

Oracle8i

interMedia Text 移行ガイド

リリース 8.1

2000 年 2 月

部品番号 : J00967-01

ORACLE®

Oracle8i *interMedia* Text 移行ガイド, リリース 8.1

部品番号 : J00967-01

原本名 : Oracle8i *interMedia* Text Migration, Release 2 (8.1.6)

原本部品番号 : A77061-01

原本著者 : Colin McGregor

原本協力者 : Shamim Alpha, Steve Buxton, Chung-Ho Chen, Yun Cheng, Paul Dixon, Mohammad Faisal, Elena Huang, Garret Kaminaga, Jacqueline Kud, Bryn Llewellyn, Wesley Lin, Kavi Mahesh, Yasuhiro Matsuda, Gerda Shank, Steve Yang

Copyright © 1996, 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）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。

当プログラムのリバース・エンジニアリング等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 Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, 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 52.227-19, Commercial Computer Software - Restricted Rights (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	vii
対象読者	viii
前提条件	viii
関連マニュアル	viii
このマニュアルの構成	viii
このマニュアルの表記規則	ix
 1 新機能および変更の概要	
用語	1-2
全般	1-2
8.1 の用語	1-2
8.1 の新機能および拡張	1-3
Oracle との統合	1-3
新しいデフォルト・システム	1-3
フィルタ処理	1-3
拡張知識ベース	1-4
階層問合せフィードバック	1-4
ユーザー・データストア	1-4
ストップリストの拡張	1-5
セクション検索の拡張	1-5
代替スペル	1-6
8.1.6 の新機能および拡張	1-7
データストアの改善	1-7
フィルタの改善	1-7
セクション検索の改善	1-7

レクサーの改善	1-8
ワードリストおよび記憶域の改善	1-8
ドキュメント・サービスの改善	1-8
シソーラスの改善	1-8
索引付けおよび問合せの改善	1-9
8.1 で変更された機能	1-10
SQL	1-11
変更	1-11
新しい SQL コマンド	1-11
演算子	1-12
廃止された演算子	1-12
変更された演算子	1-12
新しい演算子	1-12
索引プリファレンス・オブジェクト	1-13
索引プリファレンス・オブジェクトの変更	1-13
新しい索引オブジェクト	1-13
PL/SQL	1-14
PL/SQL の変更	1-14
新しい PL/SQL	1-14
実行可能ファイル	1-15
変更	1-15
新しい実行可能ファイル	1-15
ビュー	1-16

2 アプリケーションの移行

移行パス	2-2
Oracle8 ユーザー	2-2
Oracle7 ユーザー	2-2
移行計画	2-3
索引の移行	2-3
アプリケーション・コードの移行	2-4
データの移行：Oracle8i へのアップグレード	2-5
移行ステップ	2-6
ディスク領域要件	2-6
ハードウェアの考慮点	2-6

ステップ	2-7
------------	-----

3 管理

サーバー	3-2
8.1 より前	3-2
8.1	3-2
サーバーのステータスの表示	3-2
ロールおよびユーザー	3-3
8.1 より前	3-3
8.1	3-3
ユーザーへのロールおよび権限の付与	3-4
ドキュメントのロード	3-5
8.1 より前	3-5
8.1	3-5
キューおよびパイプ	3-6
8.1 より前	3-6
8.1	3-6
1 段階問合せの使用可能化	3-7
8.1 より前	3-7
8.1	3-7
管理ツール	3-8
8.1 より前	3-8
8.1	3-8

4 索引プリファレンス

プリファレンス・オブジェクト	4-2
索引プリファレンス・オブジェクトの変更	4-2
新しい索引オブジェクト	4-5
システム定義プリファレンス	4-6
索引移行スクリプトの使用	4-7
ユーザー定義プリファレンスの変換	4-7
システム定義プリファレンスの変換	4-8
索引作成の警告	4-10

5 索引付け

テキスト索引の概要	5-2
マージされたワードおよびテーマ索引（英語のみ）	5-2
複数の索引付きの列	5-2
ビューの索引付け	5-2
索引作成の手順	5-3
プリファレンスの作成	5-5
索引の作成	5-6
8.1 より前	5-6
8.1	5-6
プリファレンスの削除	5-7
8.1 より前	5-7
8.1	5-7
例	5-7
索引の削除	5-8
8.1 より前	5-8
8.1	5-8
失敗した索引の再開	5-9
8.1 より前	5-9
8.1	5-9
索引の再構築	5-10
例	5-10
索引の最適化	5-11
8.1 より前	5-11
8.1.5	5-11
8.1.6	5-11
索引の更新 - バックグラウンド DML	5-12
索引の更新 - バッチ DML	5-13
8.1 より前	5-13
8.1.5	5-13
8.1.6	5-14
ストップリストおよびストップワード	5-14
8.1 より前	5-14
8.1	5-14
ドキュメント・セクション	5-16
8.1 より前	5-16

8.1	5-16
8.1.6 でのドキュメント・セクションの改善	5-17

6 問合せ

テキスト問合せの概要	6-2
テキスト問合せ式	6-2
テキスト問合せ方法	6-2
テキスト問合せ	6-3
8.1 より前の方法	6-3
8.1 の方法	6-4
カーソル問合せ	6-5
8.1 より前の方法	6-5
8.1 の方法	6-6
構造化テキスト問合せ	6-7
8.1 より前の方法	6-7
8.1 の方法	6-8
テーマ問合せ（英語のみ）	6-9
8.1 より前の方法	6-9
8.1 の方法	6-9
コンポジット・テキストキーの問合せ	6-10
8.1 より前の方法	6-10
8.1 の方法	6-10
MAX 演算子および FIRST/NEXT 演算子	6-11
8.1 より前の方法	6-11
8.1 の方法	6-12
PL/SQL 演算子	6-14
8.1 より前の方法	6-14
8.1 の方法	6-14
ヒット数のカウント	6-15
8.1 より前の方法	6-15
8.1 の方法	6-16
ストアド・クエリー式	6-17
8.1 より前の方法	6-17
8.1 の方法	6-19
問合せ実行計画	6-20

実行計画情報の取得	6-20
-----------------	------

7 ドキュメントの表示方法

ハイライト表示	7-2
8.1 より前の方法	7-2
新しい 8.1 のソリューション	7-3
8.1.6 の機能	7-4
テーマのリスト、要旨およびテーマ・サマリーの取得	7-5
8.1 より前の方法	7-6
新しい 8.1 のソリューション	7-8
8.1.6 の機能	7-10

A PL/SQL の変更

CTX_ADM パッケージ	A-2
CTX_DML パッケージ	A-3
CTX_DDL パッケージ	A-4
CTX_INFO パッケージ	A-6
CTX_LING パッケージ	A-7
CTX_QUERY パッケージ	A-8
CTX_SVC パッケージ	A-10
CTX_THES パッケージ	A-11

索引

はじめに

このマニュアルでは、ConText Cartridge 2.x アプリケーションの *interMedia Text 8.1* アプリケーションへの移行について説明します。

対象読者

このマニュアルは、*interMedia Text* アプリケーション開発者およびシステム管理者を対象としています。

前提条件

このマニュアルは、ユーザーが Oracle リレーショナル・データベース管理システム (RDBMS)、SQL、SQL*Plus および PL/SQL の経験者であることを前提とします。詳細は、ご使用のハードウェアおよびソフトウェアのドキュメントを参照してください。

Oracle RDBMS や関連 Tools に慣れていない場合は、『Oracle8i 概要』の第 1 章「Oracle Server の基礎知識」をお読みください。Oracle ドキュメント全体で使用される概念および用語が広く紹介されています。

関連マニュアル

interMedia Text の詳細は、次のマニュアルを参照してください。

- 『Oracle8i *interMedia Text* リファレンス』

Oracle8i の詳細は、次のマニュアルを参照してください。

- 『Oracle8i 概要』
- 『Oracle8i 管理者ガイド』
- 『Oracle8i ユーティリティ・ガイド』
- 『Oracle8i パフォーマンスのための設計およびチューニング』
- 『Oracle8i SQL リファレンス』
- 『Oracle8i リファレンス・マニュアル』
- 『Oracle8i アプリケーション開発者ガイド 基礎編』

PL/SQL の詳細は、次のマニュアルを参照してください。

- 『Oracle8i PL/SQL ユーザーズ・ガイドおよびリファレンス』

このマニュアルの構成

目次を参照してください。

このマニュアルの表記規則

このマニュアルでは、次の表記規則を使用します。

表記	意味
大文字	大文字は、Oracle コマンド、標準データベース・オブジェクトや定数、および Oracle PL/SQL の標準プロシージャを示します。
イタリック体	イタリック体は、CONTAINS 問合せの問合せ語句を示します。
固定幅フォント	固定幅フォントは、SQL*Plus コマンドの例および PL/SQL コードの例を示します。コマンドまたはコードは、記載されているとおり正確に入力してください。

新機能および変更の概要

この章では、Oracle8i *interMedia* Text リリース 8.1（以前は ConText として知られていた製品）の新機能、拡張および変更の概要について説明します。

この章の内容は次のとおりです。

- [用語](#)
- [8.1 の新機能および拡張](#)
- [8.1.6 の新機能および拡張](#)
- [8.1 で変更された機能](#)
- [SQL](#)
- [演算子](#)
- [索引プリファレンス・オブジェクト](#)
- [PL/SQL](#)
- [実行可能ファイル](#)
- [ビュー](#)

用語

このマニュアルでは、次の用語を使用します。

全般

Oracle8i *interMedia* Text

以前の Oracle ConText Cartridge 製品。

8.1 より前の ConText または ConText 2.x

Oracle ConText Cartridge リリース 2.x。

移行

この語句は、8.1 より前の ConText アプリケーションから *interMedia* Text 8.1 アプリケーションへの移動を意味します。

古い機能とは、8.1 より前の ConText または ConText 2.x の機能です。新機能とは、Oracle8i *interMedia* Text リリース 8.1 の機能です。

参照： 移行プロセスの詳細は、[第 2 章「アプリケーションの移行」](#)を参照してください。

8.1 の用語

テキスト索引

CONTEXT 型のドメイン・インデックス。

テキスト問合せ

CONTAINS 演算子を使用してテキスト索引を問い合わせる SELECT 文。

ワード問合せ

ワードまたは句と完全に一致するテキスト問合せ。

ABOUT 問合せ

ABOUT 演算子を使用して概念を検索するテキスト問合せ。このタイプの問合せは、以前はテーマ問合せでした。

8.1 の新機能および拡張

この項では、Oracle8i *interMedia Text* リリース 8.1 の新機能を簡単に説明します。これらの新機能は、*interMedia Text* リリース 8.1.5 および 8.1.6 に共通です。

Oracle との統合

Oracle8i *interMedia Text* リリース 8.1 の索引は、Oracle に対して拡張可能な（ドメイン）索引として作成されます。その結果、ユーザーは標準 SQL を使用してテキスト索引を作成し、テキスト問合せを発行します。

統合のメリットは、その他に次のものがあります。

- *interMedia Text* ドメイン・インデックスは名前で識別されるため、ポリシーが不要です。
- ワード情報（テーマ情報を含む場合もある）を含む単一の索引です。
- 索引の改名が可能です。
- 問合せパフォーマンスが向上します。
- 拡張問合せオプティマイザが使用可能になります。

参照： 索引の改名および拡張可能オプティマイザの使用の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

新しいデフォルト・システム

優れた新しいデフォルト・システムによって、索引付けのためにユーザー独自のカスタム・プリファレンスを明示的に作成および設定しなくても、すぐにテキスト索引を作成できます。

たとえば、*interMedia Text* は、インストール時にデフォルトの索引付けオプションを確立し、レクサーやストップリストなどの言語固有のプリファレンスを、データベース設定で指定した言語に自動的に設定します。また、Oracle はテキスト列形式を検出し、これらの列に含まれる整形済のテキストを自動的にフィルタ処理します。

参照： デフォルト・システムの詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

フィルタ処理

Oracle8i *interMedia Text* は、Inso Corporation のフィルタ処理テクノロジーを使用します。このフィルタ処理テクノロジーによって、システムはほとんどのドキュメント形式を自動的に検出およびフィルタ処理できます。この単一のフィルタ処理テクノロジーを使用して、システムは単一書式列または複合書式列を索引付けできます。

また、独自のユーザー・フィルタを使用するようにシステムを設定することもできます。

Inso フィルタ処理テクノロジーは、ドキュメント表示用のプレーン・テキストおよび HTML 出力の作成にも使用されます。

参照： サポートされているドキュメント形式、フィルタ処理およびドキュメント表示方法の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

拡張知識ベース

カスタムの概念、カテゴリ、ワードまたは句を知識ベースに追加して、テーマ機能を改善できます。拡張語をシソーラスとして作成することによって行います。ctxload でシソーラスをインポートし、次に ctxkbtct でシソーラスをコンパイルして、既存の知識ベースを補強します。

参照： シソーラスのローダー ctxload および知識ベースのコンパイラ ctxkbtct の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

階層問合せフィードバック

任意の問合せ式では、関連する問合せ語句（上位語、下位語、関連語句）の情報を取得できます。アプリケーションは、この情報をユーザーに示して、ユーザーが問合せを詳細化することを助けます。

参照： CTX_DOC.HFEEDBACK プロシージャの詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

ユーザー・データストア

データストア方法として、USER_DATASTORE が追加されています。このデータ格納方法によって、索引付けのときにドキュメントを合成するプロシージャを定義できます。このような仮想ドキュメントは索引付けのときにのみ存在し、内容および構造は索引に保存されます。

たとえば、ユーザー定義のプロシージャは、1つのドキュメント内に日付、作成者およびテキスト列を合成し、単一のドキュメントの一部としてその作成者および日付の情報を索引付けできます。

参照： USER_DATASTORE の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

ストップリストの拡張

ストップリストのストップテーマおよびストップクラスのサポート

ストップワードの定義に加えて、ストップリストに追加できるストップテーマおよびストップクラスを定義できます。

ストップテーマは、索引付けされないテーマです。ストップクラスとは、数などの索引付けされない文字（英数字）のクラスを定義します。

参照： ストップテーマおよびストップクラスの追加の詳細は、『Oracle8i *interMedia Text* リファレンス』の CTX_DDL パッケージを参照してください。

ストップリストへのオブジェクトの動的な追加

索引の作成後、ストップワード、ストップテーマおよびストップクラスをストップリストに追加できます。

参照： 詳細は、『Oracle8i *interMedia Text* リファレンス』の ALTER INDEX コマンドを参照してください。

セクション検索の拡張

フィールド・セクション

フィールド・セクションは、8.1 の新しい機能です。これらは、サブドキュメントとして索引付けされるセクションです。これらのセクションには、ゾーン・セクションより優れたパフォーマンスの利点があります。

参照： ドキュメント・フィールド・セクションの詳細は、『Oracle8i *interMedia Text* リファレンス』の CTX_DDL.ADD_FIELD_SECTION プロシージャを参照してください。

XML セクション・グループ

XML_SECTION_GROUP を使用すると、XML スタイルのタグ付きドキュメントにセクションを定義できます。

ニュース・グループ・セクション

新しいセクション・グループ・オブジェクト NEWS_SECTION_GROUP は、RFC 1036 仕様に従ったニュース・グループ形式に設定されたドキュメントへのセクションの定義をサポートします。

参照： XML でのセクションの定義およびニュース・グループ形式に設定されたドキュメントの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

代替スペル

ドイツ語、デンマーク語およびスウェーデン語では、*interMedia* Text は、受け入れられた問合せ語句の代替スペルを認識します。BASIC_LEXER で代替スペルを使用できます。

参照： 代替スペルの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

8.1.6 の新機能および拡張

この項では、Oracle8i *interMedia* Text リリース 8.1.6 のすべての新機能および拡張について示します。

参照： 新機能の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

データストアの改善

- 新しい NESTED_DATASTORE によって、NESTED TABLE の行を索引付けできます。
- 拡張された USER_DATASTORE によって、ユーザー・プロシージャの出力タイプを定義できます。

フィルタの改善

- CREATE INDEX によってドキュメント FORMAT 列を指定できます。行単位レベルでドキュメント形式を定義すると、バイナリ・ドキュメントを選択してフィルタ処理を行い、テキストおよび HTML ドキュメントに対するフィルタ処理をバイパスできるため、複合書式のテキスト表の索引付けに有効です。
- CREATE INDEX によってキャラクタ・セット (CHARSET) 列を指定できます。行単位レベルで文書のキャラクタ・セットを定義すると、複合キャラクタ・セットのテキスト表の索引付けに有効です。

セクション検索の改善

- 拡張された BASIC_SECTION_GROUP によって、属性を使用してタグを解析できます。
- 拡張された HTML_SECTION_GROUP によって、META タグ NAME および CONTENT 属性を索引付けできます。
- 拡張された XML_SECTION_GROUP によって、ドキュメント・タイプ指定を使用したセクション定義をサポートします。
- 拡張された XML_SECTION_GROUP によって、属性セクションの定義および索引付けをサポートします。
- 新しい AUTO_SECTION_GROUP によって、XML セクションおよび属性を自動的に索引付けできます。
- ALTER INDEX によって、有効な索引を再構築できます。

レクサーの改善

- 新しい MULTI_LEXER によって、英語、フランス語および日本語ドキュメントを含む表などの、マルチ言語テキスト表を索引付けできます。
- ドキュメントごとに生成されるテーマの数に、制限がありません。
- 拡張された日本語および韓国語レクサーは、UTF-8 をサポートします。

ワードリストおよび記憶域の改善

- 拡張された BASIC_WORDLIST には、ファジー・マッチングおよびワード・ステミングに対する自動言語検出用の新しい属性が追加されています。
- 拡張された BASIC_WORDLIST には、語句の拡張制限の設定に使用できる、WILDCARD_MAXTERMS 属性が追加されています。
- 拡張された BASIC_WORDLIST には、新しい SUBSTRING_INDEX ブール属性が追加されています。この属性を使用可能にすると、ダブル・トランケーテッド問合せのパフォーマンスが向上します。
- 拡張された BASIC_STORAGE には、サブ文字列索引作成用の新しい P_TABLE_CLAUSE 記憶域句が追加されています。

