

Oracle Ultra Search

ユーザーズ・ガイド

リリース 9.0.3

2002 年 12 月

部品番号 : J06888-01

ORACLE®

Oracle Ultra Search ユーザーズ・ガイド, リリース 9.0.3

部品番号 : J06888-01

原本名 : Oracle Ultra Search User's Guide, Release 9.0.3

原本部品番号 : B10043-01

原本著者 : Michele Cyran

原本協力者 : Sandeepan Banerjee, Stefan Buchta, Eddy Chee, Chung-Ho Chen, Will Chin, Jack Chung, Cindy Hsin, Yasuhiro Matsuda, Colin McGregor, Paul Yang, Steve Yang, David Zhang, Valarie Moore

Copyright © 2002 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
このマニュアルの構成	viii
関連文書	ix
表記規則	x
 Oracle Ultra Search の新機能	 xv
 1 Oracle Ultra Search の概要	
Oracle Ultra Search の概要	1-2
Oracle Ultra Search のコンポーネント	1-2
Oracle Ultra Search Crawler	1-2
Oracle Ultra Search Server コンポーネント	1-3
Oracle Ultra Search Administration ツール	1-3
Oracle Ultra Search の API とサンプル・アプリケーション	1-3
Oracle Ultra Search のシステム構成	1-4
 2 Oracle Ultra Search のインストールと構成	
Oracle Ultra Search の要件	2-2
表記規則	2-2
Oracle Ultra Search のシステム要件	2-2
Oracle Ultra Search Server コンポーネントのインストール	2-4
Oracle Ultra Search Server コンポーネントのインストールと構成	2-4

Oracle Ultra Search Server コンポーネントの既存データベースまたは メタデータ・リポジトリへのインストール	2-5
データベース要件	2-5
Oracle Ultra Search Server コンポーネントの既存の Oracle9i データベースへのインストール	2-6
Oracle Ultra Search Server コンポーネントの既存の Oracle9iAS メタデータ・リポジトリへの インストール	2-7
Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール	2-9
Web アプリケーションの概要	2-10
ブラウザの要件	2-11
Oracle Ultra Search Middle-Tier コンポーネントのインストール	2-12
Oracle Ultra Search Middle-Tier コンポーネントの構成	2-13
Oracle Ultra Search Server コンポーネントのリモート Crawler ホストへのインストール	2-29
Oracle Ultra Search Server コンポーネントをリモート Crawler ホストにインストールする場合	2-29
Oracle Ultra Search Server コンポーネントをリモート Crawler ホストで構成する場合	2-30
誤った値を入力した場合	2-31

3 インストール後の作業とアップグレードに関する情報

Oracle Ultra Search 用の Oracle データベース・サーバーの構成	3-2
手順 1: Oracle データベースのチューニング	3-2
手順 2: CTXSYS ユーザー用の一時表領域の作成および割当て	3-4
手順 3: Oracle Ultra Search インスタンスの各ユーザー用の大規模な表領域の作成	3-4
手順 4: 各 Oracle Ultra Search インスタンス用の新規データベース・ユーザーの作成および構成 ...	3-5
手順 5: 表の統計の収集	3-6
手順 6: 索引設定の変更	3-7
ストップリストの管理	3-8
Oracle Ultra Search のデフォルトのストップリスト	3-8
インスタンスのストップリストの変更	3-8
最新の Oracle Ultra Search へのアップグレード	3-10
Oracle Ultra Search リリース 1.0.3 (Oracle9i データベース リリース 1 (9.0.1)) からの アップグレード	3-10
Oracle Ultra Search リリース 9.0.2 からリリース 9.0.3 へのアップグレード	3-14
Oracle Ultra Search リリース 9.2 からリリース 9.0.3 へのアップグレード	3-14

4 Oracle Ultra Search Crawler とデータ・ソースについて

Oracle Ultra Search Crawler の概要	4-2
Crawler 設定	4-2
データ・ソース	4-2
Crawler エージェントの使用	4-3
データ・ソースの同期	4-3
表示 URL とアクセス URL	4-3
ドキュメント属性	4-3
スケジュールのクロール・プロセス	4-4
ドキュメントのキューイングおよびキャッシュ	4-4
ドキュメントの索引付け	4-8
データ同期	4-9
Oracle Ultra Search のリモート Crawler	4-10

5 Oracle Ultra Search Administration ツールについて

Oracle Ultra Search Administration ツール	5-2
Crawler パラメータの設定	5-2
検索オプションの設定	5-3
異なる言語でのオンライン・ヘルプ	5-3
Oracle Ultra Search へのログオン	5-4
SSO ユーザーとしてのインスタンスへのログオンと管理	5-5
Oracle9iAS Portal による Oracle Ultra Search へのログオン	5-5
「インスタンス」タブ	5-6
インスタンスの作成	5-6
インスタンスの選択	5-9
インスタンスの削除	5-10
インスタンスの編集	5-10
「Crawler」タブ	5-11
設定	5-11
リモート Crawler プロファイル	5-14
Crawler 統計	5-14
「Web アクセス」タブ	5-15
URL 認証	5-15
プロキシ	5-15

「属性」タブ	5-16
検索属性	5-16
マッピング	5-17
「ソース」タブ	5-18
Web ソース	5-18
表ソース	5-20
電子メール・ソース	5-22
ファイル・ソース	5-23
ユーザー定義ソース	5-24
Oracle9iAS Portal ソース	5-25
「スケジュール」タブ	5-26
データ同期	5-26
索引最適化	5-30
「問合せ」タブ	5-30
データ・グループ	5-30
URL 発行	5-31
検索オプション	5-31
問合せ統計	5-32
「ユーザー」タブ	5-34
設定	5-34
スーパー・ユーザー	5-34
権限	5-35
「グローバルゼーション」タブ	5-35
検索属性名	5-35
LOV 表示名	5-36
データ・グループ名	5-36

6 Oracle Ultra Search 開発者ガイドおよび API リファレンス

Oracle Ultra Search API の概要	6-2
Oracle Ultra Search の Query API	6-2
Oracle Ultra Search Crawler エージェント API	6-3
Crawler エージェントの概要	6-4
Crawler エージェントの機能	6-6
サンプル・エージェント・ファイル	6-8
サンプル Crawler エージェントの設定	6-8

Oracle Ultra Search の Java Email API	6-11
JavaMail の実装	6-12
Java Email API	6-12
サンプル・メーリング・リスト・ブラウザ・アプリケーション・ファイル	6-12
サンプル・メーリング・リスト・ブラウザ・アプリケーションの設定	6-13
Oracle Ultra Search の URL リライター API	6-13
URL リンクのフィルタ処理	6-14
URL リンクのリライト	6-14
URL リライタの作成および使用	6-16
Oracle Ultra Search のサンプル検索アプリケーション	6-17
JSP のサンプル検索アプリケーション	6-18
JSP の概要	6-18
Oracle Ultra Search の Query タグ・ライブラリ	6-19
Query タグの説明	6-21
問合せ構文拡張のカスタマイズ	6-28
デフォルトの問合せ構文拡張の実装	6-28
ルールのカスタマイズ	6-33

A Web クロール・プロセスのチューニング

Web クロールの方針	A-2
クロール・プロセスの監視	A-2
URL ルーピング	A-2

B 問合せパフォーマンスのチューニング

C リモート Crawler の使用方法

拡張性およびロード・バランシング	C-2
起動方法	C-2
インストールと構成の順序	C-3

D 表データ・ソースの同期化

Oracle データベースのクロールの同期化	D-2
ログ表の作成	D-2
ログ・トリガーの作成	D-3
Oracle 以外のデータベースのクロールの同期化	D-5

E メタデータの Oracle Ultra Search へのロード

ロード・ツールの起動	E-2
ドキュメントおよび関連性スコアのロード	E-3
入力 XML ファイル	E-3
検索属性 LOV および LOV 表示名のロード	E-4
LOV XML ファイル	E-4
ドキュメント検索オプション用の XML Schema	E-5
LOV および LOV 表示名用の XML Schema	E-6

F Crawler の Java CLASSPATH の変更

Crawler の Java CLASSPATH を変更するケース	F-2
Crawler CLASSPATH とリモート Crawler CLASSPATH の違い	F-2
Oracle Ultra Search Server ホストでの Crawler の Java CLASSPATH 変更	F-2
リモート Crawler ホストでの Crawler の Java CLASSPATH 変更	F-3

G 問合せ構文拡張 9.0.1 のカスタマイズ

デフォルトの問合せ構文拡張の実装	G-2
エンド・ユーザーの問合せ構文	G-2
スコア付け	G-4
拡張ルール	G-4
ルールのカスタマイズ	G-6

索引

はじめに

このマニュアルでは、Oracle Ultra Search のインストール、構成および使用方法について説明します。また、Oracle Ultra Search の API を使用してカスタム・アプリケーションを作成する方法についても説明します。

ここでは次の項目について説明します。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連文書](#)
- [表記規則](#)

対象読者

このマニュアルは、次の作業を行うデータベース管理者およびアプリケーション開発者を対象としています。

- Oracle Ultra Search のインストールとインスタンスの管理
- Oracle Ultra Search アプリケーションの開発

このマニュアルを活用するには、Oracle データベース管理システム、SQL、SQL*Plus および PL/SQL での作業経験が必要です。

このマニュアルの構成

このマニュアルは、次の章と付録で構成されています。

「Oracle Ultra Search の新機能」

Oracle Ultra Search の新機能について説明し、追加情報の参照先を示します。

第 1 章「Oracle Ultra Search の概要」

Oracle Ultra Search の概要およびシステム構成について説明します。

第 2 章「Oracle Ultra Search のインストールと構成」

Oracle Ultra Search のインストールおよび構成方法について説明します。

第 3 章「インストール後の作業とアップグレードに関する情報」

Oracle Ultra Search 用の Oracle データベース・サーバーの構成方法、ストップリストの管理方法など、インストール後の作業について説明します。最新の Oracle Ultra Search へのアップグレード方法についても説明します。

第 4 章「Oracle Ultra Search Crawler とデータ・ソースについて」

Crawler の機能について説明します。Crawler の設定、データ・ソース、ドキュメント属性、データ同期およびリモート Crawler についても説明します。

第 5 章「Oracle Ultra Search Administration ツールについて」

Oracle Ultra Search Administration ツールを使用して Oracle Ultra Search Crawler を構成およびスケジュールする方法を説明します。

第 6 章「Oracle Ultra Search 開発者ガイドおよび API リファレンス」

Oracle Ultra Search の API（Query API、Crawler エージェント API、Email API、URL リライタ API）について説明します。サンプル検索アプリケーション、Query タグ・ライブラリ、問合せ構文拡張のカスタマイズの詳細など、API 関連情報も示します。

付録 A 「Web クロール・プロセスのチューニング」

Web クロールの方針、URL ルーピングおよびクロール・プロセスの監視方法について説明します。

付録 B 「問合せパフォーマンスのチューニング」

Oracle Ultra Search の問合せパフォーマンスを改善する方法をいくつか説明します。

付録 C 「リモート Crawler の使用方法」

リモート Crawler を使用する理由と使用方法について説明します。

付録 D 「表データ・ソースの同期化」

表ソースのクロールを最適化するために Oracle Ultra Search が使用するロギング機能について説明します。

付録 E 「メタデータの Oracle Ultra Search へのロード」

Oracle Ultra Search データベースにメタデータをロードするためのコマンドライン・ツールについて説明します。

付録 F 「Crawler の Java CLASSPATH の変更」

Crawler の Java CLASSPATH を変更する理由と変更方法について説明します。

付録 G 「問合せ構文拡張 9.0.1 のカスタマイズ」

Oracle Ultra Search リリース 9.0.1 の問合せ構文拡張をカスタマイズする方法について説明します。

関連文書

詳細は、次の Oracle マニュアルを参照してください。

- 『Oracle9i データベース概要』
- 『Oracle9i データベース管理者ガイド』
- 『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』

このマニュアルに記載されている例の多くでは、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストレーション・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J（Oracle Technology Network Japan）に接続すれば、無償でダウンロードできます。OTN-Jを使用するには、オンラインでの登録が必要です。次の URL で無料で登録できます。

<http://otn.oracle.co.jp/membership/>

OTN-J のユーザー名とパスワードを取得済みの場合は、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

表記規則

このマニュアル・セットの本文とコード例に使用されている表記規則について説明します。

- [本文の表記規則](#)
- [コード例の表記規則](#)
- [Windows オペレーティング・システムの表記規則](#)

本文の表記規則

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表は、本文の表記規則と使用例を示しています。

規則	意味	例
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。	この句を指定する場合は、 索引構成表 を作成します。
固定幅フォントの大文字	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージとメソッドの他、システム指定の列名、データベース・オブジェクトと構造体、ユーザー名、およびロールがあります。	この句は、NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用すると、データベースのバックアップを作成できます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。

規則	意味	例
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびサンプルのユーザー指定要素を示します。この要素には、コンピュータ名とデータベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトと構造体、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニット、およびパラメータ値があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。	sqlplus と入力して SQL*Plus をオープンします。 パスワードは orapwd ファイルに指定されています。 データ・ファイルと制御ファイルのバックアップを /disk1/oracle/dbs ディレクトリに作成します。 department_id、department_name および location_id の各列は、hr.departments 表にあります。 初期化パラメータ QUERY_REWRITE_ENABLED を true に設定します。 oe ユーザーで接続します。 これらのメソッドは JRepUtil クラスに実装されます。
固定幅フォントの小文字のイタリック	固定幅フォントの小文字のイタリックは、ブレースホルダまたは変数を示します。	parallel_clause を指定できます。 Uold_release.SQL を実行します。 old_release は、アップグレード前にインストールしたリリースです。

コード例の表記規則

コード例は、SQL、PL/SQL、SQL*Plus またはその他のコマンドラインを示します。次のように、固定幅フォントで、通常の本文とは区別して記載されています。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表は、コード例の記載上の表記規則と使用例を示しています。

規則	意味	例
[]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。	DECIMAL (digits [, precision])
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体は入力しないでください。	{ENABLE DISABLE}

規則	意味	例
	縦線は、大カッコまたは中カッコ内の複数の選択肢を区切るために使用します。オプションのうち1つを入力します。縦線自体は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	水平の省略記号は、次のどちらかを示します。 <ul style="list-style-type: none">■ 例に直接関係のないコード部分が省略されていること。■ コードの一部が繰り返し可能であること。	CREATE TABLE ... AS subquery; SELECT col1, col2, ... , coln FROM employees;
.	垂直の省略記号は、例に直接関係のない数行のコードが省略されていることを示します。	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fs1/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fs1/dbs/tbs_09.dbf 9 rows selected.
その他の表記法	大カッコ、中カッコ、縦線および省略記号以外の記号は、示されているとおりに入力してください。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システムにより指定される要素を示します。これらの用語は、ユーザー定義用語と区別するために大文字で記載されています。大カッコで囲まれている場合を除き、記載されているとおりの順序とスペルで入力してください。ただし、この種の用語は大 / 小文字区別がないため、小文字でも入力できます。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
小文字	小文字は、ユーザー指定のプログラム要素を示します。たとえば、表名、列名またはファイル名を示します。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則とその使用例を示しています。

規則	意味	例
「スタート」 → を選択します。	プログラムを起動する方法です。	Database Configuration Assistant を起動するには、「スタート」 → 「プログラム」 → 「Oracle - HOME_NAME」 → 「Configuration and Migration Tools」 → 「Database Configuration Assistant」 を選択します。
ファイル名と ディレクトリ名	ファイル名とディレクトリ名は、大 / 小文字区別がありません。左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ () およびハイフン (-) の特殊文字は使用できません。特殊文字の円記号 (¥) は、引用符内にある場合でも要素セパレータとして扱われます。ファイル名が ¥¥ で開始する場合、Windows では汎用命名規則を使用しているとみなされます。	c:¥winnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現在のハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプト内のエスケープ文字はカレット (^) です。プロンプトは、現在作業中のサブディレクトリを示します。このマニュアルでは、コマンド・プロンプトと呼びます。	C:¥oracle¥oradata>
特殊文字	Windows のコマンド・プロンプトでは、特殊文字の二重引用符 (") 用のエスケープ文字として、円記号 (¥) が必要な場合があります。カッコおよび一重引用符 (') にエスケープ文字は不要です。エスケープ文字および特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は 16 文字以内の英数字で指定できます。ホーム名で利用できる特殊文字はアンダースコア (_) のみです。	C:¥> net start OracleHOME_NAME_TNSListener

規則	意味	例
<code>ORACLE_HOME</code> と <code>ORACLE_BASE</code>	<p>Oracle8 リリース 8.0 以前では、Oracle のコンポーネントをインストールすると、すべてのサブディレクトリはトップレベルの <code>ORACLE_HOME</code> ディレクトリの下に、次のいずれかの名前（デフォルトの場合）で配置されていました。</p> <ul style="list-style-type: none">■ Windows NT の場合は、<code>C:\%orant</code>■ Windows 98 の場合は、<code>C:\%orawin98</code> <p>このリリースでは、Optimal Flexible Architecture (OFA) ガイドラインに準拠しています。トップレベルの <code>ORACLE_HOME</code> ディレクトリの下にすべてのサブディレクトリが配置されるわけではありません。デフォルトでは、<code>C:\%oracle</code> に <code>ORACLE_BASE</code> というトップレベル・ディレクトリがあります。他の Oracle ソフトウェアがインストールされていないコンピュータに Oracle の最新リリースをインストールする場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は、<code>C:\%oracle%\orann</code> (<code>nn</code> は最新のリリース番号) です。Oracle ホーム・ディレクトリは、<code>ORACLE_BASE</code> の直下にあります。</p> <p>このマニュアルでのディレクトリ・パスの例はすべて、OFA の表記規則に準拠しています。</p>	<code>%ORACLE_HOME%\rdbms\admin</code> ディレクトリにアクセスします。

Oracle Ultra Search の新機能

ここでは、Oracle Ultra Search の新機能について説明し、追加情報の参照先を示します。

注意： Oracle Ultra Search リリース 9.0.3 は、Oracle Collaboration Suite リリース 9.0.3 の一部です。

Oracle Ultra Search リリース 9.2 は、Oracle9i リリース 2 (9.2) の一部です。Oracle Ultra Search リリース 1.0.3 は、Oracle9i リリース 1 (9.0.1) の一部です。

