	2 ビット RGB	読取り
	4 ビット RGB	読取り
	8 ビット RGB	読取り
	24 ビット RGB	読取り
	圧縮形式	サポート
	RLE	読取り
	データ記述	サポート
	なし	なし

サポートしているイメージ・ファイル形式および圧縮形式

表 B-6 PICT データ形式

形式	ピクセル形式	サポート
PICT v.1 および 2 ファイル形式: 'PICT' ファイル拡張子: .pct MIME: image/pict	モノクロ	読取り / 書込み
	2 ビット LUT	読取り
	4 ビット LUT	読取り
	8 ビット LUT	読取り / 書込み
	16 ビット RGB	読取り
	24 ビット RGB	読取り / 書込み
	圧縮形式	サポート
これらの圧縮形式のいずれかを選択:	パックビット	読取り / 書込み
	JPEG (8 ビット・グレースケールおよび RGB)	読取り / 書込み
	データ記述	サポート
これらのコンテンツ形式のいずれか (複数可) を選択:	ベクトル / オブジェクト・グラフィックス	未サポート

表 B-7 ロー・ピクセル・データ形式

形式	ピクセル形式	サポート
ロー・ピクセル ファイル形式: 'RPIX' ファイル拡張子: .rpx MIME: image/x-ora-rpix	モノクロ	読取り / 書込み
	8 ビット・グレースケール	読取り / 書込み
	24 ビット RGB	読取り / 書込み
	圧縮形式	サポート
これらの圧縮形式のいずれかを選択します。	非圧縮	8 ビット・グレースケールおよび RGB の読取り / 書込み
	FAX3 (CCITT G3)	モノクロのみの読取り / 書込み
	FAX4 (CCITT G4)	モノクロのみの読取り / 書込み

表 B-7 ロー・ピクセル・データ形式 (続き)

	データ記述	サポート
	INVERSE 走査線オーダー	読取り / 書込み
	REVERSE ピクセル・オーダー	読取り / 書込み
	BIP、BIL または BSQ インタリーブ	読取り / 書込み
	代替バンド・オーダー	読取り / 書込み
	>3 バンド	読取り

表 B-8 Sun ラスター・データ形式

形式	ピクセル形式	サポート
Sun ラスター	モノクロ	読取り / 書込み
	8 ビット・グレースケール	読取り / 書込み
ファイル形式: 'RASf'	8 ビット LUT	読取り / 書込み
ファイル拡張子: .ras	24 ビット RGB	読取り / 書込み
MIME: image/x-ora-rasf		
	圧縮形式	サポート
これらの圧縮形式のいずれかを選択します。	非圧縮	読取り / 書込み
	SUNRLE (RLE)	読取り / 書込み
	データ記述	サポート
	なし	なし

表 B-9 Targa データ形式

形式	ピクセル形式	サポート
Targa	8 ビット・グレースケール	読取り / 書込み
	8 ビット LUT	読取り / 書込み
ファイル形式: 'TGAF'	16 ビット RGB	読取り
ファイル拡張子: .tga	24 ビット RGB	読取り / 書込み
MIME: image/x-ora-tgaf	32 ビット RGB	読取り
	圧縮形式	サポート

サポートしているイメージ・ファイル形式および圧縮形式

表 B-9 Targa データ形式 (続き)

これらの圧縮形式のいずれかを選択します。	非圧縮	読取り / 書込み
	TARGARLE (RLE)	読取り / 書込み
	データ記述	サポート
	なし	なし

表 B-10 TIFF データ形式

形式	ピクセル形式	サポート
TIFF v.4/5/6 ファイル形式: 'TIFF' ファイル拡張子: .tif MIME: image/tiff	モノクロ	読取り / 書込み
	8 ビット・グレースケール	読取り / 書込み
	4 ビット LUT	読取り
	8 ビット LUT	読取り / 書込み
	24 ビット RGB	読取り / 書込み
	圧縮形式	サポート
これらの圧縮形式のいずれかを選択します。	非圧縮	読取り / 書込み
	パックビット	読取り / 書込み
	ハフマン	読取り / 書込み
	FAX3 (CCITT G3)	読取り / 書込み
	FAX4 (CCITT G4)	読取り / 書込み
	LZW	読取り / 書込み
	LZWHDIFF	読取り / 書込み
	JPEG (8 ビット・グレースケールおよび RGB)	読取り / 書込み
	データ記述	サポート
これらのコンテンツ形式のいずれか (複数も可) を選択します。	プレーナ・データ	未サポート
	タイル・データ	未サポート
	照度差解釈	読取り / 書込み
	MSB/LSB	読取り / 書込み

C

Process() および ProcessCopy() の イメージ演算子

この付録では、`process()` メソッドおよび `processCopy()` メソッドで使うコマンド・オプション、つまり演算子について説明します。

使用可能な演算子は大きく 3 つのカテゴリに分類されます。それぞれを次の項で説明します。

- C.2 項「イメージ・フォーマット演算子」
- C.3 項「イメージ処理演算子」
- C.4 項「形式固有の演算子」

C.1 項の「共通概念」では、これらの演算子の相対順序を説明します。

C.1 共通概念

この項では、すべてのイメージ演算子と `process()` メソッドおよび `processCopy()` メソッドに共通する概念について説明します。

C.1.1 ソース・イメージおよび宛先イメージ

`process()` メソッドおよび `processCopy()` メソッドは、ソース・イメージと呼ばれるあるイメージを操作して、宛先イメージと呼ばれる別のイメージを作成します。`process()` メソッドの場合、宛先イメージはソース・イメージと同一の記憶領域に書き込まれ、永続的にソース・イメージと置き換えられます。`processCopy()` メソッドの場合、宛先イメージとソース・イメージの記憶領域は異なります。

C.1.2 `process()` および `processCopy()`

`process()` メソッドおよび `processCopy()` メソッドは、機能上は同一です。ただし、`process()` が入力元と同一の BLOB に出力を書き込むのに対し、`processCopy()` は出力を別の BLOB

に書き込む点が異なります。`process()` と `processCopy()` のコマンド文字列オプションは同一で、違いはありません。

この付録では、`process()` という名前と `processCopy()` という名前を同義で使います。`process()` という名前を使用しているときには、とくに明示的な注意がないかぎり、`process()` と `processCopy()` の両方を指しています。

C.1.3 演算子および値

`process()` の演算子はすべて、`<operator> = <value>` という形式でコマンド文字列内で使われます。演算子は、単にコマンド文字列内に存在するだけでは有効になりません。この式の右側は演算子の値と呼ばれ、演算子の適用方法を決定します。

C.1.4 演算子の組合せ

一般に、演算子の組合せに意味がある場合は、任意の数の演算子を組み合わせて、`process()` メソッドに渡されるコマンド文字列内で使うことができます。ただし、一部の演算子は、他の演算子が存在する場合または他の条件が満たされている場合にのみサポートされます。たとえば、`compressionQuality` 演算子は、宛先イメージの圧縮形式が `JPEG` の場合にのみサポートされます。その他の演算子では、ソース・イメージまたは宛先イメージがロー・ピクセルまたは外部イメージである必要があります。

演算子を柔軟に組み合わせることができるため、単一の操作でイメージ・フォーマットの変更、カラー数の増減、データの圧縮、および結果イメージのカットやスケール変更を行うことができます。これらの各操作を順次行うには、複数のコールを作成することを強くお勧めします。

C.2 イメージ・フォーマット演算子

最も抽象的なレベルでは、イメージ・フォーマット演算子はイメージ記憶域内でデータのレイアウトを変更するのに使います。イメージの意味内容を変更しないため、宛先イメージに格納可能なデータ量を超える情報がソース・イメージに含まれていないかぎり、イメージの視覚的外観はまったく変更されません。宛先イメージに格納可能な情報量を超えるソース・イメージの例は次のとおりです。

- 24 ビット・イメージを 8 ビット・イメージに変換する場合（1 ピクセルに対するビット数が多すぎる場合）
- カラー・イメージをグレースケールまたはモノクロ・イメージに変換する場合（カラー・プレーンが多すぎる場合）
- 非圧縮イメージまたは可逆圧縮形式で格納されたイメージを非可逆圧縮形式に変換する場合（詳細が多すぎる場合）

C.2.1 FileFormat

FileFormat 演算子は、出力ファイルのイメージ・ファイル・タイプ、つまり形式を決定します。この演算子の値は 4 文字コードで、新規ファイル形式名のニーモニックです。

FileFormat 演算子の使用可能な値のリストを表 4-1 に示します。付録 B には、各ファイル形式のニーモニック（ファイル形式）、代表的なファイル拡張子、使用可能な圧縮形式およびコンテンツ形式、その他の主な機能などの基本情報を示します。

FileFormat 演算子に指定する値は、**process()** で出力を指定するときに唯一最も重要な点です。この値は、圧縮品質が有用かどうか、および形式固有の演算子が有用かどうかにかかわらず、可能なコンテンツ形式および圧縮形式の範囲を決定します。

FileFormat 演算子が **process()** コマンド文字列で使われない場合、**interMedia image** はソース・イメージのファイル形式を判別し、その形式をデフォルトのファイル形式値として使います。ソース・イメージのファイル形式が出力をサポートしていない場合には、エラーが発生します。ソース・イメージが外部イメージの場合は、出力イメージはロー・ピクセルとして書き込まれます。

C.2.2 ContentFormat

ContentFormat 演算子は、イメージ・コンテンツの形式を決定します。イメージ・コンテンツとは、イメージがサポートしているカラー数およびそのサポート方法です。出力イメージの格納に使われるファイル形式にもよりますが、コンテンツ形式の一部またはほとんどがサポートされない場合があります。

サポートしている **ContentFormat** 演算子の値は、大きく 3 つのクラスに分類され、さらに特別な値が 3 つあります。これらの値の実際のニーモニックを、表 4-1 に示しています。

名前に **grayscale** または **greyscale** という単語が含まれているコンテンツ形式では、グレーの階調のみをサポートします。このコンテンツ形式間の違いは、使用可能な階調の数です。「4 ビット」形式は 16 階調をサポートし、「8 ビット」形式は 256 階調をサポートします。

grayscale と **greyscale** は同じ意味です。

名前に **LUT** という単語が含まれているコンテンツ形式では、カラー参照表を使用してさまざまなカラーをサポートします。「1 ビット LUT」形式では、2 つのカラーをサポートします。「2 ビット LUT」形式では、4 つのカラーをサポートします。「4 ビット LUT」形式では、16 の一意なカラーをサポートします。「8 ビット LUT」形式では、256 のカラーをサポートします。

名前に **RGB** という単語が含まれているコンテンツ形式では、カラーの値を赤、緑、青の 3 つの組み合わせでピクセル・データに直接格納します。**RGB** データ全体のビット数は形式名に指定されており、個々の形式では、ビットが赤、緑および青に個々の方法で割り当てられます。ただし、データのビット数が大きいほど、異なる階調間の違いを微細に表現することができます。一部のイメージ・フォーマットでは、一部のビットが使用されません。

モノクロのコンテンツ形式では、黒および白のみ格納でき、グレーの階層は格納されません。「ロー」および「24 ビット・プレーナ」形式は、現在サポートされていません。

ContentFormat 演算子が **process()** メソッドに渡されない場合は、ソース・イメージのコンテンツ形式が宛先イメージのファイル形式によってサポートされていれば、**interMedia image** はソース・イメージのコンテンツ形式の複製を試行します。ソース・イメージのコンテンツ形式がサポートされていなければ、宛先ファイル形式に応じて、デフォルトのコンテンツ形式が選択されます。

C.2.3 CompressionFormat

CompressionFormat 演算子は、イメージ・データの圧縮に使われる圧縮アルゴリズムを決定します。サポートされる圧縮形式の範囲は、出力イメージのファイル形式に大きく左右されます。一部のファイル形式は単一の圧縮形式しかサポートしていません。また、一部の圧縮形式は単一のファイル形式でしかサポートしていません。

サポートしている **CompressionFormat** 演算子の値を、表 4-1 に示しています。

ニーモニックに **RLE** という単語が含まれる圧縮形式はすべて、ランレングス・コード化圧縮方式で、同一カラーの領域が大きいイメージにのみ効果的です。**PACKBITS** 圧縮タイプはランレングス・コード化方式です。この方式は **Macintosh** システムから生まれましたが、他のシステムでサポートされています。この方式には、他のランレングス・コード化圧縮方式に類似した制限事項があります。**LZW** または **HUFFMAN** という単語が含まれる形式は、イメージの冗長情報を検査するより複雑な圧縮方式で、より広範囲のイメージに役立ちます。**FAX3** および **FAX4** は、ファクシミリ・データを圧縮するための **CCITT Group 3** および **Group 4** 規格で、モノクロ・イメージにのみ役立ちます。このパラグラフで説明した圧縮形式はすべて可逆圧縮方式です。つまり、イメージの圧縮によってデータが廃棄されることはありません。可逆形式に圧縮された後で解凍されたイメージの外観は、元のイメージと同一です。

JPEG 圧縮形式は特殊なケースです。写真イメージの圧縮用に開発された **JPEG** 形式は非可逆形式です。つまり、通常は、重要でない詳細を廃棄することによってイメージを圧縮します。この形式は写真イメージやそれと同様に細かいイメージの圧縮に最適であるため、多くの場合、線画イメージや類似カラーの領域が大きいイメージなどのイメージ・タイプではよい結果が出ません。**JPEG** は、現在 **interMedia image** でサポートしている唯一の非可逆圧縮方式です。

CompressionFormat 演算子が使われない場合は、ソース・イメージのファイル形式が宛先イメージのファイル形式と同一であれば、**interMedia image** はソース・イメージの圧縮形式の使用を試行します。宛先イメージのコンテンツ形式が処理中に変更され、新しいコンテンツ形式がソース・イメージの圧縮形式をサポートしていない場合は、デフォルトの圧縮形式が選択されます。デフォルトは多くの場合 **"None"** または **"No Compression"** です。

CompressionFormat 演算子が使われず、宛先イメージのファイル形式がソース・イメージのファイル形式と異なる場合は、宛先イメージのファイル形式に応じて、デフォルトの圧縮形式が選択されます。このデフォルトの圧縮形式は多くの場合 **"None"** または **"No Compression"** です。

C.2.4 CompressionQuality

CompressionQuality 演算子は、非可逆圧縮形式で圧縮されるイメージの相対的な品質を決定します。この演算子は、可逆圧縮形式にとっては意味がないので、現在は JPEG 以外の圧縮形式ではサポートされていません。

CompressionQuality 演算子は、最大圧縮イメージ（最低の視覚的品質）から最小圧縮イメージ（最高の視覚的品質）まで、**MAXCOMPRATIO**、**HIGHCOMP**、**MEDCOMP**、**LOWCOMP** および **MAXINTEGRITY** の 5 つの値を受け入れます。**MAXCOMPRATIO** 値を使うと、イメージを最小の領域に格納できますが、イメージの視覚的逸脱が生じる可能性があります。**MAXINTEGRITY** 値を使うと、結果イメージを元のイメージにより忠実に保持できますが、格納に必要な領域は大きくなります。

CompressionQuality 演算子が使われず、宛先イメージの圧縮形式で圧縮品質管理をサポートしている場合は、デフォルトで **MEDCOMP** が使われ、標準の品質になります。

C.3 イメージ処理演算子

interMedia image でサポートしているイメージ処理演算子は、イメージのディスプレイ上の外観を直接変更します。*interMedia image* でサポートしているイメージ処理演算子は、すべての可能なイメージ処理操作の一部のみであり、複雑なイメージ解析を実行するユーザー向けではありません。

C.3.1 Cut

Cut 演算子は、元のイメージのサブセットの作成に使います。**Cut** 演算子に指定する値は、ソース・イメージ内のカット・ウィンドウの原点座標 (x,y)、およびカット・ウィンドウのピクセル単位の幅と高さです。この演算子は、要求されたスケール変更の前に適用されます。

Cut 演算子を使わない場合は、ソース・イメージ全体が使われます。

C.3.2 Scale

Scale 演算子は、演算子の値として指定された比率で、イメージを拡大または縮小します。値が 1.0 より大きい場合は、宛先イメージのスケールは大きくなります。（拡大されます。）値が 1.0 より小さい場合は、出力のスケールは小さくなります。（縮小されます。）スケールの値が 1.0 の場合はスケールは変更されず、エラーは発生しません。**Scale** 演算子が **process()** メソッドに渡されない場合は、ソース・イメージにスケール変更が適用されません。

interMedia image で使われるスケール変更手法は 2 つあります。1 つめの手法は「サンプリングによるスケール変更」で、要求された圧縮品質が **MAXCOMPRATIO** または **HIGHCOMP** の場合か、イメージが両ディメンションに拡大される場合にのみ使われます。このスケール変更手法は、スケール変更アルゴリズムによって計算されるピクセルに最も近いソース・イメージ・ピクセルを選択して、そのピクセルのカラーを使うことによって実現されます。この手法は高速ですが、結果イメージの品質は低くなります。

2 つめのスケール変更手法は「平均値計算によるスケール変更」で、1 つめの手法のケース以外のすべてのケースで使われます。この手法は、スケール変更アルゴリズムによって計算されるピクセルに近いいくつかのピクセルを選択して、その平均カラーを計算することによって実現されます。この手法は低速ですが、結果イメージの品質は高くなります。

Scale 演算子が使われない場合の、デフォルトのスケール変更値は **1.0** です。この演算子は、他のスケール変更演算子と組み合わせることはできません。

C.3.3 XScale

XScale 演算子は、**Scale** 演算子に類似していますが、イメージの幅（**x** ディメンション）にのみ影響します。**Xscale** と **Scale** の重要な違いは、**Xscale** では、サンプリングによるスケール変更が、イメージ品質が **MAXCOMPRATIO** または **HIGHCOMP** に指定されたときに必ず使われ、イメージが拡大されるか縮小されるかは関係ないことです。

この演算子は **Yscale** 演算子と組み合わせて使い、各軸を個別にスケール変更することができます。その他のスケール変更演算子（**Scale**、**FixedScale**、**MaxScale**）と組み合わせることはできません。

C.3.4 YScale

YScale 演算子は、**Scale** 演算子に類似していますが、イメージの高さ（**y** ディメンション）にのみ影響します。**Yscale** と **Scale** の重要な違いは、**YScale** では、サンプリングによるスケール変更が、イメージ品質が **MAXCOMPRATIO** または **HIGHCOMP** に指定されたときに必ず使われ、イメージが拡大されるか縮小されるかは関係ないことです。

この演算子は **XScale** 演算子と組み合わせて使い、各軸を個別にスケール変更することができます。その他のスケール変更演算子（**Scale**、**FixedScale**、**MaxScale**）と組み合わせることはできません。

C.3.5 FixedScale

FixedScale 演算子は、スケール変更値を指定するための代替方法を提供します。**Scale**、**Xscale** および **Yscale** 演算子はすべて、浮動小数点のスケール比率を受け入れるのに対し、**FixedScale**（および **MaxScale**）演算子では、スケール変更値をピクセル単位で指定します。この演算子は、サムネール・イメージなど、特定のサイズのイメージを簡単に作成するためのものです。

FixedScale 演算子に指定する 2 つの整数値は、宛先イメージの希望のディメンション（幅および高さ）です。指定するディメンションは、ソース・イメージのディメンションより大きくても小さくても（あるいは一方が大きくもう一方が小さくても）構いません。

この演算子によって使われるスケール変更方法は、すべてのケースで **Scale** 演算子によって使われる方法と同じです。この演算子は、他のスケール変更演算子と組み合わせることはできません。

C.3.6 MaxScale

MaxScale 演算子は、**FixedScale** 演算子の別形で、ソース・イメージの縦横比率（幅と高さの相対）を保持します。**MaxScale** も 2 つの整数ディメンション値を受け入れますが、これらの値はスケール変更後の該当するディメンションの最大値を示します。最終的なディメンションは、実際には指定された値より小さくなる場合があります。

FixedScale 演算子と同様に、この演算子も特定のサイズのイメージを簡単に作成するためのものです。**MaxScale** を使用して作成されたサムネール・イメージは縦横比率が正確であるため、**MaxScale** は、**FixedScale** 演算子よりもさらにサムネール・イメージの作成に適しています。