ドキュメント・サービスの改善

- 拡張された CTX_DOC プロシージャによって、CTX_DOC.HIGHLIGHT や CTX_DOC.GIST のようなプロシージャの使用時に、主キーに加えて ROWID によってドキュメントを識別できます。
- 拡張された CTX_DOC プロシージャによって、オプションで結果のメモリー内格納が可能になり、応答時間を短縮できます。

シソーラスの改善

PL/SQL パッケージ CTX_THES には、次の新しいプロシージャが追加されています。

- ALTER_THESAURUS
- ALTER_PHRASE
- CREATE_RELATION
- DROP_PHRASE
- DROP_RELATION
- SN
- THES_TT

すべての CTX_THES 拡張ファンクションおよびプロシージャは、表への出力拡張の格納をサポートします。

索引付けおよび問合せの改善

- 拡張された ALTER INDEX には、並列度を指定できるパラレル索引付け句が追加されています。
- 索引の同期化および最適化の機能が、PL/SQL インタフェース CTX_DDL に追加されています。このパッケージには、次の新機能があります。
 - CTX_DDL.SYNC_INDEX
 - CTX_DDL.OPTIMIZE_INDEX
- 新しい CTX_QUERY.BROWSE_WORDS プロシージャによって、索引をブラウズできます。
- ACCUM 演算子は、新しいスコア・アルゴリズムを使用して、重み付けされたオペランドを正しくランク付けします。
- 拡張された WITHIN 演算子は、階層セクション検索（ネストされた WITHIN 句を使用したセクション内でのセクションの問合せ）をサポートします。

8.1 で変更された機能

テキスト索引は、現在 Oracle ドメイン・インデックスであるため、索引付け、問合せ、ドキュメント表示などのほとんどの機能は、Oracle8i *interMedia Text* リリース 8.1 では変更されています。これらの機能の移行方法は、このマニュアルで説明します。

次の表に、このマニュアルで説明するトピックおよびそれらの詳細の参照先を示します。

トピック	移行情報の詳細の参照先
移行の計画	第 2 章「アプリケーションの移行」
管理	第 3 章「管理」
索引表およびオブジェクト	第 4 章「索引プリファレンス」
索引作成および管理	第 5 章「索引付け」
問合せ	第 6 章「問合せ」
ドキュメントの表示方法	第 7 章「ドキュメントの表示方法」

SQL

変更

CONTAINS 演算子および SCORE 演算子は、8.1 で変更されていません。

参照： テキスト問合せの移行の詳細は、[第 6 章「問合せ」](#)を参照してください。

これらの演算子の構文については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

新しい SQL コマンド

次の標準 SQL コマンドを使用して、CONTEXT 型のドメイン・インデックスである 8.1 テキスト索引を作成および管理します。

- CREATE INDEX を使用して、テキスト索引を作成します。
- ALTER INDEX を使用して、テキスト索引を管理します。
- DROP INDEX を使用して、テキスト索引を削除します。

参照： CREATE INDEX および ALTER INDEX を使用したテキスト索引の移行の詳細は、[第 5 章「索引付け」](#)を参照してください。

構文については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

演算子

8.1 より前で使用可能な演算子のほとんどは、8.1 でも使用可能です。ただし、廃止および変更された演算子もあります。

廃止された演算子

次の表に、8.1 で廃止された演算子を示します。「移行方法」という列に、移行方法および詳細の参照先を示します。

演算子	等価記号	状態	移行方法
EXECUTE	@	廃止	SELECT 文内で関数をコールします。 第 6 章の「PL/SQL 演算子」 を参照してください。
FIRST/NEXT	#	廃止	結果セットの選択はカーソル問合せにとってかわります。 第 6 章の「MAX 演算子および FIRST/NEXT 演算子」 を参照してください。
MAX	:	廃止	結果セットの選択はカーソル問合せにとってかわります。 第 6 章の「MAX 演算子および FIRST/NEXT 演算子」 を参照してください。

変更された演算子

次の演算子は、8.1 で変更されています。

演算子	等価記号	状態	移行の注意
ACCUMulate	,	変更	スコア付けの方法が異なります。 『Oracle8i <i>interMedia</i> Text リファレンス』を参照してください。

新しい演算子

Oracle8i *interMedia* Text リリース 8.1 には、次の新しい演算子があります。

- ABOUT
- TR
- TRSYN

参照： ABOUT 演算子の詳細は、第 6 章の「テーマ問合せ（英語のみ）」を参照してください。

これらの演算子および既存の演算子の完全な構文については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

索引プリファレンス・オブジェクト

索引プリファレンス・オブジェクトの変更

索引プリファレンス・オブジェクトおよびシステム定義プリファレンスの多くが、8.1 で変更されています。

参照： 索引オブジェクトの変更の詳細は、第 4 章「索引プリファレンス」を参照してください。

新しい索引オブジェクト

次の索引オブジェクトが追加されています。

データ記憶域

- USER_DATASTORE
- NESTED_DATASTORE (8.1.6)

レクサー

- MULTI_LEXER (8.1.6)

セクション・グループ

- NEWS_SECTION_GROUP
- XML_SECTION_GROUP
- AUTO_SECTION_GROUP (8.1.6)

参照： 新規および既存のオブジェクトの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

PL/SQL

PL/SQL の変更

PL/SQL パッケージおよびプロシージャの多くが変更されています。

参照： PL/SQL の変更のリストは、[付録 A 「PL/SQL の変更」](#) を参照してください。

新しい PL/SQL

Oracle8i *interMedia* Text リリース 8.1 には、新しいパッケージおよびプロシージャがあります。

参照： Oracle8i *interMedia* Text リリース 8.1 のパッケージおよびプロシージャの完全なリストは、『Oracle8i *interMedia* Text リファレンス』を参照してください。

実行可能ファイル

変更

次の実行可能ファイルは、8.1 で変更されています。

ctxsrv サーバー

この実行可能ファイルには、次の変更があります。

- ctxsrv は M パーソナリティとしてのみ機能し、バックグラウンドで稼働して DML を処理します。
- 問合せ (Q)、リーダー (R)、言語処理 (L) および DDL (D) パーソナリティは廃止されています。

ctxload ロードー

このロードーは、8.1 より前のすべての機能を保持しています。このロードーには、BLOB および CLOB 列のエクスポート / 更新のサポートが追加されています。

参照： ctxsrv および ctxload コマンドの構文の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

ctxctl ユーティリティ

ctxsrv を監視および停止するこのシェル・スクリプトは廃止されました。

参照： 8.1 のサーバーの監視方法の詳細は、[第 3 章の「サーバー」](#)を参照してください。

新しい実行可能ファイル

8.1 の新しい実行可能ファイルを次に示します。

ctxkbtc

この実行可能ファイルは、複数のシソーラスから拡張された知識ベースをコンパイルします。

参照： ctxkbtc コマンドの構文の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

ビュー

ConText 2.x からのビューの多くが改名されています。また、新しいビューがリリース 8.1 に追加されています。

参照： リリース 8.1 のすべてのビューの完全なリストは、『Oracle8i *interMedia Text* リファレンス』を参照してください。

アプリケーションの移行

この章では、*interMedia Text* の移行プロセスについて説明します。この章の内容は次のとおりです。

- 移行パス
- 移行計画
- 移行ステップ

移行パス

このマニュアルでは、Oracle8 データベースで動作する ConText 2.x アプリケーションを、Oracle8i データベースで動作する *interMedia* Text 8.1 アプリケーションへ移行する方法について説明します。

注意： ConText 2.x アプリケーションは、Oracle8i データベース上では実行できません。

Oracle8 ユーザー

ConText 2.x を Oracle8 データベース上で実行している場合、アプリケーションを *interMedia* Text 8.1 に移行する必要があります。また、Oracle8 データベースをアップグレードする必要もあります。

参照： この章の「[移行計画](#)」を参照してください。

Oracle7 ユーザー

ConText 2.x を Oracle7 データベース上で実行している場合、アプリケーションを *interMedia* Text 8.1 に移行する前に、データベースを Oracle8i に移行する必要があります。

注意： Oracle7 データベースを Oracle8i データベースへ移行する場合の詳細は、『Oracle8i 移行ガイド』を参照してください。

移行計画

移行では、ConText 2.x がインストールされ、ConText アプリケーションが実行中であることを前提としています。移行の目的は、ConText 2.x/Oracle 8.0 本番アプリケーションを、*interMedia Text 8.1/Oracle8i* 環境の同一アプリケーションの実行中のバージョンに移すことです。

ConText アプリケーションの移行は、Oracle7 データベースから Oracle8 データベースへの移行のような、機械的な処理ではありません。これは、ConText 2.x アプリケーションが *interMedia Text 8.1* 環境では実行できないためです。

interMedia Text 8.1 へ移行するための一般的な計画は、次のとおりです。

1. [索引の移行](#)
2. [アプリケーション・コードの移行](#)
3. [データの移行：Oracle8i へのアップグレード](#)

索引の移行

8.1 より前の移行では、古い環境の索引付け機能を新しい 8.1 環境に複製する必要があります。

8.1 より前で索引を作成する方法は、プリファレンスを作成する必要があるという点で 8.1 と似ています。ただし、プリファレンスを索引へ連結する方法は、ポリシーが 8.1 に存在しないという点で異なります。8.1 では、カスタム・プリファレンスに CREATE INDEX 文で直接名前を付けます。

注意： 8.1 でのユーザー独自のプリファレンスは、デフォルトでない索引プリファレンスを必要とする場合のみ作成してください。

デフォルト索引の特性の詳細は、『Oracle8i *interMedia Text* リファレンス』の第 1 章を参照してください。

索引プリファレンスを作成するために使用する、システム提供のオブジェクトのほとんどは改名または廃止されています。したがって、8.1 より前の索引を移行するには、システム提供のどのオブジェクトが変更または廃止されているかを知る必要があります。

索引プリファレンスおよび索引は、次の方法のいずれかまたは両方を使用して移行できます。

- [手動での索引の移行](#)

■ 索引移行スクリプト

手動での索引の移行

手動で索引を移行するには、索引作成スクリプトを書き換えて、8.1 より前のプリファレンスを 8.1 のプリファレンスで再作成する必要があります。8.1 で改名された索引オブジェクトまたは新しい索引オブジェクトで新しいプリファレンスを作成し、廃止されたオブジェクトに基づくプリファレンスを移行する方法を決定します。

参照： 置換え、廃止および変更された索引オブジェクトの詳細は、[第 4 章の「プリファレンス・オブジェクト」](#)を参照してください。

新しい構文でプリファレンスおよび索引を作成する方法の詳細は、[第 5 章「索引付け」](#)を参照してください。

手動での索引の移行は、8.1 の新機能を使用してアプリケーションを再設計する場合に有効です。

索引移行スクリプト

8.1 より前の索引を 8.1 の索引へ移行するために、8.1 インストールで提供される移行スクリプトを使用できます。migrate.sql を作成するには、8.1 より前の環境でこれらのスクリプトを実行し、その後、新しい 8.1 環境で migrate.sql を実行します。

interMedia Text 8.1 のコードで書かれた migrate.sql スクリプトは、8.1 より前の索引プリファレンスおよび索引を interMedia Text 8.1 環境にレプリケートします。8.1 で廃止されている 2.x 索引付け機能またはオブジェクトがある場合、移行スクリプトは、コメントとして警告を発行します。機能の移行方法および移行スクリプトの編集方法をそれぞれ決定する必要があります。

参照： 移行スクリプトを使用する場合の詳細は、この章の[「移行ステップ」](#)を参照してください。

移行スクリプトの使用方法については、[第 4 章の「索引移行スクリプトの使用」](#)を参照してください。

アプリケーション・コードの移行

8.1 より前の ConText アプリケーションを iMT アプリケーションへ移行するには、8.1 のコードを使用して、アプリケーションを完全書き換える必要があります。これは、8.1 より前のアプリケーションが上位互換性を持たないためです。

索引を作成するコードに加え、移行が必要なアプリケーション・コードは、問合せを発行するコードおよびユーザーにドキュメントを表示するコードをすべて含みます。

新しいコードは、テキスト検索および索引付けが Oracle Server と統合されたために追加されました。たとえば、テキスト索引がドメイン・インデックスになったため、標準 SQL で索引付けし、問い合わせます。さらに、ドキュメント表示のための出力を作成する新しい PL/SQL プロシージャがあります。

アプリケーション・コードを再作成するには、移行プロセスの一部としてテスト環境を作成する必要があります。

参照： テスト環境の作成の詳細は、この章の「[移行ステップ](#)」を参照してください。

問合せの移行の詳細は、[第 6 章「問合せ」](#)を参照してください。

ドキュメント表示の新しい解決方法の詳細は、[第 7 章「ドキュメントの表示方法」](#)を参照してください。

データの移行：Oracle8i へのアップグレード

データの移行とは、8.1 より前のデータを 8.1 データベースへ転送することです。これは、Oracle8 データベースの Oracle8i データベースへのアップグレードともいいます。*interMedia Text* をインストールした後、これを行います。

エクスポート / インポートを使用して、データを手動で移行できます。または、Oracle Data Migration Assistant を使用できます。Oracle Data Migration Assistant は、Oracle Universal Installer で *interMedia Text* をインストールした後、自動的に起動されます。

参照： Oracle8i へのアップグレードの詳細は、『Oracle8i 移行ガイド』を参照してください。

データを移行する場合の詳細は、この章の「[移行ステップ](#)」を参照してください。

LONG 列

8.1 より前のアプリケーションが、LONG および LONG RAW テキスト列を使用している場合、データの移行プロセスの一部として、これらの列を LOB へ移行することを考慮します。

参照： LONG および LONG RAW 列の移行の詳細は、『Oracle8i 移行ガイド』を参照してください。

移行ステップ

この項では、移行の要件およびアプリケーションを移行するために必要なステップについて説明します。

ディスク領域要件

移行ステップでは、実働環境に加えて、個別のテスト環境を作成するための十分なディスク領域があり、次の要素を含むことを前提としています。

- 本番データベースの全体またはサブセット
- 新しいアプリケーション・コード

ハードウェアの考慮点

Oracle8i *interMedia* Text への移行では、3 つの Oracle ホームを使用します。

- 8.1 より前の本番 Oracle ホーム
- テスト用の 8.1 Oracle ホーム
- 新しい本番 8.1 Oracle ホーム

8.1 より前の本番 Oracle ホームがあるマシンとは異なるマシン上に、テスト Oracle ホームを作成することもできます。テスト Oracle ホームを異なるマシンに作成すると、アプリケーション・コードの再作成中に、テスト環境および実働環境間でのリソース（ディスク領域や処理時間など）の競合が発生しません。

データベースをアップグレードするには、新しい 8.1 Oracle ホームを作成する必要があります。Oracle8 から Oracle8i へのアップグレード処理は、同じマシン上で行われると想定されていますが、必須ではありません。

ステップ

表 2-1 に、Oracle8 上で動作する 8.1 より前のアプリケーションを、Oracle8i 上で動作する実行中の 8.1 アプリケーションに移行する有効な計画の概要を示します。この計画では、実働環境の欄の隣にテスト環境の設定を示し、各環境で実行する手順の概要を説明します。

注意： 次の表は、推奨する計画です。移行計画の詳細は、ご使用のアプリケーションの設計によって、サイトごとに異なります。

表 2-1

ステップ	実働環境	テスト環境
1.	実行中	<i>interMedia Text 8.1</i> および Oracle8i ソフトウェアを、本番 Oracle ホームとは別の Oracle ホームにインストールします。このテスト Oracle ホームが、テスト環境です。
2.	drminst および drmrn をテスト環境からコピーします。 第 4 章の「索引移行スクリプトの使用」 を参照してください。	
3.	drminst および drmrn スクリプトを実行して、migrate.sql を生成します。 第 4 章の「索引移行スクリプトの使用」 を参照してください。	
4.	テスト環境で使用する本番データの全体またはサブセットをエクスポートします。	
5.	実行中	Oracle8i データベース・サーバーを起動します。
6.	実行中	本番データベースのデータまたはサブセットをインポートします。
7.	実行中	migrate.sql を実働環境からコピーします。
8.	実行中	migrate.sql を編集します。 第 4 章の「索引移行スクリプトの使用」 を参照してください。

表 2-1

ステップ	実働環境	テスト環境
9.	実行中	<p>migrate.sql またはユーザー独自の索引作成スクリプトのいずれかを実行して、索引を移行します。</p> <p>この章の「索引の移行」を参照してください。</p> <p>これで、8.1 より前の実働環境と同様のスキーマおよび索引を持つ <i>interMedia Text 8.1</i> 環境ができます。</p>
10.	実行中	<p>8.1 より前のアプリケーション・コードを移行します。</p> <p>この章の「アプリケーション・コードの移行」を参照してください。</p>
11.	実行中	<p><i>interMedia Text 8.1</i> 環境でアプリケーションをテストします。</p>
12.	8.1 より前のシステムのバックアップをとります。	
13.	索引および CTXSYS スキーマを削除します。	
14.	<p>Oracle Universal Installer を使用して、<i>interMedia Text 8.1</i> をインストールします。</p> <p>この処理では、Oracle8<i>i</i> <i>interMedia Text</i> インストール用に、新しい Oracle ホームを作成する必要があります。この新しい Oracle ホームが、新しい本番 Oracle ホームになります。この新しいホームは、古い本番 Oracle ホーム (Oracle8 上で実行される) およびテスト用の Oracle ホームとは異なります。</p> <p>また、このインストール・プロセスによって、新しい CTXSYS ユーザーおよび 8.1 <i>interMedia Text</i> データ・ディクショナリも作成されます。</p>	