Oracle Ultra Search リリース 9.0.2 は、Oracle9i Application Server (9iAS) リリース 2 (9.0.2) の一部です。

■ Oracle9i Application Server との統合

Oracle9i Application Server (9iAS) の Oracle Ultra Search は、Oracle9i データベースでの Oracle Ultra Search と同じ製品ですが、次の 2 つの点で異なります。

- Oracle Ultra Search は Oracle9iAS Portal と統合されています。この結果、Portal ユーザーは、強力な複数リポジトリ検索を自分のポータル・ページに追加できます。Portal 独自のリポジトリをクロールし、検索可能にする機能もあります。この機能は Oracle9iAS で使用する場合にのみ使用可能です。
- Oracle9iAS には、Single Sign-On (SSO) Server が含まれています。このため、SSO ユーザーは Oracle9iAS 製品のすべてのコンポーネントに一度でログオンでき、Oracle Ultra Search の管理インタフェースによって、データベース・ユーザーまたは SSO ユーザーのユーザー管理操作を実行できます。認証 SSO ユーザーに、Oracle Ultra Search のログイン画面が表示されることはありません。かわりに、認証 SSO ユーザーはインスタンスを即時に選択できます。SSO ユーザーに Oracle Ultra Search の管理権限がない（「ユーザー」タブで設定）場合は、エラーが戻されます。この機能は Oracle9iAS で使用する場合にのみ使用可能です。

関連項目： <http://portalstudio.oracle.com/>

■ 拡張可能な Crawler および Crawler エージェント

提供されているものの他に、独自にデータ・ソースおよびタイプを定義、編集または削除することができます。独自の Crawler エージェントを実装して、Lotus Notes や Documentum などの独自のデータベースとインタフェースを持つ独自のドキュメント・リポジトリを、クロールして索引付けできます。独自のリポジトリを、ユーザー定義データ・ソースと呼びます。Crawler によるデータ・ソースへのアクセスを可能にするモジュールを Crawler エージェントと呼びます。

関連項目：

- 6-3 ページ [「Oracle Ultra Search Crawler エージェント API」](#)
- Oracle Ultra Search Javadoc

■ サンプル検索アプリケーション

Oracle Ultra Search には、検索結果を問い合わせるための、完全な機能を備えたサンプル検索アプリケーションが含まれています。サンプル検索アプリケーションには、サンプル検索ポートレットが含まれています。Oracle Ultra Search のサンプル・ポートレットは、Oracle9iAS Portal で使用する検索ポートレットの作成方法を説明します。この同じポートレットが、Oracle9iAS Portal 製品の機能としてインストールされています。

関連項目： 6-2 ページ [「Oracle Ultra Search の Query API」](#)

■ サンプル検索ポータルレット

Oracle Ultra Search は、Oracle9iAS Portal ページに埋め込まれる検索ポータルレットを提供します。この検索ポータルレットは、JavaServer Pages (JSP) アプリケーションとして実装されます。

Oracle Ultra Search の検索ポータルレットは、Query API 完全サンプル・アプリケーションが提供するほとんどの機能をサポートしています。

関連項目：

- ポータルレットの詳細は、Oracle 9iAS Portal のマニュアルを参照してください。
- Query API 完全サンプル・アプリケーションの詳細は、「Oracle Ultra Search のサンプル検索アプリケーションの README」を参照してください。

■ Query API

Oracle Ultra Search は、検索機能をユーザーのポータル・サイトに取り込むための柔軟な API を提供します。次に、Query API の新機能を示します。

- 3 つの属性型：文字列、数値および日付
- 複数値属性
- 属性、属性の値リスト (LOV) およびデータ・グループの表示名サポート
- ドキュメント検索オプション
- 演算子 (AND、OR) を使用した属性問合せ演算子の任意のグループ化 (属性演算子の評価順序の制御を伴います)
- 問合せ結果に戻されるメタデータの選択

関連項目：

- 6-2 ページ [「Oracle Ultra Search の Query API」](#)
- Oracle Ultra Search Javadoc

■ URL のリライト

URL リライタは、Ultra Search UrlRewriter インタフェースを実装するためにユーザーが提供する Java モジュールです。Crawler は URL リライタを使用して、抽出した URL リンクが URL キューに挿入される前にフィルタ処理またはリライトします。URL のフィルタ処理によって不要なリンクが削除され、URL のリライトによって URL リンクが変換されます。この変換は、アクセス URL を使用する場合に必要です。

関連項目：

- 5-18 ページ [「Web ソース」](#)
- 6-13 ページ [「Oracle Ultra Search の URL リライター API」](#)
- Oracle Ultra Search Javadoc

■ **ロボットの除外**

ロボットの除外によって、ロボットで参照できるサイトを制御できます。ロボットの除外が有効（デフォルト）な場合、Web Crawler は、Web サーバーの robots.txt ファイルに指定されているアクセス・ポリシーに基づいてページ間を横断します。たとえば、ロボットは、<http://www.foobar.com/> を参照する場合、<http://www.foobar.com/robots.txt> をチェックします。このファイルが見つかった場合、Crawler はその内容を分析して、ドキュメントを取り出せるかどうかを確認します。Web サイトを所有している場合は、ロボットの除外を無効にすることができます。ただし、他の Web サイトをクロールするときは、ロボットの除外を有効にして robots.txt に従う必要があります。

関連項目： 5-18 ページ [「Web ソース」](#)

■ **表示 URL サポート**

データベース・ベースの Web アプリケーションから情報を収集するとき、Oracle Ultra Search では、URL を指定して、取り出されたデータをブラウザに表示できます。URL は、データベース内のデータに対応する Web アプリケーションの画面を示します。これは、表データ・ソース、ファイル・データ・ソースおよびユーザー定義データ・ソースに使用できます。

関連項目： 4-3 ページ [「表示 URL とアクセス URL」](#)

■ **ドキュメントおよび検索属性**

ドキュメント属性（メタデータ）は、ドキュメントのプロパティを示します。各データ・ソースには、ドキュメント属性の独自のセットがあります。値はクロール・プロセス中に取り出され、検索属性の 1 つにマップされた後、データベースに格納され、索引付けされます。この結果、その属性に基づいてドキュメントを問い合わせることができます。異なるデータ・ソースのドキュメント属性を同じ検索属性にマップできます。したがって、複数のデータ・ソースのドキュメントを同じ検索属性に基づいて問い合わせることができます。

検索属性の値リスト（LOV）は、検索の問合せを指定する際に有効です。属性の LOV が使用可能な場合、Crawler は、属性値、属性値の表示名およびその変換を含む LOV 定義を登録します。

関連項目： 4-3 ページ [「ドキュメント属性」](#)

■ メタデータ・ローダー

Oracle Ultra Search では、メタデータを Oracle Ultra Search データベースにロードするためのコマンドライン・ツールが提供されます。データが大量にある場合は、このツールによって HTML ベースの管理ツールより早くロードできます。このローダー・ツールは次の種類のメタデータをサポートします。

- 検索属性の値リスト (LOV) および表示名
- ドキュメント検索オプションおよびドキュメント・ロード

関連項目： [付録 E「メタデータの Oracle Ultra Search へのロード」](#)

■ ドキュメント検索オプション

ドキュメント検索オプションによって、検索結果をオーバーライドし、ドキュメント検索オプションで問合せ結果リストにランク付けされているドキュメントの順序を変更できます。この結果、重要なドキュメントを高くスコア付けでき、検索を容易にすることができます。

関連項目： [5-30 ページ「「問合せ」タブ」](#)

■ データ収集モード

初期計画の目的として、索引付けを行わず、URL の収集のみを **Crawler** で行う場合があります。クロールの終了後に、ドキュメントの URL とステータスを調査し、不要なドキュメントを削除してから、索引付けを開始できます。クロール・モードを次のように更新できます。

- 索引付けのために全ての URL を実行
- 索引付けの前に URL を調査
- 索引付けのみ

関連項目： [5-26 ページ「「スケジュール」タブ」](#)

■ インスタンス・スナップショット・サポート

マスター Oracle Ultra Search インスタンスの読取り専用スナップショットを作成できます。このスナップショットは、検索処理用またはバックアップ用に有効です。スナップショット・インスタンスも、更新可能にできます。これは、マスター・インスタンスが破損し、スナップショットを新しいマスター・インスタンスとして使用する場合に役立ちます。

関連項目： [5-6 ページ「「インスタンス」タブ」](#)

■ シングル・サインオン認証

Oracle Ultra Search Administration ツールでは、ユーザーのタイプに応じて、3 つのログオン・モードをサポートしています。次のユーザーでログオンできます。

- Oracle Internet Directory (OID) で管理され、SSO Server で認証されたシングル・サインオン (SSO) ・ユーザー
- Oracle Ultra Search データベースのローカル・データベース・スキーマ・ユーザー (non-SSO モード)
- Enterprise Manager の IAS_ADMIN ユーザー

注意： シングル・サインオン (SSO) は、Oracle9i Application Server (9iAS) でのみ使用可能です。Oracle9i データベースでは使用できません。

関連項目： 5-4 ページ [「Oracle Ultra Search へのログオン」](#)

■ 問合せ構文拡張

以前のリリースでは、デフォルトの問合せ構文拡張の実装のコードが WK_QUERYEXP PL/SQL パッケージに含まれていました。現在は、Contains 問合せによって、インターネット検索エンジンに類似した問合せ構文を指定できます。この構文は、'title' (タイトル) StringAttribute でユーザーの問合せと一致するドキュメントを高くスコア付けします。

関連項目： 6-28 ページ [「問合せ構文拡張のカスタマイズ」](#)

Oracle Ultra Search の概要

この章では、次の項目について説明します。

- [Oracle Ultra Search の概要](#)
- [Oracle Ultra Search のコンポーネント](#)
- [Oracle Ultra Search のシステム構成](#)

Oracle Ultra Search の概要

Oracle Ultra Search は、Oracle データベース・サーバーおよび Oracle Text のテクノロジー上に構築され、複数のリポジトリ間（Oracle データベース、他の ODBC 準拠のデータベース、IMAP メール・サーバー、Web サーバーで提供する HTML ドキュメント、ディスク上のファイルなど）で統一した検索機能を提供します。

Oracle Ultra Search では、Crawler を使用してドキュメントに索引付けをします。各ドキュメントはそれぞれのリポジトリに格納され、クロールした情報を使用して索引が作成されます。索引は、指定した Oracle データベースのファイアウォール内に格納されます。また、Oracle Ultra Search には、コンテンツ管理ソリューションを構築するための API も用意されています。

Oracle Ultra Search によって、次の機能が提供されます。

- データベース内でテキストを検索するための完全なテキスト問合せ言語
- Oracle データベース・サーバーと SQL 問合せ言語との完全な統合
- コンセプトの検索やテーマ分析などの拡張機能
- 150 種類以上の一般的なファイル形式の索引付け
- 中国語、日本語、韓国語（CJK）および Unicode のサポートを含む、完全なグローバリゼーション

Oracle Ultra Search のコンポーネント

Oracle Ultra Search は、次のコンポーネントで構成されています。

- [Oracle Ultra Search Crawler](#)
- [Oracle Ultra Search Server](#) コンポーネント
- [Oracle Ultra Search Administration ツール](#)
- [Oracle Ultra Search の API とサンプル・アプリケーション](#)

Oracle Ultra Search Crawler

Oracle Ultra Search Crawler は、設定されたスケジュールに従い、Oracle データベース・サーバーによってアクティブにされる Java プロセスです。アクティブになった Crawler は、様々なデータ・ソースからドキュメントをフェッチし、そのドキュメントに Oracle Text を使用して索引付けをするプロセッサ・スレッドを、構成可能な数だけ起動します。この索引は、問合せに使用されます。データ・ソースには、Web サイト、データベース表、ファイル、メーリング・リスト、Oracle9iAS Portal のページ・グループ、ユーザー定義のデータ・ソースなどが含まれます。

Crawler は、リンクをマップして関連を分析します。Crawler スケジュールは、DBMS_JOB のキュー・メカニズムと統合され、そこから起動します。Crawler は、クロール時に HTML 以外の埋込みドキュメントを検出すると、Oracle Text のフィルタ機能を使用してドキュメント・タイプを自動的に検出し、ドキュメントをフィルタ処理して索引を付けます。

関連項目： [第 4 章「Oracle Ultra Search Crawler とデータ・ソースについて」](#)

Oracle Ultra Search Server コンポーネント

Oracle Ultra Search Server コンポーネントは、Oracle Ultra Search のリポジトリと Oracle Text で構成されています。Oracle Text は、データ・ソースから取り出したデータの索引付けと問合せに必要な、テキスト索引付けと検索機能を提供します。Oracle Ultra Search Server コンポーネントは、ユーザーには表示されず、Crawler が取り出した情報を索引付けし、問合せ結果を提供します。

関連項目： [2-4 ページ「Oracle Ultra Search Server コンポーネントのインストール」](#)

Oracle Ultra Search Administration ツール

Oracle Ultra Search Administration ツールは、Oracle Ultra Search Crawler を構成およびスケジュールするための JavaServer Pages (JSP) Web アプリケーションです。このツールは通常、Web サーバーと同じマシンにインストールされています。Administration ツールには、イントラネットのブラウザからアクセスできます。Administration ツールは、Oracle Ultra Search の検索アプリケーションから独立しています。したがって、Administration ツールと検索アプリケーションは、セキュリティや拡張性を強化するために、異なるマシンでホスト管理できます。

関連項目： [第 5 章「Oracle Ultra Search Administration ツールについて」](#)

Oracle Ultra Search の API とサンプル・アプリケーション

Oracle Ultra Search には、次の API が用意されています。

- Query API。索引付けされたデータで使います。Java API は、HTML レンダリング要素を適用しません。HTML インタフェースは、アプリケーションで完全にカスタマイズできます。
- Crawler エージェント API。独自のドキュメント・リポジトリをクロールして索引付けします。
- Java Email API。アーカイブされた電子メールにアクセスし、電子メールを表示するために検索アプリケーションで使用されます。この API は、独自のカスタム検索アプリケーションを作成するときにも使用できます。

- URL リライター API。Crawler はこれを使用して、抽出した URL リンクが URL キューに挿入される前にフィルタ処理およびリライトします。

Oracle Ultra Search には、検索結果を問い合わせるための、高度な機能を備えた検索アプリケーションが含まれています。検索アプリケーションは JSP に基づいており、JSP1.1 準拠のエンジンで動作します。

関連項目：

- [第 6 章「Oracle Ultra Search 開発者ガイドおよび API リファレンス」](#)
- Oracle Ultra Search Javadoc

Oracle Ultra Search のシステム構成

Oracle Ultra Search は、Oracle データベース・サーバーに対して実行されるクライアント・プログラムです。サーバー層と中間層の 2 つの構成で配置できます。

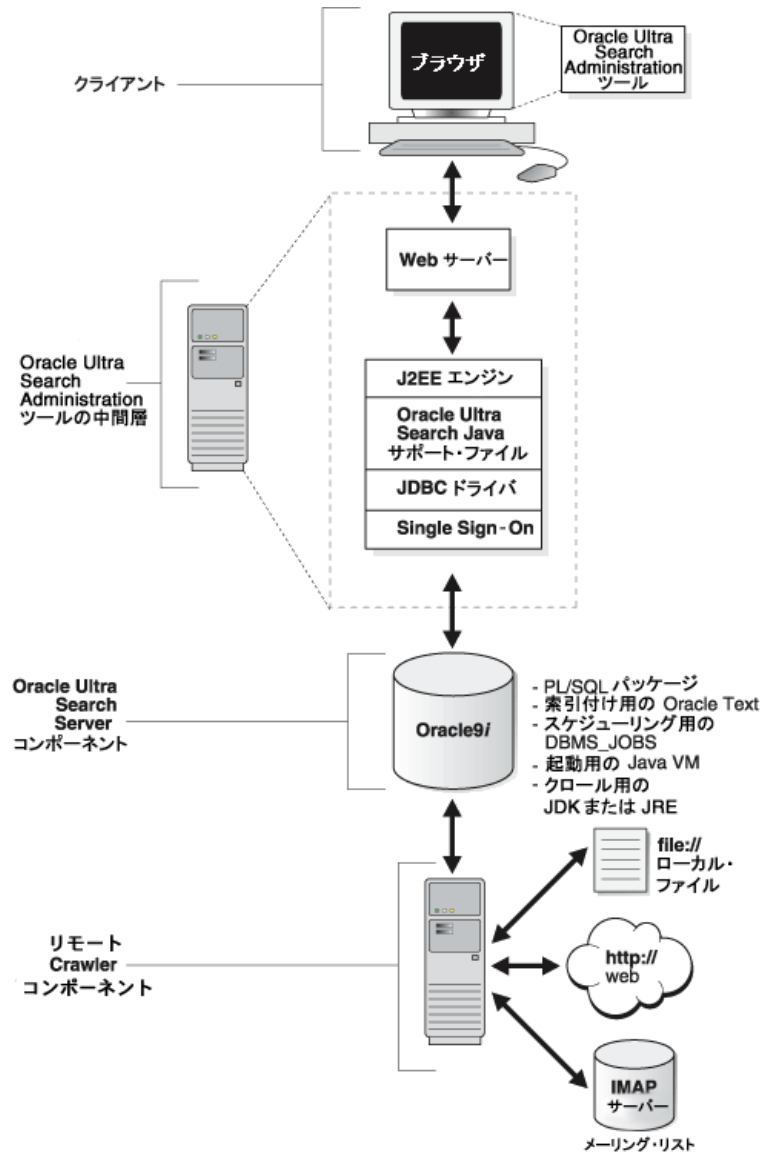
Oracle Ultra Search のデフォルトの Query インタフェースと Administration ツールは、任意の HTML ブラウザ・クライアントで実行されます。Administration ツールは、中間層の特定の Java クラスに依存しています。この論理的な中間層は、データベース・サーバーを実行する物理的なマシンと同じか、Oracle9iAS を実行している別のマシンの場合があります。

Oracle Ultra Search データベース・サーバーのコンポーネントは、Oracle Ultra Search のデータ・ディクショナリで構成されています。このデータ・ディクショナリには、異なるすべてのリポジトリ上にあるメタデータの他に、Crawler の駆動に必要なスケジュールと Java クラスも格納されています。Crawler 自体は、データベース・サーバー・マシン上で実行するか、別のマシン上でリモートで実行できます。