MaxScale 演算子は、ソース・イメージの縦横比率を保持するとともに、指定されたディメンションの範囲内に収まるようにソース・イメージのスケール変更を行います。縦横比率が保持されるため、実際には宛先イメージの一方のディメンションしか、演算子に指定された値と等しくならない可能性があります。もう一方のディメンションは、指定された値以下になる可能性があります。このスケール変更方法についての別の考え方は、**MaxScale** 演算子で指定された縦横のディメンション内に宛先イメージ全体を収めるという制約のもとで、できるだけ大きい単一のスケール変更係数でソース・イメージのスケール変更が行われるというものです。

Cut 演算子を **MaxScale** 演算子と組み合わせて使う場合は、入力イメージの縦横比率のかわりに、カット・ウィンドウの縦横比率が保持されます。

この演算子によって使われるスケール変更方法は、すべてのケースで **Scale** 演算子によって使われる方法と同じです。この演算子は、他のスケール変更演算子と組み合わせることはできません。

C.4 形式固有の演算子

次の演算子は、宛先イメージのファイル形式がロー・ピクセルの場合にのみサポートされません。ただし、**InputChannels** 演算子は例外で、ソース・イメージがロー・ピクセルまたは外部イメージの場合にのみサポートされます。ソース・イメージ・フォーマットがロー・ピクセルまたは外部イメージなので、**FileFormat** 演算子を使用して宛先イメージ・フォーマットが明示的にロー・ピクセルに設定されても、または *interMedia image* によって自動的にロー・ピクセル・フォーマットが選択されても、問題ではありません。

C.4.1 ChannelOrder

ChannelOrder 演算子は、宛先ロー・ピクセル・イメージ内の赤、緑および青のチャンネル（バンド）の相対順序を決定します。この演算子に渡される二重引用符内の文字 **R**、**G** および **B** の順序によって、出力内のこれらのチャンネルの順序が決定されます。ロー・ピクセル・イメージのヘッダーには、この順序が失われないように書き込まれます。

ロー・ピクセル・フォーマットおよびこの形式でのチャンネルの順序付けの詳細は、付録 D を参照してください。

C.4.2 Interleave

Interleave 演算子は、宛先ロー・ピクセル・イメージ内の赤、緑および青のチャンネル（バンド）のレイアウトを制御します。この演算子では、**BIP**、**BIL** および **BSQ** の 3 つのニーモニック値がサポートされており、出力イメージは、これらの値によってそれぞれ「ピクセル単位のバンドのインタリーブ」、「ライン単位のバンドのインタリーブ」、「順次バンド」のように操作されます。

ロー・ピクセル・フォーマット、この形式でのチャンネルのインタリーブ、またはこれらのインタリーブ値の意味の詳細は、付録 D を参照してください。

C.4.3 PixelOrder

PixelOrder 演算子は、ロー・ピクセル・イメージの走査線内のピクセルの方向を制御します。値 **NORMAL** は、走査線の左端のピクセルが、イメージ・データ・ストリームの先頭に現れることを示します。値 **REVERSE** を指定すると、走査線の右端のピクセルが先頭に現れます。

ロー・ピクセル・フォーマットおよびピクセルの順序付けの詳細は、付録 D を参照してください。

C.4.4 ScanlineOrder

ScanlineOrder 演算子は、ロー・ピクセル・イメージ内の走査線の順序を制御します。値 **NORMAL** は、上端のディスプレイ走査線が、イメージ・データ・ストリームの先頭に現れることを示します。値 **INVERSE** を指定すると、下端の走査線が先頭に現れます。

ロー・ピクセル・フォーマットおよび走査線の順序付けの詳細は、付録 D を参照してください。

C.4.5 InputChannels

C.4 項で説明したように、**InputChannels** 演算子は、ソース・イメージがロー・ピクセル・フォーマットまたは外部イメージの場合にのみサポートされます。

InputChannels 演算子は、後のイメージ処理用に、マルチバンド・イメージの個々のバンドを赤、緑および青のチャンネルに割り当てます。ソース・イメージ内の任意のバンドを任意のチャンネルに割り当てることができます。必要ならば、単一バンドのみを指定することもできます。その場合、選択したバンドはグレー・チャンネルとして使われ、結果の出力イメージはグレースケールになります。イメージ内の 1 つめのバンドは数値 1 です。**Input Channels** 演算子に渡されるバンド数は、1 以上で、ソース・イメージの合計バンド数以下にする必要があります。**InputChannels** 演算子によって選択されたバンドのみが出力に書き込まれます。その他のバンドは、出力イメージがロー・ピクセル・フォーマットの場合でも転送されません。

ロー・ピクセル・イメージや外部イメージにはすべて、これらの入力チャネル割当てがヘッダー・ブロックに書き込まれていますが、そのデフォルトの割当てではこの演算子によって上書きされることに注意してください。

ロー・ピクセル・フォーマットおよび入力チャネルの詳細は、付録 D を参照してください。

ロー・ピクセル・イメージ・フォーマット

この付録では、Oracle ロー・ピクセル・イメージ・フォーマットについて説明します。この付録は、サポートされていないイメージ・フォーマットを *interMedia image* にインポートするために、またはイメージ内のピクセル・データに直接アクセスする手段として、ロー・ピクセル・フォーマットを使いたい開発者および上級ユーザー向けです。

この付録の大部分は、外部イメージにも適用できます。

D.1 ロー・ピクセルの概要

interMedia image では、アートワーク、写真などのイメージを効率的に圧縮して格納するのに適した、一般的なイメージ・フォーマットを多数サポートしており、これらの形式間での変換機能を提供しています。ただし、これらの形式の大部分は少なくともある程度の独自性があり、また多くの場合そのコンテンツの形式は大きく変更されやすいので、イメージのピクセル・データに簡単にアクセスするのに適していません。

ロー・ピクセル・フォーマットは、圧縮されたデータ・ストリーム内のピクセルの位置を判別するのに必要な複雑な計算を行うことなく、ピクセル・データに直接アクセスする必要があるアプリケーションに役立ちます。この形式では、フィルタ処理やエッジ検出などのピクセル指向のイメージ処理を実行しているアプリケーションでのイメージ読取りが自動化されます。この形式は、イメージにデータを書き戻す必要があるアプリケーションにとって、さらに役立ちます。圧縮されたイメージ内の単一のピクセルを変更するだけでも、イメージ・ストリーム全体に影響を与える可能性があります。非圧縮形式を使用すると、アプリケーションでピクセル・データを直接書き込み、その後単一の `process()` コマンドでイメージを圧縮することができます。

また、この形式は、*interMedia image* で直接サポートしていない、単純な非圧縮形式のデータを使うユーザーにとって役立ちます。このようなユーザーは、データにロー・ピクセル識別子およびヘッダーを付加して、*interMedia image* にインポートできます。これらのイメージの読取りのみが必要なユーザー（インポートや変換用）のために、この機能は「外部イメージ・サポート」として *interMedia image* に組み込まれています。この機能とロー・ピクセル・フォーマットとの関連は、D.10 項で説明します。

ロー・ピクセル・フォーマットでは、現在 *interMedia image* に組み込まれていないイメージ・タイプのサポートの他に、衛星イメージなどの N バンド・イメージの解釈も行うことができます。ロー・ピクセルを使うと、N バンド・イメージの 1 つまたは 3 つのバンドが、別のイメージ・フォーマットへの変換中に選択され、他の方法で N バンド・イメージをサポートしていないプログラム内で簡単に視覚化することができます。ロー・ピクセル・フォーマットで作成されたイメージのバンドは 1 つまたは 3 つのみであることに注意してください。

ロー・ピクセル・フォーマットの現行バージョンは 1.0 です。この付録はこのバージョンのロー・ピクセル・イメージにのみ適用できます。形式の詳細は他のバージョンでは変更される可能性があります。

D.2 ロー・ピクセル・イメージの構造

ロー・ピクセル・イメージは、4 バイトのイメージ識別子、その後に続く 30 バイトのイメージ・ヘッダー、その後に続く 0 (ゼロ) バイト以上の任意のギャップ、およびその後に続くピクセル・データから構成されます。

ロー・ピクセル・イメージはカラー・マップ型ではないため、カラー参照表が含まれないことに注意してください。

ロー・ピクセルのヘッダーは「イメージ識別子」および「イメージ・ヘッダー」から構成されます。イメージ・ヘッダーは、実際には複数のフィールドから構成されています。

イメージ内の実際の先頭バイトはオフセット '0 (ゼロ)' であることに注意してください。整数フィールドはすべて符号なし整数で、ビッグ・エンディアン・バイト順序で格納されています。

名前	バイト	説明
イメージ識別子	0:3	ASCII 値 RPIX を含む 4 バイト文字配列 この配列では、イメージをロー・ピクセル・イメージとして識別します。
イメージ・ヘッダー長	4:7	識別子フィールドを除く、このヘッダーのバイト長 このフィールドの値は、ヘッダー・フィールドとイメージ内のピクセル・データ間のギャップを作成するために大きくすることができます。
メジャー・バージョン	8	イメージで使われているロー・ピクセル・フォーマットのメジャー・バージョン番号
マイナー・バージョン	9	イメージで使われているロー・ピクセル・フォーマットのマイナー・バージョン番号
イメージ幅	10:13	イメージのピクセル単位の幅
イメージ高さ	14:17	イメージのピクセル単位の高さ

圧縮タイプ	18	イメージの圧縮タイプ。None、CCITT FAX Group 3 または CCITT FAX Group 4 があります。
ピクセル・オーダー	19	イメージのピクセル・オーダー。Normal または Reverse があります。
走査線オーダー	20	イメージの走査線オーダー。Normal または Inverse があります。
インターリーブ	21	イメージのインターリーブ・タイプ。BIP、BIL または BSQ があります。
バンド数	22	イメージ内のバンド数。1 ～ 255 の範囲で指定する必要があります。
赤チャンネル番号	23	赤のデフォルトとして使うチャンネルのバンド番号 イメージがグレースケールの場合、このフィールドはグレーのチャンネル番号になります。
緑チャンネル番号	24	緑のデフォルトとして使うチャンネルのバンド番号 イメージがグレースケールの場合、このフィールドは 0 (ゼロ) になります。
青チャンネル番号	25	青のデフォルトとして使うチャンネルのバンド番号 イメージがグレースケールの場合、このフィールドは 0 (ゼロ) になります。
予約領域	26:33	現在は使われていません。すべてのバイトを 0 (ゼロ) にする必要があります。

D.3 ロー・ピクセル・ヘッダー・フィールドの説明

この項では、ロー・ピクセル・ヘッダーの各フィールドの詳細を説明します。

イメージ識別子

ロー・ピクセル・イメージの先頭の 4 バイトで、この識別子文字列を常に ASCII 値 "RPIX" (16 進 52 50 49 58) に設定する必要があります。これらの文字によって、イメージが RPIX 形式にコード化されているものとして識別されます。

この文字列は現在、ロー・ピクセルのバージョンに依存していません。

イメージ・ヘッダー長

ロー・ピクセル・リーダーでは、このフィールドに格納された値を使用して、ロー・ピクセル・イメージ内のピクセル・データ・セクションの開始点を検索します。イメージ内のピクセル・データのオフセットを検索するには、ロー・ピクセル・リーダーでイメージ識別子の長さ（常に '4'）をイメージ・ヘッダー長フィールドの値に追加します。したがって、ポスト

ヘッダー・ギャップのないロー・ピクセル 1.0 イメージの場合、ピクセル・データはオフセット 34 から開始されます。

ロー・ピクセル・バージョン 1.0 のイメージでは、通常このフィールドには整数値 '30' が含まれます。この値はロー・ピクセル・イメージ・ヘッダーの長さです。（イメージ識別子の長さは含まれません。）ただし、ロー・ピクセル・フォーマットでは、このフィールドに 30 以上の任意の値を入れることができます。ヘッダー・データの終了点とこのヘッダー長によって指定されたピクセル・データの開始点の間の領域内の情報は、ロー・ピクセル・リーダーに無視されます。これは、ロー・ピクセルと互換性があるピクセル・データ領域を含む既存のイメージにロー・ピクセル・ヘッダーを付加するユーザーにとって役立ちます。このケースでは、ヘッダー長は 30+ 既存のヘッダー長に設定されます。このヘッダーの最大長は 4,294,967,265 バイトです。（4 バイト符号なしフィールドに格納できる最大値から、ロー・ピクセルに必要な 30 バイト・ヘッダーを引いた値です。）このフィールドは、ビッグ・エンディアン・バイト順序で格納されます。

メジャー・バージョン

イメージのコード化に使われるロー・ピクセル・フォーマットのバージョンのメジャー・バージョン番号を含むシングルバイト整数です。現行のロー・ピクセル・バージョンは 1.0 なので、このフィールドは '1' になります。

マイナー・バージョン

イメージのコード化に使われるロー・ピクセル・フォーマットのバージョンのマイナー・バージョン番号を含むシングルバイト整数です。現行のロー・ピクセル・バージョンは 1.0 なので、このフィールドは '0' になります。

イメージ幅

イメージのピクセル単位の幅（x ディメンション）です。

このフィールドには 40 億ピクセルを超えるイメージ・ディメンションを格納できますが、*interMedia image* の制限事項によって、 $1 \leq \text{幅} \leq 32767$ の範囲の値を指定する必要があります。このフィールドは、ビッグ・エンディアン・バイト順序で格納されます。

イメージ高さ

イメージのピクセル単位の高さ（y ディメンション）です。

このフィールドには 40 億ピクセルを超えるイメージ・ディメンションを格納できますが、*interMedia image* の制限事項によって、 $1 \leq \text{高さ} \leq 32767$ の範囲の値を指定する必要があります。このフィールドは、ビッグ・エンディアン・バイト順序で格納されます。

圧縮タイプ

このフィールドには、ロー・ピクセル・イメージの圧縮タイプが含まれます。バージョン 1.0 の場合、このフィールドには次の値を入れることができます。

値	名前	圧縮
1	NONE	圧縮なし
2	FAX3	CCITT Group 3 圧縮
3	FAX4	CCITT Group 4 圧縮

グレースケール、RGB および N バンド・イメージの場合、イメージは常に圧縮されないもので、値 0 のみが有効です。圧縮タイプが 1 または 2 の場合は、イメージはモノクロであるとみなされます。このような場合は、イメージに単一バンドのみが含まれているとみなされるため、NORMAL ピクセル・オーダー、NORMAL 走査線オーダーおよび BIP インターリーブを指定する必要があります。

ピクセル・オーダー

このフィールドは、ロー・ピクセル・イメージ内のピクセル・オーダーを示しています。通常、走査線内のピクセルは従来の正の x 軸に沿って左から右に順序付けされます。ただし、一部のアプリケーションでは走査線を右から左に順序付けする必要があります。

このフィールドには、次の値を入れることができます。

値	名前	ピクセル・オーダー
1	NORMAL	左端のピクセルが先頭
2	REVERSE	右端のピクセルが先頭

このフィールドに 0（ゼロ）を入れることはできません。0（ゼロ）はピクセル・オーダーが指定されていないことを示し、イメージの解釈を行うことができません。CCITT G3 および G4 圧縮タイプのイメージの場合は、このフィールドに値 1 を入れる必要があります。

走査線オーダー

このフィールドは、ロー・ピクセル・イメージ内の走査線オーダーを示しています。通常、イメージ内の走査線は上から下に順序付けされます。ただし、一部のアプリケーションでは走査線を下から上に順序付けする必要があります。

このフィールドには、次の値を入れることができます。

値	名前	走査線オーダー
1	NORMAL	上端の走査線が先頭

2 INVERSE 下端の走査線が先頭

このフィールドに 0（ゼロ）を入れることはできません。0（ゼロ）は走査線オーダーが指定されていないことを示し、イメージの解釈を行うことができません。CCITT G3 および G4 圧縮タイプのイメージの場合は、このフィールドに値 1 を入れる必要があります。

インターリーブ

このフィールドは、ロー・ピクセル・イメージ内のさまざまなバンドのインターリーブを示しています。さまざまなインターリーブ・オプションの意味の詳細は、D.5.3 項を参照してください。

このフィールドには、次の値を入れることができます。

値	名前	インターリーブ
1	BIP	ピクセル単位のバンドのインターリーブ (chunky)
2	BIL	ライン単位のバンドのインターリーブ
3	BSQ	順次バンド (プレーナ)

このフィールドに 0（ゼロ）を入れることはできません。0（ゼロ）はインターリーブが指定されていないことを示し、イメージの解釈を行うことができません。CCITT G3 および G4 圧縮タイプのイメージの場合は、このフィールドに値 1 を入れる必要があります。

バンド数

このフィールドにはイメージ内のバンドまたはプレーンの数が含まれます。1 ≤ バンド数 ≤ 255 の範囲の値を指定する必要があります。このフィールドに値 0（ゼロ）を入れることはできません。

CCITT イメージの場合は、このフィールドに値 1 を入れる必要があります。

赤チャネル番号

このフィールドには、イメージ変換操作中に赤チャネルとして使われるバンドの番号が含まれます。これは、標準の RGB イメージの解釈を変更するため、または N バンド・イメージ内で赤として使われるデフォルトのバンドを指定するために使うことができます。このデフォルトは、process() メソッドまたは processCopy() メソッドで inputChannels 演算子を使用して上書きできます。

イメージに 1 バンドしかない場合や、N バンド・イメージの 1 バンドのみを表示のために選択する必要がある場合、そのバンド番号は赤チャネルとしてコード化しなければなりません。この場合、緑チャネルおよび青チャネルは 0（ゼロ）に設定する必要があります。

このフィールドには値 0（ゼロ）を入れることはできません。(1 ≤ 赤 ≤ バンド数) の範囲の値のみを指定できます。

緑チャンネル番号

このフィールドには、イメージ変換操作中に緑チャンネルとして使われるバンドの番号が含まれます。これは、標準の RGB イメージの解釈を変更するため、または N バンド・イメージ内で緑として使われるデフォルトのバンドを指定するために使うことができます。このデフォルトは、`process()` メソッドまたは `processCopy()` メソッドで `inputChannels` 演算子を使用して上書きできます。

イメージに 1 バンドしかない場合や、N バンド・イメージの 1 バンドのみを表示のために選択する必要がある場合、そのバンド番号は赤チャンネルとしてコード化しなければなりません。この場合、緑チャンネルおよび青チャンネルは 0（ゼロ）に設定する必要があります。

このフィールドには、 $0 \leq \text{緑} \leq \text{バンド数の範囲}$ の値を入れることができます。

青チャンネル番号

このフィールドには、イメージ変換操作中に青チャンネルとして使われるバンドの番号が含まれます。これは、標準の RGB イメージの解釈を変更するため、または N バンド・イメージ内で青として使われるデフォルトのバンドを指定するために使うことができます。このデフォルトは、`process()` メソッドまたは `processCopy()` メソッドで `inputChannels` 演算子を使用して上書きできます。

イメージに 1 バンドしかない場合や、N バンド・イメージの 1 バンドのみを表示のために選択する必要がある場合、そのバンド番号は赤チャンネルとしてコード化しなければなりません。この場合、緑チャンネルおよび青チャンネルは 0（ゼロ）に設定する必要があります。

このフィールドには、 $0 \leq \text{青} \leq \text{バンド数の範囲}$ の値を入れることができます。

予約領域

予約領域という名前のこれらの 8 バイトの用途は現在開発中ですが、この領域はロー・ピクセル 1.0 イメージ内でも確保されています。これらのバイトはすべて 0（ゼロ）にクリアする必要があります。0（ゼロ）にクリアしないと、予想できない結果になります。

D.4 ロー・ピクセル・ポストヘッダー・ギャップ

イメージ識別子およびイメージ・ヘッダーとは別に、ロー・ピクセル・バージョン 1.0 イメージには、実際のピクセル・データの前にオプションのポストヘッダー・ギャップが含まれています。イメージ・ヘッダーの予約領域とは異なり、このギャップ内のバイトには、任意の値を入れることができます。これは、イメージについての追加のメタデータや、場合によっては別のファイル形式の実際のイメージ・ヘッダーを格納するのに役立ちます。

ただし、このギャップに格納する情報についての基準はないので、この領域にメタデータを格納する場合は、別のユーザーがそのデータに対して異なる解釈を行う可能性があるのに注意してください。また、ロー・ピクセル・イメージを処理するときは、このギャップに格納された情報が宛先イメージにコピーされないことにも注意してください。入力と同一の位置に出力を書き込む `process()` メソッドの場合は、この処理が行われるトランザクションをロールバックしないと、ソース情報が失われます。

D.5 ロー・ピクセル・データ・セクションおよびピクセル・データ・フォーマット

ロー・ピクセル・イメージのデータ・セクションは、イメージの実際のピクセル・データが格納される場所です。この領域は、ビットマップ・データと呼ばれることがあります。この項では、ビットマップ・データのレイアウトについて説明します。

CCITT 圧縮を使用したイメージの場合、ビットマップ・データ領域には、追加ヘッダーのないロー CCITT ストリームが格納されます。この項の残りの部分は、非圧縮イメージのみに適用されます。

ロー・ピクセル・イメージ内のビットマップ・データは、8 ビットの、プレーン単位、ピクセル単位、ダイレクト・カラーのバック・データとして格納されます。ピクセル、走査線またはバンドのブロック化や埋込みはありません。走査線は、イメージ内で上端を先頭または下端を先頭にすることができます。走査線内で、ピクセルは左端を先頭または右端を先頭として順序付けすることができます。これらのオプションはすべて、比較的簡単な方法のインターリーブによって影響を受けます。例として、後続の項を参照してください。

D.5.1 走査線の順序付け

画面上で、あるイメージが次のようになっているとします。

```
1111111111...
2222222222...
3333333333...
4444444444...
```

各桁は単一のピクセルを表しています。桁の値はそのピクセルが存在する走査線です。

通常、ピクセルの上部または上端行を形成する走査線は、下部の走査線より前に、イメージ・データ・ストリーム内に格納されます。先行するイメージは、ビットマップ・データ・ストリーム内に次のように現れます。

```
...1111111111...2222222222...3333333333...4444444444...
```

先頭の走査線が、残りの走査線より先に現れることに注意してください。ロー・ピクセル・フォーマットでは、この走査線の順序付けを **NORMAL** と呼びます。

ただし、一部のアプリケーションでは、次のように下端の走査線がデータ・ストリームの先頭に現れるほうが好ましい場合があります。

```
...4444444444...3333333333...2222222222...1111111111...
```

ロー・ピクセル・フォーマットでは、この走査線の順序付けを **INVERSE** と呼びます。

D.5.2 ピクセルの順序付け

画面上で、あるイメージの走査線が次のようになっているとします。

```
...123456789...
```


各桁は単一のピクセルを表しています。桁の値はそのピクセルが存在する列です。

通常、左端のピクセルを形成するデータは、右側のピクセルより前に、イメージ・データ・ストリーム内に格納されます。先行する走査線は、ビットマップ・データ・ストリーム内に次のように現れます。

...123456789...