表 2-1

ステップ	実働環境	テスト環境
15.	<p>Oracle Data Migration Assistant を使用して、データ（Oracle8i へのアップグレードと同様）を移行します。</p> <p>Oracle Universal Installer は、<i>interMedia Text</i> のインストール後、自動的にこのアシスタントを起動します。</p> <p>この章の「データの移行：Oracle8i へのアップグレード」を参照してください。</p>	
16.	<p>テスト Oracle ホームから、<code>migrate.sql</code> および他の新しく作成された索引作成スクリプトをコピーします。</p>	
17.	<p><code>migrate.sql</code> または他の索引作成スクリプト（あるいはその両方）を実行して、索引を再作成します。</p>	
18.	<p>新しい 8.1 アプリケーション・コードをテスト Oracle ホームからコピーします。</p>	
19.	<p>8.1 アプリケーションを実行します。</p>	

この章では、2.x から 8.1 で変更された管理の概念について説明します。この章の内容は次のとおりです。

- サーバー
- ロールおよびユーザー
- ドキュメントのロード
- キューおよびパイプ
- 1 段階問合せの使用可能化
- 管理ツール

サーバー

8.1 より前

8.1 より前では、ConText サーバー `ctxsrv` には、5 つのパーソナリティ（R、Q、D、M および L）があります。単一のサーバーが複数のパーソナリティを持つことができます。パーソナリティの組合せをマスクといいます。パーソナリティ・マスクを使用すると、サーバーに異なる機能（問合せ処理、DDL 処理、言語処理など）を割り当てることができます。

8.1

8.1 では、索引付けまたは問合せに `ctxsrv` を起動する必要はありません。索引付けおよび問合せは、標準 SQL を使用して実行されます。バックグラウンド DML 処理に対してのみ、サーバーを起動します。バッチ DML の実行にサーバーは必要なく、ALTER INDEX で索引を同期化して行います。

ドキュメント・サービス出力（テーマ・サマリーや要旨など）の作成にも、`ctxsrv` の起動は必要ありません。CTX_DOC パッケージのプロシージャを使用して、ドキュメント・サービス出力を取得します。

`ctxsrv` を起動した場合に指定できるパーソナリティは M のみです。これはデフォルトです。

参照： `ctxsrv`、ALTER INDEX および CTX_DOC パッケージの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

サーバーのステータスの表示

Oracle8i *interMedia* Text リリース 8.1 では、8.1 より前で ConText サーバーのステータスを表示するために使用されていた、`ctxctl` コマンドはサポートされていません。

8.1 でサーバーのステータスを表示するには、CTX_SERVERS ビューを使用できます。また、Oracle8i *interMedia* Text Manager（Oracle Enterprise Manager で使用可能な Java アプリケーション）も使用できます。

注意： Oracle8i Enterprise Manager は、Oracle8i *interMedia* Text とは別出荷となります。

ロールおよびユーザー

8.1 より前

8.1 より前では、ConText ユーザーに、次の 3 つの事前定義済 ConText ロールのうちのいずれかを割り当てることができます。

- CTXADMIN
- CTXAPP
- CTXUSER

8.1

8.1 では、CTXADMIN ロールおよび CTXUSER ロールは廃止されました。

8.1 では、CTXSYS はインストール時に作成されるユーザーです。システムによって、CTXAPP ロールも定義されます。

CTXSYS ユーザー

CTXSYS ユーザーは、インストール時に作成されます。*interMedia Text* ユーザーを、このユーザーとして管理します。

CTXSYS は、次のことができます。

- ctxsrv サーバーの開始
- システム定義プリファレンスの変更
- 他のユーザー・プリファレンスの削除および変更
- PL/SQL パッケージ CTX_ADM でのプロシージャのコールによる、サーバーの起動およびシステム・パラメータの設定
- すべてのシステム定義ビューの問合せ
- CTXAPP ロールにより可能となるすべてのタスクの実行

CTXAPP ロール

CTXAPP ロールは、システム定義ロールです。このロールによって、次のことができるようになります。

- *interMedia Text* プリファレンスの作成および削除
- *interMedia Text* PL/SQL パッケージの使用

どのユーザーでも、テキスト索引の作成およびテキスト問合せの発行ができます。CTXAPP ロールによって、プリファレンスを作成したり、*interMedia Text* PL/SQL パッケージを使用することができます。

ユーザーへのロールおよび権限の付与

8.1 より前では、システムは、標準 SQL モデルを使用してユーザーにロールを付与します。このモデルは、*interMedia Text* 8.1 でも同じです。テキスト・ロールをユーザーに付与するには、SQL の GRANT コマンドを使用します。

また、アプリケーション開発者が *interMedia Text* PL/SQL パッケージのプロシージャをコールできるようにするには、*interMedia Text* パッケージに対する EXECUTE 権限を各ユーザーに明示的に付与する必要があります。

ドキュメントのロード

8.1 より前

ctxload 実行可能ファイルを使用して、ドキュメントを LONG または LONG RAW テキスト列にロードします。シソーラスのインポートにも、この実行可能ファイルを使用します。

8.1

ctxload

8.1 でも、ctxload が、LONG と LONG RAW のロードおよびシソーラスのインポートをサポートします。

提案： 8.1 より前のアプリケーションが LONG または LONG RAW を使用してドキュメントを格納している場合、これらの列を LOB に移行することをお薦めします。移行には、SQL 演算子 TO_LOB を使用できます。

LONG および LONG RAW を LOB へ移行する場合の詳細は、『Oracle8i 移行ガイド』を参照してください。

ctxload を使用した LONG のロードの詳細は、『Oracle8i interMedia Text リファレンス』を参照してください。

SQL*Loader

8.1 では、ドキュメントを CHAR、CLOB、BLOB、BFILE または VARCHAR2 の型の列に格納できます。ドキュメントをこれらのデータ型のいずれかにロードするには、SQL*Loader を使用できます。

参照： SQL*Loader を使用した例は、『Oracle8i interMedia Text リファレンス』を参照してください。

キューおよびパイプ

8.1 より前

8.1 より前では、ConText は、テキスト要求キューを使用してテキスト操作を処理します。テキスト要求キューは、問合せパイプ、DDL パイプ、DML キューおよびサービス・キューで構成されています。

8.1

8.1 では、テキストの索引付けおよび問合せは、標準 SQL を使用して実行されます。このため、すべてのキューおよびパイプ（DML キューを除く）は廃止されました。

DML キューは索引更新要求を格納し、Oracle によって移入されます。

保留されている DML 要求は、CTX_PENDING ビューおよび CTX_USER_PENDING ビューで表示できます。

DML エラーは、CTX_INDEX_ERRORS ビューまたは CTX_USER_INDEX_ERRORS ビューで表示できます。

参照： これらのビューを使用する場合の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

1 段階問合せの使用可能化

8.1 より前

8.1 より前では、`initSIDORA` ファイルで `text_enable` を TRUE に設定することによって、1 段階問合せを使用できます。

8.1

8.1 では、標準 SQL を使用してテキスト問合せを発行します。Oracle8i *interMedia* Text リリース 8.1 のインストール後は、その他の設定は不要です。一度、有効なテキスト索引を作成すると、テキスト問合せを発行できるようになります。

参照： テキスト索引の作成の詳細は、[第 5 章「索引付け」](#)を参照してください。

管理ツール

8.1 より前

8.1 より前では、GUI 管理ツールを使用して、索引の作成やサーバーの起動などができます。このツールは、ConText ワークベンチで使用可能です。

8.1

Oracle8i *interMedia* Text リリース 8.1 では、管理ツールは、Oracle8i *interMedia* Text Manager に置き換えられました。このツールは、別出荷の Oracle Enterprise Manager に統合された Java アプリケーションです。

Text Manager によって、管理者はプリファレンス、ストップリスト、セクションおよび索引を作成できます。また、このツールによって、管理者は ctxsrv DML サーバーを起動、監視および停止できます。

参照： Oracle8i *interMedia* Text Manager の詳細は、このツールとともに出荷されるオンライン・ヘルプを参照してください。

索引プリファレンス

この章では、8.1 より前の ConText オブジェクトを *interMedia Text 8.1* のオブジェクトに置き換える方法について説明します。この章の内容は次のとおりです。

- [プリファレンス・オブジェクト](#)
- [システム定義プリファレンス](#)
- [索引移行スクリプトの使用](#)

プリファレンス・オブジェクト

索引プリファレンス・オブジェクトの変更

[表 4-1](#) に、索引付けのためのプリファレンス作成に使用されるオブジェクトのうち、2.x から 8.1 で変更または廃止されたものを示します。

[表 4-1](#) の情報を使用して、2.x オブジェクトで定義したプリファレンスを手動で移行できます。そうしない場合は、移行スクリプトを使用できます。

参照： 移行スクリプトの使用の詳細は、この章の「[索引移行スクリプトの使用](#)」を参照してください。

プリファレンス作成の詳細は、[第 5 章「索引付け」](#)を参照してください。

表 4-1

プリファレンス・クラス	プリファレンス・オブジェクト	状態	移行の注意
データストア	DIRECT	変更	DETAIL_DATASTORE に改名
	MASTER DETAIL	廃止	8.1 では、MASTER DETAIL は廃止されています。 マスター表 / ディテール表を設定するには、DETAIL_DATASTORE オブジェクト (2.x では MASTER DETAIL NEW) を使用してください。
	MASTER DETAIL NEW	変更	DETAIL_DATASTORE に改名
	OSFILE	変更	FILE_DATASTORE に改名
	URL	変更	URL_DATASTORE に改名
フィルタ	NULL FILTER	変更	NULL_FILTER に改名
	BLASTER FILTER	廃止	8.1 では、BLASTER FILTER は廃止されています。 単一または複合書式の列を索引付けするには、INSO_FILTER オブジェクトを使用してください。Inso フィルタは自動検出するため、フォーマット属性は必要ありません。 INSO_FILTER がサポートしていない書式 (単一または複合列) を索引付けするには、ユーザー・フィルタを作成 / 指定し、USER_FILTER オブジェクトを使用して、コマンド属性で実行可能ファイル名を指定してください。
	HTML FILTER	廃止	HTML タグに索引付けせずに HTML ドキュメントを索引付けするには、HTML_SECTION_GROUP 型のセクション・グループ・プリファレンスを作成および指定してください。 2.x で HTML ドキュメントに対して日本語の CODE_CONVERSION が ON になっている場合は、CHARSET を JAAUTO に設定して CHARSET_FILTER フィルタ・プリファレンスを使用します。HTML のフィルタ処理およびセクション化は 1 つのオブジェクトに統合されるため、KEEP_TAG は必要ありません。
	USER FILTER	変更	USER_FILTER に改名
レクサー	BASIC LEXER	変更	BASIC_LEXER に改名 また、SENT_PARA が有効な属性ではなくなりました。文章または段落の検索を可能にするには、セクション・グループ・プリファレンスに特殊セクションを追加してください。
	JAPANESE V-GRAM LEXER	変更	JAPANESE_VGRAM_LEXER に改名 また、8.1 では、このオブジェクトには属性がありません。

表 4-1

プリファレンス・クラス	プリファレンス・オブジェクト	状態	移行の注意
	KOREAN LEXER	変更	KOREAN_LEXER に改名 また、8.1 では、索引付けされたものをユーザーが制御できるように、いくつかの属性が追加されています。
	CHINESE V-GRAM LEXER	変更	CHINESE_VGRAM_LEXER に改名 また、このオブジェクトには属性がありません。
	THEME LEXER	廃止	テーマを索引付けするには、INDEX_THEMES 属性を TRUE に設定して BASIC_LEXER を使用してください。
	NLS_LEXER	廃止	このオブジェクトは、2.x では実装されませんでした。8.1 では削除されました。
ワードリスト	GENERIC WORD LIST	変更	BASIC_WORDLIST に改名 また、STCLAUSE、INSTCLAUSE、SOUNDEX_AT_INDEX および SECTION_GROUP 属性が廃止されました。 SOUNDEX は自動的に使用可能になり、属性は必要ありません。 索引にセクション・グループを指定するには、まず CTX_DDL.CREATE_SECTION_GROUP を使用してセクション・グループを作成し、次に CREATE INDEX のパラメータ文字列にセクション・グループを指定してください。 BASIC_WORDLIST には、STEMMER および FUZZY 属性が必要です。
ストップリスト	GENERIC STOP LIST	廃止	CTX_DDL.CREATE_STOPLIST プロシージャを使用してストップリストを作成し、CREATE INDEX のパラメータ文字列にそのストップリストを指定してください。
エンジン	GENERIC ENGINE	変更	BASIC_STORAGE に改名 8.1 では、ENGINE プリファレンスが STORAGE プリファレンスに改名されました。 また、2.x のすべての属性が廃止されています。 INDEX_MEMORY 属性はありません。CREATE INDEX および ALTER INDEX のパラメータ文字列に索引付けメモリーを設定してください。 様々なオプションが結合されています。そのため、III_TABLESPACE、III_STORAGE などというように各表に 3 つの句を指定するのではなく、すべての句を I_TABLE_CLAUSE という 1 つの属性で指定してください。

表 4-1

プリファレンス・クラス	プリファレンス・オブジェクト	状態	移行の注意
ローダー	GENERIC LOADER	廃止	2.x では、これらのプリファレンス・オブジェクトは、ctxload を使用したテキスト・ロードを自動化するためのソースの作成に使用されていました。 8.1 では、ctxload は、テキスト・ロードをサポートしていません。そのため、これらのプリファレンス・オブジェクトは廃止されています。 データベースにテキストをロードするには、SQL*Loader を使用してください。
	NULL TRANSLATOR	廃止	
	USER TRANSLATOR	廃止	
	DIRECTORY READER	廃止	
コンプレッサ	NULL COMPRESSOR	廃止	このオブジェクトは 2.x では実装されませんでした。8.1 では削除されました。

新しい索引オブジェクト

Oracle8i *interMedia* Text リリース 8.1 には、いくつかの索引オブジェクトが新しく追加されています。

参照： Oracle8i *interMedia* Text リリース 8.1 の索引オブジェクトの完全なリストは、『Oracle8i *interMedia* Text リファレンス』を参照してください。

システム定義プリファレンス

Oracle8i *interMedia* Text リリース 8.1 では、ほとんどのシステム定義プリファレンスが改名され、いくつかの新しいプリファレンスが追加されています。

参照： Oracle8i *interMedia* Text リリース 8.1 のシステム定義プリファレンスの完全なリストは、『Oracle8i *interMedia* Text リファレンス』を参照してください。

移行スクリプトによって、システム定義プリファレンスの変換が自動化できます。

参照： 自動変換の完全なリストは、後述の「[システム定義プリファレンスの変換](#)」を参照してください。

索引移行スクリプトの使用

移行スクリプトには、次のものがあります。

- `drminst`
- `drmrn`
- `migrate.sql`

移行スクリプト `drminst` および `drmrn` は、8.1 インストールにあります。これらのスクリプトは、`$ORACLE_HOME/ctx/migrate` ディレクトリにあります。

これらのスクリプトを実働環境（8.1 より前）にコピーしてください。これらのスクリプトを実行して `migrate.sql` を生成します。

`migrate.sql` スクリプトは、8.1 より前の索引をできるだけ完璧に再作成しようとします。索引の再作成は、*interMedia Text 8.1* のコードを使用して対応するプリファレンスおよび属性を作成し、最後に索引を作成することによって行われます。

`migrate.sql` をテスト環境にコピーし、適切に編集し、編集後の `migrate.sql` を実行して索引を変換します。

参照： 移行計画の詳細は、第2章の「移行計画」を参照してください。

ユーザー定義プリファレンスの変換

移行スクリプトは、8.1 より前のオブジェクトで定義されたすべてのプリファレンスを、改名された 8.1 のオブジェクトで定義されたプリファレンスに置き換えようとします。

たとえば、OSFILE オブジェクトに設定された `mydatapref` という名前のプリファレンスを使用する、8.1 より前のポリシーがある場合、移行スクリプトは `mydatapref` という名前の 8.1 のプリファレンスを作成し、このプリファレンスが `FILE_DATASTORE` オブジェクトを使用するように設定します。この 8.1 の索引は、`mydatapref` の記憶域プリファレンスで作成されます。

廃止されたオブジェクトを元にしたプリファレンスに関する警告

8.1 より前のオブジェクトまたは属性が 8.1 で廃止されている場合、移行スクリプトは REM 文の形式で警告を発行し、対応するプリファレンスを作成しません。この場合、ご使用のシステムに合わせて、正しい置換え用のプリファレンス（存在する場合）が反映されるように移行スクリプトを編集する必要があります。

参照： 改名または廃止されたオブジェクトのリストは、この章の「プリファレンス・オブジェクト」を参照してください。

システム定義プリファレンスの変換

移行スクリプトは、8.1 より前のシステム定義プリファレンスを、8.1 のシステム定義プリファレンスに置き換えます。

変更されていないプリファレンス

8.1 より前から 8.1 でシステム定義プリファレンス名が変更されていない場合、移行スクリプトは同じプリファレンス名および属性を使用します。たとえば、8.1 より前のポリシーが `DEFAULT_LEXER` というシステム定義プリファレンスを使用していた場合、8.1 の索引は `DEFAULT_LEXER` というシステム定義レクサー・プリファレンスで作成されます。

改名されたプリファレンス

システム定義プリファレンスの名前または属性が 8.1 で改名されている場合、移行スクリプトは古い名前を新しい名前に置き換えます。

廃止されたプリファレンスに関する警告

8.1 より前のシステム定義プリファレンスまたは属性が 8.1 で廃止されている場合、移行スクリプトは `REM` 文の形式で警告を発行し、対応するプリファレンスを作成しません。この場合、ご使用のシステムに合わせて、正しい置換え用のプリファレンス（存在する場合）が反映されるように移行スクリプトを編集する必要があります。

変換表

移行スクリプトは、2.x のシステム定義プリファレンスを新しい 8.1 のプリファレンスに置き換えます。次の表に、これらの置換を示します。

2.x のシステム定義プリファレンス	8.1 のプリファレンス
DEFAULT_DIRECT_DATASTORE	DEFAULT_DATASTORE
MD_TEXT	廃止
MD_BINARY	廃止
OSFILE	FILE_DATASTORE
DEFAULT_URL	URL_DATASTORE
DEFAULT_NULL_COMPRESSOR	廃止
DEFAULT_NULL_FILTER	NULL_FILTER
AUTOB	INSO_FILTER
WW6B	INSO_FILTER
HTML_FILTER	廃止