関連項目： コンポーネントの詳細は、[第 2 章「Oracle Ultra Search のインストールと構成」](#)を参照してください。

[図 1-1](#) は、Oracle Ultra Search のシステム構成を示しています。

図 1-1 Oracle Ultra Search のシステム構成



Oracle Ultra Search のインストールと構成

この章では、次の項目について説明します。

- Oracle Ultra Search の要件
- Oracle Ultra Search Server コンポーネントのインストール
- Oracle Ultra Search Server コンポーネントの既存データベースまたはメタデータ・リポジトリへのインストール
- Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール
- Oracle Ultra Search Server コンポーネントのリモート Crawler ホストへのインストール

Oracle Ultra Search の要件

Oracle Ultra Search を使用するには、Oracle Universal Installer で次のコンポーネントをインストールする必要があります。

- Oracle Ultra Search Server コンポーネント
- Oracle Ultra Search Middle-Tier コンポーネント

注意： Oracle Ultra Search をインストールすると、ローカルとリモートの両方の構成でクローलできます。リモート構成の場合は、Oracle Ultra Search Server コンポーネントを 1 つ以上の別のマシンにインストールして、拡張性を向上させることもできます。

Oracle Ultra Search Server コンポーネントは、既存のデータベース（Oracle Text がインストールされている Oracle9i リリース 1（9.0.1）以上のデータベース）にインストールできます。インストールするには、中間層のホストから OPCA（Oracle Portal Configuration Assistant）を起動し、データベース・ホストを指定します。

関連項目： 2-5 ページ「[Oracle Ultra Search Server コンポーネントの既存データベースまたはメタデータ・リポジトリへのインストール](#)」

表記規則

ORACLE_HOME は、Oracle Ultra Search Server コンポーネントがインストールされている Oracle ホーム・ディレクトリを示します。ORACLE_HOME は、Oracle Ultra Search Middle-Tier コンポーネントがインストールされている Oracle ホーム・ディレクトリも示します。

REMOTE_ORACLE_HOME は、Oracle Ultra Search Server コンポーネントがリモート Crawler ホスト上にインストールされている場合の Oracle ホーム・ディレクトリです。

Oracle Ultra Search のシステム要件

ここでは、Oracle Ultra Search のシステム要件について説明します。

ハードウェア要件

十分な RAM Oracle Ultra Search の索引付けエンジンは、Oracle で動作します。そのため、Oracle をインストールするには、システムに十分なメモリが必要です。Oracle インスタンスのシステム・グローバル領域には、50MB 以上のメモリが必要です。

関連項目： 『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』

Oracle Ultra Search Web Crawler は、同じホスト上で個別の Java プロセスとして実行されま
す。Web Crawler 単体に、50MB のメモリーを割り当ててください。

Oracle Ultra Search Administration ツールは、J2EE 1.2 標準の Web アプリケーションです。
そのため、Oracle Ultra Search Server コンポーネントとは別のホストにインストールして実
行できますが、通常は、Oracle Ultra Search Server コンポーネントと同じホストにインス
トールして実行します。いずれの場合も、J2EE エンジンには十分なメモリーを割り当ててく
ださい。Oracle HTTP Server は、Oracle J2EE コンテナとともに使用することをお薦めしま
す。J2EE エンジンを実行する JDK と同様に、HTTP Server にも十分なメモリーを割り当て
てください。

十分なディスク領域 顧客の要件は多様なため、ディスク領域に特定の値を推奨することは
できません。ただし、一般的なガイドラインとして推奨する最小値は、次のとおりです。

- Oracle9iAS のインフラストラクチャまたはデータベース、および Oracle Ultra Search
Server コンポーネント用のディスク領域は、約 3GB。
- 各 Web サーバー・ホストについては、Oracle HTTP Server と関連製品をインストールす
るのに十分な領域。さらに、Oracle Ultra Search Middle-Tier コンポーネントのインス
トールに、15MB のディスク領域が必要です。
- 各リモート Crawler ホストについては、Oracle Ultra Search Server コンポーネントのイン
ストールに必要なディスク領域と同じ大きさの領域。
- 大きな TEMPORARY 表領域に必要なディスク領域。一般的なガイドラインでは、ホスト
上の RAM に応じて、できるだけ大きな TEMPORARY 表領域を作成します。
- Oracle Ultra Search インスタンスのユーザー表領域に必要なディスク領域。
 - Oracle Ultra Search インスタンスのユーザーは、明示的に作成する必要があるデー
タベース・ユーザーです。クロール処理と索引付け処理の一部として収集および処
理されるすべてのデータは、ユーザーのスキーマに格納されます。
 - 一般的なガイドラインでは、索引付けするデータの合計と同じ大きさの表領域を作
成します。たとえば、クロールおよび索引付けされるデータの合計を約 10GB と見
積もった場合は、Oracle Ultra Search インスタンスのユーザー用に 10GB 以上の表
領域を作成する必要があります。この表領域を、Oracle Ultra Search インスタンス
のユーザー用のデフォルト表領域として確実に割り当ててください。

ソフトウェア要件

Oracle Ultra Search Middle-Tier コンポーネントは Web アプリケーションです。これらのコ
ンポーネントを実行するには、Web サーバーが必要です。Oracle HTTP Server と JServ、ま
たは Oracle HTTP Server と Oracle J2EE コンテナの使用をお薦めします。

Oracle Ultra Search Server コンポーネントのインストール

Oracle Ultra Search Server コンポーネントは、Oracle データベースの一部です。インストール時に、Oracle Ultra Search Middle-Tier コンポーネントとともにインストールされます。また、データベース・サーバー層と同じ Oracle ホーム・ディレクトリにインストールされます。

Oracle Ultra Search Server コンポーネントは、次のもので構成されます。

- Oracle Ultra Search のデータ・ディクショナリおよび PL/SQL パッケージ
- Oracle Ultra Search Crawler の Java クラス
- Oracle Ultra Search のリモート Crawler
- Oracle Ultra Search の製品ライブラリ

Oracle Ultra Search Server コンポーネントのインストールと構成

手順 1: Oracle9iAS インフラストラクチャおよび Oracle Ultra Search Server コンポーネントのインストール

データベース・リリース：関連するホストで、Oracle Universal Installer（OUI）を起動します。宛先の Oracle ホーム名とフルパスを選択してから、オプション「Oracle9i Server」を選択します。デフォルトで、Oracle Ultra Search Server Tier が Oracle9i データベースとともにインストールされます。

Oracle9iAS インフラストラクチャ：関連するホストで OUI を起動します。宛先の Oracle ホーム名とフルパスを選択してから、オプション「Oracle9iAS Infrastructure 9.0.2.0.0」を選択します。デフォルトで、Oracle Ultra Search Server Tier が Oracle9iAS インフラストラクチャとともにインストールされます。

Oracle9iAS または Oracle データベース・サーバーのインストール時に、Oracle Ultra Search Server コンポーネントがインストールされます。このとき、次の処理が行われます。

- すべての Oracle Ultra Search Server コンポーネント・ファイルが、ultrasearch という名前のディレクトリにコピーされます。このディレクトリは、データベースのインストール時に指定された ORACLE_HOME の直下にあります。
- データベース・ユーザー WKSYS（パスワードは wksys）が作成されます。セキュリティ上、このパスワードは後で変更してください。すべての Oracle Ultra Search データベース・オブジェクトは、このユーザーのスキーマにインストールされます。
- データベースに対し、様々な PL/SQL スクリプトがユーザー WKSYS で実行されます。これらのスクリプトによって、様々なデータベース・オブジェクトがインストールおよび作成されます。

関連項目： 必要な環境変数の設定については、インストール・ガイドを参照してください。

手順 2: Oracle Ultra Search 用の Oracle データベースの構成

すべての Oracle Ultra Search コンポーネントをインストールした後、Oracle データベースを構成します。これはインストール後の作業です。

関連項目： 3-2 ページ「Oracle Ultra Search 用の Oracle データベース・サーバーの構成」

Oracle Ultra Search Server コンポーネントの既存データベースまたはメタデータ・リポジトリへのインストール

Oracle Ultra Search は、現在使用している Oracle9i データベースまたは Oracle9iAS メタデータ・リポジトリにインストールできます。

注意： この機能は、Oracle9iAS でのみサポートされています。

Oracle Ultra Search の既存データベースまたはメタデータ・リポジトリへのインストール処理には、ファイルの転送、Oracle Ultra Search スーパー・ユーザー WKSYS の作成、Oracle Ultra Search PL/SQL データベース・パッケージのロード、Oracle9i Application Server (9iAS) と Oracle Ultra Search との関連付けが含まれます。

インフラストラクチャ・データベースのインストールとは異なり、Oracle Ultra Search パッケージは、中間層インストールからロードされます。Oracle9iAS Portal をデータベースにインストールすると、Oracle Portal Configuration Assistant (OPCA) によって、選択したデータベース・ホストに Oracle Ultra Search パッケージがロードされます。その後、必要なファイルをデータベース層のファイル・システムに手動で転送します。

関連項目： Oracle9iAS Portal および OPCA の使用方法の詳細は、<http://portalstudio.oracle.com/> を参照してください。

データベース要件

- Java Development Kit (JDK) がリリース 1.2.2.07 以上であること
- データベースが Oracle9i リリース 1 (9.0.1) 以上であること
- 初期化ファイルで JOB_QUEUE_PROCESSES が 3 以上に設定されていること
- Oracle Text がデータベースにインストールされていること。Oracle Text は、データ・ソース (Web サイトやデータベース表など) から取り出されたデータの索引付けおよび問合せに必要なテキスト索引付けと検索機能を提供します。

注意： Oracle Ultra Search では、Crawler を実行するために、Java 1.2 準拠の Java Runtime Environment (JRE) がデータベース・マシンにインストールされている必要があります。また、JDK 1.2.2.07 が Oracle HTTP Server とともにすでにインストールされている必要があります。

Oracle Ultra Search Server コンポーネントの既存の Oracle9i データベースへのインストール

1. ユーザー WKSYS を作成して、Oracle Ultra Search PL/SQL パッケージをロードし、Oracle Ultra Search と Oracle9iAS を関連付けます。

Oracle Ultra Search PL/SQL データベース・パッケージは、中間層 (Oracle9iAS) インストール時にロードされます。このインストール時に、Oracle Universal Installer が必要なファイルを中間層のホストにコピーし、その後、Oracle Portal Configuration Assistant (OPCA) が Oracle Ultra Search を Oracle9iAS に関連付けます。データベース層のホスト、Oracle SID およびポートを指定すると、OPCA によって、Oracle Ultra Search パッケージがデータベース層のホストのデータベースにロードされます。

Oracle Ultra Search スーパー・ユーザー WKSYS がすでに存在していることが OPCA で検出され、使用する Oracle Ultra Search がリリース 1.0.3 (Oracle9i データベース 9.0.1 にバンドル) の場合、ユーザーは次の 3 つのオプションのうちいずれかを選択するように求められます。

- 継続 : 既存の WKSYS ユーザーを削除して、新しい WKSYS ユーザーを作成し、Oracle Ultra Search パッケージを再ロードします。Oracle Ultra Search によって収集されたデータはすべて削除されます。
- 移行 : 手動で移行するために、Oracle Ultra Search ドキュメントの指示に従うようユーザーに要求します。

関連項目： 3-10 ページ [「Oracle Ultra Search リリース 1.0.3 \(Oracle9i データベース リリース 1 \(9.0.1\)\) からのアップグレード」](#)

- 強制終了 : OPCA は、Oracle Ultra Search PL/SQL データベース・パッケージの既存のデータベースへのロードを中止します。

注意： WKSYS のデフォルトのパスワードは `wksys` です。セキュリティ上の問題が発生しないように、パスワードをただちに変更する必要があります。次に、OPCA は Oracle9iAS を Oracle Ultra Search とともに動作するように構成します。`mod_oc4j.conf`、`server.xml` および `ultrasearch.properties` ファイルが構成に含まれます。詳細は、2-9 ページの [「Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール」](#) を参照してください。

2. ファイルを転送し、データベースを構成します。Oracle Ultra Search では、特定のファイルがデータベース層のファイル・システムに存在する必要があります。データベース層は物理的にアクセス不可能な位置に存在する場合があります。また、Oracle Universal Installer と OPCA のいずれもリモート・データベース層のファイル・システムにはアクセスできないため、必要なファイルは手動でコピーする必要があります。
\$ORACLE_HOME/ultrasearch/setup/ ディレクトリには、setupDB.jar、setupDB.bat、setupDB.csh および setupDB.sh など、いくつかのファイルがあります。これらのファイルを、リモート・データベース層の ORACLE_HOME にコピーしてください。
3. オペレーティング・システムごとに次のスクリプトを実行して、正しい位置にファイルを設定します。

Windows NT の場合: テキスト・エディタを使用して setupDB.bat を編集します。"set ORACLE_HOME=" の行の後に、Oracle データベースをインストールしたディレクトリ名を入力します。変更内容を保存した後、setupDB.bat をダブルクリックするか、コマンド・プロンプトで setupDB を実行して、setupDB.bat を実行します。

Solaris の場合: Bourne シェル・プロンプトで setupDB.sh を実行します。システムによって、カレント・ディレクトリが ORACLE_HOME に設定され、JDK_HOME へのパスの入力を要求されます。jar 実行可能ファイルまたは Java 実行可能ファイルへのデフォルトのパスが正しくない場合は、JDK へのパスの入力を要求されます。JDK をインストールしていない場合は、最新の JDK をダウンロードし、再度 setupDB.sh を実行します。

注意: このスクリプトを実行するには、setupDB.sh の実行可能ビットがオンになっている必要があります。

Oracle Ultra Search Server コンポーネントの既存の Oracle9iAS メタデータ・リポジトリへのインストール

1. ユーザー WKSYS を作成して、Oracle Ultra Search PL/SQL パッケージをロードし、Oracle Ultra Search を Oracle9iAS に関連付けます。

Oracle Ultra Search PL/SQL パッケージは、中間層 (Oracle9iAS) インストール時にロードされます。このインストール時に、Oracle Universal Installer が必要なファイルを中間層のホストにコピーし、その後、Oracle Portal Configuration Assistant (OPCA) が Oracle Ultra Search を Oracle9iAS に関連付けます。メタデータ・リポジトリ層のホスト、Oracle SID およびポートを指定すると、OPCA によって、Oracle Ultra Search パッケージがメタデータ・リポジトリ層のホストのメタデータ・リポジトリにロードされます。

検出された Oracle Ultra Search がリリース 9.0.2 以上の場合、ユーザーは次の 3 つのオプションのうちいずれかを選択するように求められます。

- 継続: 既存の WKSYS ユーザーを削除して、新しい WKSYS ユーザーを作成し、Oracle Ultra Search パッケージを再ロードします。Oracle Ultra Search によって収集されたデータはすべて削除されます。
- 9.0.3 にアップグレード: 既存の Oracle Ultra Search パッケージをリリース 9.0.3 にアップグレードします。Oracle Ultra Search によって収集されたすべてのデータはそのまま保持されます。
- 強制終了: OPCA は、Oracle Ultra Search PL/SQL パッケージの既存のメタデータ・リポジトリへのロードを中止します。

注意: WKSYS のデフォルトのパスワードは `wksys` です。セキュリティ上の問題が発生しないように、パスワードをただちに変更する必要があります。次に、OPCA は Oracle9iAS を Oracle Ultra Search とともに動作するように構成します。`mod_oc4j.conf`、`server.xml` および `ultrasearch.properties` ファイルが構成に含まれます。詳細は、2-9 ページの「[Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール](#)」を参照してください。

2. ファイルを転送し、メタデータ・リポジトリを構成します。Oracle Ultra Search では、特定のファイルがメタデータ・リポジトリ層のファイル・システムに存在する必要があります。メタデータ・リポジトリ層は物理的にアクセス不可能な位置に存在する場合があります、また、Oracle Universal Installer と OPCA のいずれもリモート・メタデータ・リポジトリ層のファイル・システムにはアクセスできないため、必要なファイルは手動でコピーする必要があります。`$ORACLE_HOME/ultrasearch/setup/` ディレクトリには、`setupDB.jar`、`setupDB.bat`、`setupDB.csh` および `setupDB.sh` など、いくつかのファイルがあります。これらのファイルを、リモート・メタデータ・リポジトリ層の `ORACLE_HOME` にコピーしてください。
3. オペレーティング・システムごとに次のスクリプトを実行して、正しい位置にファイルを設定します。

Windows NT の場合: テキスト・エディタを使用して `setupDB.bat` を編集します。`"set ORACLE_HOME="` の行の後に、Oracle メタデータ・リポジトリをインストールしたディレクトリ名を入力します。変更内容を保存した後、`setupDB.bat` をダブルクリックするか、コマンド・プロンプトで `setupDB` を実行して、`setupDB.bat` を実行します。

Solaris の場合: Bourne シェル・プロンプトで `setupDB.sh` を実行します。システムによって、カレント・ディレクトリが `ORACLE_HOME` に設定され、`JDK_HOME` へのパスの入力を要求されます。`jar` 実行可能ファイルまたは `Java` 実行可能ファイルへのデフォルトのパスが正しくない場合は、`JDK` へのパスの入力を要求されます。`JDK` をインストールしていない場合は、最新の `JDK` をダウンロードし、再度 `setupDB.sh` を実行します。

注意： このスクリプトを実行するには、`setupDB.sh` の実行可能ビットがオンになっている必要があります。

Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール

Oracle Ultra Search Middle-Tier コンポーネントには、次のものが含まれています。

- Oracle Ultra Search Administration ツール
- Oracle Ultra Search の Java Query API
- Oracle Ultra Search の JavaServer Pages (JSP) サンプル検索アプリケーション

Oracle9iAS の場合、Oracle Ultra Search Middle-Tier コンポーネントは、Application Server インストールの一部です。「Oracle Universal Installer」メニューから「Oracle9iAS Portal および Wireless」オプションを選択して、Application Server をインストールするとともに Oracle Ultra Search Middle-Tier コンポーネントをインストールして構成する必要があります。

Oracle9i データベースの場合、Oracle Ultra Search Middle-Tier コンポーネントは、データベース・サーバーのインストール時に Oracle Ultra Search Server コンポーネントとともにインストールされます。このコンポーネントは、データベース・クライアントの一部でもあります。ただし、Oracle データベース・クライアント・インストールでは、Oracle Ultra Search Middle-Tier コンポーネントに必須の Oracle HTTP Server はインストールされません。Oracle HTTP Server をインストールしてから、Oracle Ultra Search Middle-Tier コンポーネントをインストールする必要があります。「Oracle Universal Installer」メニューから「Oracle9i Database」インストール・オプションを選択してインストールします。その後、「カスタム」インストールを実行して、Oracle HTTP Server のインストールを選択します。