左のピクセルが、残りのピクセルより先に現れることに注意してください。ロー・ピクセル・フォーマットでは、このピクセルの順序付けを **NORMAL** と呼びます。

ただし、一部のアプリケーションでは、次のように右端のピクセルがデータ・ストリームの先頭に現れるほうが好ましい場合があります。

...987654321...

ロー・ピクセル・フォーマットでは、このピクセルの順序付けを **REVERSE** と呼びます。

D.5.3 バンド・インターリーブ

バンド・インターリーブは、イメージ・バッファ内のピクセル・データのさまざまなバンドの相対位置を示します。

バンドはその出現ごとに、先頭のバンドを 1、最終のバンドを n として、イメージ・データ・ストリーム内で順序付けされます。バンド 0 は、バンドやデータがないことを示します。

ピクセル単位のバンドのインターリーブ (BIP)、つまり *chunky*

BIP イメージ、つまり *chunky* イメージでは、ピクセル・データのさまざまなバンドまたはチャンネルがピクセル単位で順次配置されます。このため、1 つのピクセルのすべてのデータが 1 箇所に配置されます。イメージのバンドが赤、緑および青チャンネルの場合は、BIP イメージは次のようになります。

```
scanline 1: RGBRGBRGBRGBRGBRGB...
scanline 2: RGBRGBRGBRGBRGBRGB...
scanline 3: RGBRGBRGBRGBRGBRGB...
...
```

ライン単位のバンドのインターリーブ (BIL)

BIL イメージでは、ピクセル・データのさまざまなバンドが走査線単位で順次配置されます。このため、1 つのピクセルのデータがイメージの架空の複数行にわたって配置されます。これは、データを走査線単位でバッファに格納するセンサーのデータ編成を反映しています。イメージのバンドが赤、緑および青チャンネルの場合は、BIL イメージは次のようになります。

```
scanline 1: RRRRRRRRRRRRRRRRRR...
             GGGGGGGGGGGGGGGGGG...
             BBBBBBBBBBBBBBBBBB...
scanline 2: RRRRRRRRRRRRRRRRRR...
```

```

GGGGGGGGGGGGGGGGGGGG...
BBBBBBBBBBBBBBBBBBBB...
scanline 3: RRRRRRRRRRRRRRRRRRR...
GGGGGGGGGGGGGGGGGGGG...
BBBBBBBBBBBBBBBBBBBB...
...

```

順次バンド (BSQ)、つまりプレーナ

プレーナ・イメージでは、ピクセル・データのさまざまなバンドがビット・プレーン単位で順次配置されます。このため、1つのピクセルのデータがイメージの複数プレーンにわたって配置されます。これは、メモリー内のさまざまな位置からの、異なるディスプレイ電子銃を制御するビデオ・バッファ・システムのデータ編成を反映しています。イメージのバンドが赤、緑および青チャネルの場合は、プレーナ・イメージは次のようになります。

```

plane 1: RRRRRRRRRRRRRRRRR... (part of scanline 1)
RRRRRRRRRRRRRRRRRR... (part of scanline 2)
RRRRRRRRRRRRRRRRRR... (part of scanline 3)
...
plane 2: GGGGGGGGGGGGGGGGG... (part of scanline 1)
GGGGGGGGGGGGGGGGGG... (part of scanline 2)
GGGGGGGGGGGGGGGGGG... (part of scanline 3)
...
plane 3: BBBBBBBBBBBBBBBBB... (part of scanline 1)
BBBBBBBBBBBBBBBBBB... (part of scanline 2)
BBBBBBBBBBBBBBBBBB... (part of scanline 3)
...

```

D.5.4 N バンド・データ

ロー・ピクセル・フォーマットでは、イメージ内で最大 255 バンドのデータをサポートしています。イメージ内のデータのバンドの相対位置は、データの 3 バンドのインターリーブの例を示した D.5.3 項で説明しています。

単一バンドのデータの場合、インターリーブがないため、3つの方式はすべて等価です。他のバンド数のインターリーブの例を、次の表に示します。すべてのイメージには、3つの走査線および4つの列があります。各ピクセルの各バンドは、1桁のバンド番号によって表されます。イタリックの標準テキストの数字は、イメージの2番めの走査線を表し、太字の数字はイメージの3番めの走査線を表しています。

バンド数	BIP	BIL	BSQ
2	12121212 12121212 12121212	11112222 11112222 11112222	111111111111 <i>222222222222</i>

4	1234123412341234 1234123412341234 1234123412341234	1111222233334444 1111222233334444 1111222233334444	111111111111 222222222222 333333333333 444444444444
5	12345123451234512345 12345123451234512345 12345123451234512345	11112222333344445555 11112222333344445555 11112222333344445555	111111111111 222222222222 333333333333 444444444444 555555555555

D.6 ロー・ピクセル・ヘッダーの C 構造体

次の C 言語構造体では、ロー・ピクセル・ヘッダーをプログラムに基づいた方法で記述します。この構造体は未整列状態でイメージ・ファイル内に格納されます。（つまり、各フィールドは 1 バイト境界上に配置されます。）また、整数はすべて、ビッグ・エンディアン・バイト順序で格納されます。

```
struct RawPixelHeader
{
    unsigned char identifier[4]; /* Always "RPIX" */

    unsigned long hlength; /* Length of this header in bytes */
    /* Including the hlength field */
    /* Not including the identifier field */
    /* &k.hlength + k.hlength = pixels */

    unsigned char majorversion; /* Major revision # of RPIX format */
    unsigned char minorversion; /* Minor revision # of RPIX format */

    unsigned long width; /* Image width in pixels */
    unsigned long height; /* Image height in pixels */
    unsigned char comtype; /* Compression (none, FAXG3, FAXG4, ... ) */
    unsigned char pixelorder; /* Pixel order */
    unsigned char scnorder; /* Scanline order */
    unsigned char interleave; /* Interleaving (BIP/BIL/Planar) */

    unsigned char numbands; /* Number of bands in image (1-255) */
    unsigned char rchannel; /* Default red channel assignment */
    unsigned char gchannel; /* Default green channel assignment */
    unsigned char bchannel; /* Default blue channel assignment */
    /* Grayscale images are encoded in R */
    /* The first band is '1', not '0' */
    /* A value of '0' means "no band" */

    unsigned char reserved[8]; /* For later use */
};
```

D.7 ロー・ピクセル・ヘッダーの C 定数

次の C 言語定数では、ロー・ピクセル・ヘッダー内で使われる値を定義します。

```
#define RPIX_IDENTIFIER "RPIX"

#define RPIX_HEADERLENGTH 30

#define RPIX_MAJOR_VERSION 1
#define RPIX_MINOR_VERSION 0

#define RPIX_COMPRESSION_UNDEFINED 0
#define RPIX_COMPRESSION_NONE 1
#define RPIX_COMPRESSION_CCITT_FAX_G3 2
#define RPIX_COMPRESSION_CCITT_FAX_G4 3
#define RPIX_COMPRESSION_DEFAULT RPIX_COMPRESSION_NONE

#define RPIX_PIXEL_ORDER_UNDEFINED 0
#define RPIX_PIXEL_ORDER_NORMAL 1
#define RPIX_PIXEL_ORDER_REVERSE 2
#define RPIX_PIXEL_ORDER_DEFAULT RPIX_PIXEL_ORDER_NORMAL

#define RPIX_SCANLINE_ORDER_UNDEFINED 0
#define RPIX_SCANLINE_ORDER_NORMAL 1
#define RPIX_SCANLINE_ORDER_INVERSE 2
#define RPIX_SCANLINE_ORDER_DEFAULT RPIX_SCANLINE_ORDER_NORMAL

#define RPIX_INTERLEAVING_UNDEFINED 0
#define RPIX_INTERLEAVING_BIP 1
#define RPIX_INTERLEAVING_BIL 2
#define RPIX_INTERLEAVING_BSQ 3
#define RPIX_INTERLEAVING_DEFAULT RPIX_INTERLEAVING_BIP

#define RPIX_CHANNEL_UNDEFINED 0
```

UNDEFINED 値のさまざまなマクロは例示のためのもので、必ず使われるとは限らないことに注意してください。ただし、RPIX_CHANNEL_UNDEFINED は例外で、単一バンド・イメージの緑および青チャネル用に使われます。

D.8 ロー・ピクセル PL/SQL 定数

次の PL/SQL 定数では、ロー・ピクセル情報内で使われる値を定義します。定数は、RPIX イメージ識別子の長さ + RPIX ヘッダーの長さを表します。

```
CREATE OR REPLACE PACKAGE ORDImageConstants AS
  RPIX_HEADER_LENGTH_1_0  CONSTANT INTEGER := 34;
END ORDImageConstants;
```

D.9 CCITT 圧縮を使用したロー・ピクセル・イメージ

一般に、ロー・ピクセル・フォーマットは非圧縮ダイレクト・カラー・イメージ向けですが、CCITT Fax Group 3 または Fax Group 4 圧縮を使用したモノクロ・イメージの格納もサポートしています。これは、ドキュメント管理アプリケーションなど、白黒ページの走査を格納する場合に役立ちます。通常、これらのイメージは、グレースケールとして格納しても実用的ではありません。これらのイメージで使われている高解像度に伴う、使われないデータ・ビットによって、過度のディスク領域が必要になるためです。

CCITT 圧縮を使用したロー・ピクセル・イメージは、標準のロー・ピクセル・イメージとして処理されますが、次の制限があります。

- 「圧縮タイプ」フィールドには、D.3 項 (FAX3 または FAX4) で概説したように、値 1 または 2 を入れる必要があります。
- 「ピクセル・オーダー」フィールドには、値 1 (NORMAL ピクセル・オーダー) を入れる必要があります。
- 「走査線オーダー」フィールドには、値 1 (NORMAL 走査線オーダー) を入れる必要があります。
- 「インターリーブ」フィールドには、値 1 (BIP インターリーブ) を入れる必要があります。
- 「バンド数」フィールドには、値 1 (1 バンド) を入れる必要があります。
- 「赤チャネル番号」フィールドには、値 1 を入れる必要があります。
- 「緑チャネル番号」フィールドおよび「青チャネル番号」フィールドには、値 0 (バンドなし) を入れる必要があります。

これらの制限の他に、ピクセル・データへの直接アクセスを試行するアプリケーションでは、CCITT 形式のデータの読取り / 書込み方法を認識する必要があります。

D.10 外部イメージ・サポートおよびそのロー・ピクセル・フォーマット

interMedia image では、いくつかの単純なパラメータで記述でき、データがある種の簡単な方法でイメージ・ファイル内に配置されている、一部の外部イメージの読取りをサポートしています。サポートしている形式は非常に多く、絶えず変化するため、形式のリストはありません。そのかわりに、*interMedia image* の外部イメージ・サポートを使用してイメージを読取り可能かどうかを判別するための簡単なガイドラインがあります。次の項に、その規則を要約します。

ヘッダー

外部イメージには、ヘッダー長が 4,294,967,265 バイトを超えない限り、任意の形式で任意のヘッダーを付けることができます。またはヘッダーを付けなくても構いません。前述したように、このヘッダー内の情報はすべて無視されます。

イメージ幅

外部イメージは、最大 32,767 ピクセルの幅にすることができます。

イメージ高さ

外部イメージは、最大 32,767 ピクセルの高さにすることができます。

圧縮タイプ

外部イメージは、圧縮しないか、CCITT Fax Group 3 または Fax Group 4 を使用して圧縮する必要があります。ランレングス・コード化など他の圧縮方式は、現在サポートしていません。

ピクセル・オーダー

外部イメージには、ピクセルを左から右、または右から左の順序で格納できます。犁耕型順序付けなど他のピクセル順序付け方式は、現在サポートしていません。

走査線オーダー

外部イメージには、上端先頭または下端先頭の走査線オーダーを設定できます。イメージ表示内の隣接する走査線は、イメージ記憶域内でも隣接している必要があります。イメージ・フォーマットの中には、たとえば走査線 1、5、9…が隣接し、2、6、10 が隣接するというように、イメージ走査線を互い違いに配置するものがあります。この方式は現在サポートしていません。

インターリーブ

外部イメージでは、BIP、BIL または BSQ インターリーブを使う必要があります。他のデータ・バンドの配置はサポートしていません。また、バンドにはピクセル、走査線またはバンドレベルのブロック化や埋め込みを使うことはできません。

バンド数

外部イメージでは、最大 255 バンドのデータを使うことができます。データのバンド数が 255 を超えると、イメージのインターリーブが順次バンドの場合は、最初の 255 バンドにアクセスできます。このケースでは、データの追加のバンドはアクセス可能なバンドの先に配置され、最初の 255 バンドのレイアウトには影響を与えません。他のインターリーブが使われているイメージでは、追加のバンドによってビットマップ・データのレイアウトが変更されるため、255 を超えるバンドを使うことができません。

トレーラ

外部イメージでは、ビットマップ・データの後にイメージ・トレーラを格納することができます。このトレーラは、任意の長さにすることができます。ただし、このようなデータは *interMedia image* に完全に無視されるため、このようなトレーラの存在や長さを指定する方法（や必要性）はありません。

このようなトレーラを含むイメージが `process()` メソッドまたは `processCopy()` メソッドで修正された場合は、結果のイメージにこのトレーラは含まれません。`processCopy()` メソッドの場合は、ソース・イメージは元の状態で残ります。

サンプル・プログラム

Oracle8i *interMedia* には、使用可能なスクリプトおよびサンプル・プログラムがいくつか付属しています。

サンプル Oracle8i *interMedia* スクリプトおよびプログラムは、この製品のインストール後に作成される次のディレクトリに格納されます。

```
$ORACLE_HOME/ord/aud/demo/  
$ORACLE_HOME/ord/img/demo/  
$ORACLE_HOME/ord/vid/demo/
```

E.1 サンプル・オーディオ・スクリプト

オーディオ・スクリプトを構成するファイルは、次のとおりです。

- **auddemo.sql** - オーディオのデモンストレーション（デモ）。次のオーディオ・オブジェクト機能のデモを行います。
 - *interMedia* オブジェクトをチェックします。
 - オーディオを組み込んだサンプル表を作成します。
 - オーディオ表に NULL 行を挿入します。
 - 行をチェックします。
 - すべてのオーディオ属性を直接チェックします。
 - メソッドをコールすることにより、すべてのオーディオ属性をチェックします。
 - **fplugins.sql** および **fpluginb.sql** というファイルを使用して独自のフォーマット・プラグインをインストールします。この 2 つのファイルの、*interMedia* オーディオを拡張してオーディオ・データ形式をサポートする方法については、次の 2 つの箇条書きおよび 2.1.14 項を参照してください。
- **fplugins.sql** - デモ・フォーマット・プラグインの仕様。サポート対象のフォーマット・プラグインを記述するときのガイドラインとして使います。

- `fpluginb.sql` - デモ・フォーマット・プラグインの本体。サポート対象のフォーマット・プラグインを記述するときのガイドラインとして使います。

この SQL デモを実行するときの要件および指示については、`$ORACLE_HOME/ord/aud/demo` ディレクトリの `README.txt` を参照してください。

E.2 イメージの変更またはインストール・テストのサンプル・プログラム

Oracle8i *interMedia* のイメージをインストールすると、Oracle8i *interMedia* イメージ・デモンストレーション・プログラムを実行できます。このプログラムは、インストールが成功したかどうかを確認するテストとして使うこともできます。

この項では、*interMedia* イメージ・デモの作成および実行に必要なステップについて説明します。

interMedia イメージ・デモ・ファイルは、`<ORACLE_HOME>/ord/img/demo` にあります。`<ORACLE_HOME>` は、`ORACLE_HOME` ディレクトリのことです。

E.2.1 デモンストレーション（デモ）のインストール・ステップ

1. Oracle8i *interMedia* イメージ・デモでは、SCOTT/TIGER データベース・ユーザーを使います。このユーザーが存在しない場合は作成してください。

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> create user SCOTT identified by tiger;
SVRMGR> grant connect,resource to SCOTT;
```

2. イメージ・デモ・ディレクトリを作成します。`<ORACLE_HOME>` は、`ORACLE_HOME` ディレクトリです。

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> create or replace directory imgdemodir as '<ORACLE_HOME>/ord/img/demo';
```

ディレクトリの指定は、オペレーティング・システムにより異なります。適切な位置の指定方法については、各オペレーティング・システムのインストールおよび構成ガイドを参照してください。

3. このディレクトリに対する権限を PUBLIC に割り当てます。

```
SVRMGR> grant read on directory imgdemodir to public with grant option;
```

4. 必要に応じて `imgdemo` プログラムを作成します。

```
% cd <ORACLE_HOME>/ord/img/demo
% make -f demo_ording.mk imgdemo
```

make 操作は、オペレーティング・システムにより異なります。各オペレーティング・システムでこのプログラムを作成する方法については、それぞれのインストールおよび構成ガイドを参照してください。

E.2.2 デモの実行

imgdemo ファイルは、プログラムでの **Oracle8i interMedia** イメージの使用法を示すサンプル・プログラムです。このデモは **C** 言語を使用して記述されており、**OCI (Oracle コール・インタフェース)** を使用してデータベースにアクセスし、**Oracle8i interMedia** イメージを実行します。

このプログラムでは、**imgdemo.dat** を操作します。**imgdemo.dat** は **demo** ディレクトリに保存されているビットマップ (**BMP**) イメージ・ファイルです。イメージ・ファイルは **demo** と同じディレクトリにある場合は、コマンド行にイメージ・ファイル名をオプションで指定できます。いずれの場合でも、**Oracle8i interMedia** イメージによってイメージが操作されると、結果のイメージは **imgdemo.out** ファイルに書き込まれ、ユーザーが指定した一般的なレンダリング・ツールで表示できます。

デモを実行すると、デフォルト・データベースの **SCOTT/TIGER** スキーマ内で **IMGDEMOTAB** という名前の表がいったん削除されてから再作成されます。この表を使用してデモ・データが管理されます。表が作成されると、イメージ・ファイルへの参照が表に挿入されます。次に、そのデータが表にロードされ、**ORDImage** の **processCopy()** メソッドを使用して **JFIF** に変換されます。