2.x のシステム定義プリファレンス	8.1 のプリファレンス
BASIC_HTML_FILTER	廃止
DEFAULT_LEXER	DEFAULT_LEXER
VGRAM_JAPANESE	JAPANESE_LEXER
VGRAM_JAPANESE_1	JAPANESE_LEXER
VGRAM_JAPANESE_2	JAPANESE_LEXER
KOREAN	KOREAN_LEXER
VGRAM_CHINESE	CHINESE_LEXER
VGRAM_CHINESE_1	CHINESE_LEXER
VGRAM_CHINESE_2	CHINESE_LEXER
THEME_LEXER	廃止
BASIC_HTML_LEXER	廃止
BASIC_HTML_SECTION	廃止
SOUNDEX	DEFAULT_WORDLIST
NO_SOUNDEX	DEFAULT_WORDLIST
VGRAM_JAPANESE_WORDLIST	JAPANESE_WORDLIST
KOREAN_WORDLIST	KOREAN_WORDLIST
VGRAM_CHINESE_WORDLIST	CHINESE_WORDLIST
BASIC_HTML_WORDLIST	廃止
DEFAULT_STOPLIST	DEFAULT_STOPLIST
NO_STOPLIST	EMPTY_STOPLIST
DEFAULT_INDEX	DEFAULT_STORAGE
DEFAULT_LOADER	廃止
DEFAULT_TRANSLATOR	廃止
FRENCH_STOPLIST	FRENCH_STOPLIST
ITALIAN_STOPLIST	ITALIAN_STOPLIST
SPANISH_STOPLIST	SPANISH_STOPLIST
GERMAN_STOPLIST	GERMAN_STOPLIST

索引作成の警告

属性およびプリファレンスが作成された後、移行スクリプトは 8.1 より前のポリシー情報から 8.1 より前の索引を再作成しようとします。

interMedia Text 8.1 にはポリシーは存在しません。8.1 では、拡張可能な型の Oracle 索引としてテキスト索引を作成します。また、列ごとに 1 つのみ索引を作成できます。

したがって、8.1 より前の環境に次の移行不可能な構造のいずれかが含まれている場合、移行スクリプトは REM 文として警告を発行します。

- 1 列に複数のポリシー
- 索引付けされていないポリシー
- テンプレート・ポリシー
- ビューへの索引

参照： テキスト索引の詳細は、[第 5 章の「テキスト索引の概要」](#)を参照してください。

索引付け

この章では、アプリケーションに影響を与える可能性のあるテキスト索引付けプロセスの変更について説明します。この章の内容は次のとおりです。

- [テキスト索引の概要](#)
- [索引作成の手順](#)
- [プリファレンスの作成](#)
- [索引の作成](#)
- [プリファレンスの削除](#)
- [失敗した索引の再開](#)
- [索引の再構築](#)
- [索引の最適化](#)
- [索引の更新 - バックグラウンド DML](#)
- [索引の更新 - バッチ DML](#)
- [ストップリストおよびストップワード](#)
- [ドキュメント・セクション](#)

テキスト索引の概要

8.1 より前では、最初にポリシーを作成し、次に索引を作成するためのポリシーを使用することによって、CTX_DDL パッケージで索引が作成されます。

8.1 では、標準 SQL を使用し、Oracle の特殊型の拡張可能な索引としてテキスト索引が作成されます。つまり、*interMedia Text* 8.1 の索引は、Oracle の索引として動作するという事です。この索引には参照する名前があり、ポリシーは存在しません。

参照： テキスト索引作成の詳細は、この章の「[索引作成の手順](#)」を参照してください。

マージされたワードおよびテーマ索引（英語のみ）

8.1 では、1 つのテキスト索引にテーマおよびワード情報の両方を含むことができます。テーマ問合せを発行する場合に、テキスト索引に加えてテーマ索引が必要だった 8.1 より前とは、この点で異なります。

英語でのデフォルトでは、*interMedia Text* はワード情報でテーマ情報を索引付けします。レクサー・プリファレンスによって、オプションで、テーマの索引付けを使用可能および使用禁止にできます。

参照： テーマ情報の索引付けについては、この章の「[プリファレンスの作成](#)」を参照してください。

複数の索引付きの列

8.1 より前では、1 つのテキスト列に複数の索引を作成できました。これは、1 つのテキスト列にテキスト索引およびテーマ索引を作成する必要がある場合に有効です。

8.1 では、Oracle 標準に従い、列に複数のドメイン・インデックスを付けることはできません。ただし、1 つのテキスト索引に、ワード情報に加えてテーマ情報を含めることができます。

ビューの索引付け

8.1 より前では、ビューに ConText 索引を作成できます。これは、異なる表の内容を継ぎ合わせたドキュメントに索引付けする必要がある場合に有効です。

ただし、Oracle の SQL 標準ではビューへの索引作成はサポートされていません。したがって、8.1 では、異なる表の内容を継ぎ合わせたドキュメントの作成および索引付けが必要な場合、8.1 で新しく追加された USER_DATASTORE オブジェクトを使用して、データ記憶域プリファレンスを作成できます。このオブジェクトで、索引付け時にドキュメントを合成するプロシージャを定義します。

参照： USER_DATASTORE の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

索引作成の手順

8.1 より前

8.1 より前での索引作成の手順は次のとおりです。

- 1. 索引付けプリファレンスを決定します。
- 2. 索引プリファレンスを作成します。
- 3. 索引ポリシーを作成します。
- 4. ポリシーを指定して、CTX_DDL.CREATE_INDEX プロシージャをコールします。

8.1

索引作成の手順は次の理由から、より簡単になっています。

- 8.1 ではポリシーが存在しません。
- デフォルトで、言語、テキスト列のデータ型、ドキュメントの形式がシステムによって自動検出され、必要に応じて索引付けプリファレンスが設定されます。つまり、索引の作成について、ユーザー独自のプリファレンス作成はオプションです。

デフォルトによって、システムはドキュメントがテキスト列に格納されると想定します。一度、この要件が満たされれば、プリファレンスを明示的に指定することなく、ConText 型の拡張可能な索引として CREATE INDEX コマンドを使用してテキスト索引を作成できます。

参照： 出荷時のデフォルトの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

8.1 での索引作成の手順は次のとおりです。

- 1. 任意に、デフォルトを使用しない場合は、カスタム索引付けプリファレンスを決定します。ここでは、次のプリファレンスを決定します。

プリファレンス・クラス	説明
データストア	ドキュメントの保存方法
フィルタ	ドキュメントのプレーン・テキストへの変換方法
レクサー	索引付けされる言語
ワードリスト	ステミング問合せおよびファジー問合せの拡張方法

プリファレンス・クラス	説明
記憶域	索引データの格納方法
ストップリスト	索引付けしないワードおよびテーマ
セクション・グループ	ドキュメント・セクションの定義方法

参照： リリース 8.1 で使用可能なプリファレンス・オブジェクトの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

2. ユーザー独自のカスタム・プリファレンスを作成します。この章の「[プリファレンスの作成](#)」を参照してください。
3. SQL コマンド CREATE INDEX でテキスト索引を作成し、索引に名前を付け、任意のプリファレンスを指定します。この章の「[索引の作成](#)」を参照してください。

プリファレンスの作成

8.1 では、CTX_DDL.CREATE_PREFERENCE および CTX_DDL.SET_ATTRIBUTE プロシージャの構文が変更されています。また、これらのプロシージャをコールする順序も変更されています。

8.1 では、プリファレンスを作成してから属性を設定するという、8.1 より前での順序とは逆の手順で行います。

参照： プリファレンス・オブジェクトと関連する属性の完全なリスト、および CTX_DDL.CREATE_PREFERENCE と CTX_DDL.SET_ATTRIBUTE プロシージャの構文は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

例：ファイル・データ記憶域の指定

次の例では、索引付けされるファイルがオペレーティング・システムに格納されていることをシステムに知らせる、mypref というデータ記憶域のカスタム・プリファレンスを作成します。次に、この例では CTX_DDL.SET_ATTRIBUTE を使用して、PATH 属性をディレクトリ /docs に設定します。

```
begin
ctx_ddl.create_preference('mypref', 'FILE_DATASTORE');
ctx_ddl.set_attribute('mypref', 'PATH', '/docs');
end;
```

参照： データ記憶域の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

索引の作成

8.1 より前

8.1 より前では、CTX_DDL.CREATE_INDEX を使用して索引を作成し、ポリシーに名前を付けていました。

8.1

8.1 では、CREATE INDEX コマンドを使用して、拡張可能な索引型としてテキスト索引を作成します。その索引に名前を付け、オプションでパラメータ文字列内にレクサーやフィルタなどのプリファレンスを指定します。

参照： CREATE INDEX コマンドの構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

索引作成の例

次の例では、mytable 内の news 列に、newsindex というテキスト索引を作成します。索引は、my_lexer というレクサー・プリファレンスおよび my_stop というストップリストで作成されます。未指定のプリファレンスに対しては、デフォルトの属性が使用されます。

```
create index newsindex on mytable(news) indextype is ctxsys.context
parameters('lexer my_lexer stoplist my_stop');
```

プリファレンスの削除

8.1 より前

8.1 より前では、CTX_DDL.DROP_PREFERENCE を使用してプリファレンスを削除します。この操作は、参照したすべてのポリシーがデータ・ディクショナリから削除されている場合にのみ可能です。

8.1

8.1 では、同じプロシージャ CTX_DDL.DROP_PREFERENCE で索引プリファレンスを削除します。プリファレンスは索引とは別に存在し、8.1 ではポリシーが存在しないため、プリファレンスを削除する前に索引を削除する必要はありません。

プリファレンスを削除しても、その削除したプリファレンスを使用している索引には影響ありません。

参照： CTX_DDL.DROP_PREFERENCE プロシージャの構文の詳細は、『Oracle8i interMedia Text リファレンス』を参照してください。

例

次のコードは、プリファレンス my_lexer を削除します。

```
begin
ctx_ddl.drop_preference('my_lexer');
end;
```

索引の削除

8.1 より前

8.1 より前では、CTX_DDL.DROP_INDEX を使用して索引を削除します。

8.1

8.1 では、SQL の DROP INDEX コマンドを使用して索引を削除します。

たとえば、newsindex という索引を削除するには、次の SQL コマンドを発行します。

```
drop index newsindex;
```

索引のクラッシュなどによって索引の状態が判断できない場合、前述のコマンドで索引は削除できません。かわりに次のコマンドを使用します。

```
drop index newsindex force;
```

参照： DROP INDEX コマンドの構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

失敗した索引の再開

8.1 より前

8.1 より前では、索引付け操作（作成または最適化）が失敗した場合、CTX_DDL.RESUME_FAILED_INDEX を使用してその操作を再開します。

8.1

interMedia Text 8.1 では、ALTER INDEX コマンドを使用して、失敗した索引付け操作を再開します。

8.1 では、最適化は一定の間隔でコミットされます。したがって、最適化が失敗しても、すべての最適化作業は保存されています。

参照： ALTER INDEX コマンドの構文の詳細は、『*Oracle8i interMedia Text* リファレンス』を参照してください。

例

次のコマンドは、2MB のメモリーを確保して `newsindex` の索引付け操作を再開します。

```
ALTER INDEX newsindex rebuild parameters('resume memory 2M');
```

索引の再構築

ALTER INDEX を使用して有効な索引を再構築できます。新しいプリファレンスで索引付けを行う場合に、索引を再構築することがあります。

参照： 索引再構築用の ALTER INDEX コマンドの構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

例

次のコマンドは、索引を再構築し、レクサー・プリファレンスを `my_lexer` と置き換えます。

```
ALTER INDEX newsindex rebuild parameters('replace lexer my_lexer');
```

索引の最適化

8.1 より前

8.1 より前では、索引を最適化するために CTX_DDL.OPTIMIZE_INDEX を使用し、5 つの異なる最適化方法のうちの 1 つを指定します。

8.1.5

8.1 では、索引を最適化するために、SQL の ALTER INDEX コマンドを REBUILD パラメータ付きで使用します。高速または完全モードのいずれかで索引を最適化できます。

8.1.6

8.1.6 では、ALTER INDEX を使用した最適化に加えて、PL/SQL の CTX_DDL.OPTIMIZE_INDEX を使用して索引を最適化できます。

参照： ALTER INDEX および CTX_DDL.OPTIMIZE_INDEX を使用する索引最適化の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

索引の更新 - バックグラウンド DML

8.1 より前と同様に、DML を自動的に処理するには、ctxsrv プロセスが実行している必要があります。ctxsrv が実行している場合、8.1 のテキスト索引は、ベース表への挿入、削除または更新が行われるときに必ず更新されます。これを、バックグラウンド DML 処理といいます。

次の例では、サーバーを起動し、サーバーのすべてのメッセージを ctx.log ファイルに書き込みます。

```
ctxsrv -user ctxsys/ctxsys -personality M -log ctx.log &
```

参照： ctxsrv を使用したバックグラウンド DML の詳細は、『Oracle8i *interMedia Text* リファレンス』の ctxsrv の仕様を参照してください。

索引の更新 - バッチ DML

索引の更新には、ベース表へのすべての更新、挿入および削除の処理が反映されます。これを、索引の同期化といいます。索引の同期化は、バックグラウンドまたはバッチ・モードで行えます。

8.1 より前

8.1 より前では、CTX_DML.SYNC を使用して索引を同期化します。さらに、ConText M サーバーが実行中である必要があります。

8.1.5

ALTER INDEX コマンドを *sync* パラメータで実行することによって、バッチ・モードで索引を更新できます。バッチ・モードで索引を同期化すると、DML キューに格納された保留中の更新および挿入が Oracle によって処理されます。

バッチでの索引の同期化は、挿入、更新および削除のバッチで行われるため、バッチ DML では通常、ctxsrv デーモンを実行してすぐに索引を同期化するよりも、索引の断片化の度合いが小さくなります。

注意： バッチでの索引の同期化には、バックグラウンドの ctxsrv サーバーは必要ありません。ctxsrv デーモンが実行中の場合、索引をすぐに同期化します。

参照： ALTER INDEX コマンドの構文の詳細は、『Oracle8i interMedia Text リファレンス』を参照してください。

例

次の例では、2MB の実行時メモリーで、索引をバッチで同期化します。

```
ALTER INDEX newsindex rebuild PARAMETERS('sync memory 2M');
```

8.1.6

ALTER INDEX の使用に加えて、PL/SQL の CTX_DDL.SYNC を使用して索引を最適化できます。

参照： CTX_DDL.SYNC を使用する索引同期化の詳細は、『Oracle8i
interMedia Text リファレンス』を参照してください。

ストップリストおよびストップワード

8.1 より前

8.1 より前では、ストップリストは索引付けされないワードで構成されています。これらのワードは、各ストップワードに対して CTX_DDL.SET_ATTRIBUTE をコールし、次に CTX_DDL.CREATE_PREFERENCE でストップリスト・プリファレンスを作成することによって記録されます。

サポートされているほぼすべての言語でデフォルトのストップリストが使用できます。使用する言語へストップリストを手動で設定します。

8.1

ストップリスト

ストップリストは、索引付けされないワードで構成されています。定義は、8.1 では変更されていません。

デフォルトのストップリスト

デフォルトでは、データベースの設定で指定した言語にデフォルトのストップリストが設定されます。ストップリストは、カスタマイズしない限り、作成または設定の必要はありません。

ストップテーマおよびストップクラス

8.1 では、ユーザー独自のストップワードを定義する以外に、索引付けされないテーマであるストップテーマも定義できます。これは英語にのみ使用できます。

数を索引付けしないように指定することもできます。索引付けされない数などの英数字のクラスが、ストップクラスです。

1つのストップリストを作成して、それにストップワード、ストップテーマおよびストップクラスを追加することによって、ユーザー独自のストップワード、ストップテーマ、ストップクラスを記録します。CREATE INDEX の *paramstring* にストップリストを指定します。

新しいプロシージャ

8.1 では、ストップワード、ストップテーマおよびストップクラスの管理に次のプロシージャを使用します。

- CTX_DDL.CREATE_STOPLIST
- CTX_DDL.ADD_STOPWORD
- CTX_DDL.ADD_STOPTHEME
- CTX_DDL.ADD_STOPCLASS
- CTX_DDL.REMOVE_STOPWORD
- CTX_DDL.REMOVE_STOPTHEME
- CTX_DDL.REMOVE_STOPCLASS
- CTX_DDL.DROP_STOPLIST

参照： これらのコマンドの使用については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

ドキュメント・セクション

索引付けする前にドキュメント・セクションを定義することによって、WITHIN 演算子を使用するそのセクション内での問合せが可能になります。セクションをセクション・グループの一部として定義します。

8.1 より前

8.1 より前では、セクション・グループを作成し、それをワードリスト・プリファレンスに指定します。ユーザー定義ゾーン・セクションと、文章および段落セクションのみを作成できます。

8.1

セクション・グループ

8.1 では、セクション・グループを作成し、それを CREATE INDEX に対する *paramstring* に指定します。セクション・グループを作成するには、CTX_DDL.CREATE_SECTION_GROUP を使用します。

参照： CTX_DDL.CREATE_SECTION_GROUP の使用の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

異なるタイプのセクション・グループを作成できます。基本セクション・グループを作成して独自のセクションを定義するか、索引作成中に HTML または XML ドキュメントからセクションを自動的に作成するセクション・グループを作成できます。

基本セクション・グループ内で、3 種類のセクションを作成できます。

- [ゾーン・セクション](#) (旧ユーザー定義セクション)
- [フィールド・セクション](#) (新規)
- [特殊セクション](#) (文章および段落セクション)