ここでは、Web サーバーとなるホストに、Oracle Ultra Search Middle-Tier コンポーネントをインストールして構成する方法を説明します。複数の Web サーバー・ホストに Oracle Ultra Search Middle-Tier コンポーネントをインストールすると、エンド・ユーザーによる大量の問合せを処理できます。

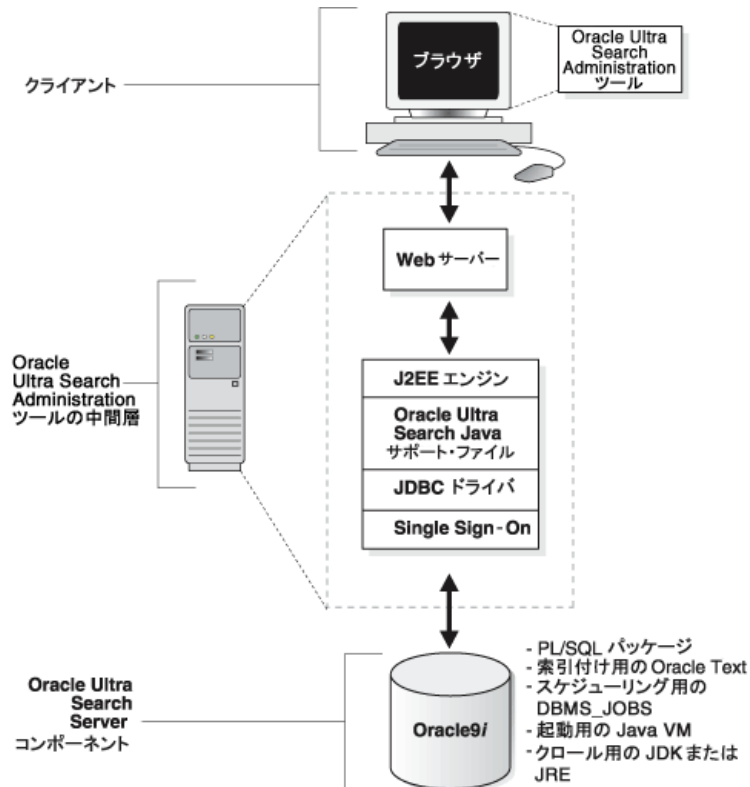
Web アプリケーションの概要

Oracle Ultra Search Administration ツールおよび Oracle Ultra Search の JSP 検索アプリケーションは、JSP の Web アプリケーションです。これらは、3 層アーキテクチャのアプリケーションです。図 2-1 に、ブラウザ・インタフェース（第 1 層）、Web サーバーと JSP エンジン（中間層）および Oracle データベース（第 3 層）の関係を示します。通常、JSP アプリケーションは、Web サーバーと同じホストにインストールします。Oracle Ultra Search の JSP アプリケーションは、Oracle Ultra Search の Java ライブラリ、JSP ファイルおよび `ultrasearch.properties` 構成ファイルで構成されています。このファイルは、JSP アプリケーションの接続先データベースを指定します。

Web サーバーはブラウザからリクエストを受け取り、Administration ツールのすべての JSP ページ・リクエストを JSP エンジンに転送して処理します。図 2-1 に示すように、Crawler 構成データを保存または取り出す場合、JSP エンジン、JDBC ドライバを使用して Oracle データベースに接続します。

Oracle Ultra Search Administration ツールまたは Oracle Ultra Search の JSP 検索アプリケーションを起動するには、これらの Web アプリケーションを起動する URL を入力します。URL については後述します。Oracle Ultra Search Administration ツールを使用して、Oracle Ultra Search インスタンスの定義、ユーザーの管理、Oracle Ultra Search Crawler の構成およびスケジュール設定、検索オプションの設定を行います。Oracle Ultra Search の JSP 検索アプリケーションを使用して、Oracle Ultra Search Server コンポーネントで収集および索引付けされたドキュメントを検索します。

図 2-1 Oracle Ultra Search のアーキテクチャ



ブラウザの要件

Administration ツールを使用するには、ブラウザは、Netscape バージョン 4.0 または Microsoft Internet Explorer バージョン 4.0 以上である必要があります。

Oracle Ultra Search Middle-Tier コンポーネントのインストール

Oracle データベースを使用してインストールする場合

Oracle Ultra Search Middle-Tier コンポーネントを構成するには、次の 2 つの方法があります。

オプション 1 Oracle Ultra Search Middle-Tier コンポーネントを、Oracle HTTP Server が事前にインストールされている Oracle ホームにインストールします。Oracle HTTP Server が事前にインストールされている場合は、インストーラによってサーバーが自動的に構成され、Oracle Ultra Search Middle-Tier コンポーネントを実行することが可能になります。Oracle HTTP Server をインストールするには、Oracle Universal Installer (OUI) のメニューから「Oracle9i Database」インストール・オプションを選択します。その後、「カスタム」インストールを実行して、Oracle HTTP Server のインストールを選択します。

オプション 2 Oracle HTTP Server を事前にインストールせずに、Oracle Ultra Search Middle-Tier コンポーネントをインストールします。

Oracle HTTP Server が事前にインストールされていない場合は、インストーラによって実行されるすべての構成作業を手動で行う必要があります。別の Web サーバーを使用する場合も、手動でサーバーを構成する必要があります。

インストール・オプションを選択します。関連するホストで OUI を起動します。Oracle9i Client のインストールを選択します。必ず「管理者」または「カスタム」インストール・オプションを選択してください。OUI から、Middle-Tier コンポーネントをインストールする Oracle ホーム・ディレクトリを入力するプロンプトが表示されます。ここでは、このディレクトリを、\$ORACLE_HOME とします。

Oracle Application Server を使用してインストールする場合

関連するホストで OUI を起動します。宛先の Oracle ホーム名とフルパスを選択してから、次の手順を実行します。

1. オプション「Oracle9iAS Application Server 9.0.2.0.0」を選択して、「次へ」をクリックします。
2. オプション「B. Portal および Wireless」を選択して、「次へ」をクリックします。
3. 「使用可能な製品コンポーネント」画面で、「Oracle Ultra Search Extension for EMD 9.0.2.0.0」がチェックされていることを確認します。「次へ」をクリックします。
4. 「コンポーネントの構成」画面で、「Oracle9iAS Portal」がチェックされていることを確認します。これにより、Oracle Portal Configuration Assistant は、Oracle Ultra Search を使用して、Oracle HTTP Server と OC4J を構成できます。このオプションのチェックを解除する場合は、「Oracle HTTP Server および JServ とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合」に記載されている指示に従って、Oracle HTTP Server と OC4J を手動で設定する必要があります。

5. Oracle9iAS が正常にインストールされるまで、インストールを続行します。

注意： サード・パーティの J2EE コンテナまたはサーブレット・エンジンを使用する場合は、Oracle Universal Installer の「コンポーネントの構成」画面でオプション「Oracle9iAS Portal」のチェックを解除し、2-20 ページの「[Oracle Ultra Search EAR ファイルのサード・パーティ Middle-Tier への配置](#)」を参照してください。この処理が完了すると、Middle-Tier コンポーネントのすべてのファイルが \$ORACLE_HOME にコピーされます。

Oracle Ultra Search Middle-Tier コンポーネントの構成

Oracle データベースを使用してインストールする場合

オプション 1 を選択した場合、OUI は Oracle HTTP Server と JServ を使用して、Oracle Ultra Search Middle-Tier コンポーネントを自動的に構成します。2-25 ページの「[ultrasearch.properties ファイルの編集](#)」に進んでください。

オプション 2 を選択した場合は、次の手順を手動で実行して、既存の Web サーバーを構成する必要があります。Oracle9i リリース 2 (9.2) のデフォルトの Web サーバー設定では、JServ とともに Oracle HTTP Server が構成されます。また、Oracle Ultra Search Middle-Tier コンポーネントを、Oracle J2EE コンテナ (OC4J) とともに Oracle HTTP Server に配置することもできます。ただし、Oracle J2EE コンテナは Oracle9i Application Server でのみ使用可能です。Oracle Ultra Search Middle-Tier コンポーネントの構成に使用する Web サーバーに対応する項に進んでください。

Oracle Application Server を使用してインストールする場合

Oracle Universal Installer の「コンポーネントの構成」画面で「Oracle9iAS Portal」オプションをチェックした場合は、後述の構成が OPCA によって自動的に実行されます。2-24 ページの「[data-sources.xml ファイルの編集](#)」に進んでください。

このオプションをチェックしなかった場合は、2-14 ページの「[Oracle HTTP Server および JServ とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合](#)」に記載されている手順を手動で実行して、既存の Web サーバーを構成する必要があります。

Oracle Enterprise Manager を使用して Oracle Ultra Search Web アプリケーションを配置することもできます。

関連項目： Oracle Enterprise Manager の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

Oracle HTTP Server および JServ とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合

1. Oracle Universal Installer (OUI) では、Administration ツールからアクセス可能な Oracle Ultra Search のオンライン・マニュアル用に、Web サーバーの別名が自動的に作成されます。インストーラが、\$ORACLE_HOME/Apache/jsp/conf/ojsp.conf を編集して、Oracle Ultra Search のドキュメント・ルートの別名を作成します。次の行が、このファイルに追加されます。

```
Alias /ultrasearch/doc/ "$ORACLE_HOME/ultrasearch/doc/"
```

2. OUI では、Oracle Ultra Search Administration ツール用に、Web サーバーの別名が自動的に作成されます。インストーラが、\$ORACLE_HOME/Apache/jsp/conf/ojsp.conf を編集して、Oracle Ultra Search の JSP ツリー・ルートの別名を作成します。次の行が、このファイルに追加されます。

```
Alias /ultrasearch/admin/ "$ORACLE_HOME/ultrasearch/webapp/isearch_admin/"
```

3. OUI では、Oracle Ultra Search JSP 検索アプリケーションと「ようこそ」ページ用に、Web サーバーの別名が自動的に作成されます。インストーラが、\$ORACLE_HOME/Apache/jsp/conf/ojsp.conf を編集して、Oracle Ultra Search の JSP ツリー・ルートの別名を作成します。次の行が、このファイルに追加されます。

```
Alias /ultrasearch/ "$ORACLE_HOME/ultrasearch/sample/"
```

注意： この別名は最後に位置づける必要があります。

4. OUI では、Oracle Ultra Search Middle-Tier ライブラリ、Oracle Ultra Search の Java Query API ライブラリおよび JGL オブジェクト・ライブラリが、Servlet Engine の Java CLASSPATH に自動的に追加されます。インストーラが、\$ORACLE_HOME/Apache/Jserv/etc/jserv.properties を編集して、これらのファイルを含めます。次の 5 行が追加されます。

```
wrapper.classpath=$ORACLE_HOME/ultrasearch/lib/ultrasearch_midtier.jar
wrapper.classpath=$ORACLE_HOME/ultrasearch/lib/ultrasearch_query.jar
wrapper.classpath=$ORACLE_HOME/ultrasearch/lib/jgl3.1.0.jar
wrapper.classpath=$ORACLE_HOME/lib/mail.jar
wrapper.classpath=$ORACLE_HOME/lib/activation.jar
```

5. OUI では、Oracle9i JDBC Thin ドライバが Servlet Engine の Java CLASSPATH に自動的に追加されます。インストーラが、\$ORACLE_HOME/Apache/Jserv/etc/jserv.properties を編集して、このディレクトリを含めます。次の 3 行が追加されます。

```
wrapper.classpath=$ORACLE_HOME/jlib/uiux2.jar
wrapper.classpath=$ORACLE_HOME/jlib/share.jar
wrapper.classpath=$ORACLE_HOME/jlib/regexp.jar
```

6. OUI では、Oracle9i JDBC Thin ドライバが Servlet Engine の Java CLASSPATH に自動的に追加されます。インストーラが、
\$ORACLE_HOME/Apache/Jserv/etc/jserv.properties を編集して、このディレクトリを含めます。次の行が追加されます。

```
wrapper.classpath=$ORACLE_HOME/jdbc/lib/classes12.zip
```

7. OUI では、ultrasearch.properties ファイルを含むディレクトリが、Servlet Engine の CLASSPATH に自動的に追加されます。インストーラが、
\$ORACLE_HOME/Apache/Jserv/etc/jserv.properties を編集して、このディレクトリを含めます。次の行が追加されます。

```
wrapper.classpath=$ORACLE_HOME/ultrasearch/webapp/config/
```

注意： Windows NT の場合、パスは
%ORACLE_HOME%\Apache\Jserv\conf\jserv.properties です。

8. Oracle Ultra Search Middle-Tier コンポーネントの現行のバージョンでは、バージョン 2.1 以上の servlet.jar が必要です。バージョン 2.2 の servlet.jar で JServ を構成するには、\$ORACLE_HOME/Apache/Jserv/etc/jserv.properties ファイルを編集し、次の行を追加します。

```
wrapper.classpath=$ORACLE_HOME/lib/servlet.jar
```

9. 2-25 ページの「[ultrasearch.properties ファイルの編集](#)」に進んでください。

Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合

注意： この機能は Oracle9iAS で使用する場合にのみ使用可能です。

Oracle Ultra Search Web アプリケーションを配置するには、J2EE 1.2 コンテナが必要です。Apache Web Server および Oracle Containers for J2EE (OC4J) の使用をお勧めします。

関連項目： サード・パーティの J2EE コンテナまたは Servlet Engine を使用する場合は、2-20 ページの「[Oracle Ultra Search EAR ファイルのサード・パーティ Middle-Tier への配置](#)」を参照してください。

1. OC4J 構成の場合は、\$ORACLE_HOME/j2ee/OC4J_Portal/config/にある、OC4J 構成ファイルの server.xml、application.xml および default-web-site.xml を変更します。OC4J の構成は、Oracle Ultra Search J2EE アプリケーションで動作します。

関連項目： EAR および WAR アプリケーションの配置方法、および OC4J の拡張機能の詳細は、OC4J のマニュアルを参照してください。

- server.xml の <application-server> タグの下に、次の行を追加します。

```
<application name="ultrasearch_admin" path="$ORACLE_
HOME/ultrasearch/webapp/ultrasearch_admin.ear" />

<application name="ultrasearch_query" path="$ORACLE_
HOME/ultrasearch/sample.ear" />

<application name="ultrasearch_portlet" path="$ORACLE_
HOME/ultrasearch/webapp/ultrasearch_portlet.ear" />
```

注意： これらの行によって、OC4J は、Oracle Ultra Search の EAR ファイルの場所を定義するだけでなく、EAR ファイルを配置する必要があることを認識します。Ultrasearch_admin.ear には、Oracle Ultra Search Administration ツールの Web アプリケーションが含まれています。sample.ear ファイルには、サンプル問合せ JSP ページが含まれています。OC4J によって sample.ear が配置されると、\$ORACLE_HOME/ultrasearch/sample ディレクトリを参照できます。このディレクトリの JSP を使用して、独自の問合せ Web ページを作成します。このディレクトリの詳細は、2-27 ページの「[Oracle Ultra Search の JSP サンプル検索アプリケーションのテスト](#)」を参照してください。

- application.xml の <orion-application> タグの下に、次の行を追加します。

```
<library path="$ORACLE_HOME/ultrasearch/lib/ultrasearch_query.jar" />
<library path="$ORACLE_HOME/ultrasearch/webapp/config" />
<library path="$ORACLE_HOME/jlib/uix2.jar" />
<library path="$ORACLE_HOME/jlib/share.jar" />
<library path="$ORACLE_HOME/jlib/regexp.jar" />
<library path="$ORACLE_HOME/lib/mail.jar" />
<library path="$ORACLE_HOME/lib/activation.jar" />
<library path="$ORACLE_HOME/lib/xmlparserv2.jar" />
<library path="$ORACLE_HOME/jdbc/lib/nls_charset12.zip" />
<library path="$ORACLE_HOME/jdbc/lib/classes12.jar" />
```

これらのライブラリは、Oracle Ultra Search Administration ツールおよび問合せ Web アプリケーションの実行に必要です。

注意： \$ORACLE_HOME/ultrasearch/webapp/config には、ultrasearch.properties ファイルが含まれています。詳細は、2-25 ページの「[ultrasearch.properties ファイルの編集](#)」を参照してください。

■ default-web-site.xml の場合

<web-site> タグに、属性として port="<ajp13 port>" と protocol="ajp13" を追加します。<web-site> タグの下に、次の行を追加します。

```
<web-app application="ultrasearch_admin" name="admin" root="/ultrasearch/admin" />
```

```
<web-app application="ultrasearch_query" name="query" root="/ultrasearch/query" />
```

```
<web-app application="ultrasearch_portlet" name="query"
root="/provider/ultrasearch" />
```

これらの行は、Oracle Ultra Search の EAR ファイル内のどの Web アプリケーション (WAR ファイル) が配置されるかを示します。

- application フィールドはアプリケーション名を示します。このアプリケーション名は、server.xml 内のアプリケーション名と一致している必要があります。
- name フィールドは Web アプリケーション名を示します。このアプリケーション名は、そのアプリケーションに対応している EAR ファイル内の WAR ファイル名と一致している必要があります。
- root は、この Web アプリケーションの仮想パスを指定します。この仮想パスは、URL の下にあるパスです。管理 Web アプリケーションには、http://<hostname.domainname>:<HTTPPort>/ultrasearch/admin/ を使用してアクセスします。

注意： 特定の Web アプリケーションの仮想パスは、次の 3 つのファイルに定義されます。EAR ファイルの META-INF ディレクトリ内の default-web-site.xml、mod_oc4j.conf および application.xml です (META-INF は、EAR ファイルを抽出すると作成されます)。各 Web アプリケーションへの仮想パス・ポイントを変更するには、default-web-site.xml の web-app のルート属性、および application.xml のタグ context-root で囲まれている値を変更する必要があります。

2. modOC4J 構成ファイルを変更します。次の行を
\$ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf に追加します。

```
OC4JMount /ultrasearch/admin/* ajp13://<host>:<ajp13 port>
OC4JMount /ultrasearch/query/* ajp13://<host>:<ajp13 port>
OC4JMount /provider/ultrasearch/* ajp13://<host>:<ajp13 port>
```