イメージのプロパティが、**setProperties()** メソッドを使用してデータベース内に抽出されます。**setProperties()** メソッドがコールされた後、**UPDATE** コマンドが発行されます。**setProperties()** のコールでは、そのタイプの属性のローカル・コピーだけが更新されるので、**UPDATE** コマンドが必要になります。

次に、**Oracle8i interMedia** イメージの **process()** メソッドを使用して、データベース内でイメージのカットおよびスケール変更が行われます。この処理は、変更をコミットする更新に続いて行われます。プログラムにより、ピクセル位置 (**100,100**) を起点として、幅 **100** ピクセル、高さ **100** ピクセルの範囲のイメージがカットされます。このサブイメージは、元のサイズの **2** 倍に拡大された後、**imgdemo.out** ファイル内のファイル・システムに書き込まれます。

デモ・プログラムを終了しても、**imgdemo.out** ファイルは現行のディレクトリに残ります。また、**IMGDEMOTAB** 表も、データベースの **SCOTT/TIGER** スキーマ内に残ります。

例 E-1 コマンド行からデモを実行する

imgdemo をコマンド行に入力してデモを実行します。オプションで、このデモで別のイメージを使うには、デモと同じディレクトリにファイルをコピーし、**imgdemo** の引数としてコマンド行でそのファイル名を指定します。

次のコマンドを使います。

```
$ imgdemo <optional-image-filename>
```

デモを実行すると、その進行状況についてのメッセージが表示され、セットアップが正しく行われなかった場合にはエラーが表示されます。表示される可能性があるメッセージは次のとおりです。

```
Dropping table IMGDEMOTAB...
Creating and populating table IMGDEMOTAB...
Loading data into cartridge...
Modifying image characteristics...
Writing image to file imgdemo.out...
Disconnecting from database...
Logged off and detached from server.
Demo completed successfully.
```

プログラムでエラーが発生する場合、**Oracle8i interMedia** のイメージ・ソフトウェアが正しくインストールされていないか、またはデータベースが起動されていない可能性があります。プログラムが正常に終了すると、元のイメージと結果のイメージ（すでに説明したようにカットとスケール変更が行われた状態）を一般的なイメージ・レンダリング・ツールで表示できます。

E.3 サンプル・ビデオ・スクリプト

ビデオ・スクリプトを構成するファイルは、次のとおりです。

- **viddemo.sql** - ビデオ・デモ。次のビデオ・オブジェクト機能を示します。
 - **interMedia** オブジェクトをチェックします。
 - ビデオを組み込んだサンプル表を作成します。
 - ビデオ表に **NULL** 行を挿入します。
 - 行をチェックします。
 - すべてのビデオ属性を直接チェックします。
 - メソッドをコールすることにより、すべてのビデオ属性をチェックします。
 - **fplugins.sql** および **fpluginb.sql** というファイルを使用して独自のフォーマット・プラグインをインストールします。この 2 つのファイルの、**interMedia** ビデオを拡張して新しいビデオ・データ形式をサポートする方法については、次の 2 つの箇条書きおよび 2.3.12 項を参照してください。
- **fplugins.sql** - デモ・フォーマット・プラグインの仕様。サポート対象のフォーマット・プラグインを記述するときのガイドラインとして使います。
- **fpluginb.sql** - デモ・フォーマット・プラグインの本体。サポート対象のフォーマット・プラグインを記述するときのガイドラインとして使います。

この SQL デモを実行するときの要件および指示については、\$ORACLE_HOME/ord/vid/demo ディレクトリの README.txt を参照してください。

E.4 Java デモ

Java デモは、オーディオおよびビデオ・クライアント側 Java クラスの使用法の習得を支援するために提供されます。このデモにより、独自のアプリケーションを作成できるようになります。この 2 つのデモでは、クライアント側でオーディオおよびビデオ・オブジェクトがインスタンス化され、多くのアクセサ・メソッドがコールされます。オーディオ Java デモ・ファイルは ORACLE_HOME/ord/aud/demo ディレクトリに、ビデオ Java デモ・ファイルは \$ORACLE_HOME/ord/vid/demo ディレクトリにあります。各 Java デモを起動するときの実行要件および指示については、各ディレクトリにある README.txt ファイルを参照してください。

この付録では、このガイドの他の部分でデモ・プログラムとして提示している追加の参照情報を説明します。次の説明を行います。

- ORDAnotations 型およびそのメソッド

F.1 オブジェクト型

Oracle8i *interMedia* のデモ・プログラムでは、次のオブジェクト型が使われます。

- ORDAnotations: 注釈情報を格納します。

ORDAnnotations オブジェクト型

ORDAnnotations オブジェクト型では、注釈情報を格納します。

このオブジェクト型は、次のリストで定義されます。

```
CREATE OR REPLACE TYPE ORDAnnotations
AS OBJECT
(
  -- TYPE ATTRIBUTES
  annotations          ORDAnnotationList,
  -- METHOD DECLARATION
  MEMBER PROCEDURE setAnnotation(key   IN VARCHAR2,
                                value IN VARCHAR2),
  MEMBER FUNCTION getAnnotation(key IN VARCHAR2)
                  RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getAnnotation, WNDS, WNPS, RNPS),

  MEMBER PROCEDURE deleteAnnotation(key IN VARCHAR2),
  MEMBER PROCEDURE appendToAnnotation(key   IN VARCHAR2,
                                       value IN VARCHAR2),
  MEMBER FUNCTION getAnnotationCount RETURN INTEGER,
  PRAGMA RESTRICT_REFERENCES(getAnnotationCount, WNDS, WNPS, RNPS),

  MEMBER FUNCTION getAnnotationObjByPosition(pos IN INTEGER)
                  RETURN ORDAnnotation,
  PRAGMA RESTRICT_REFERENCES(getAnnotationObjByPosition, WNDS, WNPS, RNPS),

  MEMBER FUNCTION getAnnotationKeyByPosition(pos IN INTEGER)
                  RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getAnnotationKeyByPosition, WNDS, WNPS, RNPS),

  MEMBER FUNCTION getAnnotationValueByPosition(pos IN INTEGER)
                  RETURN VARCHAR2,
  PRAGMA RESTRICT_REFERENCES(getAnnotationValueByPosition, WNDS, WNPS, RNPS),
```

パラメータは次のとおりです。

- **annotations:** 1 つ以上の **ORDAnnotation** オブジェクトを含む **ORDAnnotationList** 表で、各 **ORDAnnotation** オブジェクトにはキーが **VARCHAR2(256)**、値が **VARCHAR2(4000)** として定義されています。

F.2 メソッド

この項では、注釈操作で使われるメソッドの参照情報を提供します。

ORDAnnotations メソッドは、次のように定義されます。

注釈属性に関連付けられた ORDAnnotations メソッド

- `setAnnotation()`: ORDAnnotation オブジェクト・キーおよび値フィールドの値を作成します。
- `getAnnotation()`: キー値に基づいて、ORDAnnotation オブジェクトの値フィールドの値を返します。
- `deleteAnnotation()`: キー値により、ORDAnnotation オブジェクトを削除します。
- `appendToAnnotation()`: ORDAnnotation オブジェクトのキーおよび値フィールドの値に情報を追加します。
- `getAnnotationCount`: ORDAnnotationList 表の ORDAnnotation オブジェクトの数を返します。
- `getAnnotationObjByPosition()`: ORDAnnotationList 表の位置により、ORDAnnotation オブジェクトを返します。
- `getAnnotationKeyByPosition()`: ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、キー・フィールドの値を返します。
- `getAnnotationValueByPosition()`: ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、値フィールドの値を返します。

F.2.1 注釈属性に関連付けられた ORDAnnotations メソッド

この項では、注釈属性に関連付けられた ORDAnnotations メソッドについての参照情報を提供します。

この章で説明するメソッドは、TANNOT 表に基づいた例です。この章の例を参照する場合は、次の TANNOT 表定義を参照してください。

```
CREATE TABLE TANNOT(n NUMBER, a ORDSYS.ORDAnnotations)
storage (initial 100K next 100K pctincrease 0),
nested table a.annotations store as base_annot;
INSERT INTO TANNOT VALUES(1, ORDSYS.ORDAnnotations(NULL));
```

setAnnotation() メソッド

形式

```
setAnnotation(key    IN VARCHAR2,  
              value  IN VARCHAR2);
```

説明

ORDAnnotation オブジェクトのキーおよび値フィールドの値の作成または変更を行います。

パラメータ

key

ORDAnnotation オブジェクトのキー・フィールドの値

value

ORDAnnotation オブジェクトの値フィールドの値

使用方法

なし

プラグマ

なし

例外

なし

例

なし

getAnnotation() メソッド

形式

```
getAnnotation(key IN VARCHAR2) RETURN VARCHAR2;
```

説明

キー・フィールドの値に基づいて、ORDAnnotation オブジェクトの値フィールドの値を返します。

パラメータ

key

ORDAnnotation オブジェクトのキー・フィールドの値

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAnnotation, WNDS, WNPS, RNPS)

例外

値を基にしてキーを検索できなかった場合、このメソッドをコールすると、KEY_NOT_FOUND 例外が発生します。

注釈属性の値が空 (NULL) の場合、このメソッドをコールすると、NULL_ANNOTATION_LIST 例外が発生します。

例

ORDAnnotations オブジェクトの ORDAnnotationList 表の最初に位置する、ORDAnnotation オブジェクトのキーおよび値フィールドの値を返します。

```
DECLARE
    mya ORDSYS.ORDAnnotations;
BEGIN
    SELECT T.a INTO mya FROM TANNOT T WHERE N = 1;
    DBMS_OUTPUT.PUT_LINE('Annotation for key: ' || mya.getAnnotation('key'));
    DBMS_OUTPUT.PUT_LINE('Annotation for value: ' || mya.getAnnotation('note1'));
    EXCEPTION
    WHEN ORDSYS.ORDAnnotationsExceptions.KEY_NOT_FOUND THEN
```

getAnnotation() メソッド

```
        DEMS_OUTPUT.PUT_LINE('key does not exist in annotations');  
    END;
```

deleteAnnotation() メソッド

形式

```
deleteAnnotation(key IN VARCHAR2);
```

説明

キー・フィールドの値に基づいて、ORDAnnotation オブジェクトを削除します。

パラメータ

key
ORDAnnotation オブジェクトのキー・フィールドの値

使用方法

なし

プラグマ

なし

例外

値でキーが検索できなかった場合、このメソッドをコールすると、KEY_NOT_FOUND 例外が発生します。

例

なし

appendToAnnotation() メソッド

形式

```
appendToAnnotation(key    IN VARCHAR2,  
                   value  IN VARCHAR2);
```

説明

キーおよび値フィールドに指定された値を、ORDAnnotation オブジェクトの現行の値に追加します。

パラメータ

key

ORDAnnotation オブジェクトのキー・フィールドに追加された値

value

ORDAnnotation オブジェクトの値フィールドに追加された値

使用方法

なし

プラグマ

なし

例外

値でキーが検索できなかった場合、このメソッドをコールすると、KEY_NOT_FOUND 例外が発生します。

注釈属性の値が空 (NULL) の場合、このメソッドをコールすると、NULL_ANNOTATION_LIST 例外が発生します。

例

なし

getAnnotationCount メソッド

形式

```
getAnnotationCount RETURN INTEGER;
```

説明

ORDAnnotationList 表の ORDAnnotation オブジェクトの数を返します。

パラメータ

なし

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAnnotationCount, WNDS, WNPS, RNPS)

例外

なし

例

ORDAnnotation オブジェクト数を返します。

```
DECLARE
  mya ORDSYS.ORDAnnotations;
BEGIN
  SELECT T.a INTO mya FROM TANNOT T WHERE N = 1;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(mya.getAnnotationCount));
END;
```

getAnnotationObjByPosition() メソッド

形式

```
getAnnotationObjByPosition(pos IN INTEGER) RETURN ORDAnotation;
```

説明

ORDAnnotatationList 表の位置により、ORDAnnotatation オブジェクトを戻します。

パラメータ

pos
ORDAnnotatationList 表の ORDAnotation オブジェクトの位置

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAnnotationObjByPosition, WNDS, WNPS, RNPS)

例外

pos の値が ORDAnotationList 表の ORDAnotation オブジェクトの数より大きい場合、このメソッドをコールすると、POSITION_OUT_OF_RANGE 例外が発生します。

例

ORDAnnotatationList 表の位置により、ORDAnnotatation オブジェクトを戻します。

```
DECLARE
    mya ORDSYS.ORDAnnotations;
    obj ORDSYS.ORDAnnotatation;
    n    INTEGER;
BEGIN
    SELECT T.a INTO mya FROM TANNOT T WHERE N = 1;
    n := mya.getAnnotationCount;
    FOR i in 1..n LOOP
        obj := mya.getAnnotationObjByPosition(i);
        DBMS_OUTPUT.PUT_LINE('Key ' || TO_CHAR(i) || ':' || obj.key);
        DBMS_OUTPUT.PUT_LINE('Value: ' || obj.value);
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
```



```
END;
```

getAnnotationKeyByPosition() メソッド

形式

```
getAnnotationKeyByPosition(pos IN INTEGER) RETURN VARCHAR2;
```

説明

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、キー・フィールドの値を返します。

パラメータ

pos

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAnnotationKeyByPosition, WNDS, WNPS, RNPS)

例外

pos の値が ORDAnnotationList 表の ORDAnnotation オブジェクトの数より大きい場合、このメソッドをコールすると、POSITION_OUT_OF_RANGE 例外が発生します。

例

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、キー・フィールドの値を返します。

```
DECLARE
  mya ORDSYS.ORDAnnotations;
  n   INTEGER;
BEGIN
  SELECT T.a INTO mya FROM TANNOT T WHERE N = 1;
  n := mya.getAnnotationCount;
  FOR i in 1..n LOOP
    DBMS_OUTPUT.PUT_LINE('Key ' || TO_CHAR(i) || ':' || mya.getAnnotationKeyByPosition(
i));
    DBMS_OUTPUT.PUT_LINE('Value: ' || mya.getAnnotationValueByPosition(i));
    DBMS_OUTPUT.PUT_LINE('-----');
```

```
END LOOP;  
END;
```

getAnnotationValueByPosition() メソッド

形式

```
getAnnotationValueByPosition(pos IN INTEGER) RETURN VARCHAR2;
```

説明

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、値フィールドの値を返します。

パラメータ

pos

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置

使用方法

なし

プラグマ

Pragma RESTRICT_REFERENCES(getAnnotationValueByPosition, WNDS, WNPS, RNPS)

例外

pos の値が ORDAnnotationList 表の ORDAnnotation オブジェクトの数より大きい場合、このメソッドをコールすると、POSITION_OUT_OF_RANGE 例外が発生します。

例

ORDAnnotationList 表の ORDAnnotation オブジェクトの位置により、値フィールドの値を返します。

```
DECLARE
    mya ORDSYS.ORDAnnotations;
    n    INTEGER;
BEGIN
    SELECT T.a INTO mya FROM TANNOT T WHERE N = 1;
    n := mya.getAnnotationCount;
    FOR i in 1..n LOOP
        DBMS_OUTPUT.PUT_LINE('Key ' || TO_CHAR(i) || ':' || mya.getAnnotationKeyByPosition(
i));
        DBMS_OUTPUT.PUT_LINE('Value: ' || mya.getAnnotationValueByPosition(i));
        DBMS_OUTPUT.PUT_LINE('-----');
```

```
END LOOP;  
END;
```

getAnnotationValueByPosition() メソッド

G

よく寄せられる質問

Oracle8i *interMedia* のインストール後に、オンラインで FAQ のリストを含むテキスト・ファイルを表示できます。

このテキスト・ファイルは、次の場所に格納されています。

`$ORACLE_HOME/ord/img/admin/imfaq.txt`

例外およびエラー・メッセージ

H.1 例外

次の項では、*interMedia* オブジェクトの例外およびエラー・メッセージについて説明します。

H.1.1 ORDAudioExceptions 例外

次の例外は、ORDAudio オブジェクトに関連付けられています。

UNSUPPORTED_DATA_SOURCE

原因：この例外は、データ・ソースが外部にある場合に発生します。

処置：ソース情報をローカル・ソースに設定します。

DESCRIPTION_IS_NOT_SET

原因：この例外は、`getDescription` ファンクションをコールして説明属性が設定されていない場合に発生します。

処置：説明属性を設定します。

INVALID_MIME_TYPE

原因：この例外は、`setMimeType` プロシージャの MIME パラメータ値が NULL の場合に発生します。

処置：MIME パラメータ値を既知の値に設定します。

AUDIO_FORMAT_IS_NULL

原因：この例外は、`getFormat` ファンクションをコールしたときにフォーマットが NULL だった場合に発生します。

処置：オーディオ・オブジェクトのフォーマットを既知のフォーマットに設定します。

NULL_INPUT_VALUE

原因：この例外は、`setFormat` プロシージャの `knownFormat` パラメータ値が `NULL` の場合に発生します。

処置：これらのパラメータを既知の値で設定します。

METHOD_NOT_SUPPORTED

原因：この例外は、コールしたメソッドがサポートされていない場合に発生します。

処置：サポートされているメソッドをコールします。

AUDIO_PLUGIN_EXCEPTION

原因：この例外は、オーディオ・プラグインで例外が発生した場合に発生します。

処置：詳細な情報は、3.3.1 項を参照してください。

H.1.2 ORDIImageExceptions 例外

次の例外は `ORDImage` オブジェクトに関連付けられています。

NULL_LOCAL_DATA

原因：この例外は、`source.localData` の値が `NULL` の場合に発生します。

処置：`source.localData` を `empty_blob()` で初期化します。

NULL_PROPERTIES_DESCRIPTION

原因：この例外は、`SetProperties` の説明パラメータが設定されていない場合に発生します。

処置：外部イメージを使う場合は、説明属性を設定します。説明属性を設定しない場合は、説明パラメータを渡すことができません。

NULL_DESTINATION

原因：この例外は、宛先イメージの値が `NULL` の場合に発生します。

処置：初期化された宛先イメージを渡します。

DATA_NOT_LOCAL

原因：この例外は、ソース情報がローカルに設定されていない場合に発生します。

処置：ソース属性情報をローカル・イメージ・ソースにリセットします。`import()` または `importFrom()` メソッドをコールして、データをローカル `BLOB` にインポートします。

NULL_CONTENT

原因：この例外は、`ORDImgB` または `ORDImgF` イメージの `content` 属性の値が `NULL` の場合に発生します。

処置：`content` 属性を初期化します。

NULL_SOURCE

原因：この例外は、ソース・イメージの値が NULL の場合に発生します。

処置：初期化されたソース・イメージを渡します。

H.1.3 ORDVideoExceptions 例外

次の例外は、ORDVideo オブジェクトに関連付けられています。

UNSUPPORTED_DATA_SOURCE

原因：この例外は、データ・ソースが外部にある場合に発生します。

処置：ソース情報をローカル・ソースに設定します。

DESCRIPTION_IS_NOT_SET

原因：この例外は、getDescription ファクションをコールしたときに、説明属性が設定されていなかった場合に発生します。

処置：説明属性を設定します。

INVALID_MIME_TYPE

原因：この例外は、setMimeType プロシージャの MIME パラメータ値が NULL の場合に発生します。

処置：MIME パラメータ値を既知の値に設定します。

VIDEO_FORMAT_IS_NULL

原因：この例外は、getFormat ファクションをコールしたときに、フォーマットが NULL だった場合に発生します。

処置：ビデオ・オブジェクトのフォーマットを既知のフォーマットに設定します。

NULL_INPUT_VALUE

原因：この例外は、setFrameSize プロシージャの knownWidth または knownHeight パラメータ値が NULL の場合に発生します。

処置：これらのパラメータを既知の値で設定します。

METHOD_NOT_SUPPORTED

原因：この例外は、コールしたメソッドがサポートされていない場合に発生します。

処置：サポートされているメソッドをコールします。

VIDEO_PLUGIN_EXCEPTION

原因：この例外は、ビデオ・プラグインで例外が発生する場合に発生します。

処置：詳細な情報は、5.3.1 項を参照してください。

H.1.4 ORDSourceExceptions 例外

次の例外は、ORDSource オブジェクトに関連付けられています。

INCOMPLETE_SOURCE_INFORMATION

原因：この例外は、srcType が NULL でソース情報が不完全な場合、または srcType が NULL でデータが BLOB にローカルに格納されていない場合に発生します。