ゾーン・セクション

ゾーン・セクション (8.1 より前ではユーザー定義セクション) は、開始タグおよび終了タグで区切られたセクションです。たとえば、HTML の および タグは、ボールド体で表現されるワードの範囲をマークします。

ゾーン・セクションは、ゾーン・セクション内でネストやオーバーラップが可能で、1 つのドキュメント内で複数回出現させることが可能です。

ゾーン・セクションは、CTX_DDL.ADD_ZONE_SECTION でセクション・グループの一部として作成します。

参照： CTX_DDL.ADD_ZONE_SECTION の使用の詳細は、『Oracle8i
interMedia Text リファレンス』を参照してください。

フィールド・セクション

フィールド・セクションは、8.1 の新機能です。フィールド・セクションは、開始タグおよび終了タグで区切られています。デフォルトでは、フィールド・セクション内のテキストは残りのドキュメントとは別のサブドキュメントとして索引付けされます。

ゾーン・セクションとは異なり、フィールド・セクションはネストまたはオーバーラップできません。このため、フィールド・セクションは非繰返しセクション、非オーバーラップ・セクション（ニュース型ドキュメントの TITLE、AUTHOR セクションなど）に最適です。

フィールド・セクションの索引付け方法によって、フィールド・セクションでの WITHIN 問合せは、通常、ゾーン・セクションの WITHIN 問合せより高速です。

CTX_DDL.ADD_FIELD_SECTION プロシージャを使用して、フィールド・セクションをセクション・グループの一部として作成します。

参照： CTX_DDL.ADD_FIELD_SECTION の使用の詳細は、『Oracle8i
interMedia Text リファレンス』を参照してください。

特殊セクション

8.1 では、特殊セクションは、8.1 より前での段落および文章セクションと同一です。

文章および段落セクションを作成するには、CTX_DDL.ADD_SPECIAL_SECTION プロシージャを使用します。

参照： CTX_DDL.ADD_SPECIAL_SECTION の使用の詳細は、『Oracle8i
interMedia Text リファレンス』を参照してください。

8.1.6 でのドキュメント・セクションの改善

次の項では、リリース 8.1.6 での拡張について説明します。ご使用の 8.1 より前のドキュメント・セットが基本的に XML ドキュメントである場合、自動セクショナーおよび属性セクション機能の使用を検討してください。

自動セクションナー

8.1.6 では、*interMedia Text* には、自動的に XML ドキュメントをセクション化する新しい自動セクションナー `AUTO_SECTION_GROUP` があります。これによって、開始タグ / 終了タグの各組にゾーン・セクションが作成されます。

属性セクション

XML ドキュメントのセクション化に `XML_SECTION_GROUP` を使用する場合、属性テキストをセクションとして定義できます。属性セクションを定義すると、`WITHIN` 演算子を使用して XML 属性テキストを検索できます。

属性セクションは、`AUTO_SECTION_GROUP` を使用する場合も自動的に定義されます。

ドキュメント・タイプ別セクション

`XML_SECTION_GROUP` を使用する場合、ドキュメント・タイプ別セクションを作成できます。このようなセクションを XML ドキュメントに対して作成すると、`WITHIN` セクション検索を、特定のドキュメント・タイプに制限できます。

META タグの索引付け

`HTML_SECTION_GROUP` を使用する場合、META タグの `NAME` および `CONTENT` 属性を 0（ゼロ）またはフィールド・セクションとして定義できます。これによって、`WITHIN` 演算子を使用して `NAME` および `CONTENT` 属性を検索できます。

セクションの動的な追加

索引付け後、SQL コマンド `ALTER INDEX` を使用してセクション（ゾーン、フィールドおよび属性）を追加できます。ただし、新しいセクションは、ドキュメント・セットを再索引付けするまで問合せできません。

参照： この項で説明した機能については、『Oracle8i *interMedia Text* リファレンス』を参照してください。

この章では、8.1 より前の問合せを *interMedia Text 8.1* に移行する方法について説明します。
この章の内容は次のとおりです。

- テキスト問合せの概要
- テキスト問合せ
- カーソル問合せ
- 構造化テキスト問合せ
- テーマ問合せ（英語のみ）
- コンポジット・テキストキーの問合せ
- MAX 演算子および FIRST/NEXT 演算子
- PL/SQL 演算子
- ヒット数のカウント
- ストアド・クエリー式
- 問合せ実行計画

テキスト問合せの概要

基本的な *interMedia Text* 問合せでは、問合せ式（演算子を使用するものもしないものも）を入力として与えます。Oracle は、各ドキュメントの関連性スコアとともに、その式を満たすすべてのドキュメント（すでに索引付けされた）を戻します。スコアは、ドキュメントを結果セットに順序付けるために使用できます。

テキスト問合せは、複数の CONTAINS 句および複数の構造化句を含むことができます。

基本的なテキスト問合せは、8.1 では変更されていません。

テキスト問合せ式

サポートされなくなったいくつかの演算子（この章で説明します）を除き、8.1 の基本的なテキスト問合せ式の構文（単一引用符で囲まれたものすべて）は、8.1 より前と同じです。

テキスト問合せ方法

8.1 より前では、3 つの方法のうちの 1 つを使用した問合せが実行可能でした。3 つの方法とは、1 段階問合せ、2 段階問合せまたはカーソル問合せ（以前のインメモリ問合せ）です。

8.1 では、Oracle は、結果およびベース表の結合が続く PL/SQL の CONTAINS プロシージャを使用する 2 段階方法をサポートしません。2 段階問合せの機能は、SELECT 文を使用する新しいのテキスト問合せで使用可能です。

8.1 では、WHERE 句で CONTAIN 演算子を使用する標準 SQL の SELECT 文が、唯一の問合せ方法です。この問合せは標準 SQL であるため、SELECT 文を使用できる場所（PL/SQL カーソルなど）でこの問合せをプログラマ的に使用できます。

テキスト問合せ

8.1 では、8.1 より前の 2 段階問合せがテキスト問合せに置き換えられています。テキスト問合せは、単一の SELECT 文で実行する場合は、8.1 より前の 1 段階問合せと似ています。また、新しい 8.1 のテキスト問合せは結果表を使用しません。

この項では、2 段階問合せを新しいテキスト問合せに移行する方法について説明します。

8.1 より前の方法

8.1 より前の方法では、次のように結果表を作成します。

```
create table CTX_TEMP(  
    textkey varchar2(64),  
    score number,  
    conid number);
```

または、CTX_QUERY.GETTAB を使用して結果表を作成することもできます。

CONTAINS プロシージャを次のように実行します。

```
execute ctx_query.contains('ARTICLE_POLICY','petroleum','CTX_TEMP');
```

次に、結果表をベース表に結合し、次のようにドキュメント・テキストを取り出します。

```
SELECT SCORE, title  
FROM CTX_TEMP, TEXTTAB  
WHERE texttab.PK=ctx_temp.textkey  
ORDER BY SCORE DESC;
```

8.1 の方法

SQL の例

SELECT 文では、CONTAINS 演算子を使用した WHERE 句で問合せを指定します。また、ヒットリストの各ヒットのスコアを戻すには、SCORE 演算子を指定します。次の例では、問合せの発行方法を示します。

```
SELECT SCORE(1) title from news
      WHERE CONTAINS(text, 'oracle', 1) > 0;
```

スコアが高いドキュメントからスコアが低いドキュメントの順に結果を順序付けするには、次のように ORDER BY 句を使用します。

```
SELECT SCORE(1), title from news
      WHERE CONTAINS(text, 'oracle', 1) > 0
      ORDER BY SCORE(1) DESC;
```

PL/SQL の例

PL/SQL アプリケーションでは、カーソルを使用して問合せの結果をフェッチできます。次の例では、NEWS 表に対して問合せを発行し、ワード *oracle* を含むすべての記事を検索します。最初にヒットする 10 個のタイトルおよびスコアは、標準出力に出力されます。

```
declare
  rowno number := 0;
begin
  for c1 in (SELECT SCORE(1) score, title FROM news
            WHERE CONTAINS(text, 'oracle', 1) > 0
            ORDER BY SCORE(1) DESC)
  loop
    rowno := rowno + 1;
    dbms_output.put_line(c1.title||': '||c1.score);
    exit when rowno = 10;
  end loop;
end;
```

この例では、カーソル FOR ループを使用して最初の 10 個のヒットを取り出します。別名の *score* は、SCORE 演算子の戻り値に対して宣言されます。スコアおよびタイトルは、カーソル・ドット表記法を使用して標準出力に出力されます。

応答時間に対してこの問合せを最適化することもできます。

参照： 応答時間に対する最適化の詳細は、この章の「[カーソル問合せ](#)」を参照してください。

カーソル問合せ

8.1 より前では、大規模なヒットリストのごく一部のみが必要な場合、テキスト問合せにカーソル問合せ（以前はインメモリ問合せ）を使用します。

8.1 では、インメモリ問合せの PL/SQL インタフェースは廃止されました。8.1 で廃止されたプロシージャは次のとおりです。

- CTX_QUERY.OPEN_CON
- CTX_QUERY.FETCH_HIT
- CTX_QUERY.CLOSE_CON
- CTX_QUERY.COUNT_LAST

8.1 より前のインメモリ問合せを移行する場合は、カーソルを使用します。SELECT 文で FIRST_ROWS ヒントを使用して、大規模なヒットリストの最初の *n* ヒットを取得します。

8.1 より前の方法

次のインメモリ問合せは、ワード *oracle* を含むすべてのドキュメントを検索し、これらのドキュメントをスコア順で戻します。問合せのメカニズムは、ヒットを行ごとに順番に戻します。つまり、このメカニズムによって、最初にヒットリスト全体を取得するオーバーヘッドなしに、最初の *n* ヒットを抽出できます。

```
declare
  pk  varchar2(80);
  scr number;
  cur number;
begin
  cur := ctx_query.open_con('mypolicy','oracle',TRUE);
  while (ctx_query.fetch_hit(cur, pk, scr) > 0)
  loop
    -- deal with hit
  end loop;
  ctx_query.close_con(cur);
end;
```

8.1 の方法

8.1 より前のカーソル問合せプロシージャ（OPEN_CON、FETCH_HIT、CLOSE_CON、COUNT_LAST）は、8.1 では廃止されました。

大規模なヒットリストの最初の *n* ヒットを取得するには、カーソルを使用して CONTAINS 問合せを実行します。次のように SELECT 文で FIRST_ROWS ヒントを使用して、応答時間を最適化します。

```
begin
  for c1 in (select /*+ FIRST_ROWS */ pk, score(1) scr
              from basetable
              where contains(textcol, 'oracle', 1) > 0
              order by scr desc)
  loop
    -- deal with hit
    dbms_output.put_line('KEY is ' || c1.pk);
    dbms_output.put_line('SCORE is ' || c1.scr);
  end loop;
end;
```

参照： CONTAINS 問合せでの FIRST_ROWS ヒントの使用については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

構造化テキスト問合せ

構造化テキスト問合せ（複合問合せともいう）とは、テキスト列を問い合わせる CONTAINS 述語および構造化データ列を問い合わせる別の述語の両方を持つ問合せのことです。

8.1 より前では、CONTAINS プロシージャへのパラメータとして構造化述語を指定していました。

8.1 では、テキスト問合せは標準 SQL を使用します。構造化問合せを発行するには、SELECT 文の WHERE 条件に構造化句を指定します。

8.1 より前の方法

例 1: CONTAINS struct_query パラメータ

構造化列を問い合わせるには、CONTAINS プロシージャで struct_query パラメータを使用します。次の例では、1996 年 10 月 1 日以後に書き込まれたワード *oracle* を含むすべての記事を戻します。

```
exec ctx_query.contains('news','oracle','res_tab',
struct_query => 'issue_date >= (''1-OCT-1996'')')
```

例 2: 2 段階結合方法

ConText の旧リリースでは、struct_query パラメータを CONTAINS プロシージャで使用できません。これらのリリースでは、結果表とベース表を結合する場合に構造化条件を指定します。

たとえば、ワード *oracle* をベース表 CTX_TEMP に対して問い合わせると、この問合せは次のようになります。

```
execute ctx_query.contains('ARTICLE_POLICY','oracle','CTX_TEMP')
```

結果表をベース表に結合する場合は、次のように構造化条件を指定してドキュメント・テキストを取り出します。

```
SELECT score, title
FROM CTX_TEMP, TEXTTAB
WHERE texttab.PK=ctx_temp.textkey
      AND texttab.issue_date >= ('01-OCT-96')
ORDER BY score DESC;
```

8.1 の方法

SELECT 文の WHERE 条件に構造化条件を指定します。次の SELECT 文は前述の問合せと同じ内容を実行します。この SELECT 文では、1997 年 10 月 1 日以後に書き込まれたワード *oracle* を含むすべての記事が戻されます。

```
SELECT SCORE(1), title, issue_date from news
      WHERE CONTAINS(text, 'oracle', 1) > 0
      AND issue_date >= ('01-OCT-97')
      ORDER BY SCORE(1) DESC;
```

テーマ問合せ（英語のみ）

テーマ問合せは概念を問い合わせます。通常、問合せ文字列は、検索する観念を表す概念またはテーマを指定します。Oracle はテーマを含むドキュメントを戻します。

8.1 より前では、最初にテーマ・ポリシーを作成してテーマ問合せを発行し、別のテーマ索引を作成します。次に、テーマ・ポリシーを CONTAINS プロシージャに指定します。

8.1 では、単一のテキスト索引にワードおよびテーマ情報が含まれます。ABOUT 演算子を使用してテーマ問合せを発行します。

8.1 より前の方法

テーマ問合せを発行するには、まずテキスト列に対応付けられたテーマ・レクサーを持つポリシーで、テキスト列を索引付けします。問合せを発行するには、同じテーマ・ポリシーおよびテーマ問合せの文字列を指定します。

たとえば、THEME_POL がテーマ・ポリシーとします。次のように 2 段階問合せを使用して、*insects* のテーマについてのドキュメントをすべて取り出します。

```
execute ctx_query.contains('THEME_POL', 'insects', 'CTX_TEMP');
```

8.1 の方法

ワード情報およびテーマ情報は、1 つの索引に結合されています。テーマ問合せを発行するには、索引にテーマ・コンポーネントが必要です。

参照： テーマ・コンポーネントを索引に作成する方法については、[第 5 章「索引付け」](#)を参照してください。

問合せ式内に ABOUT 演算子を使用して、テーマ問合せを発行します。たとえば、*insects* についてのすべてのドキュメントを取り出すには、次のように問合せを記述します。

```
SELECT SCORE(1), title FROM news
      WHERE CONTAINS(text, 'about (insects)', 1) > 0
      ORDER BY SCORE(1) DESC;
```

参照： ABOUT 演算子の使用方法については、『Oracle8i *interMedia* Text リファレンス』を参照してください。

コンポジット・テキストキーの問合せ

コンポジット・テキストキー問合せは、コンポジット・テキストキーで索引付けされたベース表を問い合わせます。

8.1 より前の方法

コンポジット・テキストキー問合せを発行するには、次のように、まず2つの列からなるコンポジット・テキストキーを持つ結果表を手動で作成します。

```
create table CTX_TEMP2(  
    textkey varchar2(64),  
    textkey2 varchar2(64),  
    score number,  
    conid number);
```

次に、WHERE 条件で AND 演算子を使用して、結果表およびベース表を結合します。次に例を示します。

```
exec ctx_query.contains('ARTICLE2_POLICY','petroleum','CTX_TEMP2')  
SELECT score, title  
FROM CTX_TEMP2, TEXTTAB2  
WHERE texttab2.PK=ctx_temp2.textkey AND  
      texttab2.PK2=ctx_temp2.textkey2  
ORDER BY score DESC;
```

8.1 の方法

8.1 の問合せは基本的な SELECT 文です。結果表は使用されないため、8.1 より前の CONTAINS のような結果表とベース表の結合はありません。つまり、単一系列のテキストキーを持つ表に対して問合せを発行したのと同じように、コンポジット・テキストキー表に対して問合せを発行します。

texttab2 がコンポジット・テキストキー表の場合、前述の問合せは次のように記述されます。

```
SELECT SCORE(1),title FROM texttab2  
WHERE CONTAINS(text,'petroleum') > 0  
ORDER BY SCORE(1) DESC;
```

MAX 演算子および FIRST/NEXT 演算子

MAX および FIRST/NEXT 結果セット演算子は、Oracle8i *interMedia* Text リリース 8.1 ではサポートされません。

8.1 より前の方法

MAX

MAX 演算子を使用して、問合せ結果セット内での高いスコアのドキュメントを指定した数だけ取り出します。たとえば、ワード `dog` が含まれているドキュメントのうち、スコアが上位 20 までのドキュメントを取得するには、次のように記述します。

```
'dog:20'
```

FIRST/NEXT

FIRST/NEXT 演算子を使用して、ソートされていない問合せ結果セット内でのドキュメントの範囲を取得します。たとえば、ワード `dog` が含まれているドキュメントのうち、11 から 20 までのドキュメントを取得するには、次のように記述します。

```
'dog#11-20'
```

8.1 の方法

MAX 演算子および FIRST/NEXT 演算子は、Oracle8i *interMedia* Text リリース 8.1 ではサポートされません。PL/SQL の応答時間を最適化するカーソル問合せを使用して、MAX 型または FIRST/NEXT 型の問合せに対する結果を得ることができます。

MAX に対するソリューション

応答時間を最適化する問合せによって、ヒットリストからスコアの高いドキュメントが必要な場合に対するソリューションが提供されます。

次の例では、最初の 20 ヒットを標準出力に戻します。この例では、FIRST_ROWS ヒントおよびカーソルを使用します。

```
declare
cursor c is
  select /*+ FIRST_ROWS */ title, score(1) score
    from news
   where contains(txt_col, 'dog', 1) > 0
   order by score(1) desc;
begin
  for c1 in c
  loop
    dbms_output.put_line(c1.score||':'||substr(c1.title,1,50));
    exit when c%rowcount = 20;
  end loop;
end;
/
```