<host> は OC4J がインストールされているコンピュータのホスト名、<ajp13 port> は ajp13 プロトコルのポートです。このポートは、default-web-site.xml の下に記述されている <ajp13 port> と一致している必要があります、OC4Jmount は、default-web-site.xml 内の web-app と一致している必要があります。

注意： この modOC4J 構成では、\$ORACLE_HOME/j2ee/home/ ディレクトリでコマンド「java -jar oc4j.jar -config \$ORACLE_HOME/j2ee/OC4J_Portal/config/server.xml」を使用して、OC4J サーバーを手動で起動する必要があります。モジュール OC4J では、ロード・バランシングやフェイルオーバーなどの高度な機能が提供されます。必要に応じてこの機能をオンにしてください。この機能の構成に関しては、モジュール OC4J のマニュアルを参照してください。

3. Oracle HTTP Web Server に別名を追加します。
\$ORACLE_HOME/Apache/jsp/conf/ojsp.conf ファイルを変更して、Oracle Ultra Search のオンライン・ヘルプ・マニュアルおよび「Oracle Ultra Search へようこそ」ページを、次のディレクトリに配置します。

Solaris の場合: Alias /ultrasearch/doc/ "\$ORACLE_HOME/ultrasearch/doc/"

NT の場合: Alias /ultrasearch/doc/ "%ORACLE_HOME%\ultrasearch\doc/"

Solaris の場合: Alias /ultrasearch/ "\$ORACLE_HOME/ultrasearch/sample/"

NT の場合: Alias /ultrasearch/ "%ORACLE_HOME%\ultrasearch\sample/"

既存のデータベースに Oracle Ultra Search をインストールする場合は、この手順に加えて、ユーザーのデータベースに Oracle Ultra Search をインストールする必要があります。

関連項目： 2-5 ページ「[Oracle Ultra Search Server コンポーネントの既存データベースまたはメタデータ・リポジトリへのインストール](#)」

「[Single Sign-On Server とともに Administration ツールを構成する場合](#)」の指示に従ってください。それ以外の場合は、2-24 ページの「[data-sources.xml ファイルの編集](#)」に進んでください。

4. Oracle Ultra Search のサンプル・ページでは、インスタンス所有者としてデータベースへの JDBC 接続が必要です。JServ の制限事項により、JDBC 接続を作成するユーザー名、パスワードおよび接続文字列は、サンプルの JSP コード内でコード化されています。特定のインスタンスを問い合わせるために JSP を構成するには、JSP ソース・コードを編集し、ユーザー名、パスワードおよび接続文字列の値を置換します。

サンプル JSP ソース・コードは、すべて

`$ORACLE_HOME/ultrashow/sample/query` ディレクトリに含まれています。

次のファイルには、ユーザー名、パスワードおよび接続文字列の値が含まれています。

- `common_customize_instance.jsp` (`search.jsp` で使用)
- `usearch.jsp`
- `tag/tsearch.jsp`
- `9i/gsearch.jsp`
- `9i/display.jsp`
- `9i/gsearchf.jsp`
- `9i/gutil.jsp`
- `9i/mail.jsp`

注意： Oracle9i の JSP ファイルは使用できません。これらのファイルを使用する予定がない場合は、構成する必要はありません。

Single Sign-On Server とともに Administration ツールを構成する場合

注意： この機能は Oracle9iAS で使用する場合にのみ使用可能です。

Oracle Single Sign-On (SSO) Server とともに Oracle Ultra Search Administration ツールを構成するには、2-15 ページの「[Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合](#)」での構成以外に、次の手順を実行する必要があります。

1. OC4J 構成の場合は、`$ORACLE_HOME/j2ee/OC4J_Portal/config/`にある、OC4J 構成ファイルの `application.xml` および `default-web-site.xml` を変更します。
 - `application.xml` の `<orion-application>` タグの下に、次の行を追加します。


```
<library path="$ORACLE_HOME/jlib/repository.jar" />
<library path="$ORACLE_HOME/jlib/jndi.jar" />
<library path="$ORACLE_HOME/jlib/ldapjclnt9.jar" />
```

```
<library path="$ORACLE_HOME/j2ee/home/jazn.jar" />
<library path="$ORACLE_HOME/j2ee/home/jaas.jar" />
```

- default-web-site.xml の <web-site> タグの下に、次の行を追加します。

```
<web-app application="ultrasearch_admin" name="admin"
root="/ultrasearch/admin_sso" />
```

2. modOC4J 構成ファイルを変更します。次の行を mod_oc4j.conf に追加します。

```
Oc4jMount /ultrasearch/admin_sso/* ajp13://<host>:<ajp13 port>
```

3. \$ORACLE_HOME/Apache/Apache/conf ディレクトリの下での Oracle HTTP Server 構成ファイル httpd.conf を変更し、次の行を追加します。

```
<IfModule mod_ossso.c>
<Location /ultrasearch/admin_sso>
require valid-user
authType Basic
</Location>
</IfModule>
```

URL `http://<hostname.domainname>:<HTTPport>/ultrasearch/admin_sso/` にアクセスし、SSO 認証を使用して Oracle Ultra Search Administration ツールにログインします。

2-24 ページの「[data-sources.xml ファイルの編集](#)」に進んでください。

Oracle Ultra Search EAR ファイルのサード・パーティ Middle-Tier への配置

Oracle Ultra Search の EAR ファイルに含まれているのは Web アプリケーション (WAR ファイル) のみであるため、すべての J2EE 1.2 コンテナに配置できます。配置するには、Oracle Ultra Search の WAR ファイル名、事前定義 URL ルートおよび必要な Java ライブラリを認識する必要があります。ここでは、標準の J2EE 1.2 コンテナに配置するために認識する必要がある Oracle Ultra Search の EAR ファイルについて説明します。各 J2EE 1.2 コンテナの構成に関する情報は含まれていません。

関連項目：

- それぞれの構成については、サード・パーティの J2EE コンテナのドキュメントを参照してください。
- Oracle HTTP Server および Oracle J2EE コンテナの構成の詳細は、2-13 ページの「[Oracle Ultra Search Middle-Tier コンポーネントの構成](#)」を参照してください。

Administration ツールの配置 Oracle Ultra Search Administration ツールは、
\$ORACLE_HOME/ultrasearch/webapp/ultrasearch_admin.ear ファイルに含まれて
いる Web アプリケーションです。次に、ultrasearch_admin のファイル構造を示します。
アーカイブ・ファイルを抽出するには、コマンド「jar -xvf ultrasearch_admin.ear」
を実行します。

```
ultrasearch_admin.ear
  META-INF/
    application.xml
    orion-application.xml
  admin.war
  admin_sso.war
  ultrasearch_midtier.jar
  jgl3.1.0.jar
```

注意： admin.war は Oracle Ultra Search Administration ツールです。
admin_sso.war は SSO（シングル・サインオン）がサポートされている
Oracle Ultra Search Administration ツールです。
ultrasearch_midtier.jar と jgl3.1.0.jar は、admin.war および
admin_sso.war が使用する Java ライブラリです。これらは EJB ではありません。
admin.war および admin_sso.war のコンテキスト・ルートは、
それぞれ /ultrasearch/admin および /ultrasearch/admin_sso で
す。これらは、META-INF/application.xml に定義されています。

次に、admin.war および admin_sso.war の構造を示します。これらには同じファイル・
セットが含まれます。このファイル・セットは、コマンド「jar -xvf admin.war」およ
び「jar -xvf admin_sso.war」を実行すると参照できます。

```
admin_sso.war and admin.war
  WEB-INF/
    lib/
    web.xml
  index.jsp
  */*.jsp
```

Oracle Ultra Search の EAR ファイルと WAR ファイルを J2EE 1.2 に準拠させるには、
ultrasearch_midtier.jar および jgl3.1.0.jar を、admin.war および admin_sso.war
の WEB-INF/lib/ ディレクトリに移動する必要があります。次に、コマンド
「jar -cvf <war file or EAR file name> <files to jar>」を使用して、
admin.war、admin_sso.war および ultrasearch_admin.ear を再度 jar コマンドで圧縮
します。

注意： 作成された新しい admin.war ファイルは、サーブレット 2.2 標準の Web アプリケーションです。サーブレット 2.2 のエンジンを使用すると、admin.war を単体で配置できます。

次に、Oracle Ultra Search Administration ツールに必要な Java ライブラリを示します。このライブラリは、Oracle Ultra Search Administration ツールの配置に使用される J2EE コンテナまたは Servlet Engine のライブラリ・パスに追加する必要があります。

```
$ORACLE_HOME/ultrasearch/webapp/config
$ORACLE_HOME/jlib/uix2.jar
$ORACLE_HOME/jlib/share.jar
$ORACLE_HOME/jlib/regexp.jar
$ORACLE_HOME/jdbc/lib/nls_charset12.zip
$ORACLE_HOME/jdbc/lib/classes12.jar
$ORACLE_HOME/lib/xmlparserv2.jar
```

SSO とともに Oracle Ultra Search を構成するには、Oracle9iAS インフラストラクチャがインストールされていて、Oracle Internet Directory 機能がオンになっている必要があります。前述のライブラリ・パスの他に、次のライブラリ・パスも必要です。

```
$ORACLE_HOME/jlib/repository.jar
$ORACLE_HOME/jlib/jndi.jar
$ORACLE_HOME/jlib/ldapjclnt9.jar
$ORACLE_HOME/j2ee/home/jazn.jar
$ORACLE_HOME/j2ee/home/jaas.jar
```

サンプル検索アプリケーションの配置 Oracle Ultra Search のサンプル検索アプリケーションは、\$ORACLE_HOME/ultrasearch/sample.ear ファイルに含まれている Web アプリケーションです。このファイルは、J2EE1.2 標準に準拠しています。配置のために変更する必要はありません。

次に、sample.ear のファイル構造を示します。アーカイブ・ファイルを抽出するには、コマンド「jar -xvf sample.ear」を実行します。

```
sample.ear
    META-INF/
        application.xml
    query.war
    agent/
    index.html
```

問合せ JSP ページはすべて query.war に含まれます。このファイルは、サーブレット 2.2 標準の Web アプリケーションです。サーブレット 2.2 のエンジンを使用して、このファイルを単体で配置します。query.war のコンテキスト・ルートは、/ultrasearch/query です。これは、sample.ear ファイルの META-INF/application.xml に定義されています。コンテキスト・ルートを変更するには、このファイルを編集します。

次に、Oracle Ultra Search のサンプル検索アプリケーションに必要な Java ライブラリを示します。

```
$ORACLE_HOME/ultrasearch/webapp/config
$ORACLE_HOME/jdbc/lib/classes12.jar
$ORACLE_HOME/jdbc/lib/nls_charset12.zip
$ORACLE_HOME/lib/xmlparserv2.jar
$ORACLE_HOME/lib/activation.jar
$ORACLE_HOME/lib/mail.jar
```

Oracle Ultra Search の検索アプリケーションは、J2EE コンテナの接続プーリング機能も使用します。コンテナ認証済みのデータ・ソースを定義する必要があります。このデータ・ソースは、Oracle 接続に戻す必要があります。このデータ・ソースには、`oracle.jdbc.pool.OracleConnectionCacheImpl` と同等の Java クラスの使用をお勧めします。

また、このデータ・ソースには、フィールド位置に `jdbc/UltraSearchPooledDS`、ユーザー名とパスワードに Oracle Ultra Search のインスタンス所有者のデータベース・ユーザー名とパスワード、URL に「`jdbc:oracle:thin:@<database host>:<oracle port>:<oracle sid>`」の形式の JDBC 接続文字列が含まれている必要があります。

関連項目： Oracle J2EE コンテナのデータ・ソース構成の詳細は、2-24 ページの「[data-sources.xml ファイルの編集](#)」を参照してください。

Oracle Ultra Search ポートレットの配置 Oracle Ultra Search ポートレットは、`$ORACLE_HOME/ultrasearch/webapp/ultrasearch_portlet.ear` ファイルに含まれている Web アプリケーションです。このファイルは、J2EE1.2 標準に準拠しています。このファイルは、構造の点では `sample.ear` に類似しています。アーカイブ・ファイルを抽出するには、コマンド「`jar -xvf ultrasearch_portlet.ear`」を実行します。

```
ultrasearch_portlet.ear
  META-INF/
    application.xml
  query.war
  agent/
  index.html
```

問合せ JSP ページはすべて `query.war` に含まれます。このファイルは、サーブレット 2.2 準拠の Web アプリケーションです。サーブレット 2.2 のエンジンを使用すると、`query.war` を単体で配置できます。`query.war` のコンテキスト・ルートは、`/provider/ultrasearch/` です。これは、`ultrasearch_portlet.ear` ファイルの `META-INF/application.xml` に定義されています。コンテキスト・ルートを変更するには、このファイルを編集します。

次に、Oracle Ultra Search ポートレットに必要な Java ライブラリを示します。

```
$ORACLE_HOME/jdbc/lib/classes12.jar
$ORACLE_HOME/jdbc/lib/nls_charset12.zip
```

```
$ORACLE_HOME/lib/xmlparserv2.jar  
$ORACLE_HOME/lib/activation.jar  
$ORACLE_HOME/lib/mail.jar
```

Oracle Ultra Search ポートレットは、J2EE コンテナの接続プーリング機能を使用します。コンテナ認証済みのデータ・ソースを定義する必要があります。このデータ・ソースは、Oracle 接続を戻す必要があります。このデータ・ソースには、`oracle.jdbc.pool.OracleConnectionCacheImpl` と同等の Java クラスの使用をお勧めします。

また、このデータ・ソースには、フィールド位置に `jdbc/UltraSearchPooledDS`、ユーザー名とパスワードに Oracle Ultra Search のインスタンス所有者のデータベース・ユーザー名とパスワード、URL に「`jdbc:oracle:thin:@<database host>:<oracle port>:<oracle sid>`」の形式の JDBC 接続文字列が含まれている必要があります。

関連項目： Oracle J2EE コンテナのデータ・ソース構成の詳細は、2-24 ページの「[data-sources.xml ファイルの編集](#)」を参照してください。

data-sources.xml ファイルの編集

注意： この機能は Oracle9iAS で使用する場合にのみ使用可能です。

注意： `data-sources.xml` に格納されている消去されたデータベース・スキーマのパスワードは保護されていません。パスワードを保護するには、独自のソリューションを実装する必要があります。

Oracle9iAS ベースの Oracle Ultra Search Query API は、J2EE コンテナのデータ・ソース機能を使用します。`$ORACLE_HOME/j2ee/OC4J_Portal/config` の下の、ファイル `data-sources.xml` を編集します。タグ `<data-sources>` の下に、次の行を追加します。

```
<data-source  
  class="oracle.jdbc.pool.OracleConnectionCacheImpl"  
  name="UltraSearchDS"  
  location="jdbc/UltraSearchPooledDS"  
  username="<username>"  
  password="<password>"  
  url="jdbc:oracle:thin:@<database_host>:<oracle_port>:<oracle_sid>"  
</>
```

`<username>` と `<password>` は Oracle Ultra Search のインスタンス所有者のデータベース・ユーザー名とパスワード、`<database_host>` はバックエンド・データベース・マシンのホスト名、`<oracle_port>` はユーザーの Oracle データベースへのポート、`<oracle_sid>` はユーザーの Oracle データベースの SID です。ユーザー名、パスワードお

よび JDBC URL の他に、data-sources.xml では、接続キャッシュ・サイズとキャッシュ・スキームも構成できます。次のタグは、キャッシュ・サイズの最小値と最大値、非アクティブのタイムアウト間隔およびキャッシュ・スキームを指定します。

```
<data-source
  class="oracle.jdbc.pool.OracleConnectionCacheImpl"
  name="UltraSearchDS"
  location="jdbc/UltraSearchPooledDS"
  username="wk_test"
  password="wk_test"
  url="jdbc:oracle:thin:@localhost:5521:isearch"
  min-connections="3"
  max-connections="30"
  inactivity-timeout="30">
  <property name="cacheScheme" value="1"/>
</data-source>
```

注意： JDBC データ・ソースの URL は、
 jdbc:oracle:thin:@[hostname]:[port]:[sid] の形式、または TNS
 キーワード - 値の構文の形式で指定できます。たとえば、次のように指定
 します。"jdbc:oracle:thin:@(DESCRIPTION=(LOAD_
 BALANCE=yes)(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
 (HOST=c1s02a)(PORT=3999))(ADDRESS=(PROTOCOL=TCP)
 (HOST=c1s02b)(PORT=3999)))(CONNECT_DATA=(SERVICE_
 NAME=acme.us.com)))"

キャッシュ・スキームには、次の 3 種類があります。

- DYNAMIC_SCHEME = 1
- FIXED_WAIT_SCHEME = 2
- FIXED_RETURN_NULL_SCHEME = 3

関連項目：『Oracle9iAS Containers for J2EE サービス・ガイド』

ultrasearch.properties ファイルの編集

ultrasearch.properties ファイルには、Web アプリケーションおよび JSP アプリケーションが接続するデータベースを指定します。

\$ORACLE_HOME/ultrasearch/webapp/config/ultrasearch.properties を編集して、Oracle インスタンスおよびリスナーのホスト名、ポートおよび SID を指定します。

指定するには、connection.url で始まる行を

connection.url=jdbc:oracle:thin:hostname:port:SID のように編集します。

- `hostname` には、Oracle Ultra Search を実行している Oracle ベース・インスタンスのホスト名をフルネームで入力します。
- `port` には、Oracle9i データベース・インスタンスのリスナー・ポート番号を入力します。
- `SID` には、Oracle9i データベース・インスタンスの ID を入力します。

次に、`connection.url` 文字列の例を示します。

```
connection.url=jdbc:oracle:thin:@ultrasearch.us.oracle.com:1521:myInstance
```

Oracle HTTP Server と JServ を使用して Oracle Ultra Search Middle-Tier コンポーネントを構成するように選択した場合は、`admin.srchome` で始まる行を `admin.srchome=<jsp_src_home>` のように編集する必要もあります。

- Solaris の場合は、「`admin.srchome=$ORACLE_HOME/ultrasearch/webapp/isearch_admin`」と入力します。
- NT の場合は、「`admin.srchome=%ORACLE_HOME%\ultrasearch\webapp\isearch_admin`」と入力します。

`$ORACLE_HOME` および `%ORACLE_HOME%` は、Oracle Ultra Search Server コンポーネントがインストールされている Oracle ホーム・ディレクトリです。