処置：ソース情報をチェックして、必要に応じて srcType、srcLocation または srcName 属性を設定します。

INCOMPLETE_SOURCE_LOCATION

原因：この例外は、srcLocation の値が NULL の場合に発生します。

処置：ソース位置をチェックして srcLocation 属性を設定します。

INCOMPLETE_SOURCE_NAME

原因：この例外は、srcName の値が NULL の場合に発生します。

処置：ソース名をチェックして srcName 属性を設定します。

EMPTY_SOURCE

原因：この例外は、ローカル・ソースが NULL の場合に発生します。この例外は、オーディオ形式プラグインから発生することがあります。

処置：初期化されたソースを渡します。

NULL_SOURCE

原因：この例外は、ローカル・ソースの値が NULL の場合に発生します。

処置：初期化されたソースを渡します。

UNSUPPORTED_DATA_SOURCE

原因：この例外は、データ・ソースが外部にある場合に発生します。

処置：ソース情報をローカル・ソースに設定します。

INVALID_SOURCE_TYPE

原因：この例外は、getBFile メソッドで 'FILE' 以外のソース・タイプが検出された場合に発生します。

処置：ソース・タイプが 'FILE' になっていることを確認します。

SOURCE_PLUGIN_EXCEPTION

原因：この例外は、ソース・プラグインで例外が発生する場合に発生します。

処置：詳細な情報は、6.3.1 項、6.3.2 項および 6.3.3 項を参照してください。

H.2 ORDAudio エラー・メッセージ

AUD-00001 Audio Cartridge 環境を初期化できませんでした。

原因：オーディオ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00511 string

原因：オーディオ・データへのアクセス中にエラーが検出されました。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00599 内部エラーが発生しました。

原因：内部エラーが発生しました。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00601 オーディオのコピー中にメモリー不足が発生しました。

原因：オーディオのコピー中に、オペレーティング・システムのプロセス・メモリーを使い切っています。

処置：データベース管理者またはオペレーティング・システム管理者に依頼して、プロセスのメモリー割当てを増やしてください。

AUD-00602 オーディオ・データにアクセスできません。

原因：オーディオ・データの読み込みまたは書き込み中にエラーが発生しました。

処置：システム管理者に連絡してください。

AUD-00603 オーディオ・データのソースにアクセスできません。

原因：SOURCE 属性が無効です。

処置：確実にソースの SOURCE 属性をオーディオ・データとともに移入してください。

AUD-00604 オーディオ・データの宛先にアクセスできません。

原因：宛先の SOURCE 属性が無効です。

処置：宛先ソースの SOURCE 属性を有効にしてください。

AUD-00606 オーディオ・データにアクセスできません。

原因：無効なオーディオにアクセスしようとしてしました。

処置：オーディオの SOURCE 属性をオーディオ・データとともに移入してください。

AUD-00607 オーディオの宛先に書き込みできません。

原因：宛先オーディオの SOURCE 属性が無効です。

処置：宛先オーディオの SOURCE 属性を初期化された BLOB ロケータおよび十分な表領域とともに移入してください。

AUD-00702 オーディオ処理環境の初期化ができません。

原因：オーディオ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00703 オーディオ・データの読み込みができません。

原因：ソースにオーディオ・データがありません。

処置：オーディオ・データの移入方法は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

AUD-00704 オーディオ・データの読み込みができません。

原因：ソースにオーディオ・データがありません。

処置：オーディオ・データの移入方法は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

AUD-00705 入力形式がサポートされていないか壊れています。

原因：これは内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00706 出力形式がサポートされていないか壊れています。

原因：これは内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00707 オーディオ・データへアクセスできません。

原因：オーディオ・データの読み込みまたは書き込み中にエラーが発生しました。

処置：システム管理者に連絡してください。

AUD-00710 オーディオの宛先に書き込みできません。

原因：オーディオの宛先が無効です。

処置：オーディオの宛先が有効で、十分な記憶領域があることを確認してください。

AUD-00711 オーディオの宛先のプロパティを設定できません。

原因：これは内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00712 オーディオの宛先に書き込みできません。

原因：オーディオの宛先が無効です。

処置：オーディオの宛先が有効で、十分な記憶領域があることを確認してください。

AUD-00713 オーディオ宛先のフォーマットがサポートされていません。

原因：サポートされていない形式にオーディオを変換する要求になっています。

処置：サポートされている形式については、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

AUD-00714 内部エラーが発生しました。

原因：これは内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

AUD-00715 現在のソースに格納されているオーディオのオープンに失敗しました。

原因：ソース・オブジェクトに格納されているオーディオを読み取るためにオープンできません。

処置：オーディオ・ソースのアクセス権限が、コール者に正しく定義されていることを確認してください。

H.3 ORDImage エラー・メッセージ

IMG-00001 Oracle8i *interMedia* 環境を初期化できません。

原因：イメージ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00502 位取りの値が無効です。

原因：イメージ・プロセス機能用のパラメータ解析中に無効な位取りの値が検出されました。

処置：有効な位取りの値を使用して、誤りのある文を修正してください。正しい使用方法およびイメージ処理コマンド文字列についての構文の詳細は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00505 CUT 四角形を指定する値の数が足りません。

原因：指定している四角形に、不正な値が使用されました。

処置：左下および右上の頂点に、4 つの整数値を正しく使用してください。

IMG-00506 CUT 四角形を指定する値の数が余分にあります。

原因：指定している四角形に、不正な値が使用されました。

処置：左下および右上の頂点に、4 つの整数値を正しく使用してください。

IMG-00510 *string*

原因：イメージ・プロセス機能用のパラメータ解析中に構文エラーが検出されました。

処置：有効なパラメータ値を使って、誤りのある文を修正してください。正しい使用方法およびイメージ処理コマンド文字列についての構文の詳細は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00511 string

原因: イメージ・データへのアクセス中にエラーが検出されました。

処置: オラクル社カスタマ・サポートに連絡してください。

IMG-00512 適合しないスケール変更パラメータが複数見つかりました。

原因: イメージ・プロセス・コマンド文字列に、適合しないスケール変更パラメータが複数見つかりました。プロセス・コマンド文字列で利用できる XSCALE および YSCALE を除き、スケール関数は相互に排他的であり、結合できません。

処置: スケール関数を削除して 1 つのみにしてください (XSCALE および YSCALE の場合は 2 つ)。

IMG-00513 スケール変更操作で値が欠落しています。

原因: イメージ・ディメンションの指定に、不正な値を使用しています。FixedScale および MaxScale の場合、要求されたイメージの X および Y ディメンションに 2 つの整数値が必要です。

処置: FixedScale および MaxScale に 2 つの値を使用してください。

IMG-00514 スケール変更操作で余分な値があります。

原因: イメージ・ディメンションの指定に、不正な値を使用しています。FixedScale および MaxScale の場合、要求されたイメージの X および Y ディメンションに 2 つの整数値が必要です。

処置: FixedScale および MaxScale に 2 つの値を使用してください。

IMG-00515 入力チャネル数が正しくありません。

原因: 入力チャネルの指定に、不正な値を使用しています。入力チャネルには、灰色または赤、緑および青のチャネル割当てに対して、1 つまたは 3 つのチャネル番号が必要です。

処置: 1 つまたは 3 つの値を使用して、入力チャネルを指定してください。

IMG-00516 デフォルトのチャネルが範囲外です。

原因: デフォルトのチャネル選択の指定に、不正な値を使用しています。

処置: バンド数以下で 0 を超えるチャネル数を使用してください。

IMG-00517 パラメータ文字列に高さまたは幅が指定されていません。

原因: setProperties パラメータ文字列に高さまたは幅が指定されていません。

処置: 高さおよび幅の両方を指定してください。

IMG-00518 高さまたは幅の値が無効です。

原因: 高さまたは幅は正の整数である必要があります。

処置: 高さおよび幅の両方を正の整数として指定してください。

IMG-00519 パラメータの組合せが無効です。

原因: CCITTG3 または CCITTG4 が `compressionFormat` として使用されている場合、`setProperties` パラメータ文字列に、高さ、幅、データ・オフセットおよび `userString` 以外のパラメータを指定できません。

処置: `compressionFormat` が CCITTG3 または CCITTG4 のいずれかである場合、高さおよび幅のみを指定してください。データ・オフセットおよび `userString` もオプションで指定できます。

IMG-00531 イメージ処理コマンドが空または NULL です。

原因: 空または NULL のイメージ処理コマンドが、イメージ・プロセス機能に渡されました。

処置: 正しい使用方法およびイメージ処理コマンド文字列についての詳細は、『*Oracle8i interMedia Audio, Image, Video ユーザーズ・ガイド*およびリファレンス』を参照してください。

IMG-00599 内部エラー

原因: 内部エラーが発生しました。

処置: オラクル社カスタマ・サポートに連絡してください。

IMG-00601 イメージのコピー中にメモリー不足になりました。

原因: イメージのコピー中に、オペレーティング・システムのプロセス・メモリーを使い切っています。

処置: データベース管理者またはオペレーティング・システム管理者に依頼して、プロセスのメモリー割当てを増やしてください。

IMG-00602 イメージ・データにアクセスできません。

原因: イメージ・データの読み込みまたは書き込み中にエラーが発生しました。

処置: システム管理者に連絡してください。

IMG-00603 ソースのイメージ・データにアクセスできません。

原因: ソース・イメージの `SOURCE` 属性が無効です。

処置: ソース・イメージの `SOURCE` 属性をイメージ・データとともに移入してください。

IMG-00604 宛先のイメージ・データにアクセスできません。

原因: 宛先イメージの `SOURCE` 属性が無効です。

処置: 宛先イメージの `SOURCE` 属性をイメージ・データとともに移入してください。

IMG-00606 イメージ・データにアクセスできません。

原因: 無効なイメージにアクセスしようとした。

処置: イメージの `SOURCE` 属性をイメージ・データとともに移入してください。

IMG-00607 宛先イメージへの書き込みができません。

原因：宛先イメージの SOURCE 属性が無効です。

処置：宛先イメージの SOURCE 属性が正しく初期化され、表領域が十分にあることを確認してください。

IMG-00609 BFILE に格納されているイメージの読み込みができません。

原因：BFILE に格納されているイメージを読み込むためにオープンできません。

処置：イメージ・ファイルのアクセス権限およびイメージ・ファイルのあるディレクトリに読み込みアクセス許可があるかどうかを確認してください。

IMG-00701 空のイメージのプロパティは設定できません。

原因：イメージ・オブジェクトにデータがありません。

処置：イメージ・データのイメージ・オブジェクトへの移入方法は、『Oracle8i *interMedia Audio, Image, Video* ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00702 イメージ処理環境を初期化できません。

原因：イメージ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00703 イメージ・データの読み込みができません。

原因：イメージ・オブジェクトにイメージ・データがありません。

処置：イメージ・データのイメージ・オブジェクトへの移入方法は、『Oracle8i *interMedia Audio, Image, Video* ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00704 イメージ・データの読み込みができません。

原因：イメージ・オブジェクトにイメージ・データがありません。

処置：イメージ・データのイメージ・オブジェクトへの移入方法は、『Oracle8i *interMedia Audio, Image, Video* ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00705 サポートされていないか、または破損している入力形式です。

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00706 サポートされていないか、または破損している出力形式です。

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00707 イメージ・データにアクセスできません。

原因：イメージ・データの読み込みまたは書き込み中にエラーが発生しました。

処置：システム管理者に連絡してください。

IMG-00710 宛先イメージへの書き込みができません。

原因：宛先イメージが無効です。

処置：宛先イメージの SOURCE 属性が正しく初期化され、表領域が十分にあることを確認してください。

IMG-00711 宛先イメージのプロパティを設定できません。

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00712 宛先イメージへの書き込みができません。

原因：宛先イメージが無効です。

処置：宛先イメージの SOURCE 属性が正しく初期化され、表領域が十分にあることを確認してください。宛先イメージを含む行がロックされていることを確認してください（一時 BLOB には適用されません）。

IMG-00713 サポートされていない宛先イメージ・フォーマットです。

原因：サポートされていない形式にイメージを変換する要求になっています。

処置：サポートされる形式については、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00714 内部エラー

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

IMG-00715 BFILE に格納されているイメージをオープンできません。

原因：BFILE に格納されているイメージを読み込むためにオープンできません。

処置：イメージ・ファイルのアクセス権限およびイメージ・ファイルがあるディレクトリに読み込み許可があるかどうかを確認してください。

IMG-00716 ソースのイメージ・フォーマットはプロセス・オプションをサポートしません。

原因：ソース・イメージ・フォーマットでサポートされていないプロセス・オプションを適用するよう要求されました。

処置：サポートされているプロセス・オプションの詳細は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00717 宛先のイメージ・フォーマットはプロセス・オプションをサポートしません。

原因：宛先イメージ・フォーマットでサポートされていないプロセス・オプションを適用するように要求されました。

処置：サポートされているプロセス・オプションの詳細は、『Oracle8i interMedia Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

IMG-00718 同じ一時 LOB をソースと宛先両方に使用することはできません。

原因：ソースおよび宛先の両方に指定されている同じ一時 LOB で processCopy にコピーされました。

処置：パラメータの宛先に異なる LOB を指定してください。

H.4 ORDVideo エラー・メッセージ

VID-00001 Video Cartridge 環境の初期化に失敗しました。

原因：ビデオ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00511 string

原因：ビデオ・データへのアクセス中にエラーが検出されました。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00599 内部エラーが発生しました。

原因：内部エラーが発生しました。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00601 ビデオのコピー中にメモリー不足が発生しました。

原因：ビデオのコピー中に、オペレーティング・システムのプロセス・メモリーを使い切っています。

処置：データベース管理者またはオペレーティング・システム管理者に依頼して、プロセスのメモリー割当てを増やしてください。

VID-00602 ビデオ・データへアクセスできません。

原因：ビデオ・データの読み込みまたは書き込み中にエラーが発生しました。

処置：システム管理者に連絡してください。

VID-00603 ビデオ・データのソースへアクセスできません。

原因：ソース属性が無効です。

処置：ソースの SOURCE 属性をビデオ・データとともに移入してください。

VID-00604 ビデオ・データの宛先へアクセスできません。

原因：宛先ソースの SOURCE 属性が無効です。

処置：宛先ソースの SOURCE 属性が有効であることを確認してください。

VID-00606 イメージ・データへアクセスできません。

原因：無効なイメージにアクセスしようとしてしました。

処置：イメージの SOURCE 属性をイメージ・データとともに移入してください。

VID-00607 イメージの宛先へ書込みできません。

原因：宛先イメージの SOURCE 属性が無効です。

処置：宛先イメージの SOURCE 属性を初期化された BLOB ロケータおよび十分な表領域とともに移入してください。

VID-00702 ビデオ処理環境の初期化に失敗しました。

原因：ビデオ処理の外部プロシージャの初期化プロセスに失敗しました。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00703 ビデオ・データの読み込みができません。

原因：ソースにビデオ・データがありません。

処置：ビデオ・データの移入方法は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

VID-00704 ビデオ・データの読み込みができません。

原因：ソースにビデオ・データがありません。

処置：ビデオ・データの移入方法は、『Oracle8i *interMedia* Audio, Image, Video ユーザーズ・ガイドおよびリファレンス』を参照してください。

VID-00705 入力形式がサポートされていないか壊れています。

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00706 出力形式がサポートされていないか壊れています。

原因：これは、内部エラーです。

処置：オラクル社カスタマ・サポートに連絡してください。

VID-00707 ビデオ・データへアクセスできません。

原因：ビデオ・データの読み込みまたは書込み中にエラーが発生しました。

処置：システム管理者に連絡してください。

VID-00710 ビデオの宛先へ書込みができません。

原因：ビデオの宛先が無効です。

処置: ビデオの宛先が有効で、十分な記憶領域があることを確認してください。

VID-00711 ビデオの宛先のプロパティを設定できません。

原因: これは、内部エラーです。

処置: オラクル社カスタマ・サポートに連絡してください。

VID-00712 ビデオの宛先へ書き込みができません。

原因: ビデオの宛先が無効です。

処置: ビデオの宛先が有効で、十分な記憶領域があることを確認してください。

VID-00713 ビデオ宛先の形式がサポートされていません。

原因: サポートされていない形式にビデオを変換する要求になっています。

処置: サポートされている形式については、『*Oracle8i interMedia Audio, Image, Video ユーザーズ・ガイド*および*リファレンス*』を参照してください。

VID-00714 内部エラーが発生しました。

原因: これは、内部エラーです。

処置: オラクル社カスタマ・サポートに連絡してください。

VID-00715 現在のソースに格納されているビデオをオープンできません。

原因: ソース・オブジェクトに格納されているビデオを、読み取るためにオープンできません。

処置: ビデオ・ソースのアクセス権限が、コール者に正しく定義されていることを確認してください。

使用されなくなったイメージ・オブジェクト型およびメソッド

8.1.4 以前のリリースでは、Oracle8 Image Cartridge に一連のイメージ・オブジェクト型およびメソッドが記述されていましたが、これらはリリース 8.1.5 では使用されていません。使用されなくなった機能は出荷時にソフトウェア・キットに付属し、現行リリースでは利用できます。ただし、将来のリリースで拡張されることはなく、使われなくなったり、削除される可能性もあります。ここでは、リリース 8.1.4 以前のイメージ・アプリケーションからリリース 8.1.5 の *interMedia* イメージ・アプリケーションへの移行に役立つ参照情報として、これらの使用されなくなった機能について説明します。この付録では、これらの使用されなくなったイメージ・オブジェクト型およびメソッドについて説明します。

使用されなくなった機能は 2 種類のオブジェクト型で構成されています。

- ORDIImgB では、Oracle8 外部バイナリ・ラージ・オブジェクト (BLOB) に格納されたイメージがサポートされます。
- ORDIImgF では、Oracle8 外部バイナリ・ファイル (BFILE) に格納されたイメージがサポートされます。

cartridge (リリース 8.1.4) には次のファンクションおよびプロシージャが含まれています。

表 I-1 ファンクションおよびプロシージャ

ファンクションまたはプロシージャ	説明
checkProperties	格納されているイメージ属性が実際のイメージと一致することを確認します。
copyContent	別の BLOB にイメージのコピーを作成します。(BLOB でのみ利用でき、BFILE では利用できません。)
deleteContent	イメージ・コンテンツを削除します。
getMimeType	イメージの MIME タイプを戻します。
getCompressionFormat	イメージに使用されている圧縮タイプを戻します。

表 I-1 ファンクションおよびプロシージャ (続き)

ファンクションまたはプロシージャ	説明
getContent	イメージを含む BLOB または BFILE を戻します。
getContentType	イメージの形式を戻します。
getContentLength	イメージのサイズをバイト単位で戻します。
getFormat	イメージのファイル・タイプを戻します。
getHeight	イメージの高さをピクセル単位で戻します。
getWidth	イメージの幅をピクセル単位で戻します。
process	BLOB 上のインプレース・イメージ処理を実行します。
processCopy	別の BLOB にイメージをコピーしている間にイメージ処理を実行します。
setProperties	イメージの属性フィールドを設定します (ORDImB または ORDImF データ型)。

ORDImB オブジェクトにイメージを格納またはコピーする場合、最初に表に空の BLOB を作成する必要があります。この章の例では、3 つのイメージを格納するために、次の表 **ordimgtab** が作成済みであることを想定しています。3 つの空の行は次のように作成します。

```
create table ordimgtab(col1 number, col2 ORDSYS.ORDImB);
insert into ordimgtab values
  (1, ORDSYS.ORDImB(empty_blob(), NULL, NULL, NULL, NULL, NULL, NULL));
insert into ordimgtab values
  (2, ORDSYS.ORDImB(empty_blob(), NULL, NULL, NULL, NULL, NULL, NULL));
insert into ordimgtab values
  (3, ORDSYS.ORDImB(empty_blob(), NULL, NULL, NULL, NULL, NULL, NULL));
commit;
```

ORDImF オブジェクトにイメージを格納する場合、型を初期化指定子とともに移入する必要があります。

```
create table ordimgtab(col1 number, col2 ORDSYS.ORDImF);
insert into ordimgtab values
  (1, ORDSYS.ORDImF(bfilename
    ('ORDIMGDIR', 'jdoe.gif'), NULL, NULL,
    NULL, NULL, NULL, NULL));
```