参照： 応答時間を最適化する問合せの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

FIRST/NEXT に対するソリューション

応答時間を最適化する問合せによって、スコアでソートされたヒットリストからドキュメントの一部が必要な場合のソリューションが提供されます。

このソリューションは、カーソルで FIRST_ROWS ヒントを使用する MAX ドキュメントのソリューションに似ています。コードはカーソル内をループし、要求された範囲のヒットのみを処理します。次の例では、ソートされたドキュメント 11 ～ 20 を標準出力に戻します。

```
declare
cursor c is
  select /*+ FIRST_ROWS */ title, score(1) score
    from news
   where contains(txt_col, 'dog', 1) > 0
   order by score(1) desc;
begin
  for c1 in c
  loop
    if (c%rowcount > 10) then
      dbms_output.put_line(c1.score||':'||substr(c1.title,1,50));
    end if;
    exit when c%rowcount = 20;
  end loop;
end;
/
```

参照： 応答時間を最適化する問合せの詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

PL/SQL 演算子

Oracle8i *interMedia* Text リリース 8.1 では、問合せで PL/SQL をコールする EXECUTE 演算子はサポートされません。

interMedia Text が Oracle8 に統合された結果、標準 SQL を使用できるようになり、これによって、ファンクションが SQL 文で指定された要件を満たす場合に、SELECT 文内でファンクションをコールできます。

8.1 より前の方法

問合せ内での PL/SQL ファンクションのコールは、別の形式にワードを変換する場合に有効です。たとえば、ファンクション *french* が英語のワードと等価なフランス語を戻すとします。次を発行して、*cat* に対するフランス語のワードを *ctxuser* として検索できます。

```
'@ctxuser.french(cat)'
```

8.1 の方法

ファンクションが SQL 文で指定された要件を満たす限り、ユーザー・ファンクションを CONTAINS 句で直接コールできます。また、コールした側にはファンクションの EXECUTE 権限が必要です。

たとえば、ファンクション *french* が英語のワードと等価なフランス語を戻すとします。次のように記述すると、*cat* に対するフランス語のワードを検索できます。

```
SELECT SCORE(1), title from news
      WHERE CONTAINS(text, french('cat'), 1) > 0
      ORDER BY SCORE(1);
```

ファンクションを次のようにバインド変数として CONTAINS に渡すことによって、パフォーマンスを向上できます。

```
variable qry varchar2(80);
exec :qry := french('cat');

SELECT SCORE(1), title from news
      WHERE CONTAINS(text, :qry, 1) > 0
      ORDER BY SCORE(1);
```

参照： ユーザー・ファンクションの作成および SQL からのユーザ・ファンクションのコールについては、『Oracle8i SQL リファレンス』を参照してください。

ヒット数のカウント

Oracle8i *interMedia* Text リリース 8.1 では、CTX_QUERY.COUNT_HITS ファンクションがサポートされています。8.1 より前では、このファンクションは、問合せの発行前に問合せでヒット数をカウントするために使用されます。ただし、8.1 では、ポリシーではなく索引名を指定します。また、8.1 より前では構造化述語の指定に使用されていた `struct_query` パラメータは廃止されました。

8.1 では、CONTAINS 述語のみでの問合せから戻されたヒット数をカウントするために、SELECT 文で CTX_QUERY.COUNT_HITS または COUNT(*) を使用できます。

構造化述語を含む問合せから戻されたヒット数をカウントするには、SELECT 文で COUNT(*) 関数を使用します。

インメモリ問合せは 8.1 では廃止されているため、8.1 より前の、CTX_QUERY.COUNT_LAST プロシージャも 8.1 では廃止されています。

8.1 より前の方法

問合せのヒットは、次のように COUNT_HITS を使用してカウントします。

```
declare count number;
begin
  count := ctx_query.count_hits(policy_name => my_pol, text_query => 'oracle',
                                exact => TRUE);
  dbms_output.put_line('Number of docs with oracle:');
  dbms_output.put_line(count);
end;
```

8.1 の方法

CONTAINS 述語のみ

ワード *oracle* を含むドキュメント数を検索するには、次のいずれかを行います。

- 次のように SQL の COUNT 関数で問合せを発行します。

```
SELECT count(*) FROM news WHERE CONTAINS(text, 'oracle', 1) > 0;
```

- 次のように COUNT_HITS を使用します。

```
declare count number;  
begin  
  count := ctx_query.count_hits(index_name => my_index, text_query => 'oracle',  
                                exact => TRUE);  
  dbms_output.put_line('Number of docs with oracle:');  
  dbms_output.put_line(count);  
end;
```

参照： CTX_QUERY.COUNT_HITS の構文の詳細は、『Oracle8i
interMedia Text リファレンス』を参照してください。

構造化述語

構造化述語での問合せによって戻されるドキュメント数を検索するには、次のように COUNT(*) を使用します。

```
SELECT count(*) FROM news WHERE CONTAINS(text, 'oracle', 1) > 0 and author =  
'jones';
```

ストアド・クエリー式

8.1 より前では、定義および問合せの結果を格納できます。そのため、問合せ式に SQE 演算子を使用して結果を取得できます。ワイルド・カード問合せなどの問合せに対してストアド・クエリー式を使用すると、結果が格納されるため、パフォーマンスが向上します。

8.1 では、プロシージャ CTX_QUERY.STORE_SQE は問合せの定義のみを格納します。結果は格納されません。SQE 演算子での問合せの参照は、問合せの定義を参照するのみです。このようにして、SQE は長い問合せ式または頻繁に使用される問合せ式の定義を容易にします。

ストアド・クエリー式は索引に連結されていません。CTX_QUERY.STORE_SQE をコールする場合は、ストアド・クエリー式および問合せ式の名前のみを指定します。

8.1 より前のセッション SQE の概念はなくなりました。問合せの定義はテキスト・データ・ディクショナリに格納されます。どのユーザーもストアド・クエリー式を参照できます。

参照： CTX_QUERY.STORE_SQE の構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

REFRESH_SQE および PURGE_SQE の管理プロシージャは、*interMedia* Text 8.1 では廃止されました。

8.1 より前の方法

8.1 より前では、ストアド・クエリー式を次のように定義および使用します。

1. テキスト列またはポリシーに対する結果を格納するために、CTX_QUERY.STORE_SQE をコールします。STORE_SQE を使用して、SQE の名前、(SQE に対するテキスト列を識別する) ポリシー、問合せ式を指定します。また、SQE がセッション SQE かまたはシステム SQE かを指定します。
2. テキスト (またはテーマ) 問合せの問合せ式内で、ストアド・クエリー式をコールします。ConText は、標準の問合せ結果を戻すのと同じ方法で SQE の結果を戻します。SQE の結果が古い場合、ConText は結果を戻す前に自動的に SQE を再評価します。

ストアド・クエリー式の管理は、PL/SQL パッケージ CTX_QUERY の REFRESH_SQE、REMOVE_SQE および PURGE_SQE プロシージャを使用して行います。

8.1 より前の例

PROG_LANG という名前のセッション SQE を作成するには、次のように CTX_QUERY.STORE_SQE を使用します。

```
exec ctx_query.store_sqe('emp_resumes', 'prog_lang', 'computer science', 'session');
```

この SQE が EMP_RESUMES ポリシーのテキスト列を問い合わせ、語句 *computer science* が含まれているすべてのドキュメントを戻します。結果はポリシーの SQE 表に格納されます。

これで、*prog_lang* は次のような問合せ式内でコールできます。

```
select score, docid from emp
where contains(resume, 'sqe(prog_lang)')>0
order by score;
```

8.1 の方法

ストアド・クエリー式を次のように定義および使用します。

1. テキスト列に対する結果を格納するために、CTX_QUERY.STORE_SQE をコールします。STORE_SQE を使用して、SQE および問合せ式の名前を指定します。SESSION パラメータおよび SYSTEM パラメータは廃止されました。
2. SQE 演算子を使用して、問合せ式でストアド・クエリー式をコールします。Oracle は、標準の問合せ結果を戻す場合と同じ方法で SQE の結果を戻します。問合せは SQE がコールされたときに評価されます。

プロシージャ REFRESH_SQE および PURGE_SQE は廃止されました。REMOVE_SQE を使用して削除します。

例

次の例では、ワード *tornado*、*hurricane* または *earthquake* を含むドキュメントを検索する、*disaster* というストアド・クエリー式を作成します。

```
begin
ctx_query.store_sqe('disaster', 'tornado | hurricane | earthquake');
end;
```

式でこの問合せを実行するには、次のように問合せを記述します。

```
SELECT SCORE(1), title from news
WHERE CONTAINS(text, 'SQE(disaster)', 1) > 0
ORDER BY SCORE(1);
```

参照： CTX_QUERY.STORE_SQE の構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

問合せ実行計画

問合せ実行計画（以前は問合せ式フィードバック）では、実際に問合せを発行する前に、テキスト問合せの実行計画を取得できます。Oracle は、表から解析ツリーを構築できる表に実行計画情報を戻します。

interMedia Text 8.1 では、この機能が問合せ式フィードバックから問合せ実行計画に改名されました。階層問合せフィードバックは新しい異なる機能であり、問合せ実行計画とは異なることに注意してください。

問合せ計画のユーザー・インタフェースは、次のように変更されました。

- プロシージャ `CTX_QUERY.FEEDBACK` は、`CTX_QUERY.EXPLAIN` に改名されました。
- `CTX_QUERY.EXPLAIN` では、ポリシーではなく索引名を指定します。
- 結果表の `FEEDBACK_ID` 列は、`EXPLAIN_ID` に改名されました。
- 実行計画表の `OPERATION` 列は、新しい `ABOUT` 演算子の値を持ちます。
- `FIRST_NEXT_DOC` 演算子および `MAX_DOC` 演算子の列の値は削除されました。
- `WEIGHT` および `THRESHOLD` に対応付けられた係数は、`OPTIONS` 列から `OBJECT_NAME` 列に移動しました。

参照： `CTX_QUERY.EXPLAIN` の構文および実行計画表の構造の詳細は、『*Oracle8i interMedia Text* リファレンス』を参照してください。

実行計画情報の取得

`CTX_QUERY.EXPLAIN` を使用して、式のフィードバックを取得します。この情報を取得するためのプロシージャは、8.1 より前から変更されていません。次のことを行う必要があります。

1. 実行計画表を作成します。
2. `CTX_QUERY.EXPLAIN` を実行します。
3. 実行計画表からデータを取り出します。
4. オプションとして、表情報から拡張ツリーを組み立てます。

最初の3つの手順をコード化する場合は、新しい8.1の`EXPLAIN`構文を使用する必要があります。

手順 4 の拡張ツリーを実行計画表から組み立てる方法は、8.1 より前の方法と同じです。

参照： 拡張ツリーの組立て例は、『Oracle8i *interMedia* Text リファレンス』の CTX_QUERY.EXPLAIN コマンド構文を参照してください。

ドキュメントの表示方法

この章では、ドキュメントの表示方法の移行方法について説明します。この章の内容は次のとおりです。

- [ハイライト表示](#)
- [テーマのリスト、要旨およびテーマ・サマリーの取得](#)

ハイライト表示

interMedia Text 問合せアプリケーションでは、テキスト問合せに対してハイライト表示した問合せ語句、または ABOUT 問合せに対してハイライト表示したテーマとともに選択されたドキュメントを表示できます。

ハイライト表示に対応付けられた3つのタイプの出力を生成できます。それらは、ドキュメントのマークアップ形式、ドキュメントのプレーン・テキスト形式（フィルタ処理された出力）およびドキュメントのハイライト・オフセット情報です。

8.1 より前では、プロシージャ `CTX_QUERY.CTX_HIGHLIGHT` を使用して前述の3つのタイプの出力、つまり、ドキュメントのマークアップ形式、ドキュメントのプレーン・テキスト形式（フィルタ処理された出力）およびドキュメントのハイライト・オフセット情報を生成しました。

interMedia Text 8.1 では、これらの3つのタイプの出力は、`CTX_DOC`（ドキュメント・サービス）パッケージの異なる3つのプロシージャによって生成されます。また、各タイプの出力に対するプレーン・テキスト形式および `HTML` 形式も生成できます。

8.1 でこの出力を格納するために使用する結果表も、8.1 より前の結果表と異なります。

interMedia Text 8.1 では、テーマ・ハイライトに対する出力が8.1 より前と異なります。8.1 より前では、システムが問合せを最も的確に表しているドキュメントの段落をハイライト表示します。*interMedia Text* 8.1 では、ワードや句などの個々のテーマがハイライト表示されます。

8.1 より前の方法

`CTX_QUERY.HIGHLIGHT` を使用して、ハイライト情報、マークアップされたドキュメントおよびフィルタ処理されたドキュメントを取得します。

たとえば、テキストキー 14 で識別されるドキュメントで語句 *dog*（犬）のすべてのオカレンスをハイライト表示するには、次の文を発行します。

```
ctx_query.highlight (
    cspec=> 'text_policy',
    textkey => '14',
    query => 'dog',
    id=> 14,
    hightab => 'highlight_ascii',
    mutab   => 'mu_ascii' );
```

この例では、オフセット情報を `HIGHTAB` 表に格納し、ハイライト表示されたマークアップ・ドキュメントを `MU_ASCII` 表に格納します。

新しい 8.1 のソリューション

テキスト・ハイライト

テキスト・ハイライトに対する動作は 8.1 より前と同じです。問合せを指定すると、Oracle が問合せを満たすドキュメントのワードをハイライト表示します。プレーン・テキストまたは HTML でハイライト表示を取得できます。

テーマ・ハイライト

テーマ問合せでは、*interMedia Text 8.1* はテーマ問合せを最も的確に表すワードまたは句をハイライト表示およびマークアップします。この動作は、テーマ問合せで段落がハイライト表示される 8.1 より前とは異なります。

ハイライト・プロシージャ

ハイライト・オフセット情報は、ドキュメントを表示するためのユーザー独自のカスタム・ルーチンを作成する場合に有効です。

ハイライト・オフセット情報を取得するには、CTX_DOC.HIGHLIGHT プロシージャを使用します。このプロシージャは問合せおよびドキュメントを取得し、プレーン・テキストまたは HTML 形式のいずれかに対するハイライト・オフセット情報を戻します。

オフセット情報によって、ハイライト表示されたドキュメントを自由に表示できます。たとえば、CTX_DOC.MARKUP から取得される標準プレーン・テキストのマークアップを使用せずに、異なるフォント・タイプまたは色で自由にハイライト表示できます。

参照： CTX_DOC.HIGHLIGHT の使用の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

マークアップ・プロシージャ

CTX_DOC.MARKUP プロシージャはドキュメントの参照および問合せを取り、ドキュメントのマークアップ形式を戻します。出力は、マークアップされたプレーン・テキストまたはマークアップされた HTML のいずれかです。

8.1 では、HTML ナビゲーションに対するマークアップ・シーケンスをカスタマイズできます。

参照： CTX_DOC.MARKUP の詳細は、『Oracle8i *interMedia Text* リファレンス』を参照してください。

フィルタ・プロシージャ

ドキュメントが Microsoft Word など固有の形式で格納されている場合、フィルタ・プロシージャ CTX_DOC.FILTER を使用して、ドキュメントのプレーン・テキスト形式または HTML 形式のいずれかを取得できます。

参照： CTX_DOC.FILTER の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

8.1.6 の機能

リリース 8.1.6 では、リリース 8.1.5 のハイライト表示、フィルタ処理およびマークアップ機能のすべてを使用できます。

8.1.6 では、ドキュメント・サービスに次の新しい機能が追加されています。

- 主キーの他に、ROWID でドキュメントを識別できます。
- 結果をメモリー内に格納して、パフォーマンスを向上できます。

参照： ROWID 入力および結果のメモリー内格納の使用例は、『Oracle8i *interMedia* Text リファレンス』の CTX_DOC パッケージを参照してください。

テーマのリスト、要旨およびテーマ・サマリーの取得

8.1 では、次のように変更されています。

- CTX_LING パッケージはサポートされなくなりました。テーマのドキュメント・サービス出力のリスト、テーマ・サマリーおよび要旨は、CTX_DOC パッケージから取得します。
- CTX_LING.REQUEST_THEMES および CTX_LING.REQUEST_GIST は、新しいプロシージャ CTX_DOC.THEMES および CTX_DOC.GIST に置き換えられました。8.1 の新しいプロシージャには異なる仕様があり、結果表には異なるスキーマがあります。
- Oracle は現在、ドキュメント・サービス要求を同期的に処理するため、サービス・キューに要求を送る必要はありません。
- CTX_DOC.GIST をコールするときに、要旨またはテーマ・サマリーのサイズを指定できます。

8.1 では、ドキュメント・サービスに次の新しい機能が追加されています。

- 主キーの他に、ROWID でドキュメントを識別できます。
- 結果をメモリー内に格納して、パフォーマンスを向上できます。

次の表では、テーマのリスト、要旨およびテーマ・サマリーを説明します。これらの定義は 8.1 では変更されていません。

表 7-1

出力タイプ	説明
テーマのリスト	ドキュメントの主な概念のリスト 各テーマが単一のワードまたは句であるか、または各テーマが親テーマの階層リストであるテーマのリストを生成できます。
要旨	ドキュメント全体の内容を最も的確に表しているテキスト
テーマ・サマリー	ドキュメントで指定したテーマを最も的確に表しているテキスト

8.1 より前の方法

出力表の作成

テーマのリスト、テーマ・サマリーまたは要旨を生成する前に、CTX_LING 出力を格納するために結果表を作成する必要があります。

REQUEST_THEMES からテーマのリストを格納するために、CTX_THEMES というテーマ表を作成するには、次の SQL 文を発行します。

```
create table ctx_themes (  
    cid          number,  
    pk           varchar2(64),  
    theme        varchar2(2000),  
    weight       number);
```

REQUEST_GIST から要旨またはテーマ・サマリーを格納するために、CTX_GIST という要旨表を作成するには、次の SQL 文を発行します。

```
create table ctx_gist (  
    cid          number,  
    pk           varchar2(64),  
    pov          varchar2(80),  
    gist         long);
```