これは、Oracle Ultra Search Administration ツールの JSP ページの場所です。

注意： 1 行目は Oracle JDBC ドライバの名前であるため、変更する必要はありません。

関連項目： Oracle JDBC ドライバのマニュアルを参照してください。

Web サーバーの起動

Oracle9iAS の場合のみ、Oracle Portal Configuration Assistant で Web サーバーを構成するために、Enterprise Manager コンソールを使用して Web サーバーを起動する必要があります。

関連項目： Enterprise Manager コンソールの詳細は、『Oracle9i Application Server 管理者ガイド』を参照してください。

Oracle9i データベースの場合、または Oracle HTTP Server と OC4J を手動で構成するように選択した場合は、次の手順を実行します。

- `$ORACLE_HOME/Apache/Apache/bin/spachectl stop` の後で `$ORACLE_HOME/Apache/Apache/bin/spachectl start` を起動して、Oracle HTTP Server を起動します。

- デフォルトの HTTP ポートは 7777 です。これを変更するには、`$ORACLE_HOME/Apache/Apache/conf/httpd.conf` を編集します。
- Oracle HTTP Server が起動しているかどうかをテストするには、「Oracle Ultra Search へようこそ」ページの
`http://<hostname.domainname>:<HTTPport>/ultrasearch/index.html` を表示します。このページには、Oracle Ultra Search に関する一般的な情報が掲載されています。また、Oracle Ultra Search Administration ツールおよび Oracle Ultra Search のサンプル問合せ JSP ページへのリンクも含まれています。ただし、Oracle Ultra Search Administration ツール、Oracle Ultra Search のサンプル問合せページおよびディレクトリ `$ORACLE_HOME/ultrasearch/sample` は、OC4J の起動後は使用できません。
- OC4J とともに Oracle Ultra Search Middle-Tier コンポーネントを配置する場合は、`java -jar $ORACLE_HOME/j2ee/home/oc4j.jar -config $ORACLE_HOME/j2ee/OC4J_Portal/config/server.xml` を起動して OC4J を起動します。
- Oracle HTTP Server の起動時に OC4J が自動的に起動するように構成できます。詳細は、OC4J のマニュアルを参照してください。

Oracle Ultra Search Administration ツールのテスト

Web サーバーが実行中であることを確認します。

Administration ツールにログオンして、変更をテストします。

- `http://<hostname.domainname>:<HTTPport>/ultrasearch/admin/index.jsp` を表示します。

`<hostname.domainname>` は Oracle Ultra Search Middle-Tier コンポーネントをインストールしているホストのフルネーム、`<HTTPport>` はデフォルトの Web サーバー・ポートです。同じホストで Web ブラウザを実行している場合は、「localhost」を入力します。
- Oracle Ultra Search Server コンポーネントのインストール時に、Oracle Ultra Search インスタンスの所有者を新しく作成しておく必要があります。インスタンスの所有者は、「Oracle Ultra Search 用の Oracle データベース・サーバーの構成」の手順 4 で作成されます。Oracle Ultra Search インスタンスの所有者のデータベース・ユーザー名とパスワードを入力して、Oracle Ultra Search Administration ツールにログオンします。
- JSP ページでは、初めてページにアクセスしたとき、コンパイルに数秒かかります。2 回目以降のアクセスでは、これより速くなります。
- Oracle Ultra Search Administration ツールに正常にログオンできた場合は、Oracle Ultra Search Administration ツールの構成処理が完了しています。

Oracle Ultra Search の JSP サンプル検索アプリケーションのテスト

Oracle Ultra Search Administration ツールが動作していることを確認できた場合は、Oracle Ultra Search の JSP サンプル検索アプリケーションを実行できます。

Oracle Ultra Search の JSP サンプル検索アプリケーションをテストするには、次のいずれかを実行します。

- `http://<hostname.domainname>:<HTTPport>/ultrasearch/query/search.jsp` を表示します。
- 「Oracle Ultra Search へようこそ」 ページ (`http://<hostname.domainname>:<HTTPport>/ultrasearch/index.html`) のリンクに従います。

関連項目： 特定のインスタンスを問い合わせるために JSP を構成する方法については、2-15 ページの「[Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier コンポーネントを構成する場合](#)」を参照してください。

次に、サンプル検索アプリケーションの位置を示します。次に示すディレクトリに移動すると、サンプルのソース・コードにアクセスできます。また、URL ルートを使用して各サンプル問合せ JSP ページの動作デモを参照することもでき、URL ルートの最後に正しい JSP ファイル名を追加できます。

検索アプリケーション用ルート・ディレクトリは、
`$ORACLE_HOME/ultrasearch/sample/query/` です。

問合せの URL ルートは、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/` です。

Oracle9iAS の問合せ (Oracle9iAS の Ultra Search Query API と、`usearch.jsp` および `search.jsp` を含む検索サンプル JSP ページ) は、
`$ORACLE_HOME/ultrasearch/sample/query/` にあります。

Oracle9iAS の問合せの URL ルートは、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/` です。

(たとえば、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/search.jsp` と指定すると、`search.jsp` にアクセスできます。)

Oracle9i の問合せ (Oracle9i データベースリリース 1 (9.0.1) の Ultra Search Query API と、`gsearch.jsp` を含む問合せ JSP) は、
`$ORACLE_HOME/ultrasearch/sample/query/9i/` にあります。

Oracle9i の問合せの URL ルートは、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/9i/` です。

ポートレットは、`$ORACLE_HOME/ultrasearch/sample/query/portlet/` にあります。

ポートレットの URL ルートは、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/portlet/` です。
taglib は、`$ORACLE_HOME/ultrasearch/sample/query/tag/` にあります。
taglib の URL ルートは、
`http://<hostname.domainname>:<HTTPport>/ultrasearch/query/tag/` です。

Oracle Ultra Search Server コンポーネントのリモート Crawler ホストへのインストール

Oracle Ultra Search のリモート Crawler では、異なるホスト上で複数の Crawler をパラレルに実行できます。ただし、すべてのリモート Crawler ホストは、共有ディレクトリや Oracle Ultra Search の共有データベースなどのリソースを共有する必要があります。

Oracle Ultra Search Server コンポーネントをリモート Crawler ホストにインストールする場合

Oracle Ultra Search のリモート Crawler は、Oracle Ultra Search Server コンポーネントの一部です。そのため、リモート Crawler のインストール手順は、Oracle Ultra Search Server コンポーネントのインストール手順と同じです。

Oracle Ultra Search Server コンポーネントは、各リモート Crawler ホスト上で、Oracle ホームと呼ばれる共通ディレクトリにインストールされます。このディレクトリを入力するプロンプトが表示されます。Oracle ホーム・ディレクトリは `$REMOTE_ORACLE_HOME` とします。

Oracle9i Application Server (9iAS) のインストール時に Oracle HTTP Server をインストールしなかった場合は、リモートのクロールを行うために次の一連の手順を手動で実行する必要があります。

- UNIX システムの場合は
`$REMOTE_ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/unix/define_env` を指定し、Windows NT システムの場合は
`$REMOTE_ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/winnt/define_env.bat` を指定します。
- `%ORACLE_HOME%` を環境変数 `REMOTE_ORACLE_HOME` の値に置き換えます。
- `%s_jreLocation%` を Java Runtime Environment (JRE) バージョン 1.2.2 以上のディレクトリ・パスに置き換えます。JRE のルート・ディレクトリを指定する必要があります。
- `%s_jreJDBCclassfile%` を Oracle JDBC Thin ドライバ (バージョン 12) のフルパスおよびファイル名に置き換えます。

Oracle Ultra Search Server コンポーネントをリモート Crawler ホストで構成する場合

Oracle Ultra Search のリモート Crawler ホストに必要な構成は、Oracle Ultra Search システムにホストを登録することのみです。この登録処理は、Oracle Ultra Search のリモート Crawler のホスト上で SQL スクリプトを実行することによって行われます。SQL スクリプトによって、SQL*Plus で Oracle9iAS 中間層に接続し、リモート Crawler ホストを登録します。

1. Oracle ホームを正確に指定します。

Oracle Ultra Search Middle-Tier コンポーネントは、共通ディレクトリである Oracle ホームにインストールされます。Oracle Ultra Search Middle-Tier コンポーネントの前に他の Oracle 製品をインストールした場合は、使用中のホストに複数の Oracle ホームが存在している可能性があります。登録スクリプト (register.sql) により、Oracle Ultra Search Middle-Tier コンポーネントがインストールされている Oracle ホームのディレクトリを入力するように求められます。

2. WKSYS スーパー・ユーザー・パスワードを指定します。

WKSYS スーパー・ユーザーまたは WKADMIN ロールを付与されたデータベース・ユーザーとして登録スクリプトを実行する必要があります。

3. SQL*Plus を起動します。

この SQL スクリプトは、\$REMOTE_ORACLE_HOME の下の
/ultrasearch/tools/remotecrawler/scripts/common/register.sql にあります。

SQL*Plus を実行します。他の Oracle 製品がインストールされている場合は、複数のバージョンの SQL*Plus が同じホスト上に存在している可能性があるため、必ず適切なバージョンの SQL*Plus を実行してください。UNIX プラットフォームでは、変数 PATH、ORACLE_HOME および TNS_ADMIN に適切な値が設定されていることを確認する必要があります。Windows プラットフォームでは、「スタート」メニューから適切なメニュー項目を選択します。

適切な SQL*Plus クライアントの実行方法を確認してから、Oracle Ultra Search データベースにログオンする必要があります。そのためには、Oracle Ultra Search データベース用に Oracle Net Services の設定を構成しておく必要があります。

関連項目： Oracle Net Services の設定を構成する方法は、『Oracle9i Net Services 管理者ガイド』を参照してください。

SQL*Plus を実行した後、手順 2 で位置を指定したスキーマとパスワードを使用して、データベースにログオンします。

4. 登録スクリプトを起動します。

SQL*Plus を WKSYS スーパー・ユーザーで起動し、次のように入力します。

```
@<full path name of registration script>
```

たとえば、UNIX ホスト上で \$REMOTE_ORACLE_HOME の値が /home/oracle9i となっている場合は、SQL*Plus プロンプトで次のように入力します。

```
@/home/oracle9i/ultrasearch/tools/remotecrawler/scripts/unix/register.sql
```

Windows NT 上で SQL*Plus を実行中で、\$REMOTE_ORACLE_HOME が d:\Oracle\Oracle9i にある場合は、SQL*Plus プロンプトで次のように入力します。

```
@d:\Oracle\Oracle9i\ultrasearch\tools\remotecrawler\scripts\winnt\register.sql
```

登録スクリプトによって、2 つの変数を入力するプロンプトが表示されます。該当する変数とその説明のリストを次に示します。

REMOTE_CRAWLER_HOSTNAME: リモート Crawler ホストの DNS ホスト名

ORACLE_HOME: 手順 1 で位置を指定した Oracle ホーム。たとえば、UNIX ホストにおける /u01/oracle9i または Windows NT ホストにおける D:\u01\Oracle9i などです（Windows NT ホストでは、スラッシュは注意して使用してください）。

登録スクリプトにより、wk_crw.register_remote_crawler PL/SQL API が起動します。wk_crw.register_remote_crawler の API の引数を構成するには、変数 REMOTE_CRAWLER_HOSTNAME および ORACLE_HOME を使用します。

5. リモート Crawler プロファイル構成を検証し、完了します。

2 つの変数には、必ず適切な値を入力してください。登録が正しく完了したかどうかを検証するには、Oracle Ultra Search Administration ツールにログオンします。

「Crawler」タブの「リモート Crawler プロファイル」サブタブをクリックします。リモート Crawler プロファイル・リストに登録したリモート Crawler のホスト名を確認します。「編集」をクリックして、リモート Crawler プロファイルの構成を完了します。

誤った値を入力した場合

register.sql スクリプトに誤った値を入力した場合は、unregister.sql スクリプトを使用してリモート Crawler の登録を解除する必要があります。登録解除のスクリプトを、登録用スクリプトと同じ方法で起動します。unregister.sql スクリプトは wk_crw.unregister_remote_crawler の PL/SQL API をコールします。

リモート Crawler の登録を正しく解除すると、register.sql スクリプトを再実行できます。

インストール後の作業とアップグレードに関する情報

この章では、次の項目について説明します。

- [Oracle Ultra Search 用の Oracle データベース・サーバーの構成](#)
- [ストップリストの管理](#)
- [最新の Oracle Ultra Search へのアップグレード](#)

Oracle Ultra Search 用の Oracle データベース・サーバーの構成

データベース管理の操作について説明します。この操作は、Oracle Enterprise Manager または SQL*Plus を使用して行います。

ここでは、次の手順について説明します。

- Oracle Ultra Search 用の Oracle データベース・サーバーの構成
- Oracle Ultra Search の各インスタンス用の表領域およびユーザーの作成

手順 1: Oracle データベースのチューニング

Oracle REDO ログのサイズの拡張（必要な場合）

Oracle データベースのすべてのインスタンスには、関連するオンライン REDO ログがあります。これは、データベースに対してコミットされた変更をすべて記録する複数のオンライン・ログ・ファイルの集合です。オンライン REDO ログは、インスタンスに障害が発生したときにデータベースを保護します。REDO ログ・ファイルのサイズによって、REDO ログ・ファイルの切替え頻度が決まります。この頻度は、テキストの索引付けのスピードにも大きく影響します。ログ・ファイルの切替え頻度を減らすために、REDO ログ・ファイルのサイズはそれぞれ 10MB 以上であることを確認してください。

ここでは、REDO ログ・ファイルのサイズを拡張する方法を簡単に説明します。適切な Oracle の管理者権限を使用して、次のセクションに文を入力します。

関連項目：

- 『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』
- 『Oracle9i データベース管理者ガイド』

1. REDO ログ・ファイルの位置およびサイズを決定します。

```
SELECT v$logfile.member, v$logfile.group#, v$log.status, v$log.bytes
FROM v$log, v$logfile
WHERE v$log.group# = v$logfile.group#
```

2. サイズの大きい REDO ログ・ファイルを追加します。

```
ALTER DATABASE ADD LOGFILE '<redo log directory>/newredo1.log' size 10m;
ALTER DATABASE ADD LOGFILE '<redo log directory>/newredo2.log' size 10m;
ALTER DATABASE ADD LOGFILE '<redo log directory>/newredo3.log' size 10m;
```

本番データベースでは、ログ・グループごとのログ・メンバーを増加し、異なる記憶デバイスを使用してパフォーマンスと信頼性を向上させます。

3. 元のログ・ファイルを削除します。元の各 REDO ログ・ファイルに対して ALTER SYSTEM SWITCH LOGFILE 文を入力し、ステータスを INACTIVE にします。これは、ログ・ファイルの削除時に Oracle がこのログ・ファイルを使用しないようにするためです。

次の文を入力して、元の REDO ログ・ファイルを削除します。

```
ALTER DATABASE DROP LOGFILE '<redo log directory>/redo01.log';
ALTER DATABASE DROP LOGFILE '<redo log directory>/redo02.log';
ALTER DATABASE DROP LOGFILE '<redo log directory>/redo03.log';
```

4. 元のログ・ファイルをファイル・システムから手動で削除します。元の各 REDO ログ・ファイルについては、適切なオペレーティング・システム文を使用して不要なログ・ファイルをファイル・システムから削除します。

UNDO 領域のサイズの拡張

すべての Oracle データベースには、データベースに対する変更のロールバック (UNDO) に使用する情報を保持する方法が必要です。このような情報は、主にコミットされる前のトランザクション処理のレコードで構成されます。これらのレコードは集散的に UNDO と呼ばれます。Oracle Universal Installer によって作成される UNDO 領域は、通常は小さすぎます。

従来、Oracle では UNDO の保存にロールバック・セグメントを使用していました。現在は別の方法で UNDO を保存しているため、ロールバック・セグメント領域の管理方法が容易になり、UNDO が上書きされるまでの時間を DBA が制御できます。この方法では、UNDO 表領域が使用されます。

自動 UNDO 管理を使用し、UNDO_TABLESPACE を使用して UNDO 領域を大きくすることをお勧めします。

関連項目： 自動 UNDO 管理の使用の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

init<SID>.ora ファイルの Oracle 初期化パラメータのチューニング

PROCESSES: 50 以上にします。

SORT_AREA_SIZE: 5MB 以上にします。

SORT_AREA_RETAINED_SIZE: 5MB 以上にします。

JOB_QUEUE_PROCESSES: 3 にします (必ず 1 以上に設定してください)。Oracle Ultra Search Crawler は、データベース・ジョブをスケジューリングして起動するため、このパラメータが必要になります。0 (ゼロ) に設定すると、データベース・ジョブは実行されません。そのため、Oracle Ultra Search Crawler が起動しません。

初期化パラメータの最新情報については、Oracle Ultra Search の README を参照してください。

手順 2: CTXSYS ユーザー用の一時表領域の作成および割当て

Oracle Universal Installer によって作成された初期データベースが作成する一時表領域は、通常は小さすぎます。Oracle Ultra Search は、Oracle Text エンジンを集团的に使用します。このため、Oracle Text システムの CTXSYS ユーザー用に大規模な一時表領域を作成する必要があります。

読取りおよび書込みのパフォーマンスを向上させる場合は、RAW 表領域を作成します。

作成した一時表領域を、CTXSYS ユーザーに一時表領域として割り当てます。割り当てるには、SYSTEM または SYS ユーザーでログオンする必要があります。次の文を入力して、CTXSYS ユーザーに一時表領域を割り当てます。

```
ALTER USER CTXSYS TEMPORARY TABLESPACE <NEW_TEMPORARY_TABLESPACE>
```

関連項目： 一時表領域の作成方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

手順 3: Oracle Ultra Search インスタンスの各ユーザー用の大規模な表領域の作成

Oracle Ultra Search の各インスタンス用に、クロールおよび索引付けの処理中に取得するすべてのデータを格納するために、十分な大きさの表領域を作成する必要があります。表領域の大きさは、クロールおよび索引付けするデータ量によって異なります。ただし、ほとんどの場合、収集するデータ量は事前にわかりません。クロールするすべてのデータの累積サイズを見積もってください。

サイズを見積もることができない場合は、できるだけ大きい領域を割り当ててください。後でディスク領域がなくなった場合でも、インスタンスの表領域にデータ・ファイルを追加すると Oracle Ultra Search はクロールを再開できます。