'bfilename' 引数 'ORDIMGDIR' は、ファイル・システム・ディレクトリを参照するディレクトリを表しています。bfilename コンストラクタのディレクトリ名は大文字でなければならないことに注意してください。次のシーケンスにより **ORDIMGDIR** という名前のディレクトリが作成されます。

```
connect internal
create or replace directory ORDIMGDIR as '<myimage directory>';
grant read on directory ORDIMGDIR to <user-or-role> with grant option;
```

ORDImgB オブジェクト型

ORDImgB オブジェクト型は、基本の記憶域および Oracle データベース内のイメージ・データの検索で使います。このオブジェクト型の定義は次のとおりです。

```
CREATE TYPE ORDImgB AS OBJECT
(
  -- TYPE ATTRIBUTES
  content          BLOB,
  height           INTEGER,
  width            INTEGER,
  contentLength    INTEGER,
  fileFormat       VARCHAR2(64),
  contentFormat    VARCHAR2(64),
  compressionFormat VARCHAR2(64),
  --- METHOD DECLARATION
  MEMBER PROCEDURE copyContent(dest IN OUT NOCOPY BLOB),
  MEMBER PROCEDURE setProperties(SELF IN OUT ORDImgB),
  MEMBER PROCEDURE process      (SELF      IN OUT ORDImgB,
                                command IN   VARCHAR2)
  MEMBER PROCEDURE processCopy(command IN   VARCHAR2,
                                dest   IN OUT NOCOPY BLOB)

  MEMBER FUNCTION getMimeType RETURN VARCHAR2,
  MEMBER FUNCTION getContent RETURN BLOB,
  MEMBER FUNCTION getContentLength RETURN INTEGER,
  MEMBER PROCEDURE deleteContent (SELF IN OUT ORDImgB),
  MEMBER FUNCTION getHeight RETURN INTEGER,
  MEMBER FUNCTION getWidth RETURN INTEGER,
  MEMBER FUNCTION getFileFormat RETURN VARCHAR2,
  MEMBER FUNCTION getContentFormat RETURN VARCHAR2,
  MEMBER FUNCTION getCompressionFormat RETURN VARCHAR2,
  MEMBER FUNCTION checkProperties RETURN BOOLEAN
);
```

パラメータは次のとおりです。

- **content** : 格納されているイメージ
- **height** : ピクセル単位のイメージの高さ
- **width** : ピクセル単位のイメージの幅
- **contentLength** : ディスク上のイメージ・ファイルのバイト単位のサイズ
- **fileFormat** : イメージのファイル・タイプ (TIFF、JFIF など)
- **contentFormat** : イメージの種類 (モノクロ、8 ビット・グレースケールなど)
- **compressionFormat** : イメージの圧縮タイプ

PL/SQL では、データは DBMS LOB パッケージとともに移動されます。クライアントから、OCI LOB コールを使用してデータを移動します。ORDImgB オブジェクト型では、データ移動用にピース単位のルーチンは提供されません。

ORDImgF オブジェクト型

ORDImgF オブジェクト型は、外部ファイルに格納するイメージ・データの検索に使用します。BFILE イメージは読み込み専用で、オブジェクト型に定義されるメンバー・プロシージャにもそれが反映されていると想定されています。

```
CREATE TYPE ORDImgF AS OBJECT
(
  -- TYPE ATTRIBUTES
  content          BFILE,
  height           INTEGER,
  width            INTEGER,
  contentLength    INTEGER,
  fileFormat       VARCHAR2(64),
  contentFormat    VARCHAR2(64),
  compressionFormat VARCHAR2(64),

  -- METHOD DECLARATION
  MEMBER PROCEDURE copyContent(dest IN OUT NOCOPY BLOB),
  MEMBER PROCEDURE setProperties(SELF IN OUT ORDImgF),
  MEMBER PROCEDURE processCopy(command IN VARCHAR2,
                                dest IN OUT NOCOPY BLOB),

  MEMBER FUNCTION getMimeType RETURN VARCHAR2,
  MEMBER FUNCTION getContent RETURN BFILE,
  MEMBER FUNCTION getContentLength RETURN INTEGER,
  MEMBER FUNCTION getHeight RETURN INTEGER,
  MEMBER FUNCTION getWidth RETURN INTEGER,
  MEMBER FUNCTION getFileFormat RETURN VARCHAR2,
  MEMBER FUNCTION getContentFormat RETURN VARCHAR2,
  MEMBER FUNCTION getCompressionFormat RETURN VARCHAR2,
  MEMBER FUNCTION checkProperties RETURN BOOLEAN
);
```

パラメータは次のとおりです。

- **content** : 格納されたイメージ
- **height** : ピクセル単位のイメージの高さ
- **width** : ピクセル単位のイメージの幅
- **contentLength** : ディスク上のイメージ・ファイルのバイト単位のサイズ
- **fileFormat** : イメージのファイル・タイプ (TIFF、JFIF など)
- **contentFormat** : イメージの種類 (モノクロ、8ビット・グレースケールなど)

- `compressionFormat` : イメージの圧縮タイプ

checkProperties メソッド

形式

```
checkProperties RETURN BOOLEAN;
```

説明

イメージ・オブジェクトの属性に格納されたプロパティが、**BLOB** または **BFILE** に格納されたイメージのプロパティと一致することを確認します。このメソッドは外部イメージには使用しません。

パラメータ

なし

戻り値

BOOLEAN

使用方法

このメソッドを使用して、イメージ属性が実際のイメージと一致することを確認します。

例

イメージ属性をチェックします。

```
imgb1          ORDSYS.ORDImB;  
properties_match BOOLEAN;  
  
...  
properties_match := imgb1.checkProperties;
```

copyContent メソッド

形式

```
copyContent (dest IN OUT NOCOPY BLOB);
```

説明

イメージを変更せずにコピーします。

パラメータ

dest
新しいイメージの宛先

使用方法

このメソッドでは、指定した **BLOB** にイメージ・データをコピーします。

例

タイプ **image1** のイメージのコピーを **myblob** という **BLOB** に作成します。

```
image1.copyContent (myblob);
```

deleteContent メソッド

形式

```
deleteContent;
```

説明

イメージのコンテンツを削除します。

パラメータ

なし

使用方法

このメソッドを使うと、イメージ **BLOB** のコンテンツを削除できます。このメソッドは **BFILES** ではなく、**BLOBS** でのみ実行できます。

例

```
イメージを削除します。  
  
imgb1  ORDSYS.ORDImgB;  
  
...  
imgb1.deleteContent;
```

getCompressionFormat メソッド

形式

```
getCompressionFormat RETURN VARCHAR2;
```

説明

イメージの圧縮タイプを戻します。このメソッドは `compressionFormat` 属性の値を戻す単純なアクセサ・メソッドで、実際には LOB を読み込みません。

パラメータ

なし

戻り値

VARCHAR2

使用方法

`compressionFormat` 属性に直接アクセスすると `ORDImB` または `ORDImF` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージの圧縮タイプを取得します。

```
imgb1          ORDSYS.ORDImB;  
compressionFormat VARCHAR2(64);  
  
...  
compressionFormat := imgb1.getCompressionFormat;
```

getContent メソッド

形式

```
getContent RETURN BLOB;  
getContent RETURN BFILE;
```

説明

イメージを含む **BLOB** または **BFILE** の **LOB** ロケータを戻します。このメソッドはコンテンツ属性の値を戻す単純なアクセサ・メソッドで、実際には **LOB** を読み込みません。

パラメータ

なし

戻り値

イメージの格納方法に応じて **BLOB** または **BFILE** を戻します。

使用方法

コンテンツ属性に直接アクセスすると **ORDImgB** または **ORDImgF** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにメソッドを使います。

例

イメージの **LOB** ロケータを取得します。

```
imgb1  ORDSYS.ORDImgB;  
content BLOB;  
...  
content := imgb1.getContent;
```

getContentTypeFormat メソッド

形式

```
getContentTypeFormat RETURN VARCHAR2;
```

説明

イメージの種類を返します (モノクロまたは 8 ビット・グレースケールなど)。このメソッドは `contentTypeFormat` 属性の値を返す単純なアクセサ・メソッドで、実際には LOB を読み込みません。

パラメータ

なし

戻り値

VARCHAR2

使用方法

`contentTypeFormat` 属性に直接アクセスすると `ORDImageB` または `ORDImageF` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージの種類を取得します。

```
imgb1          ORDSYS.ORDImageB;  
contentTypeFormat VARCHAR2 (64);  
  
...  
contentTypeFormat := imgb1.getContentTypeFormat;
```

getLength メソッド

形式

```
getLength RETURN INTEGER;
```

説明

ディスク上イメージのサイズをバイト単位で返します。このメソッドは `length` 属性の値を返す単純なアクセサ・メソッドで、実際には **LOB** を読み込みません。

パラメータ

なし

戻り値

INTEGER

使用方法

`length` 属性に直接アクセスすると **ORDImage** または **ORDImageF** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージのコンテンツの長さを取得します。

```
imgb1          ORDSYS.ORDImage;  
length INTEGER;  
  
...  
length := imgb1.getLength;
```

getFileFormat メソッド

形式

```
getFileFormat RETURN VARCHAR2
```

説明

イメージのファイル・タイプ (TIFF または JFIF など) を戻します。このメソッドは `fileFormat` 属性の値を戻す単純なアクセサ・メソッドで、実際には LOB を読み込みません。

パラメータ

なし

戻り値

VARCHAR2

使用方法

`fileFormat` 属性に直接アクセスすると `ORDImGB` または `ORDImGF` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージのファイル・タイプを取得します。

```
imgb1      ORDSYS.ORDImGB;  
fileFormat VARCHAR2(64);  
  
...  
fileFormat := imgb1.getFileFormat;
```

getHeight メソッド

形式

```
getHeight RETURN INTEGER;
```

説明

ピクセル単位のイメージの高さを戻します。このメソッドは高さ属性の値を戻す単純なアクセス・メソッドで、実際には **LOB** を読み込みません。

パラメータ

なし

戻り値

INTEGER

使用方法

高さ属性に直接アクセスすると **ORDImB** または **ORDImF** オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージの高さを取得します。

```
imgb1 ORDSYS.ORDImB;  
height INTEGER;  
  
...  
height := imgb1.getHeight;
```

getMimeType メソッド

形式

```
getMimeType RETURN VARCHAR2;
```

説明

イメージの MIME (Multipurpose Internet Mail Extension) タイプ (image/jpeg、image/tiff など) を戻します。このメソッドは、イメージの `fileFormat` に基づいて MIME タイプを戻します。サポートされる各ファイル形式に対応付けられた MIME タイプについては、付録 B を参照してください。

パラメータ

なし

戻り値

VARCHAR2

使用方法

イメージの MIME タイプを取得するためにこのメソッドを使います。Web ブラウザにより、イメージ・コンテンツとともに MIME タイプが要求されます。MIME タイプによって、イメージ・コンテンツの解釈方法を Web ブラウザに指定します。

ファイル形式が認識されない場合、このメソッドでは `image/binary` が戻されます。

例

MIME タイプのイメージを取得します。

```
imgb1      ORDSYS.ORDImgB;  
mimeType   VARCHAR2(64);  
  
...  
mimeType := imgb1.getMimeType;
```

getWidth メソッド

形式

```
getWidth RETURN INTEGER;
```

説明

イメージの幅をピクセル単位で戻します。このメソッドは幅属性の値を戻す単純なアクセサ・メソッドで、実際には LOB を読み込みません。

パラメータ

なし

戻り値

INTEGER

使用方法

幅属性に直接アクセスすると `ORDImgB` または `ORDImgF` オブジェクトの内部表現を変更してしまう可能性があるため、それを避けるためにこのメソッドを使います。

例

イメージの幅を取得します。

```
imgb1  ORDSYS.ORDImgB;  
width  INTEGER;  
  
...  
width := imgb1.getWidth;
```


process メソッド

形式

```
process (command IN VARCHAR2 );
```

説明

BLOB 上で 1 つ以上のイメージ処理技術を実行し、イメージをそのイメージ自体に上書きします。

パラメータ

command
イメージに対して実行するイメージの処理変更のリスト

使用方法

表 I-2 に示す 1 つ以上のイメージ属性を変更できます。表 I-3 は、ロー・ピクセルおよび外部イメージに対してのみ実行できる追加の変更を示しています。サポートされているすべての形式の組合せの詳細は、付録 B を参照してください。各オペレータにの詳細は、付録 C を参照してください。

表 I-2 イメージ処理オペレータ

オペレータ名	使用方法	値
compressionFormat	圧縮タイプ / 形式	JPEG、SUNRLE、BMPRLE、TARGARLE、LZW、LZWHDIFF、FAX3、FAX4、HUFFMAN3、Packbits、GIFLZW
compressionQuality	圧縮の質	MAXCOMPRATIO、MAXINTEGRITY、LOWCOMP、MEDCOMP、HIGHCOMP
contentFormat	イメージの種類 / ピクセル / データ形式	MONOCHROME、8 BITGRAYSCALE、8 BITGREYSCALE、8BITLUT、24BITRGB
cut	カットまたはクロップするウィンドウ (起点 .x 起点 .y 幅 高さ)	(INTEGER INTEGER INTEGER INTEGER) 最大値は 65535
fileFormat	イメージのファイル形式	BMPF、CAL5、GIF5、JFIF、PICT、RAS5、RPIX、TGAF、TIFF
fixedScale	特定のサイズにピクセル単位でサイズ変更 (幅、高さ)	(INTEGER INTEGER)

表 I-2 イメージ処理オペレータ (続き)

オペレータ名	使用方法	値
maxScale	アスペクト比を保って、ピクセル単位でサイズ変更 (maxWidth、maxHeight)	(INTEGER INTEGER)
scale	スケール要素 (0.5 または 2.0 など)	<FLOAT> 正の数
xScale	X 軸スケール要素 (デフォルトは 1)	<FLOAT> 正の数
yScale	Y 軸スケール要素 (デフォルトは 1)	<FLOAT> 正の数

表 I-3 ロー・ピクセル・イメージおよび外部イメージの追加イメージ処理オペレータ

オペレータ名	使用方法	値
ChannelOrder	イメージ内の赤、緑および青チャネル (バンド) の相対的な位置を示します。	RGB (デフォルト)、RBG、GRB、GBR、BRG、BGR
InputChannels	マルチバンド・イメージでは、1 つの整数 (グレースケール) または赤 (第 1)、緑 (第 2) および青 (第 3) を示す 3 つの整数を指定します。このパラメータは宛先ではなくソース・イメージに影響することに注意してください。	INTEGER または INTEGER INTEGER INTEGER
Interleave	イメージ内のバンド・レイアウトを制御します。 ピクセル単位のバンド・インターリーブ 行単位のバンド・インターリーブ バンド順序	BIP (デフォルト)、BIL、BSQ
PixelOrder	NORMAL の場合、イメージの左端のピクセルが先頭になります。	NORMAL (デフォルト)、REVERSE
ScanlineOrder	NORMAL の場合、イメージの上端の走査線が先頭になります。	NORMAL (デフォルト)、INVERSE

注意: 浮動小数点を含む値を指定する場合、値の前後に二重引用符 ("") を使う必要があります。二重引用符を使わないと、値が正しく渡されず、間違った結果が取得されます。

例

image1 のファイル形式を GIF に変更します。

```
image1.process('fileFormat=GIF');
```

image1 をより低い品質の JPEG 圧縮に変更し、イメージの長さを X 軸にそって倍にします。

```
image1.process('compressionFormat=JPEG, compressionQuality=LOWCOMP, xScale="2.0"');  
image1.setproperties;
```

一方の軸の長さを変更しても (**xScale=2.0** など) 他方の軸の長さは影響を受けず、イメージがゆがむ結果になることに注意してください。また、1 回の操作では、**xScale** と **yScale** パラメータのみを組み合わせることができます。それ以外のスケール演算子の組み合わせではエラーが発生します。

maxScale および **fixedScale** 演算子はさまざまなサイズのオリジナルからサムネイル・イメージを作成する場合に特に便利です。次の行は元のアスペクト比を保って **32 × 32** ピクセルのサムネイル・イメージを作成します。

```
image1.process(maxScale="32 32");
```

processCopy メソッド

形式

```
processCopy (command IN VARCHAR2,  
             dest IN OUT NOCOPY BLOB);
```

説明

イメージ BLOB または BFILE を処理して別の BLOB に移します。

パラメータ

command

新しいコピーのイメージに対して行うイメージ処理変更のリスト

dest

新しいイメージの宛先

使用方法

表 I-2 の「イメージ処理オペレータ」および表 I-3 の「ロー・ピクセル・イメージおよび外部イメージの追加イメージ処理オペレータ」を参照してください。

テンポラリ LOB を使う場合、ソースおよび宛先の両方に同じテンポラリ LOB を指定することはできません。

例

イメージをコピーして、ファイル形式、宛先イメージの圧縮形式およびデータ形式を変更します。

```
create or replace procedure copyit is  
  imgB1      ORDSYS.ORDImgB;  
  imgB4      ORDSYS.ORDImgB;  
  mycommand  VARCHAR2(400);  
begin  
  select col2 into imgB1 from ordimgtab where col1 = 1;  
  select col2 into imgB4 from ordimgtab where col1 = 4 for update;  
  command:= 'fileFormat=tiff compressionFormat = packbits  
            contentFormat = 8bitlut';  
  imgB1.processcopy(mycommand,imgB4.content);  
  imgB4.setproperties;  
  update ordimgtab set col2 = imgB4 where col1 = 4;  
end;
```

setProperties メソッド

形式

```
setProperties( );
```

説明

イメージの特性（BLOB または BFILE）を適切な属性フィールドに書き込みます。

パラメータ

なし

使用方法

システム固有形式のイメージのコピー、格納または処理を行った後にこのプロシージャをコールすると、新しいコンテンツの現行の特性が設定されます。

このプロシージャを使用して、イメージに関する次の情報を設定します。

- ピクセル単位の高さ
- ピクセル単位の幅
- ディスク上のイメージのバイト単位のデータ・サイズ
- ファイル・タイプ（TIFF、JFIF など）
- イメージの種類（モノクロ、8 ビット・グレースケールなど）
- 圧縮タイプ（JPEG、LZW など）

例

イメージを選択し、setProperties メソッドを使用して属性を設定します。

```
imgB1 ORDSYS.imgB;  
.  
.  
.  
select col2 into imgB1 from ordimgtab where col1 = 1 for update;  
imgB1.setProperties;  
dbms_output.put_line('image width = '|| imgB1.width );  
dbms_output.put_line('image height = '|| imgB1.height );  
dbms_output.put_line('image size = '|| imgB1.contentLength );  
dbms_output.put_line('image file type = '|| imgB1.fileFormat );  
dbms_output.put_line('image type = '|| imgB1.contentType );
```

setProperty メソッド

```
dbms_output.put_line('image compression = ' || imgB1.compressionFormat );
```

出力例：

```
image width = 360  
image height = 490  
image size = 59650  
image file type = JFIF  
image type = 24BITRGB  
image compression = JPEG
```

外部イメージの setProperties() メソッド

形式

```
SetProperties(description IN VARCHAR2);
```

説明

外部イメージの特性（BLOB または BFILE）を適切な属性フィールドに書き込むことができます。

パラメータ

description

外部イメージに設定するイメージ特性を指定します。

使用方法

外部イメージのコピー、格納または処理を行った後にこのメソッドをコールして、新しいイメージ・コンテンツの特性を設定します。付録 B で説明するシステム固有のイメージの種類とは異なり、外部イメージにはファイルのビットを解釈する方法についての情報がないか *interMedia image* で情報を認識できません。この場合、情報を明示的に設定する必要があります。

外部イメージには表 I-4 に示す次のイメージ特性を設定できます。

表 I-4 ヘッダーなしファイルのイメージ特性

フィールド	データ型	説明
CompressionFormat	STRING	値は CCITG3、CCITG4 または NONE（デフォルト）にする必要があります。
DataOffset	INTEGER	オフセットでは、イメージに <i>interMedia image</i> では解釈されないヘッダーを設定できます。オフセットにはヘッダーになる可能性のあるものを避けて設定します。値は LOB 長よりも小さい正の整数でなければなりません。デフォルトは 0（ゼロ）です。
DefaultChannelSelection	INTEGER	マルチバンド・イメージでは、1 つの整数（グレースケール）または赤（第 1）、緑（第 2）および青（第 3）を示す 3 つの整数を指定します。
Height	INTEGER	ピクセル単位のイメージの高さ。値は正の整数でなければなりません。デフォルトはないため、値は必ず指定する必要があります。