テーマのリスト

CTX_LING.REQUEST_THEMES を使用してテーマを生成します。

例 次の無名 PL/SQL ブロックは、CTX_LING.REQUEST_THEMES をコールし、次に CTX_LING.SUBMIT をコールすることによって、ドキュメント 20 に対するテーマのリストを生成します。

```
declare handle number;  
begin  
    ctx_ling.request_themes('CTXSYS.DOC_POLICY','20','CTX_THEMES');  
    handle := ctx_ling.submit;  
end;
```

テーマ・サマリーおよび要旨

CTX_LING.REQUEST_GIST を使用して、テーマ・サマリーおよび要旨を生成します。

例 次の無名 PL/SQL ブロックは、ドキュメント 20 に対するテーマ *insects* のテーマ・サマリーを生成します。テーマ・サマリーは、CTX_LING.REQUEST_GIST をコールし、次に CTX_LING.SUBMIT をコールすることによって生成されます。

```
declare handle number;
begin
  ctx_ling.request_gist('CTXSYS.DOC_POLICY','20','CTX_GIST',
                      'PARAGRAPH', 'insects');
  handle := ctx_ling.submit;
end;
```

全テーマ出力

リストの各エレメントが親テーマの階層リストである、テーマのリストを取得できます。これを行うには、次の文を発行します。

```
SQL> exec ctx_ling.set_full_themes(TRUE)
SQL> exec ctx_ling.request_themes('ctx_thidx', pk, 'ctx_themes')
SQL> exec ctx_ling.submit(200)
```

要旨サイズの変更

ConText ワークベンチ管理ツールを使用して、デフォルトの要旨サイズを変更します。

新しい 8.1 のソリューション

CTX_LING パッケージはサポートされなくなりました。要旨およびテーマ生成プロシージャは、CTX_DOC パッケージにあります。要求は同期であるため、明示的にドキュメント・サービス要求を送る必要はありません。サーバーを稼働する必要はありません。

テーマのリスト

テーマのリストは、ドキュメントの主な概念のリストです。

CTX_DOC.THEMES プロシージャを使用してテーマのリストを生成します。

参照： CTX_DOC.THEMES のコマンド構文の詳細は、『*Oracle8i interMedia Text リファレンス*』を参照してください。

テーマ表 テーマ表を作成するには、次の文を発行します。

```
create table ctx_themes (query_id number,
                        theme varchar2(2000),
                        weight number);
```

単一テーマ リストの各エレメントが単一テーマであるテーマのリストを取得するには、次の文を発行します。

```
begin
ctx_doc.themes('newsindex',34,'CTX_THEMES',1,full_themes => FALSE);
end;
```

全テーマ リストの各エレメントが親テーマの階層リストであるテーマのリストを取得するには、次の文を発行します。

```
begin
ctx_doc.themes('newsindex',34,'CTX_THEMES',1,full_themes => TRUE);
end;
```

要旨およびテーマ・サマリー

要旨およびテーマ・サマリーの定義は 8.1 で変更されていません。要旨は、ドキュメント全体の内容を最も的確に表しているテキストです。テーマ・サマリーは、ドキュメントの単一テーマを最も的確に表しているテキストです。

8.1 では、プロシージャをコールするときに、要旨またはテーマ・サマリーのサイズを指定できます。

CTX_DOC.GIST プロシージャを使用して、要旨およびテーマ・サマリーを生成します。

参照： CTX_DOC.GIST のコマンド構文の詳細は、『Oracle8i *interMedia* Text リファレンス』を参照してください。

要旨表 要旨表を作成するには、次の文を発行します。

```
create table ctx_gist (query_id number,
                      pov      varchar2(80),
                      gist      CLOB);
```

要旨 次の例では、ドキュメント 34 に対するデフォルト・サイズの段落レベルの要旨を戻します。

```
begin
ctx_doc.gist('newsindex',34,'CTX_GIST',1,'PARAGRAPH', pov =>'GENERIC');
end;
```

次の例では、10 個の段落のデフォルト・サイズでない要旨を生成します。

```
begin
ctx_doc.gist('newsindex',34,'CTX_GIST',1,'PARAGRAPH', pov =>'GENERIC',
numParagraphs => 10);
end;
```

次の例では、段落数がドキュメントの合計段落数の 10 パーセントである要旨を生成します。

```
begin
ctx_doc.gist('newsindex',34,'CTX_GIST',1, 'PARAGRAPH', pov =>'GENERIC', maxPercent
=> 10);
end;
```

テーマ・サマリー 次の例では、テキストキー 34 でドキュメントの *insects* のテーマのテーマ・サマリーを戻します。デフォルトの要旨サイズが戻されます。

```
begin
ctx_doc.gist('newsindex',34,'CTX_GIST',1, 'PARAGRAPH', pov => 'insects');
end;
```

8.1.6 の機能

リリース 8.1.6 では、リリース 8.1.5 のテーマおよび要旨機能のすべてを使用できます。

8.1.6 では、ドキュメント・サービスに次の新しい機能が追加されています。

- 主キーの他に、ROWID でドキュメントを識別できます。
- 結果をメモリー内に格納して、パフォーマンスを向上できます。

参照： ROWID 入力および結果のメモリー内格納の詳細は、『*Oracle8i interMedia Text* リファレンス』の CTX_DOC パッケージを参照してください。

PL/SQL の変更

この付録では、8.1 より前の PL/SQL パッケージおよびプロシージャの変更について説明します。この付録の表には、8.1 では廃止されたパッケージおよびプロシージャも含まれています。

この付録では、8.1 より前のパッケージおよびプロシージャの変更についてのみ説明します。8.1 の新しいパッケージおよびプロシージャについては記載していません。

参照： Oracle8i *interMedia* Text リリース 8.1 の新しいものを含むすべてのプロシージャおよびパッケージの完全なリストは、『Oracle8i *interMedia* Text リファレンス』を参照してください。

この章の内容は次のとおりです。

- [CTX_ADM パッケージ](#)
- [CTX_DML パッケージ](#)
- [CTX_DDL パッケージ](#)
- [CTX_INFO パッケージ](#)
- [CTX_LING パッケージ](#)
- [CTX_QUERY パッケージ](#)
- [CTX_SVC パッケージ](#)
- [CTX_THES パッケージ](#)

CTX_ADM パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_ADM	CHANGE_MASK	廃止	サーバーは、M パーソナリティとしてのみ稼働します。 『Oracle8i interMedia Text リファレンス』を参照してください。
	GET_QUEUE_STATUS	廃止	DML キューを除くすべてのキューは廃止されました。 DML キューのステータス（使用可能または使用不可）は変更できません。
	RECOVER	変更なし	ありません。
	SET_QUERY_BUFFER_SIZE	廃止	問合せパイプは廃止されました。問合せは、Oracle によって処理されます。
	SHUTDOWN	変更なし	ありません。
	UPDATE_QUEUE_STATUS	廃止	DML キューを除くすべてのキューは廃止されました。 DML キューのステータス（使用可能または使用不可）は変更できません。

CTX_DML パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_DML	REINDEX	変更	CTX_DML パッケージは廃止されました。 SQL コマンド ALTER INDEX ... REBUILD を使用して、テキスト索引の DML を管理してください。 第 5 章の「索引の更新 - バッチ DML」 を参照してください。
	SYNC	廃止	
	SYNC_QUERY	廃止	

CTX_DDL パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_DDL	ADD_SECTION	変更	セクションの新しいタイプです。セクションの開始タグのみ指定してください。 第 5 章の「ドキュメント・セクション」 を参照してください。
	CLEAR_ATTRIBUTES	廃止	CTX_DDL.UNSET_ATTRIBUTE を使用してください。 『Oracle8i interMedia Text リファレンス』を参照してください。
	CREATE_INDEX	廃止	SQL コマンド CREATE INDEX を使用して、テキスト索引を作成してください。 第 5 章の「索引の作成」 を参照してください。
	CREATE_POLICY	廃止	ポリシーは廃止されました。テキスト列に対して索引を直接作成してください。
	CREATE_PREFERENCE	変更	プリファレンスの新しい作成方法です。 第 5 章の「プリファレンスの作成」 を参照してください。
	CREATE_SECTION_GROUP	変更	セクション・グループ名およびセクション・グループ・タイプを指定してください。 第 5 章の「ドキュメント・セクション」 を参照してください。
	CREATE_SOURCE	廃止	SQL*Loader を使用して、テキストをロードしてください。
	CREATE_TEMPLATE_POLICY	廃止	ポリシーは廃止されました。テキスト列に対して索引を直接作成してください。
	DROP_INDEX	廃止	SQL コマンド DROP INDEX を使用して、テキスト索引を削除してください。
	DROP_INTTTRIG	廃止	DML トリガーの削除はサポートされなくなりました。
	DROP_POLICY	廃止	ポリシーは廃止されました。テキスト列に対して索引を直接作成してください。
	DROP_PREFERENCE	変更なし	ありません。
	DROP_SECTION_GROUP	変更なし	ありません。
	DROP_SOURCE	廃止	テキストのロードは、ctxload でサポートされなくなりました。SQL*Loader を使用してください。

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_DDL (続き)	OPTIMIZE_INDEX	廃止	SQL コマンド ALTER INDEX ... REBUILD を使用して、テキスト索引を最適化してください。 第 5 章の「索引の最適化」 を参照してください。
	REMOVE_SECTION	変更	セクションの新しいタイプです。 『Oracle8i interMedia Text リファレンス』を参照してください。
	RESUME_FAILED_INDEX	廃止	SQL コマンド ALTER INDEX ... REBUILD を使用して、索引付け処理を再開してください。 第 5 章の「失敗した索引の再開」 を参照してください。
	SET_ATTRIBUTE	変更	プリファレンスの新しい作成方法です。 第 5 章の「プリファレンスの作成」 を参照してください。
	UPGRADE_INDEX	廃止	ありません。移行時に索引を再作成してください。
	UPDATE_POLICY	廃止	ポリシーは廃止されました。テキスト列に対して索引を直接作成してください。
	UPDATE_SOURCE	廃止	テキストのロードは、ctxload でサポートされなくなりました。SQL*Loader を使用してください。

CTX_INFO パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_INFO	GET_INFO	廃止	CTX_INFO パッケージは廃止されました。
	GET_STATUS	廃止	
	GET_VERSION	廃止	

CTX_LING パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_LING	CANCEL	廃止	<p>CTX_LING パッケージは廃止されました。サービス・キューは廃止されました。設定ラベルは廃止されました。</p> <ul style="list-style-type: none">■ 要旨およびテーマの生成は同期的に実行されま す。エラーは、リクエストに直接戻されます。■ REQUEST_GIST および REQUEST_THEMES は CTX_DOC（新しいパッケージ）に移動され、 GIST および THEMES に改名されました。■ CTX_DOC.GIST は、要旨サイズ設定を指定する 引数を含んでいます。■ CTX_DOC.THEMES は、全テーマ出力の引数を 含んでいます。 <p>第7章の「テーマのリスト、要旨およびテーマ・サマ リーの取得」を参照してください。</p>
	GET_COMPLETION_CALLBACK	廃止	
	GET_ERROR_CALLBACK	廃止	
	GET_FULL_THEMES	廃止	
	GET_LOG_PARSE	廃止	
	GET_SETTINGS_LABEL	廃止	
	REQUEST_GIST	廃止	
	REQUEST_THEMES	廃止	
	SET_COMPLETION_CALLBACK	廃止	
	SET_ERROR_CALLBACK	廃止	
	SET_FULL_THEMES	廃止	
	SET_LOG_PARSE	廃止	
	SET_SETTINGS_LABEL	廃止	
	SUBMIT	廃止	

CTX_QUERY パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_QUERY	CLOSE_CON	廃止	インメモリ問合せは廃止されました。 SELECT 文の FIRST_ROWS ヒントを使用して、インメモリ問合せを移行します。 第 6 章の「カーソル問合せ」を参照してください。
	CONTAINS	廃止	2 段階問合せはサポートされなくなりました。 CONTAINS 関数付きで SELECT 文を使用して、問合せを実行してください。 第 6 章の「テキスト問合せ」を参照してください。
	COUNT_HITS	変更	ポリシーではなく、索引を指定してください。 struct_query パラメータも廃止されました。 第 6 章の「ヒット数のカウント」を参照してください。
	COUNT_LAST	廃止	CTX_QUERY.COUNT_HITS または COUNT(*) を使用して、ヒット数をカウントしてください。 第 6 章の「ヒット数のカウント」を参照してください。
	FEEDBACK	変更	プロシージャは EXPLAIN に改名されました。プロシージャの動作は変更されていません。 第 6 章の「問合せ実行計画」を参照してください。
	FETCH_HIT	廃止	インメモリ問合せは廃止されました。 SELECT 文の FIRST_ROWS ヒントを使用して、インメモリ問合せを移行してください。 第 6 章の「カーソル問合せ」を参照してください。
	GETTAB	廃止	CREATE TABLE コマンドを使用して、結果表を作成してください。 第 7 章「ドキュメントの表示方法」を参照してください。
	HIGHLIGHT	置換え	HIGHLIGHT は、CTX_DOC（新しいパッケージ）のプロシージャによって置き換えられました。 第 7 章の「ハイライト表示」を参照してください。
	OPEN_CON	廃止	インメモリ問合せは廃止されました。 SELECT 文の FIRST_ROWS ヒントを使用して、インメモリ問合せを移行してください。 第 6 章の「カーソル問合せ」を参照してください。

2.x パッケージ	プロシージャ	状態	移行の注意
	PKDECODE	廃止	Oracle がカーソル問合せに ROWID を戻すため、コンポジット・テキストキーのデコードは必要なくなりました。 第 6 章の「カーソル問合せ」 を参照してください。
	PKENCODE	変更なし	ありません。
	PURGE_SQE	廃止	SQE の動作は変更されました。 第 6 章の「ストアド・クエリー式」 を参照してください。
	REFRESH_SQE	廃止	SQE 動作は変更されました。システムは、SQE に対する結果を格納しません。 第 6 章の「ストアド・クエリー式」 を参照してください。
	RELTAB	廃止	DROP TABLE を使用して、結果表を削除してください。
	REMOVE_SQE	変更なし	ありません。
	STORE_SQE	変更	問合せ定義のみ格納します。SQE は索引に連結されていません。 第 6 章の「ストアド・クエリー式」 を参照してください。

CTX_SVC パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_SVC	CANCEL	廃止	CTX_SVC パッケージは廃止されました。サービス・キューは廃止されました。 要旨およびテーマの生成はインラインで実行されます。エラーは、リクエストに直接戻されます。 索引エラーは、新しいビュー CTX_INDEX_ERRORS でレポートされます。
	CANCEL_ALL	廃止	
	CANCEL_USER	廃止	
	CLEAR_ALL_ERRORS	廃止	
	CLEAR_ERROR	廃止	
	CLEAR_INDEX_ERRORS	廃止	
	CLEAR_LING_ERRORS	廃止	
	REQUEST_STATUS	廃止	

CTX_THES パッケージ

2.x パッケージ	プロシージャ	状態	移行の注意
CTX_THES	CREATE_PHRASE	変更	CREATE_PHRASE はファンクションではなくなりました。現在は、他の PL/SQL プロシージャと同じ方法でコールする必要があるプロシージャです。CREATE_PHRASE の構文は変更されていません。
	CREATE_THESAURUS	変更	CREATE_THESAURUS はファンクションではなくなりました。現在は、他の PL/SQL プロシージャと同じ方法でコールする必要があるプロシージャです。CREATE_THESAURUS の構文は変更されていません。
	DROP_THESAURUS	変更なし	ありません。

数字

- 1 段階問合せ (8.1 より前), 3-7
- 2 段階問合せ
 - 移行, 6-2
- 8.1 の拡張, 1-3

A

- ABOUT 問合せ, 6-9
 - 定義, 1-2
- ACCUM 演算子
 - スコア・アルゴリズム, 1-9
- ADD_FIELD_SECTION プロシージャ, 5-17
- ADD_SPECIAL_SECTION プロシージャ, 5-17
- ADD_STOPCLASS プロシージャ, 5-15
- ADD_STOPTHEME プロシージャ, 5-15
- ADD_ZONE_SECTION プロシージャ, 5-16
- ALTER INDEX コマンド
 - 索引の再構築, 5-10
 - 索引の最適化, 5-11
 - 索引の同期化, 5-13
 - 失敗した索引の再開, 5-9
- AUTO_SECTION_GROUP, 1-7, 5-18

B

- BASIC LEXER (8.1 より前), 4-3
- BASIC_SECTION_GROUP, 1-7
- BASIC_WORDLIST, 1-8
- BASIC_WORDLIST オブジェクト, 4-4
- BFILE 列
 - ロード, 3-5
- BLASTER FILTER (8.1 より前), 4-3

- BLOB 列
 - ロード, 3-5
- BROWSE_WORDS プロシージャ, 1-9

C

- CHARSET 列, 1-7
- CHINESE V-GRAM LEXER (8.1 より前), 4-4
- CLOB 列
 - ロード, 3-5
- CLOSE_CON プロシージャ (8.1 より前), 6-6
- CONTAINS 演算子, 6-4
- ConText 問合せ
 - 移行, 6-2
- ConText ワークベンチ (8.1 より前), 3-8
- COUNT_HITS プロシージャ, 6-15
- COUNT_LAST プロシージャ (8.1 より前), 6-6, 6-15
- CREATE INDEX コマンド, 5-6
- CREATE INDEX の paramstring, 5-6
- CREATE_INDEX プロシージャ (8.1 より前), 5-6
- CREATE_PREFERENCE プロシージャ, 5-5
- CREATE_SECTION_GROUP プロシージャ, 5-16
- CREATE_STOPLIST プロシージャ, 5-15
- CTX_ADM パッケージ, A-2
- CTX_DDL パッケージ, A-4
- CTX_DML パッケージ (8.1 より前), A-3
- CTX_DOC に対する結果のメモリー内格納, 1-8
- CTX_DOC パッケージ, 7-2, 7-5
 - 8.1 での改善, 1-8
- CTX_INDEX_ERRORS ビュー, 3-6
- CTX_INFO パッケージ (8.1 より前), A-6
- CTX_LING 出力表 (8.1 より前)
 - 作成, 7-6
- CTX_LING パッケージ (8.1 より前), 7-5, A-7
- CTX_PENDING ビュー, 3-6