CREATE TABLESPACE 文の STORAGE 句に注意してください。表領域に格納されるデータ量が非常に大きくなることがあります。この場合、より多くの記憶領域が必要になるたびに、新しいエクステンツを割り当てることになります。

extent management 句で新しいエクステンツが現在のエクステンツより大きくなるように指定した場合は（つまり、PCTINCREASE の設定が 0（ゼロ）以外の場合）、割り当てられる次のエクステンツが、使用可能なエクステンツより大きくなることがあります。このような場合は、表領域に新しいエクステンツが追加されるまで索引付けが停止されます。

この問題を回避するには、特定のインスタンス固有の表に記憶域パラメータを明示的に設定する必要があります。これらのインスタンス固有の表には、初期エクステンツ・サイズ、次のエクステンツ・サイズおよび PCTINCREASE 設定を定義します。新しいインスタンスが作成されると、これらの表が作成されます。表および STORAGE 句の設定は、次のとおりです。


```
DR$WK$DOC_PATH_IDX$I
      (initial extent size 5M, next extent size 50M, PCTINCREASE 1)
DR$WK$DOC_PATH_IDX$K
      (initial extent size 5M, next extent size 50M, PCTINCREASE 1)
```

読取りおよび書込みのパフォーマンスを向上させる場合は、RAW 表領域を作成します。

Oracle Ultra Search インスタンスの各ユーザー用に、大規模な表領域を新しく作成する必要があります。

関連項目：

- 表領域の作成と記憶域設定の管理に関する詳細は、『Oracle9i SQL リファレンス』を参照してください。
- 表領域の作成方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

手順 4: 各 Oracle Ultra Search インスタンス用の新規データベース・ユーザーの作成および構成

Oracle Ultra Search システムは、Oracle のファイングレイン・アクセス・コントロール機能を使用して、複数の Oracle Ultra Search インスタンスを 1 つの物理データベースでサポートします。この機能は、1 つの Oracle の物理インストールで複数の非結合検索の索引をホストする大きな組織やアプリケーション・サービス・プロバイダ（ASP）で特に有効です。

重要：Oracle Ultra Search システムでは、Oracle Ultra Search の各仮想インスタンスが、一意のデータベース・ユーザーに対応している必要があります。そのため、インストール・プロセスの一部として、Oracle Ultra Search インスタンス用にすべてのデータを所有する 1 つ以上の新しいデータベース・ユーザーを作成する必要があります。

注意： 2 つ以上のデータベース・インスタンスを作成する場合は、各ユーザーに対してユーザー表領域を 1 つずつ作成する必要があります。

各 Oracle Ultra Search ユーザーには、特定のロールと権限を付与する必要があります。便宜上、WKUSER ロールには必要なすべての権限が付与されています。インスタンス所有者には、WKUSER ロールを付与する必要があります。

関連項目： 5-34 ページ「[「ユーザー」タブ](#)」

次の文を入力して、新しいユーザーを作成および構成します。これらの文は、WKSYS、SYSTEM または SYS データベース・ユーザーで実行します。

```
CREATE USER <username>
        IDENTIFIED BY <password> DEFAULT TABLESPACE <default_tbs>
        TEMPORARY TABLESPACE <temporary_tbs> QUOTA UNLIMITED
        ON <default_tbs>;
```

<username> = Oracle Ultra Search インスタンス所有者の名前

<password> = Oracle Ultra Search インスタンス所有者のパスワード

<default_tbs> = 手順 3 で作成した Oracle Ultra Search インスタンスのデフォルト表領域

<temporary_tbs> = 手順 2 で作成した一時表領域

```
GRANT WKUSER TO <username>;
```

これらの手順を完了すると、WKSYS または Oracle Ultra Search スーパー・ユーザーは、このユーザー・スキーマに Oracle Ultra Search インスタンスを作成できます。

このユーザーに Oracle Ultra Search の一般管理権限またはスーパー・ユーザー権限を付与する場合は、Oracle Ultra Search スーパー・ユーザーまたは WKSYS としてログインし、「ユーザー」タブをクリックして該当する権限を付与します。

Oracle9i データベースの場合: データベースをインストールすると、すべてのユーザー・スキーマ・アカウントがロックされます。ユーザー WKSYS としてログインするには、SYSTEM または SYS データベース・ユーザーとして次の文を実行し、WKSYS をロック解除します。

```
ALTER USER WKSYS ACCOUNT UNLOCK;
ALTER USER WKSYS IDENTIFIED BY <password>;
```

Oracle9iAS の場合: インフラストラクチャ・データベースをインストールすると、すべてのユーザー・スキーマ・パスワードがランダム化されます。ユーザー WKSYS としてログインするには、SYSTEM または SYS データベース・ユーザーとして次の文を実行し、WKSYS スキーマ・パスワードを変更します。

```
ALTER USER WKSYS IDENTIFIED BY <password>;
```

手順 5: 表の統計の収集

Crawler のパフォーマンスが低下する場合、表の統計が収集されていない可能性があります。表 WK\$URL、WK\$DOC および DR\$WK\$DOC_PATH_IDX\$I の統計を収集する必要があります。WK\$URL 表の統計が最も重要です。最適な実行計画を生成するためにコストベース・オプティマイザで統計が使用されるため、定期的に統計を収集する必要があります。パフォーマンスのチューニング中は、Crawler の実行を避けてください。

PL/SQL パッケージ DBMS_STATS または ANALYZE プロシージャを使用して、統計を収集します。DBMS_STATS パッケージは、表レベルまたはスキーマ・レベルで実行できます。ス

キーマ・レベルで実行すると、表および索引を含むスキーマ内のすべてのオブジェクトについて統計が計算されます。DBMS_STATS パッケージの使用をお勧めします。

Oracle Ultra Search インスタンスを所有するスキーマに接続します。次に例を示します。

```
EXEC DBMS_STATS.GATHER_TABLE_STATS('<schema_name>', '<table_name>', null, DBMS_STATS.AUTO_SAMPLE_SIZE);
```

```
EXEC DBMS_STATS.GATHER_SCHEMA_STATS('<schema_name>', DBMS_STATS.AUTO_SAMPLE_SIZE);
```

または

```
ANALYZE TABLE <table_name> ESTIMATE STATISTICS SAMPLE 20 percent;
```

<schema_name> は Oracle Ultra Search インスタンスの所有者、<table_name> は統計を収集する対象の表 (wk\$url など) です。

B ツリー索引を再構築すると、ディスク領域が解放されてパフォーマンスが向上することがあります。次に例を示します。

```
ALTER INDEX <index_name> REBUILD;
```

<index_name> は、再構築する対象の索引です。

索引のリストを取得するには、次の文を実行します。

```
SELECT index_name FROM user_indexes WHERE index_type='NORMAL';
```

手順 6: 索引設定の変更

このステップはオプションです。

Oracle Ultra Search インスタンスの作成時に、空の索引が作成されます。既存の索引設定 (言語固有のパラメータなど) は、\$ORACLE_HOME/ultrasearch/admin/wk0pref.sql ファイルに定義されています。

設定を変更して、すべての新しい Oracle Ultra Search インスタンスで変更後の設定を使用できます。インスタンスの作成直後に独自の設定を使用して索引を変更することもできます。索引は SQL を使用して変更します。

注意： Crawler は、索引付けを開始する前にバイナリ・ドキュメントをフィルタ処理し、すべてのドキュメントを HTML ファイルに変換します。

関連項目：

- 『Oracle Text アプリケーション開発者ガイド』
- 『Oracle Text リファレンス』

ストップリストの管理

すべての Oracle Ultra Search インスタンスには、対応するストップリストがあります。ストップリストとは、索引付け処理中に無視される単語のリストです。これらの単語をストップワードといいます。ストップワードに索引付けを行っても有効ではなく、索引付けのパフォーマンスおよび正確さが低下するため、ストップワードには索引付けを行いません。

Oracle Ultra Search のデフォルトのストップリスト

インストール時に、Oracle Ultra Search 製品のデフォルトのストップリストが作成されます。その後、Oracle Ultra Search インスタンスが作成され、デフォルトのストップリストのコピーが Oracle Ultra Search インスタンスに作成されます。

デフォルトのストップリストは、WKSYS スキーマに作成されます。デフォルトのストップリスト名は、`wk_stoplist` です（このリストは、インストール時に実行されるファイル `$ORACLE_HOME/ultrasearch/admin/wk0pref.sql` に定義されています）。

ストップリストのストップワードを追加または削除して、デフォルトのストップリストを変更します。ただし、これらの変更は既存の Oracle Ultra Search インスタンスには適用されません。変更後に作成された Oracle Ultra Search インスタンスにのみ適用されます。

インスタンスのストップリストの変更

インスタンスのストップリストの変更は、あまりお勧めしません。次のいずれかの方法を使用します。

- インスタンスを作成する前に、デフォルトのストップリストを変更する。
- インスタンスを作成した直後に、インスタンスのストップリストを置き換える。

デフォルトのストップリストに対して行われた変更は、変更後に作成された他のすべてのインスタンスのストップリストに影響します。

インスタンスの作成直後にストップリストを置き換えると、そのインスタンスのストップリストのみが変更されます。そのため、ユーザー定義のストップリストを作成する必要があります。

どちらの場合も、Oracle Ultra Search インスタンスのストップリストは、初回のクロール前に変更および定義されます。つまり、Oracle Ultra Search Crawler によって収集されたすべてのドキュメントは、正しいストップリストと照合して評価されます。すべてのドキュメントの再クロールを回避するために、初回のクロール前にストップリストを変更してください。

初回のクローल前にインスタンスのストップリストを変更する場合

1. インスタンスの作成前に、デフォルトのストップリストを変更する場合

デフォルトのストップリストにストップワード「web」を追加するには、SQL*Plus を使用してユーザー WKSYS でログインし、次の文を実行します。

```
EXEC ctx_ddl.add_stopword('wk_stoplist','web');
```

デフォルトのストップリストからストップワード「web」を削除するには、SQL*Plus を使用してユーザー WKSYS でログインし、次の文を実行します。

```
EXEC ctx_ddl.remove_stopword('wk_stoplist','web');
```

これによって、すべての新しいインスタンスのストップリストには、デフォルトのストップリストに対して行われた変更が反映されます。

2. インスタンスの作成直後に、インスタンスのストップリストを置き換える場合

ユーザー定義のストップリストを新しく作成する必要があります。SQL*Plus を使用してインスタンスの所有者としてログインし、次の文を実行します。

```
BEGIN  ctx_ddl.create_stoplist('example_stoplist');
        ctx_ddl.add_stopword('example_stoplist','example_stopword');
        ... (add more stopwords by repeated the previous
             line with new stopwords) ...
END;
/
```

インスタンスのストップリストを新しいストップリストと置き換えるには、SQL*Plus を使用してインスタンスの所有者としてログインし、次の文を実行します。

```
ALTER INDEX wk$doc_path_idx rebuild parameters('replace stoplist example_
stoplist');
```

初回のクロール後にインスタンスのストップリストを変更する場合

必要に応じて、次のいずれかの方法で、初回のクロール後にインスタンスのストップリストを変更します。

1. インスタンスのストップリストにストップワードを追加する場合

インスタンスのストップリストにストップワードを追加しても、クローलまたは索引付けされたドキュメントには影響しません。この処理には、コストはかかりません。

インスタンスのストップリストにストップワード「web」を追加するには、SQL*Plus を使用してインスタンスの所有者としてログインし、次の文を実行します。

```
ALTER INDEX wk$doc_path_idx rebuild parameters('add stopword web');
```

2. 初回のクロール後に、インスタンスのストップリストを置き換える場合

新しいストップリストを定義して、インスタンスのストップリストと置き換えると、索引全体が無効になります。**この方法を選択すると、Oracle Ultra Search Crawler で索引のすべてのドキュメントを再クロールする必要があります。**これを行うには、「スケジュール」タブの「編集」ページの「全ドキュメントの処理」をクリックします。この処理には、非常にコストがかかります。そのため、できるだけこの方法ではストップリストを変更しないでください。

最新の Oracle Ultra Search へのアップグレード

Oracle Ultra Search では、次のアップグレードをサポートしています。

- [Oracle Ultra Search リリース 1.0.3 \(Oracle9i データベース リリース 1 \(9.0.1\)\) からのアップグレード](#)
- [Oracle Ultra Search リリース 9.0.2 からリリース 9.0.3 へのアップグレード](#)
- [Oracle Ultra Search リリース 9.2 からリリース 9.0.3 へのアップグレード](#)

アップグレードの基本となるのは、Oracle Ultra Search Server コンポーネントのみです。中間層でのアップグレードはサポートされていません。リリース 9.0.3 の中間層は別の Oracle ホームにインストールします。

関連項目： Oracle Ultra Search のリリース番号については、「[Oracle Ultra Search の新機能](#)」を参照してください。

Oracle Ultra Search リリース 1.0.3 (Oracle9i データベース リリース 1 (9.0.1)) からのアップグレード

Oracle Ultra Search リリース 1.0.3 をリリース 9.0.3 にアップグレードするには、次の手順を実行します。

1. Oracle Portal Configuration Assistant (OPCA) を「ALL」モードで起動すると、Oracle Ultra Search Configuration Assistant (USCA) が起動します。USCA では、既存のデータベース内の Oracle Ultra Search のリリースをチェックします。
2. Oracle Ultra Search リリース 1.0.3 であることが USCA で検出された場合は、次の 3 つのオプションのうちいずれかを選択するように求められます。
 - 継続
 - 移行
 - 強制終了
3. 「移行」オプションを選択します。アップグレード・スクリプトを実行し、いくつかの手順を手動で実行する必要があります。

Oracle Ultra Search のアップグレード・スクリプトにより、まず現行システムのバージョンが検証され、システムのアップグレード、ユーザー・データの移行が行われます。ユーザー・データには、メタデータに関する情報、データ・ソース、マッピング、Crawler スケジュール、認証および問合せ統計など、すべてのディクショナリおよび表データが含まれます。

以前のバージョンで作成されたすべての Crawler スケジュールとジョブは、データおよびシステムを移行するまで使用できません。**移行完了後、システム管理者はクロール・スケジュールを再度アクティブにして、ドキュメントを再度索引付けする必要があります。**システムを再構成したり、データを再入力する必要はありません。ユーザーは、以前のバージョンでクロールおよび索引付けしたドキュメントを移行後も問い合わせることができます。

関連項目： 2-5 ページ「[Oracle Ultra Search Server コンポーネントの既存データベースまたはメタデータ・リポジトリへのインストール](#)」

Oracle Ultra Search の移行方法

ユーザー・データを移行する方法には、インプレース方式と ETL (Extract-Transform-Load) 方式の 2 通りがあります。インプレース方式では現行の ORACLE_HOME が使用されます。ETL 方式では新しい ORACLE_HOME が作成されます。

Oracle Ultra Search のインプレース移行 インプレース移行では、既存の構成およびユーザー・データが最新の Oracle Ultra Search リリースにアップグレードされます。アップグレードされたファイルは同じ場所にそのまま残り、ソース・インストールが変更されます。この方式の利点は、ディスク領域が節約される点です。インプレース方式では、データの移行が次の 6 つの手順で行われます。

1. ユーザー・データのバックアップ
2. 既存のデータベース・オブジェクトのアンインストール
3. 新しいデータベース・オブジェクトのインストール
4. ユーザー・インスタンスの再作成
5. データのリストア
6. 索引の再作成

SQL スクリプト `wk0upgrade.sql` を使用して、前述のインプレース移行手順の 1 から 5 を実行します。このスクリプトは `%ULTRASEARCH_HOME%/admin/` ディレクトリにあります。次の入力パラメータが必要です。

- SYS PW: ユーザー SYS のパスワード
- WKSYS PW: ユーザー WKSYS のパスワード
- HOST: データベース・ホスト・マシン
- PORT: データベース・ポート番号

- ORACLE_SID: データベース SID
- WK_TABLESPACE: Oracle Ultra Search の表領域
- WK_TEMP表SPACE: 一時表領域
- CONN_STRING: データベース接続文字列
- ORACLE_HOME: Oracle ホームのパス
- JAVA_EXE_PATH: Java 実行可能ファイルのパス
- PATH_SEPARATOR: Java CLASSPATH セパレータ。UNIX には「:」、Windows NT には「;」を使用します。

6 番目の手順では、システム管理者が Oracle Ultra Search Administration ツールを使用して、すべてのクロール・スケジュールを再度アクティブにする必要があります。

Oracle Ultra Search の ETL 移行 ETL 移行は、ソース・インストールから構成データの有効なサブセットを抽出し、必要なデータを変換し、このデータを Oracle Ultra Search の新しいインストールにロードまたはマージします。この方式ではより大きなディスク領域が必要になりますが、次の利点があります。

- ソース・インストールが不安定にならないこと
- ターゲット・インストールが安定すること
- インストーラの統合要件がないこと

ETL 方式では、データの移行が次の 5 つの手順で行われます。

1. 新しいシステム（リリース 9.0.3 など）の新しい ORACLE_HOME へのインストール
2. ユーザー・インスタンス・スキーマおよび関連データベース・オブジェクトの再作成
3. ユーザー・インスタンスの再作成
4. データのリストア
5. 索引の再作成

ETL 方式の最初の 2 つの手順は手動で実行する必要があります。

- Oracle Ultra Search リリース 9.0.3 を、同じマシンまたは異なるマシンの別の ORACLE_HOME にインストールします。新しい 9.0.3 システムを旧 9.0.1 システムと同じマシンにインストールする場合は、データベースのリスナー・ポート番号を旧 9.0.1 データベースと異なる番号に構成する必要があります。この構成によって、旧データベースと新規データベースを同時に実行できます。
- Oracle Ultra Search のすべての 1.0.3 ユーザー・インスタンス・スキーマを、新しいデータベースに再作成します。また、Oracle Ultra Search リリース 1.0.3 で作成された各表データ・ソースについて、ローカル・データベースにある実表を新しいデータベースにコピーする必要があります。表データ・ソースの実表がリモート・データベース表に設

定されている場合は、新しいデータベースからリモート・データベースへのデータベース・リンクを再作成する必要があります。

SQL スクリプト `wk0migrate.sql` を使用して、ETL 移行手順の 3 と 4 を実行します。このスクリプトは `%ULTRASEARCH_HOME%/admin/` ディレクトリにあります。次の入力パラメータが必要です。