表 I-4 ヘッダーなしファイルのイメージ特性 (続き)

フィールド	データ型	説明
Interleaving	STRING	イメージ内のバンド・レイアウト。有効なスタイルは次のとおりです。 <ul style="list-style-type: none"> ■ BIP (デフォルト) ピクセル単位のバンド・インターリーブ ■ BIL 行単位のバンド・インターリーブ ■ BSQ バンド順序
NumberOfBands	INTEGER	イメージのカラーバンドを指定する値は 255 よりも小さい正の整数でなければなりません。デフォルトは 3 です。
PixelOrder	STRING	NORMAL (デフォルト) の場合、ファイルの左端のピクセルが先頭になります。REVERSE の場合、右端のピクセルが先頭になります。
ScanlineOrder	STRING	NORMAL (デフォルト) の場合、ファイルの上端の走査線が先頭になります。INVERSE の場合、下端の走査線が先頭になります。
UserString	STRING	4 文字で説明する文字列。使用する場合、文字列は fileFormat フィールドに格納されファイル形式 ("OTHER : ") に追加されます。デフォルトはブランクです。
Width	INTEGER	ピクセル単位のイメージの幅。値は正の整数でなければなりません。デフォルトはないため、値は必ず指定する必要があります。

setProperties() に入力された値は既存の ORDDimgB および ORDDimgF オブジェクト属性に書き込まれます。fileFormat は "OTHER : " に設定され、入力されるユーザー文字も含まれます。

例

イメージの種類を選択し、setProperties メソッドを使用して属性を設定します。

```
imgB1 ORDSYS.ORDimgB;
select col2 into imgB1 from ordimgtab where col1 = 1 for update;
imgB1.setProperties('width=380 height=407 dataOffset=128 bandOrder=BIL
userString="LSAT"');
```

用語集

この用語集の項目の一部はフリー・オンラインの Dictionary of Computing から引用しています。URL は次のとおりです。 <http://wombat.doc.ic.ac.uk/foldoc/index.html>

AAF

アドバンスド・オーサリング・フォーマット (Advanced Authoring Format)。Windows オペレーティング・システムで AVI ファイル形式を継承するために、Microsoft Corporation により定義されたプロフェッショナル・マルチメディア・オーサリング・フォーマットです。AAF はオープンな業界標準フォーマットで、デジタル製品ツールとコンテンツ作成アプリケーションの間のメディア交換を行うことができます。

A-Law

非一様量子化対数圧縮の欧州および世界標準（北米および日本を除きます）。A-Law は、欧州のテレフォニー標準として使われます。

AC-3 (MPEG-2 オーディオ) /MP3

サラウンド音声オーディオ

ADPCM

適応的差分 PCM (Adaptive Differential Pulse Code Modulation)。サンプル間の相違のみを記録し、相違の大小に応じてコーディング・スケールを動的に調整する圧縮技術です。ADPCM の導入は簡単ですが、ホワイト・ノイズが多く発生します。

ADT

抽象データ型 (Abstract data type)。オブジェクト型も参照してください。

AIFF

オーディオ交換ファイル形式 (Audio Interchange File Format)。オーディオ情報および楽器の高品質なサンプル情報を格納するために Apple Computer, Inc. によって開発されたファイル形式です。

AIFF-C

圧縮機能付きオーディオ交換ファイル形式 (Audio Interchange File Format with Compression)。オーディオ情報および楽器情報の高品質な圧縮サンプルを格納するために Apple Computer, Inc. によって開発されたファイル形式です。

ASF

拡張文字列フォーマット (Advanced Streaming Format)。Windows オペレーティング・システムで AVI ファイル形式を継承するために、Microsoft Corporation により定義されたマルチメディア文字列フォーマットです。ASF はオープンな業界標準フォーマットで、特に文字列メディア配布用に調整されています。

AU

UNIX プラットフォーム、特に Sun および NeXT 計算プラットフォーム共通のオーディオ・ファイル形式

AVI

オーディオ・ビデオ・インターリーブ (Audio Video Interleaved)。Microsoft Corporation により定義されたファイル・コンテナ・フォーマット。AVI ファイルには、通常オーディオおよびビデオの順序が含まれています。オーディオおよびビデオ・データには、Intel Indeo、Iterated Systems ClearVideo (フラクタル) または Motion JPEG など、いくつかの形式があります。詳細については、AAF および ASF も参照してください。

BFILE (バイナリ・ファイル)

オペレーティング・システム・ファイル内のデータベースの外部に格納されるバイナリ・データで値が構成されるラージ・オブジェクト。ファイル自体はデータベースに格納されず、ファイルへのポインタがデータベースに格納されます。BFILE は、データベースの外部にあるため、読み込み専用です。

BLOB

バイナリ・ラージ・オブジェクト (Binary large object)。ラージ・オブジェクトは、領域を最適化し効率的にアクセスできるようにデータベース表領域に格納されます。実際に行に格納されるのは、オブジェクトへのポインタだけです。

B フレーム

双方向予測コード・フレーム。MPEG ビデオでは、B フレームは、最も近い I フレームまたは P フレームの 2 つのピクチャ (過去と未来から 1 つずつ) からのピクチャの変化の程度を示すデータを使用してエンコードされます。

CLOB

Oracle8 データベースに定義されたデータベース・キャラクタ・セットに対応する、シングルバイト固定幅文字データで値が構成されるラージ・オブジェクト

CODEC

圧縮展開プログラム (Compressor DECompressor)。マルチメディア・データの圧縮および展開を実行するソフトウェア・モジュールです。代表的なオーディオ CODEC またはエンコーディング・タイプには、ADPCM、Mu-Law、Two's Complement、AC-3 などがあります。代表的なビデオ CODEC またはエンコーディング・タイプには、MPEG-1、MPEG-2、Intel Indeo、Iterated Systems ClearVideo (フラクタル)、Radius Cinepak、Motion JPEG などがあります。代表的な画像 CODEC には、JPEG、Packbits、HUFFMAN3 などがあります。

HTTP ヘッダー

Web ページを要求するときに、URL とともにブラウザから送信する情報の本体。ブラウザ・タイプ、MIME タイプなどの情報が含まれます。

IDL

Interface Definition Language (インタフェース定義言語)

I フレーム

イントラコード・フレーム。MPEG ビデオでは、I フレームは、過去の履歴を一切使用せずに静止イメージとしてコーディングされたフレームです。

I フレームの規則

適切なレート制御パフォーマンスを実現するには、ビデオ・ストリームをとおして I フレームが一定間隔で表示される必要があります。たとえば、米国および日本の場合、順序内の各 I フレーム・ペア間に、通常 11 フレーム (B フレームおよび P フレーム) が存在します。IBBPBBPBBPBBIBBPBBP.... これは、I フレーム (先頭) が 0.4 秒ごとに 1 つ以上必要であるなど、ランダム・アクセス要件に基づいています。MPEG エンコーディング・ソフトウェアでは、多くの場合、I フレーム周波数を設定できます。

JPEG

合同写真専門家グループ (Joint Photographic Experts Group)。元は標準イメージ圧縮アルゴリズムを設計した委員会の名前です。ファイル名拡張子は .jpg または .jpeg です。JPEG は、静的イメージ圧縮形式です。

LOB

ラージ・オブジェクト (Large object)。LOB は、大量のロー・データかつバイナリ・データを格納する際に使用します。Oracle8i *interMedia* は、LOB をサポートします。

localData

Oracle データベース内に格納された 4 ギガバイトまでのイメージ、オーディオ、またはビデオ・データを保持することのできる BLOB 属性。localData 内のすべてのデータは、Oracle のセキュリティおよびトランザクション・サポートにより保護されています。

MIME

多目的インターネット・メール拡張仕様（Multipurpose Internet Mail Extensions）。Multipurpose Internet Mail Extensions（MIME）を参照してください。

MIME タイプ

多目的インターネット・メール拡張仕様標準により定義されるファイル形式。この一意識別子は、MIME 電子メールまたは HTTP などの MIME ベース・プロトコルを介して送受信されたときに、異なるファイル・タイプを扱う場合に使われます。

Moving JPEG、Motion JPEG、M-JPEG

動画順序の各フレームに、JPEG 静止イメージ圧縮を適用する動画圧縮技術

再生時には、各 JPEG イメージの展開および表示を高速で行うことができる装置を使い、必要なピクチャ順序のフレーム・レートを実現する必要があります。

Moving JPEG を使うときには、Moving JPEG 用の標準はありません。ただし、JPEG 圧縮チップを使うことができます。このチップは、テレビのフレーム・レートおよび解像度で動作するように設計されています。

MPEG

動画専門家のグループ（Moving Picture Expert's Group）。フルモーション・ビデオとオーディオのシグナル、および再生中にこれらのシグナルを同期化するときを使う圧縮形式の開発を担当した ISO 組織です。

MPEG-1

CD-ROM 用に最適化した最初の MPEG 形式。離散コサイン変換（DCT）およびホフマン・コーディングを使用してフレーム内の空間的に冗長なデータを削除し、ブロック・ベースの動画補間予測（MCP）を使用してフレーム間の一時的に冗長になるデータを削除します。オーディオは、サブバンド・エンコーディングを使用して圧縮されます。

MPEG-2

MPEG ビデオおよびオーディオ圧縮アルゴリズムおよびファイル形式の 1 つ。最大 4.0 メガビット / 秒まで、デジタル記憶メディア用ブロードキャスト品質ビデオに対して最適化されます。ファイル拡張子は .MP2 です。

MPEG-2 は、国際標準 IS-13818 として承認されています。

MPEG-3

MPEG ビデオおよびオーディオの圧縮アルゴリズムおよびファイル形式の 1 つ。ファイル拡張子は .MP3 です。このアルゴリズムおよびファイル形式は現在存在せず、MPEG-2 に統合されました。

MPEG-4

低帯域幅ビデオ・テレフォニーに使われる MPEG ビデオおよびオーディオの、圧縮アルゴリズムおよびファイル形式の 1 つ。ファイル拡張子は .MP4 です。国際標準機構 (ISO) では、MPEG-4 仕様の統一デジタル・メディア記憶域形式の出発点として、QuickTime 3 ファイル形式を採用しています。詳細は、QuickTime 3 を参照してください。

Mu-Law または U-Law

非一様量子化対数圧縮の北米および日本標準

OCI

Oracle コール・インタフェース (Oracle Call Interface)

オブジェクト型は、抽象データ型 (ADT) と呼ばれることもあります。

Oracle Video Server

OVS を参照してください。

Oracle Web Application Server

分散されたプラットフォーム間動的アプリケーションを作成および実行できる HTTP サーバー。モジュール分散アーキテクチャを包含するフレームワークを提供します。移植可能アプリケーション、さまざまなアプリケーション・モデルまたはアプリケーション・パラダイムを作成できる、オープン API です。Oracle Web Application Server は、HTTP デモン (UNIX) またはリスナー、Web Request Broker (WRB)、オプションまたはサーバー側アプリケーションという、3 つのコンポーネントで構成されます。

ORDAnnotations

interMedia で定義されたオブジェクト型。これは、オーディオ注釈データの格納と管理に必要なすべての機能をカプセル化したデモ・プログラムの一部です。

ORDAudio

interMedia audio で定義されたオブジェクト型。オーディオ・データの格納と管理に必要なすべての機能が、カプセル化されています。

ORDImage

interMedia image で定義されたオブジェクト型。イメージ・データの格納と管理に必要なすべての機能が、カプセル化されています。

ORDSource

interMedia で定義されたオブジェクト型。ソース・データだけを認識します。*interMedia* オブジェクト型に対応した共通オブジェクト型です。The **ORDSource** オブジェクト型は、ソースからマルチメディア・データを取得するのに使用されます。ソースには、Oracle データベースにローカルに格納された BLOB、外部ファイル・システム、メディア・サーバー、FTP サーバー、および HTTP サーバーにローカルに格納された BFILES などが含まれます。

ORDVideo

interMedia video で定義されたオブジェクト型。ビデオ・データの格納と管理に必要なすべての機能が、カプセル化されています。

OVS

Oracle Video Server。デジタル・ビデオおよびオーディオの格納、管理、送受信および表示をオンデマンドで行うことができる、ネットワーク・クライアントとサーバー・コンピュータ用エンドツーエンド・ソフトウェア・ソリューションを提供する **Oracle** 製品です。クライアント・アプリケーションは、さまざまなユーザー・プラットフォームおよび企業プラットフォームで実行できます。**OVS** は、さまざまなサーバー・プラットフォームでサポートされており、多数の同時ユーザーに対する拡張性を提供します。

P フレーム

予測コード・フレーム (**Predictive-coded frames**)。MPEG ビデオでは、P フレームは、前の I フレームまたは P フレームのピクチャの変化の程度を示すデータを使用してエンコードされます。

QuickTime

Apple Computer, Inc. により定義されたファイル・コンテナ・フォーマット。フルモーション・ビデオおよびデジタル音声、アプリケーション・プログラムに統合されます。

QuickTime 3

Apple Computer, Inc. により定義された業界標準ファイル形式。**Mac OS**、**Java** および **Windows** の操作環境用のデジタル・メディアの作成および発行を行います。国際標準化機構 (ISO) では、MPEG-4 仕様の統一デジタル・メディア記憶域形式の出発点として、**QuickTime 3** ファイル形式を採用しています。このため、次世代の MPEG では、大部分の既存のハードウェア、ソフトウェアおよびデジタル・コンテンツがシームレスに動作することができます。詳細は、MPEG-4 を参照してください。

SND

音声。**UNIX** プラットフォーム、特に **Sun** および **NeXT** 計算プラットフォームで使われるオーディオ・ファイル・フォーマットです。

WAV

Waveform オーディオ。**WAV** とは、デジタル・オーディオ・ファイルを格納するために、**Microsoft Corporation** により開発された標準ファイル・コンテナ・フォーマットのことで、変換ツールを使うと、他のオペレーティング・システムのほとんどで **WAV** ファイルを再生できます。

アスペクト比

高さとの比率

圧縮

記憶領域または送信時間を少なくするデータのコーディング。圧縮されたデータは、使用前に展開する必要があります。

圧縮形式

CODEC により作成されるエンコード・オーディオ、イメージまたはビデオ・ファイルの形式。Oracle Video Server では、MPEG-1、MPEG-2、Intel Indeo、Iterated Systems ClearVideo（フラクタル）、Radius Cinepak および Motion JPEG など、いくつかの圧縮形式がサポートされます。

イメージ

グラフィック画像。ソースには、写真、図形、生成したイメージがあります。

イメージ処理

スケール変更、回転、圧縮など、イメージのプロパティを変更すること

インポート

データを外部記憶領域（外部ファイル・サーバー、メディア・サーバー、HTTP サーバー、FTP サーバーなど）から Oracle データベース内に移動すること

エクスポート

データを、Oracle データベースから、ローカル・ファイル・システムや HTTP サーバーなどの外部記憶領域にファイルとして移動すること

エンコーディング・タイプ

オーディオおよびビデオ・データのエンコードに使う特定の CODEC アルゴリズム。代表的なオーディオ CODEC には、ADPCM、Mu-Law、Two's Complement、AC-3 などがあります。代表的なビデオ CODEC には、Indeo、MPEG などがあります。オーディオおよびビデオ・データには、通常コンピュータ・プラットフォーム固有のエンコーディング・タイプを使います。

オーディオ

音声。コンピュータ（およびオーディオ・コンパクト・ディスクおよびデジタル・オーディオ・テープ）では、離散サンプル順序を格納することによって音声を処理します。元のソースの連続音声 **Waveform** は、毎秒数万回サンプリングが行われます。各回のサンプルでは、音圧 **WAVE** の強度を表しています。サンプリング周波数以外のパラメータに、各サンプルの使用ビット数を含むデジタル・エンコーディングがあります。エンコーディング方法には、線形、対数または **Mu-Law** のエンコーディングがあります。ソースには、音楽、音声録音または生成されたオーディオがあります。**mu-law** や **u-law** も参照してください。

オーディオ処理

フォーマット、エンコーディング・タイプ、チャンネル数、サンプリング・レートまたはサンプル・サイズなど、オーディオ・データ・プロパティに対する変更

オブジェクト型

データ操作に関係したデータ属性およびメソッド（ファンクションとプロシージャ）を含むデータ型

オブジェクト・リレーショナル・データベース

オブジェクト指向とリレーショナル特性の両方を備えたデータベース。オブジェクトは、定義および保管を行った後で、標準リレーショナル・メソッドを使用して取り出すことができます。

可逆圧縮

データ内のすべての情報を保持し、展開によって完全に回復できるデータ圧縮アルゴリズム
イメージに必要な記憶領域を減少させる方法の 1 つ。展開されたイメージは、元のイメージとビット単位で完全に一致します。

Unix 圧縮および GNU gzip では、可逆圧縮が実行されます。

カット

指定した四角形の内側のイメージを選択して、サブイメージを作成すること。サブイメージを新しいイメージにコピーした場合も、カットになります。サブイメージでオリジナルを上書きすると、クロップになります。

クロップ

指定した四角形の内側のイメージを選択して、四角形の外側をすべて削除すること。カットも参照してください。

コンテナ・フォーマット

エンコーダでビデオおよびオーディオを多重化する方法。たとえば、Oracle Video Server によりサポートされるコンテナ・フォーマットには、MPEG-1 システム、MPEG-2 トランスポート、Oracle ビデオ・ストリーム、ロー・キー・フレーム要件を満たすコンテナ・フォーマットなどがあります。

コンテンツ形式

ピクセルまたは色形式など、イメージ・データの説明

再生時間

オーディオまたはビデオ・データ全体の再生に要する時間

サンプリング・レート

オーディオ・データが記録されたサンプル毎秒のレート。サンプリング・レートの範囲は、5500 (Mac サンプリング・レートの 4 分の 1) ～ 4800 (デジタル・オーディオ・テープ (DAT) サンプリング・レート) です。

サンプル・サイズ (sample size)

オーディオ・データに存在するオーディオ・データのサンプル数、つまり、サンプル単位のビット数。サンプル単位のビット数は、8 (8 ビット) または 16 (16 ビット) です。

スケール変更

一方または両方のディメンションで、イメージの比率を変更すること。イメージを拡大するには、1 より大きい係数でスケールを変更します。イメージを縮小するには、0 (ゼロ) ～ 1 の係数でスケールを変更します。

属性

実社会の構造をモデルにした、組込みデータ型またはその他のユーザー定義データ型のオブジェクト型の構成要素。詳細は、『Oracle8i 概要』を参照してください。

帯域幅

送信チャネルの周波数の高低差

多目的インターネット・メール拡張仕様 (MIME)

ファイル・タイプを規定したインターネット仕様。サーバーおよびブラウザによって、ファイル・ヘッダーの MIME タイプが読み取られ、ビューアで表示するかまたはオーディオ・ファイルとして再生するかなど、ファイルの処理方法が決定されます。MIME は HTTP サーバーにより使われ、配布されるファイルのタイプを指定します。

チャネル数

フォーマットされたオーディオ・データに存在するオーディオ・チャネル数。チャネル・カウントとも呼ばれます。チャネル数の範囲は、モノラル・オーディオ・データ用の 1 から、AC3 エンコード・サラウンド音声オーディオ・データ用の 6 まであります。

デフォルト・コンストラクタ

オブジェクトの空のインスタンスを作成するメソッド

統一リソース識別子 (URI)

抽象または物理リソースの識別に使われる、位置を表す文字列の簡略表示 (URL)。URI はアドレス指定方法またはプロトコルの 1 つで、エンコード・アドレス文字列を使用してオブジェクトにアクセスするために、インターネット用に開発されました。

統一リソース・ロケータ (uniform resource locator) (URL)

URI の 1 つ。インターネット上で利用可能なリソースの位置を表す文字列の簡略表示です。これは、クライアントが Oracle Application Server へ送信するリクエストをエンコードする際に使用する、テキスト文字列フォーマットでもあります。

バイナリ・ラージ・オブジェクト

BLOB を参照してください。

非可逆圧縮

データ内の情報の表示に使われるビット数だけでなく、情報量を実際に減少させるデータ圧縮アルゴリズム。失われる情報は通常削除されます。これは、削除されるデータ（通常イメージまたはサウンド）は主観的に見てデータの品質にさほど重要ではないか、残りのデータから補間することにより適度に回復可能なためです。