CTX_QUERY パッケージ, A-8
CTX_SERVERS ビュー, 3-2
CTX_SVC パッケージ (8.1 より前), A-10
CTX_THES パッケージ
 8.1 での新しいプロシージャ, 1-8
CTX_USER_INDEX_ERRORS ビュー, 3-6
CTX_USER_PENDING ビュー, 3-6
CTXADMIN ロール (8.1 より前), 3-3
CTXAPP ロール, 3-3
ctxctl (8.1 より前), 1-15, 3-2
ctxkbtz コンパイラ, 1-4, 1-15
ctxload, 1-4, 3-5
 変更, 1-15
ctxsrv, 5-12
 起動および停止, 3-8
 ステータスの表示, 3-2
 変更, 1-15, 3-2
ctxsrv の M パーソナリティ, 3-2, 5-12
CTXSYS ユーザー, 3-3
 インストール, 2-8
CTXUSER ロール (8.1 より前), 3-3

D

DDL パイプ (8.1 より前), 3-6
DIRECT (8.1 より前), 4-3
DIRECTORY READER (8.1 より前), 4-5
DML キュー, 3-6
DML 処理
 バックグラウンド, 1-15, 3-2, 5-12
 バッチ, 5-13
drminst スクリプト
 概要, 4-7
 実行する場合, 2-7
drmrnrun スクリプト
 概要, 4-7
 実行する場合, 2-7
DROP INDEX コマンド, 5-8
DROP_INDEX プロシージャ (8.1 より前), 5-8
DROP_PREFERENCE プロシージャ, 5-7
DROP_STOPLIST プロシージャ, 5-15

E

EXECUTE 演算子
 移行, 6-14

F

FETCH_HIT プロシージャ (8.1 より前), 6-6
FILTER プロシージャ, 7-4
FIRST_ROWS ヒント, 6-6
FIRST/NEXT 演算子, 6-11
 移行, 6-13
FORMAT 列, 1-7

G

GENERIC ENGINE (8.1 より前), 4-4
GENERIC LOADER (8.1 より前), 4-5
GENERIC STOP LIST (8.1 より前), 4-4
GENERIC WORD LIST (8.1 より前), 4-4
GIST プロシージャ, 7-5, 7-8
 例, 7-9

H

HIGHLIGHT プロシージャ
 8.1, 7-3
 8.1 より前, 7-2
HTML
 フィルタ処理, 1-4, 7-4
HTML FILTER (8.1 より前), 4-3
HTML_SECTION_GROUP, 1-7
 META タグの索引付け, 5-18
HTML の META タグ, 5-18

I

initsid.ora ファイル, 3-7
Inso フィルタ
 概要, 1-3
interMedia Text
 関連マニュアル, viii

J

JAPANESE V-GRAM LEXER (8.1 より前), 4-3

K

KOREAN LEXER (8.1 より前), 4-4

L

LOB 列
 ロード, 3-5
LONG 列
 移行, 2-5
 ロード, 3-5

M

MARKUP プロシージャ, 7-3
MASTER DETAIL NEW (8.1 より前), 4-3
MASTER DETAIL (8.1 より前), 4-3
MAX 演算子, 6-11
migrate.sql, 2-4
 位置および機能, 4-7
 警告, 4-7, 4-8, 4-10
 索引作成, 4-10
 実行する場合, 2-7
 使用方法, 4-7
 編集, 4-7, 4-8
MULTI_LEXER レクサー, 1-8

N

NESTED_DATASTORE, 1-7
NEWS_SECTION_GROUP オブジェクト, 1-6, 1-13
NLS_LEXER (8.1 より前), 4-4
NULL COMPRESSOR (8.1 より前), 4-5
NULL FILTER (8.1 より前), 4-3
NULL TRANSLATOR (8.1 より前), 4-5

O

OPEN_CON プロシージャ (8.1 より前), 6-6
OPTIMIZE_INDEX プロシージャ, 1-9
OPTIMIZE プロシージャ (8.1 より前), 5-11
Oracle Data Migration Assistant, 2-9
Oracle Enterprise Manager, 3-2, 3-8
Oracle7 ユーザー
 移行パス, 2-2
Oracle8i
 インストール, 2-7
Oracle8i interMedia Text Manager, 3-2, 3-8
Oracle8i のインストール, 2-7
Oracle8 のアップグレード, 2-2

Oracle8 ユーザー
 移行パス, 2-2
Oracle との統合
 新機能, 1-3
Oracle ホーム
 移行に必要な, 2-6
ORDER BY 句, 6-4
OSFILE (8.1 より前), 4-3

P

p_table_clause 属性, 1-8
PL/SQL 演算子
 移行, 6-14
PL/SQL パッケージ
 新規, 1-14
 変更, A-1
PURGE_SQE プロシージャ (8.1 より前), 6-17

R

REFRESH_SQE プロシージャ (8.1 より前), 6-17
REMOVE_SQE プロシージャ
 8.1, 6-19
REMOVE_STOPCLASS プロシージャ, 5-15
REMOVE_STOPTHEME プロシージャ, 5-15
REMOVE_STOPWORD プロシージャ, 5-15
REQUEST_GIST プロシージャ (8.1 より前), 7-5, 7-7
REQUEST_THEMES プロシージャ (8.1 より前), 7-5, 7-6
RESUME_FAILED_INDEX プロシージャ (8.1 より前), 5-9

S

SECTION_GROUP 属性 (8.1 より前), 4-4
SET_ATTRIBUTE プロシージャ, 5-5
SOUNDEX_AT_INDEX 属性 (8.1 より前), 4-4
SQE 演算子, 6-17
SQL
 変更および新規のコマンド, 1-11
SQL*Loader, 3-5, 4-5
STCLAUSE 属性 (8.1 より前), 4-4
STORE_SQE プロシージャ, 6-17
 8.1, 6-19
SUBMIT プロシージャ (8.1 より前), 7-7
substring_index 属性, 1-8

SYNC_INDEX プロシージャ, 1-9
SYNC プロシージャ (8.1 より前), 5-13

T

Text Manager ツール, 3-8
text_enable 変数, 3-7
THEME LEXER (8.1 より前), 4-4
THEMES プロシージャ, 7-5, 7-8

U

URL (8.1 より前), 4-3
USER FILTER (8.1 より前), 4-3
USER TRANSLATOR (8.1 より前), 4-5
USER_DATASTORE オブジェクト, 1-4, 1-13, 5-2
8.1 での拡張, 1-7
UTF-8
日本語および韓国語レクサーのサポート, 1-8

W

WHERE 句, 6-4
wildcard_maxterms 属性, 1-8
WITHIN 演算子, 5-16
ネストされた, 1-9

X

XML_SECTION_GROUP オブジェクト, 1-5, 1-13
8.1 での拡張, 1-7
ドキュメント・タイプ・セクション, 5-18
XML ドキュメント
索引付け, 5-17
セクションのサポート, 1-5
属性セクション, 5-18

あ

アップグレード
Oracle8 から Oracel8i へ, 2-5
アプリケーション・コード
移行, 2-4, 2-8

い

移行

Oracle7, 2-2
アプリケーション・コード, 2-8
計画, 2-3
索引, 2-3, 2-8
索引プリファレンス・オブジェクト, 4-2
システム定義プリファレンス, 4-8
ステップ, 2-7
セクション・グループ, 5-16
定義, 2-3
ディスク要件, 2-6
データ, 2-5
テーマ・サマリー, 7-8
テーマのリスト, 7-8
問合せ, 6-2
ドキュメントの表示方法, 7-1
ハイライト表示, 7-2
パス, 2-2
ポリシー, 4-10
ユーザー定義プリファレンス, 4-7
要旨, 7-8
移行スクリプト
概要, 2-4
使用方法, 4-7
移行スクリプトの位置, 4-7
移行スクリプトの警告
移行不可能なポリシーおよび構造, 4-10
廃止されたシステム定義プリファレンス, 4-8
廃止されたプリファレンス・オブジェクト, 4-7
移行ステップ, 2-7
移行の計画, 2-3
インメモリ問合せ
「カーソル問合せ」を参照

え

エラー
DML, 3-6
演算子
8.1 での変更, 1-12
SQE, 6-17
廃止, 6-11
エンジン・オブジェクト
移行, 4-4

お

オフセット情報
 ハイライト, 7-3

か

カーソル問合せ
 移行, 6-5
階層問合せフィードバック, 1-4
拡張知識ベース, 1-4
 コンパイル, 1-15
拡張問合せオブティマイザ, 1-3
韓国語レクサー
 8.1 での改善, 1-8
管理
 移行, 3-1
管理ツール, 3-2, 3-8

き

記憶域オブジェクト
 移行, 4-4
記憶域クラス
 8.1 での改善, 1-8
 概要, 5-4
機能
 新規, 1-3
キュー
 変更, 3-6

こ

更新の保留, 3-6
構造化問合せ
 8.1 の方法, 6-8
 8.1 より前, 6-7
 移行, 6-7
コード
 移行, 2-4
コンプレッサ・オブジェクト (8.1 より前), 4-5
コンボジット・テキストキー問合せ
 移行, 6-10

さ

サーバー
 「ctxsrv」を参照
サーバーのパーソナリティ, 1-15, 3-2
サービス・キュー (8.1 より前), 3-6, 7-5
索引
 概要, 5-2
 再構築, 5-10
 最適化, 5-11
 削除, 5-8
 作成, 5-6
 作成の手順, 5-3
 テーマ・コンポーネント, 6-9
 デフォルト動作, 1-3
 同期化, 3-2, 5-13
 バックグラウンド DML, 5-12
 複数, 5-2
 ブラウズ, 1-9
索引移行, 4-7
 移行スクリプトによる, 2-4
 計画, 2-3
 手動, 2-4
 ステップ, 2-8
索引オブジェクト
 新規, 1-13, 4-5
 変更, 4-2
索引作成
 移行スクリプトの警告, 4-10
索引付け
 8.1 での改善, 1-9
 失敗後の再開, 5-9
索引付けされていないポリシー
 警告, 4-10
索引の更新
 バックグラウンド DML, 5-12
 バッチ DML, 5-13
索引の再構築, 5-10
索引の最適化, 1-9, 5-11
索引の削除, 5-8
索引の同期化, 1-9, 3-2, 5-13
サポートされていない演算子, 1-12, 6-14

し

システム定義ブリファレンス
 移行, 4-8

- 新規, 4-6
- シソーラス
 - インポート, 3-5
 - コンパイル, 1-4
- 実行可能ファイル
 - 変更および新規, 1-15
- 実行計画, 6-20
- 実行計画情報
 - 取得, 6-20
- 失敗した索引の再開, 5-9
- 新機能, 1-3

す

- スウェーデン語の代替スペル, 1-6
- ステミング, 1-8
- ストアド・クエリー式
 - 移行, 6-17
- ストップクラス, 1-5, 5-14
- ストップテーマ, 1-5, 5-14
- ストップリスト
 - 新しいプロシージャ, 5-15
 - 移行, 4-4, 5-14
 - 概要, 5-4
 - 拡張, 1-5
 - 管理ツールでの作成, 3-8
 - デフォルト, 5-14
- ストップワード, 5-14
- スペル
 - 代替, 1-6

せ

- セクション, 5-16
 - ゾーン, 5-16
 - 属性, 5-18
 - 動的な追加, 5-18
 - 特殊, 5-17
 - フィールド, 1-5, 5-17
- セクション・グループ
 - 新しいオブジェクト, 1-13
 - 移行, 5-16
 - 概要, 5-4
 - 管理ツールでの作成, 3-8
- セクション検索
 - 8.1 での改善, 1-7, 5-17
 - 拡張, 1-5

- ネストされた, 1-9
- 全テーマ
 - 取得, 7-8

そ

- ゾーン・セクション, 5-16
- 属性セクション
 - XML に定義, 5-18

た

- 代替スペル, 1-6
- ダブル切捨て検索, 1-8
- 単一テーマ
 - 取得, 7-8
- 段落セクション, 5-17

ち

- 知識ベース
 - 拡張, 1-4
- 知識ベースのコンパイラ, 1-15

て

- ディスク要件, 2-6
- データ記憶域
 - 8.1 での改善, 1-7
 - USER_DATASTORE オブジェクト, 1-13
 - 概要, 5-3
 - プリファレンスの例, 5-5
 - ユーザー, 1-4
- データストア・オブジェクト
 - 移行, 4-3
- データの移行
 - 概要, 2-5
 - 実行する場合, 2-9
- テーマ, 1-8
- テーマ・サマリー
 - 8.1, 7-8
 - 8.1 より前, 7-7
 - 違いの概要, 7-5
- テーマ問合せ
 - 移行, 6-9
- テーマのハイライト表示, 7-2, 7-3
- テーマのみの索引概要, 5-2

- テーマのリスト
 - 8.1 より前, 7-6
 - 取得, 7-8
 - 違いの概要, 7-5
- テーマ・ハイライト, 7-2, 7-3
- テーマ表
 - 8.1, 7-8
 - 8.1 より前, 7-6
- テーマ・ベース
 - 拡張, 1-4
- テキスト索引
 - 概要, 1-3
 - 定義, 1-2
- テキスト問合せ
 - 定義, 1-2
- テキストのハイライト表示, 7-3
- テキストのみの索引
 - 概要, 5-2
- テキスト・ハイライト, 7-3
- テキスト要求キュー (8.1 より前), 3-6
- テスト環境
 - 作成, 2-7
- デフォルトの索引動作, 1-3
- テンプレート・ポリシー
 - 警告, 4-10
- デンマーク語の代替スペル, 1-6

と

- 問合せ
 - 8.1 PL/SQL の例, 6-4
 - 8.1 SQL の例, 6-4
 - 8.1 での改善, 1-9
 - 移行, 6-2
 - カーソル (インメモリ), 6-5
 - 構造化, 6-7
 - コンボジット・テキストキー, 6-10
 - テーマ, 6-9
 - ヒット数のカウント, 6-15
- 問合せオブティマイザ, 1-3
- 問合せ式
 - 変更, 6-2
- 問合せ実行計画
 - 移行, 6-20
- 問合せの最適化, 1-3
- 問合せパイプ (8.1 より前), 3-6

- 問合せフィードバック
 - 階層, 1-4
 - 「実行計画」を参照
- 問合せ方法
 - 変更, 6-2
- ドイツ語の代替スペル, 1-6
- ドキュメント
 - 索引付けのときの合成, 1-4
- ドキュメント・セクション, 5-16
- ドキュメントの表示方法
 - 移行, 7-1
- ドキュメントのロード, 1-15, 3-5
- ドキュメント・フィルタ処理, 1-3
- 特殊セクション, 5-17

に

- 日本語レクサー
 - 8.1 での改善, 1-8
- ニュース・グループのドキュメント
 - セクションのサポート, 1-6

ね

- ネストされた WITHIN, 1-9

は

- ハードウェア要件, 2-6
- 廃止
 - PL/SQL パッケージおよびプロシージャ, A-1
 - 演算子, 1-12
 - 索引オブジェクト, 4-2, 4-7
 - システム定義プリファレンス, 4-8
 - ユーティリティ (ctxctl), 1-15
- パイプ
 - 変更, 3-6
- ハイライト表示
 - 違いの概要, 7-2
- パス
 - 移行, 2-2
- バックグラウンド DML, 3-2
- バッチ処理, 5-13
- パラレル索引付け, 1-9

ひ

ヒット数のカウント, 6-15
ビュー
 索引付け, 5-2
 新規および変更されたシステム提供の, 1-16
ビューの索引付け, 5-2
 警告, 4-10
表
 CTX_LING 8.1 より前, 7-6
 テーマ, 7-8
 要旨, 7-9
表記規則, ix

ふ

ファジー・マッチング, 1-8
フィードバック
 問合せ, 1-4
フィールド・セクション
 概要, 5-17
 新機能, 1-5
フィルタ・オブジェクト
 移行, 4-3
フィルタ・クラス
 概要, 5-3
フィルタ処理
 8.1 での改善, 1-7
 HTML およびプレーン・テキスト, 7-4
 概要, 1-3
複合書式列
 フィルタ処理, 1-3
複数の索引, 5-2
複数の索引付きの列, 5-2
複数のポリシー, 4-10
プリファレンス, 2-3
 新しいシステム定義, 4-6
 管理ツールでの作成, 3-8
 削除, 5-7
 作成 (例), 5-5
 システム定義の移行, 4-8
 ユーザー定義の移行, 4-7
プリファレンス・オブジェクト
 「索引オブジェクト」を参照
プレーン・テキスト
 フィルタ処理, 1-4, 7-4
文章セクション, 5-17

ほ

ポリシー, 2-3
 移行, 4-10
ポリシーの警告, 4-10

ま

マークアップされたドキュメント
 取得, 7-3
マシン要件, 2-6
マルチ言語テキスト表, 1-8

ゆ

ユーザー
 変更, 3-3
ユーザー定義プリファレンス
 移行, 4-7
ユーザーのデータ記憶域, 1-4

よ

用語, 1-2
要旨
 8.1, 7-8
 8.1 より前, 7-7
 違いの概要, 7-5
要旨表, 7-9

れ

レクサー
 8.1 での改善, 1-8
レクサー・オブジェクト
 移行, 4-3
レクサー・クラス
 概要, 5-3

ろ

ローダー・オブジェクト (8.1 より前), 4-5
ロール
 付与, 3-4
 変更, 3-3
ロールの付与, 3-4

わ

ワード問合せ

定義, 1-2

ワードリスト・オブジェクト

移行, 4-4

ワードリスト・クラス

8.1 での改善, 1-8

概要, 5-3