イメージに必要な記憶領域を減少させる方法の 1 つ。展開されたイメージは、元のイメージとビット単位では一致しません。イメージの詳細が一部失われるか変更されますが、通常は認識されません。

MPEG および JPEG が、非可逆圧縮技術の例です。

ビデオ

文字列イメージ。ソースは、ビデオ・カメラ、録画、または生成されたアニメーションの場合があります。フルモーション・ビデオも参照してください。

ビデオ圧縮

イメージ順序の圧縮

ファイル形式

オーディオ・データのファイル形式には、AIFF、AIFF-C、AU、WAV、MIDI、RealAudio および SND などがあります。ビデオ・データのファイル形式には、QuickTime、AVI、MPEG1、および Real Video などがあります。イメージ・データのファイル形式には、TIFF、JFIF などがあります。サポートするオーディオ・ファイルおよび圧縮形式については、付録 A を参照してください。サポートするイメージ・ファイルおよび圧縮形式については、付録 B を参照してください。

フォーマット

特定のデータ構造または編成。通常、オーディオ・データの場合は AIFF-C、ビデオ・データの場合は QuickTime、AVI、MPEG1 など、業界標準のデータ構造または編成が使われます。

フレーム・サイズ

ビデオ・データの各フレームの、ピクセル単位の高さおよび幅。帯域幅要件を見積もり、指定した表示画面サイズに対して適切な表示サイズを決定するときこの情報を使います。

フレーム数

ビデオ・オブジェクト内のフレーム数

フレームの解像度

ビデオ・データのフレームの、インチ単位のピクセル数。フレームの解像度は、帯域幅要件を満たすように調整できます。フレームの解像度を低くすると、ビデオの質が低下します。

フレーム・レート

毎秒表示されるフレーム数。通常は、30 フレーム / 秒が適正レートです。フレーム・レートを低くすると帯域幅は小さくなりますが、ビデオ・データの質が低下します。フレーム・レートを高くすると、より滑らかで高品質なビデオを作成できますが、処理能力と帯域幅を大きくする必要があります。

メソッド

PL/SQL で作成されデータベースに格納されるか、C などの言語で作成され外部に格納されたファンクションまたはプロシージャで構成される、オブジェクト型の構成要素。メソッドにより、アプリケーションからデータに対して実行する特定の操作をインプリメントできます。詳細は、『Oracle8i 概要』を参照してください。

ローカル・ソース

Oracle データベース内のイメージ、オーディオ、またはビデオ・データの位置

ローカル・ファイル・システム

BFILES の格納など、ローカル・アクセスが可能なファイル・システム

ロー・キー・フレーム (RKF)

ビデオ・コンテナ・フォーマットのコンテンツが準拠する規則セット。Oracle Video Server などの一部の Video Server では、RKF の規則に準拠するコンテナ・フォーマットのコンテンツ以外は再生できません。次の RKF 規則があります。

- **状態なし** - ビデオ・フレームでピクチャを表示するデータは、そのフレーム内にすべて含まれ、過去のフレームには含まれません。
- **連続性** - フレームのデータは、すべてビデオ・ファイルに格納されます。他のデータは混在して格納されません。

ロー・ピクセル・フォーマット

単純な固定サイズのヘッダーを持つ、非圧縮イメージ・フォーマット

索引

A

AIFF-C データ形式, A-2
AIFF データ形式, A-1
appendToAnnotation() メソッド, F-8
appendToComments() メソッド, 3-60, 5-63
AU データ形式, A-2

B

BFILE, 2-15, 2-16, I-6
BLOB, I-4
BMP データ形式, B-1

C

CALS ラスター・データ形式, B-2
checkProperties() メソッド, 3-109, 4-25, 5-121
clearLocal メソッド, 3-29, 4-11, 4-35, 5-30, 6-11
close() メソッド, 6-43
closeSource() メソッド, 3-51, 5-54
compareComments() メソッド, 3-72, 5-75
copy() メソッド, 4-11
copyCommentsOut() メソッド, 3-70, 5-73
copyContent() メソッド, I-9

D

DBA
 チューニング・ヒント, 7-1
deleteAnnotation() メソッド, F-7
deleteComments() メソッド, 5-70
deleteComments メソッド, 3-67
deleteContent メソッド, 3-47, 4-47, 5-50
deleteLocalContent メソッド, 6-39

E

eraseFromComments() メソッド, 3-66, 5-69
export() メソッド, 3-41, 4-58, 5-43, 6-29

G

getAllAttributes() メソッド, 3-113, 5-125
getAnnotation() メソッド, F-5
getAnnotationCount メソッド, F-9
getAnnotationKeyByPosition() メソッド, F-12
getAnnotationObjByPosition() メソッド, F-10
getAnnotationValueByPosition() メソッド, F-14
getAttribute() メソッド, 3-111, 5-123
getAudioDuration() メソッド, 3-103
getAudioDuration メソッド, 3-102
getBFILE メソッド, 4-46
getBFile メソッド, 6-23
getBitRate() メソッド, 5-115
getBitRate メソッド, 5-114
getCommentLength() メソッド, 3-74, 5-77
getCompressionFormat() メソッド, 4-32
getCompressionType() メソッド, 3-99, 5-107
getCompressionType メソッド, 3-98, 5-106
getContentFormat() メソッド, 4-31
getContentInLob() メソッド, 3-44, 5-46
getContentInTempLob() メソッド, 6-37
getContentLength() メソッド, 3-43, 5-45, 6-32
getContentLength メソッド, 4-29
getContent メソッド, 3-46, 4-45, 5-48
getDescription メソッド, 3-19, 5-20
getEncoding() メソッド, 3-83
getEncoding メソッド, 3-82
getFileFormat() メソッド, 4-30
getFormat() メソッド, 3-79, 5-81

getFormat メソッド, 3-78, 5-80
getFrameRate() メソッド, 5-95
getFrameRate メソッド, 5-94
getFrameResolution() メソッド, 5-91
getFrameResolution メソッド, 5-90
getFrameSize() メソッド, 5-85, 5-87
getHeight() メソッド, 4-27
getLocalContent メソッド, 6-36
getMimeType() メソッド, 4-41
getMimeType メソッド, 3-23, 5-24
getNumberOfChannels() メソッド, 3-87
getNumberOfChannels メソッド, 3-85, 3-86
getNumberOfColors() メソッド, 5-111
getNumberOfColors メソッド, 5-110
getNumberOfFrames() メソッド, 5-103
getNumberOfFrames メソッド, 5-102
getSampleSize() メソッド, 3-94, 3-95
getSourceAddress() メソッド, 6-34
getSourceInformation メソッド, 6-19
getSourceLocation メソッド, 3-35, 4-52, 5-36, 6-21
getSourceName メソッド, 3-36, 4-53, 5-37, 6-22
getSourceType メソッド, 3-33, 4-51, 5-34, 6-20
getSource メソッド, 3-32, 4-50, 5-33
getUpdateTime メソッド, 3-14, 4-38, 5-15, 6-14
getVideoDuration() メソッド, 5-99
getVideoDuration メソッド, 5-98
getWidth() メソッド, 4-28
GIF データ形式, B-2

I

import() メソッド, 3-37, 4-54, 5-38, 6-25
importFrom() メソッド, 3-39, 4-56, 5-40, 6-27
interMedia の拡張
 AIFC オーディオ・フォーマット, 3-126
 AIFF オーディオ・フォーマット, 3-123
 AUFF オーディオ・フォーマット, 3-120
 AVI ビデオ・フォーマット, 5-132
 MOOV ビデオ・フォーマット, 5-135
 MPEG ビデオ・フォーマット, 5-137
 WAVE オーディオ・フォーマット, 3-128
 オーディオのデフォルト・フォーマット, 3-118
新規イメージ・オブジェクト型, 2-19
新規オーディオ・オブジェクト型, 2-10
新規オーディオ形式, 2-10, 3-131
新規データ・ソース, 2-32, 6-59
新規ビデオ・オブジェクト型, 2-31

新規ビデオ・フォーマット, 2-30, 5-139
ビデオ・デフォルト・フォーマット, 5-130
isLocal メソッド, 3-27, 4-36, 5-28, 6-12

J

JFIF データ形式, B-3

L

loadCommentsFromFile() メソッド, 3-68, 5-71
locateInComments() メソッド, 3-64, 5-67

M

migrateFromORDImgB() メソッド, 4-61
migrateFromORDImgF() メソッド, 4-63

O

open() メソッド, 6-41
openSource() メソッド, 3-49, 5-52
ORDAnnotations メソッド
 注釈属性, F-3
ORDAudio, 3-1
 参照情報, 3-3
ORDAudio オブジェクト型
 参照情報, 3-3
ORDAudio メソッド
 mimeType 属性, 3-20
 updateTime 属性, 3-13
 オーディオ属性, 3-59, 3-75
 オーディオ・データの処理, 3-115
 説明属性, 3-16
 ソース属性, 3-24
 ソース・ファイル操作メソッド, 3-48
ORDImage, 4-1
 参照情報, 4-3, 4-6
ORDImage オブジェクト型
 参照情報, 4-3
ORDImage メソッド
 updateTime 属性, 4-10
ORDImgB オブジェクト型, I-4
ORDImgF オブジェクト型, I-6
ORDPLUGINS.ORDX_, 6-59
ORDPLUGINS.ORDX_AIFC_AUDIO パッケージ,
 3-126

ORDPLUGINS.ORDX_AIFF_AUDIO パッケージ,
3-123
ORDPLUGINS.ORDX_AUFF_AUDIO パッケージ,
3-120
ORDPLUGINS.ORDX_AVI_VIDEO パッケージ, 5-132
ORDPLUGINS.ORDX_DEFAULT_VIDEO パッケージ,
5-130
ORDPLUGINS.ORDX_FILE_SOURCE パッケージ,
6-55
ORDPLUGINS.ORDX_HTTP_SOURCE パッケージ,
6-57
ORDPLUGINS.ORDX_MOOV_VIDEO パッケージ,
5-135
ORDPLUGINS.ORDX_MPEG_VIDEO パッケージ,
5-137
ORDPLUGINS.ORDX_WAVE_AUDIO パッケージ,
3-128
ORDSource, 6-1
 参照情報, 6-3
ORDSource メソッド
 localData、srcType、srcLocation、srcName 属性,
 6-16
 updateTime 属性, 6-13
 インポートおよびエクスポート操作, 6-24
 処理コマンド, 6-52
 ソース・コンテンツ・オペレーション, 6-31
 ソース・ファイル操作メソッド, 6-40
 読み込み / 書き込み操作, 6-47
 ローカル属性, 6-9
ORDVideo, 5-1
 参照情報, 5-3
ORDVideo メソッド
 mimeType 属性, 5-21
 updateTime 属性, 5-14
 コメント属性, 5-62
 説明属性, 5-17
 ソース属性, 5-25
 ソース・ファイル操作メソッド, 5-51
 ビデオ属性, 5-78
 ビデオ・データの処理, 5-127
ORDX_DEFAULT_AUDIO パッケージ, 3-118

P

PCX データ形式, B-3
PICT データ形式, B-4
process() メソッド, 4-14, I-19

processAudioCommand() メソッド, 3-116
processCommand() メソッド, 6-53
processCopy() メソッド, 4-17, I-22
processSourceCommand() メソッド, 3-25, 5-26
processVideoCommand() メソッド, 5-128

R

read() メソッド, 6-48
readFromComments() メソッド, 3-63, 5-66
readFromSource() メソッド, 3-55, 5-58
replaceAnnotation() メソッド, F-8

S

setAnnotation() メソッド, F-4
setAudioDuration() メソッド, 3-101
setBitRate() メソッド, 5-113
setCompressionType() メソッド, 3-97, 5-105
setDescription() メソッド, 3-17, 5-18
setEncoding() メソッド, 3-81
setFormat() メソッド, 3-76, 5-79
setFrameRate() メソッド, 5-93
setFrameResolution() メソッド, 5-89
setFrameSize() メソッド, 5-83
setKnownAttributes() メソッド, 3-105, 5-117
setLocal メソッド, 3-28, 4-34, 5-29, 6-10
setMimeType() メソッド, 3-21, 4-42, 5-22
setNumberOfColors() メソッド, 5-109
setNumberOfFrames() メソッド, 5-101
setProperties()
 参照情報, I-25
setProperties() メソッド, 3-107, 5-119, I-23
setProperties メソッド, 4-20
setSampleSize() メソッド, 3-93
setSamplingRate() メソッド, 3-89, 3-90
setSource() メソッド, 3-30, 4-48, 5-31
setSourceInformation() メソッド, 6-17
setUpdateTime() メソッド, 3-15, 4-39, 5-16, 6-15
setVideoDuration() メソッド, 5-97
Sun ラスター・データ形式, B-5

T

Targa データ形式, B-5
TIFF データ形式, B-6
trimComments() メソッド, 3-65, 5-68

trimSource() メソッド, 3-53, 5-56
trim メソッド, 6-45

W

WAV データ形式, A-3
write() メソッド, 6-50
writeToComments() メソッド, 3-62, 5-65
writeToSource() メソッド, 3-57, 5-60

あ

圧縮
形式, A-1, B-1
圧縮イメージ, I-19

い

一時的な変換, 2-19
イメージのカット, I-19
イメージのクロップ, I-19
イメージのコピー, 2-18
イメージのサイズ変更, I-20
イメージの挿入, 2-14
イメージの追加, 2-13
イメージの変換, 2-18

お

オブジェクト型, I-4, I-6
ORDAudio, 3-3
ORDImage, 4-3
オブジェクト・ビュー, 2-11, 2-21, 2-31
オブジェクト・リレーショナル・テクノロジー, 1-8

か

外部イメージの setProperties() メソッド, 4-22
空の BLOB, I-2
関連ドキュメント, xxii

き

行の移入, 2-14
行の問合せ, 2-17

け

形式
圧縮, A-1, B-1
ファイル, A-1, B-1

さ

サポートしているイメージ・フォーマット, B-1
サムネイル・イメージ, 4-16, I-20, I-21
参照情報, 3-1, 4-1, 5-1, 6-1, F-1
サンプル・プログラム, E-1, I-1

し

取得
テーブルからビデオ・データを, 2-30

せ

設定
プロパティ, I-25

は

パッケージ
OEDPLUGINS.ORDX_DEFAULT_VIDEO, 5-130
ORDPLUGINS.ORDX_, 6-59
ORDPLUGINS.ORDX_AIFC_AUDIO, 3-126
ORDPLUGINS.ORDX_AIFF_AUDIO, 3-123
ORDPLUGINS.ORDX_AUFF_AUDIO, 3-120
ORDPLUGINS.ORDX_AVI_VIDEO, 5-132
ORDPLUGINS.ORDX_FILE_SOURCE, 6-55
ORDPLUGINS.ORDX_HTTP_SOURCE, 6-57
ORDPLUGINS.ORDX_MOOV_VIDEO, 5-135
ORDPLUGINS.ORDX_MPEG_VIDEO, 5-137
ORDPLUGINS.ORDX_WAVE_AUDIO, 3-128
ORDX_DEFAULT_AUDIO, 3-118
パッケージまたは PL/SQL プラグイン, 3-118, 5-130, 6-55

ふ

ファイル形式, A-1, B-1
プロパティ
設定, I-25

め

メソッド, 3-8, 4-6, 5-9, 6-7, F-3

appendToAnnotation(), F-8
appendToComments(), 3-60, 5-63
checkProperties(), 3-109, 4-25, 5-121
clearLocal, 3-29, 4-11, 4-35, 5-30, 6-11
close(), 6-43
closeSource(), 3-51, 5-54
compareComments(), 3-72, 5-75
copy(), 4-11
copyCommentsOut(), 3-70, 5-73
deleteAnnotation(), F-7
deleteComments, 3-67
deleteComments(), 5-70
deleteContent, 3-47, 4-47, 5-50
deleteLocalContent, 6-39
eraseFromComments(), 3-66, 5-69
export(), 3-41, 4-58, 5-43, 6-29
getAllAttributes(), 3-113, 5-125
getAnnotation(), F-5
getAnnotationCount, F-9
getAnnotationKeyByPosition(), F-12
getAnnotationObjByPosition(), F-10
getAnnotationValueByPosition(), F-14
getAttribute(), 3-111, 5-123
getAudioDuration, 3-102
getAudioDuration(), 3-103
getBFILE, 4-46
getBFile, 6-23
getBitRate, 5-114
getBitRate(), 5-115
getCommentLength(), 3-74, 5-77
getCompressionFormat(), 4-32
getCompressionType, 3-98, 5-106
getCompressionType(), 3-99, 5-107
getContent, 3-46, 4-45, 5-48
getContentFormat(), 4-31
getContentInLob(), 3-44, 5-46
getContentInTempLob(), 6-37
getContentLength, 4-29
getContentLength(), 3-43, 5-45, 6-32
getDescription, 3-19, 5-20
getEncoding, 3-82
getEncoding(), 3-83
getFileFormat(), 4-30
getFormat, 3-78, 5-80

getFormat(), 3-79, 5-81
getFrameRate, 5-94
getFrameRate(), 5-95
getFrameResolution, 5-90
getFrameResolution(), 5-91
getFrameSize(), 5-85, 5-87
getHeight(), 4-27
getLocalContent, 6-36
getMimeType, 3-23, 5-24
getMimeType(), 4-41
getNumberOfChannels, 3-85, 3-86
getNumberOfChannels(), 3-87
getNumberOfColors, 5-110
getNumberOfColors(), 5-111
getNumberOfFrames, 5-102
getNumberOfFrames(), 5-103
getSampleSize(), 3-94, 3-95
getSource, 3-32, 4-50, 5-33
getSourceAddress(), 6-34
getSourceInformation, 6-19
getSourceLocation, 3-35, 4-52, 5-36, 6-21
getSourceName, 3-36, 4-53, 5-37, 6-22
getSourceType, 3-33, 4-51, 5-34, 6-20
getUpdateTime, 3-14, 4-38, 5-15, 6-14
getVideoDuration, 5-98
getVideoDuration(), 5-99
getWidth(), 4-28
import(), 3-37, 4-54, 5-38, 6-25
importFrom(), 3-39, 4-56, 5-40, 6-27
isLocal, 3-27, 4-36, 5-28, 6-12
loadCommentsFromFile(), 3-68, 5-71
locateInComments(), 3-64, 5-67
migrateFromORDImgB(), 4-61
migrateFromORDImgF(), 4-63
open(), 6-41
openSource(), 3-49, 5-52
process(), 4-14
processAudioCommand(), 3-116
processCommand(), 6-53
processCopy(), 4-17
processSourceCommand(), 3-25, 5-26
processVideoCommand(), 5-128
read(), 6-48
readFromComments(), 3-63, 5-66
readFromSource(), 3-55, 5-58
replaceAnnotation(), F-8
setAnnotation(), F-4

setAudioDuration(), 3-101
setBitRate(), 5-113
setCompressionType(), 3-97, 5-105
setDescription(), 3-17, 5-18
setEncoding(), 3-81
setFormat(), 3-76, 5-79
setFrameRate(), 5-93
setFrameResolution(), 5-89
setFrameSize(), 5-83
setKnownAttributes(), 3-105, 5-117
setLocal, 3-28, 4-34, 5-29, 6-10
setMimeType(), 3-21, 4-42, 5-22
setNumberOfColors(), 5-109
setNumberOfFrames(), 5-101
setProperties, 4-20
setProperties(), 3-107, 5-119
setSampleSize(), 3-93
setSamplingRate(), 3-89, 3-90
setSource(), 3-30, 4-48, 5-31
setSourceInformation(), 6-17
setUpdateTime(), 3-15, 4-39, 5-16, 6-15
setVideoDuration(), 5-97
trim, 6-45
trimComments(), 3-65, 5-68
trimSource(), 3-53, 5-56
write(), 6-50
writeToComments(), 3-62, 5-65
writeToSource(), 3-57, 5-60
外部イメージの setProperties(), 4-22
メッセージ、エラー、例外, H-1

よ

よく寄せられる質問 (FAQ), G-1

れ

例

ビデオ・データの取得 (単純な読み込み), 2-30

例外およびエラー・メッセージ, H-1

ろ

ロー・ピクセル・データ形式, B-4

ロール・バック, 2-19