- `WKSYS_PW`: ユーザー `WKSYS` のパスワード
- `CONN_STRING`: データベース接続文字列
- `SRC_WKSYS_PW`: ソース・データベース (9.0.1 データベース) ・ユーザー `WKSYS` のパスワード
- `SRC_CONN_STRING`: ソース・データベース接続文字列

5 番目の手順では、システム管理者が **Oracle Ultra Search Administration** ツールを使用して、すべてのクロール・スケジュールを再度アクティブにする必要があります。

注意： アップグレード・スクリプトでは、電源障害やシステム障害などの予期しないエラーが発生した場合、**Oracle Ultra Search** システムの旧リリースへのロールバックは行われません。インプレース移行の場合は、移行を開始する前にデータベースをバックアップしてください。ETL 移行の場合、以前のデータがすべて保持されるため、以前のシステム (9.0.1 など) に戻すことが可能です。

Oracle Ultra Search の移行ログ

アップグレード・スクリプトでは、移行で実行された処理がログ・ファイルに示されます。次の内容がログ・ファイルに書き込まれます。

- 現在実行中のステップ
- ストアド・プロシージャからのエラー・メッセージ
- バックアップされたデータ・レコードの数
- コピーまたは移行されたデータ・レコードの数

インプレース移行では、`wk0upgrade.sql` スクリプトによって、`%ULTRASEARCH_HOME%/admin/` ディレクトリのファイル `wk0upgrade.log` に実行ログが書き込まれます。

ETL 移行では、`wk0migrate.sql` スクリプトによって、`%ULTRASEARCH_HOME%/admin/` ディレクトリのファイル `wk0migrate.log` に実行ログが書き込まれます。

Oracle Ultra Search リリース 9.0.2 からリリース 9.0.3 へのアップグレード

Oracle Ultra Search リリース 9.0.2 をリリース 9.0.3 にアップグレードするには、次の手順を実行します。

1. Oracle Portal Configuration Assistant (OPCA) を「ALL」モードで起動すると、Oracle Ultra Search Configuration Assistant (USCA) が起動します。USCA では、既存のデータベース内の Oracle Ultra Search のリリースをチェックします。
2. Oracle Ultra Search リリース 9.0.2 であることが USCA で検出された場合は、次の 3 つのオプションのうちいずれかを選択するように求められます。
 - 継続
 - 9.0.3 にアップグレード
 - 強制終了
3. 「9.0.3 にアップグレード」オプションを選択します。USCA では、既存の Oracle Ultra Search Server コンポーネントを自動的にリリース 9.0.3 にアップグレードします。手動の手順はなく、既存の Oracle Ultra Search によって収集されたすべてのデータはそのまま保持されます。
4. USCA でアップグレードが終了すると、Oracle Ultra Search リリース 9.0.3 の Server コンポーネントが既存のデータベースにインストールされます。

Oracle Ultra Search リリース 9.2 からリリース 9.0.3 へのアップグレード

Oracle Ultra Search リリース 9.2 は、Oracle Ultra Search リリース 9.0.2 と同じデータベース・スキーマを使用するため、アップグレード手順は同じです。

関連項目： 3-14 ページ [「Oracle Ultra Search リリース 9.0.2 からリリース 9.0.3 へのアップグレード」](#)

Oracle Ultra Search Crawler と データ・ソースについて

この章では、次の項目について説明します。

- [Oracle Ultra Search Crawler の概要](#)
- [Crawler 設定](#)
- [データ・ソース](#)
- [ドキュメント属性](#)
- [スケジュールのクロール・プロセス](#)
- [データ同期](#)
- [Oracle Ultra Search のリモート Crawler](#)

関連項目： [付録 B 「問合せパフォーマンスのチューニング」](#)

Oracle Ultra Search Crawler の概要

Oracle Ultra Search Crawler は、設定されたスケジュールに従い、Oracle データベース・サーバーによってアクティブにされる Java プロセスです。Crawler がアクティブになると、様々なデータ・ソースからドキュメントをフェッチするプロセッサ・スレッドを起動します。このドキュメントは、ローカル・ファイル・システムにキャッシュされます。キャッシュがフルの場合、Crawler は Oracle Text を使用して、キャッシュされたファイルに索引を付けます。この索引は、問合せに使用されます。

注意： Oracle Ultra Search インスタンスの作成時に、空の索引が作成されます。索引が SQL を使用して変更できます。既存の設定（言語固有のパラメータなど）は、
\$ORACLE_HOME/ultrasearch/admin/wk0pref.sql ファイルに定義されています。

Crawler 設定

Crawler を使用する前に、Crawler スレッド数、Crawler タイムアウトしきい値、データベース接続文字列、デフォルト・キャラクタ・セットなどの処理用パラメータを設定する必要があります。ログ・ファイル・ディレクトリや一時ディレクトリなどの一部のパラメータにはデフォルト値がないため、クロールの前に値を設定する必要があります。Administration ツールの「Crawler」タブを使用して設定します。

関連項目： 5-11 ページ「[「Crawler」タブ](#)」

データ・ソース

Administration ツールの「[ソース](#)」タブでは、Web アクセス・パラメータ以外に、特定のデータ・ソースも定義できます。次に示す 1 つ以上のデータ・ソースを定義できます。

- Web サイト
- データベース表
- ファイル
- メーリング・リスト
- Oracle9iAS Portal のページ・グループ
- ユーザー定義データ・ソース（Crawler エージェントが必要）

Crawler エージェントの使用

ユーザー定義データ・ソースを定義して、独自のドキュメント・リポジトリまたは管理システム（Lotus Notes や Documentum など）をクロールして索引付けする場合は、Crawler エージェントを Java クラスとして実装する必要があります。エージェントは、独自のドキュメント・ソースからドキュメント URL および関連付けられたメタデータを収集し、Oracle Ultra Search Crawler に情報を戻します。Oracle Ultra Search Crawler では、後でクロールするためにその情報をエンキューします。新規データ・ソース・タイプの定義の詳細は、Administration ツールの「[ソース](#)」タブの「ユーザー定義ソース」サブタブを参照してください。

データ・ソースの同期

1 つ以上のデータ・ソースが連結された同期スケジュールを作成できます。同期スケジュールは、Oracle Ultra Search の索引を、関連付けられているデータ・ソースの最新情報で更新する頻度を定義します。同期スケジュールを定義するには、Administration ツールの「[スケジュール](#)」タブを使用します。

表示 URL とアクセス URL

一部のアプリケーションについては、セキュリティ上の理由から、クロールされる URL とエンド・ユーザーが参照する URL が異なる場合があります。たとえば、ファイアウォール内にある内部 Web サイトでのクロールはセキュリティ・チェックなしで行われますが、エンド・ユーザーによる問合せがあると、ファイアウォール外の対応するミラー URL を使用する必要があります。このミラー URL は、表示 URL と呼ばれます。

別のアクセス URL が提供されないかぎり、デフォルトでは、表示 URL がアクセス URL として処理されます。表示 URL はデータ・ソース内で一意である必要があり、2 つの異なるアクセス URL を同一の表示 URL にすることはできません。

関連項目： 5-18 ページ「[「ソース」タブ](#)」

ドキュメント属性

ドキュメント属性（メタデータ）は、ドキュメントのプロパティを示します。各データ・ソースには、ドキュメント属性の独自のセットがあります。値はクロール・プロセス中に取り出され、検索属性の 1 つにマップされた後、データベースに格納され、索引付けされます。この結果、その属性に基づいてドキュメントを問い合わせることができます。異なるデータ・ソースのドキュメント属性を同じ検索属性にマップできます。したがって、複数のデータ・ソースのドキュメントを同じ検索属性に基づいて問い合わせることができます。

ドキュメントが Web ページの場合、属性は HTTP ヘッダーから取得されるか、メタ・タグの HTML 内に格納されています。ドキュメント属性は、ドキュメント管理、アクセス制御、バージョン管理など、多くの処理で使用できます。複数のデータ・ソースに、使用目的が同じ属性を異なる名前を持たせることができます（たとえば、"version"（バージョン）と

"revision" (改訂))。また、使用目的が異なる場合でも同じ属性名を付けることができます。たとえば、"language" (言語) という属性名は、あるデータ・ソースでは使用言語として使用され、他のデータ・ソースではプログラミング言語として使用されます。

検索属性は次の 3 つの方法で作成されます。

- タイトル、作成者、説明、件名および MIME タイプなどのシステム定義の検索属性
- システム管理者が作成した検索属性
- Crawler が作成した検索属性 (Crawler エージェントは、クロール時に、ドキュメント属性を、同じ名前とデータ型を持つ検索属性にマップします。マップ対象の検索属性が見つからない場合、Crawler は、Crawler エージェントで定義されているドキュメント属性と同じ名前と型で新しい検索属性を作成します)。

検索属性の値リスト (LOV) は、検索の問合せを指定する際に有効です。属性の LOV が使用可能な場合、Crawler は、属性値、属性値の表示名およびその変換を含む LOV 定義を登録します。

スケジュールのクロール・プロセス

Crawler を最初に行う場合は、データ・ソースに基づいて Web ページ、表の行、ファイルなどをフェッチする必要があります。次に、Oracle Ultra Search の索引にドキュメントを追加します。スケジュールのクロール・プロセスは、次の 2 つの段階に分けられます。

1. [ドキュメントのキューイングおよびキャッシュ](#)
2. [ドキュメントの索引付け](#)

ドキュメントのキューイングおよびキャッシュ

[図 4-1](#) および [図 4-2](#) では、9 つの手順でクロール・サイクルのインスタンスを示します。この例では Web データ・ソースを使用していますが、Crawler では他のデータ・ソース・タイプをクロールすることもできます。

[図 4-1](#) では、Crawler および Crawler スレッドをアクティブにする方法を示します。また、Crawler キューのハイパーテキスト・リンクがナビゲーションを制御する方法も示します。この図は、手順 1 ～ 5 に対応します。

[図 4-2](#) では、Crawler が Web ページをキャッシュする方法を示します。この図は、手順 6 ～ 8 に対応します。

手順は、次のとおりです。

1. Oracle は、Administration ツールで指定したスケジュールに従って、Crawler を起動します。クロールを最初に初期化する場合、URL キューはシード URL に移入されます ([図 4-1](#))。
2. Crawler はマルチ Crawler スレッドを初期化します。

3. Crawler スレッドは、キュー内の次の URL を削除します。
4. Crawler スレッドは、Web からドキュメントをフェッチします。通常、ドキュメントは、テキストおよびハイパーテキスト・リンクを含む HTML ファイルです。
5. Crawler スレッドは、ハイパーテキスト・リンクの HTML ファイルをスキャンし、URL キューに新しいリンクを挿入します。ドキュメント表で重複しているリンクは、廃棄されます。
6. Crawler は、ローカル・ファイル・システムの HTML ファイルをキャッシュします (図 4-2)。
7. Crawler は、ドキュメント表に URL を登録します。
8. Crawler スレッドは、手順 3 から繰り返します。

ネットワークの通信量または低速な Web サイトが原因で、手順 4 に示したドキュメントのフェッチに時間がかかる場合があります。最大スループットの場合、マルチ・スレッドは、指定された時間にページをフェッチします。

注意： URL は、次のクロールが実行されるまで表示されています。URL がすでに存在しないことが Crawler で検出されると、Oracle Text がこのドキュメントを削除対象として自動的にマークしている wk\$doc 表から、その URL が削除されます (索引データが存在していても削除されます)。索引最適化の実行時にはクリーンアップが実行されます。索引最適化は個別にスケジュールできます。

図 4-1 URL のキューイング

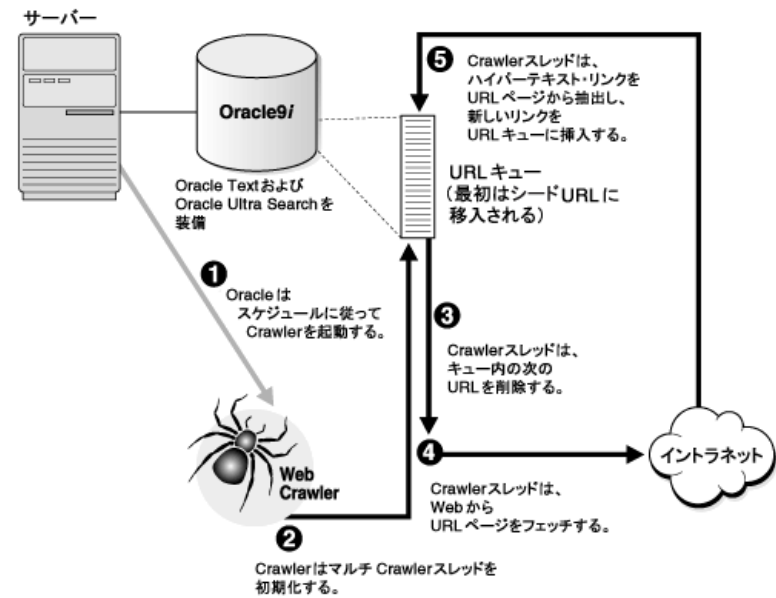
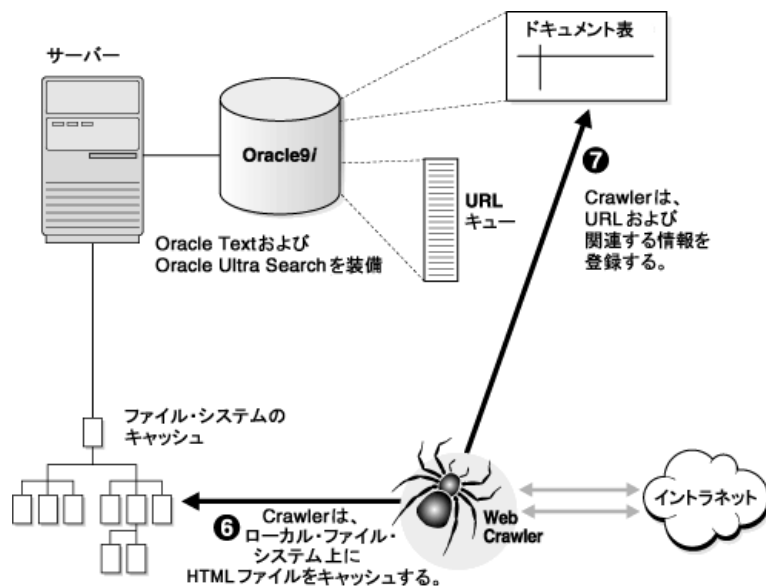


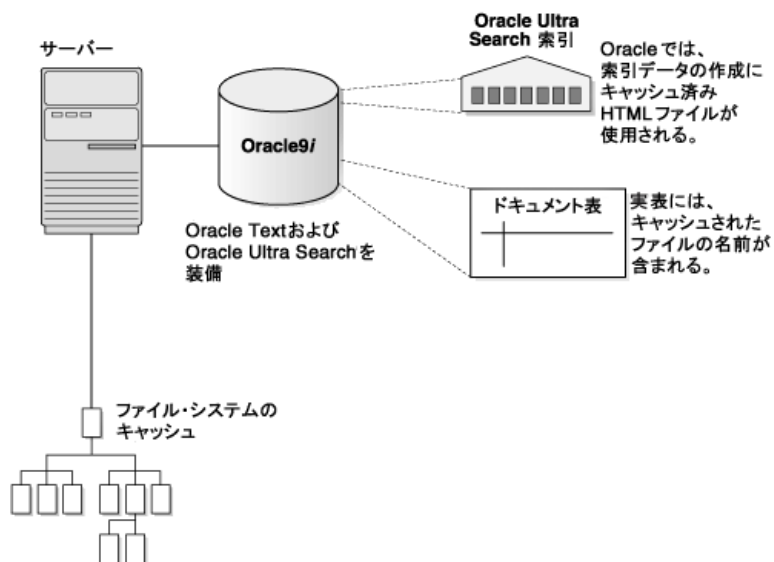
図 4-2 URL のキャッシュ



ドキュメントの索引付け

ファイル・システムのキャッシュがフル（最大サイズのデフォルトは 20MB）になると、ドキュメントのキャッシュは停止し、索引付けが始まります。次に、ドキュメント表が参照するキャッシュ済みファイルを使用して、Oracle Text の索引を補強します。図 4-3 を参照してください。

図 4-3 ドキュメントの索引付け



データ同期

初期クロールの後、URL ページが最後のクロール以降に変更された場合、URL ページはクロールおよび索引付けされます。Crawler は、HTTP の「If-Modified-Since」ヘッダー・フィールド、またはページのチェックサムを使用して、URL ページが変更されたかどうかを判断します。すでに存在しない URL はこのようにマークされ、索引から削除されます。

変更されたドキュメントを更新するために、Crawler は内部のチェックサムを使用して、新しい Web ページをキャッシュ済み Web ページと比較します。変更された Web ページはキャッシュされ、再度索引付けするためにマークされます。

データ同期の手順は、次のとおりです。

1. Oracle は、Administration ツールで指定した同期スケジュールに従って、Crawler を起動します。URL キューは、スケジュールに割り当てられたデータ・ソースの URL に移入されます。
2. Crawler はマルチ Crawler スレッドを初期化します。
3. Crawler スレッドは、キュー内の次の URL を削除します。
4. Crawler スレッドは、Web からドキュメントをフェッチします。通常、ページは、テキストおよびハイパーテキスト・リンクを含む HTML ファイルです。
5. Crawler スレッドは、新しく取得されたページのチェックサムを計算し、キャッシュ・ページのチェックサムと比較します。チェックサムが同じ場合、ページは廃棄され、Crawler は手順 3 に戻ります。それ以外の場合は、次の手順に進みます。
6. Crawler スレッドは、ハイパーテキスト・リンクのドキュメントをスキャンし、URL キューに新しいリンクを挿入します。ドキュメント表で重複しているリンクは、廃棄されます。
7. Crawler は、ローカル・ファイル・システムのドキュメントをキャッシュします。[図 4-2](#)を参照してください。
8. Crawler は、ドキュメント表に URL を登録します。
9. ファイル・システムのキャッシュがフルになる、または URL キューが空になると、Web ページのキャッシュは停止し、索引付けが始まります。それ以外の場合、Crawler スレッドは、手順 3 から繰り返します。

Oracle Ultra Search のリモート Crawler

クロールのパフォーマンスを向上させるには、データベースとは別の 1 つ以上のマシン上で Oracle Ultra Search Crawler を設定します。これらのマシンは、リモート Crawler と呼ばれます。ただし、各マシンは、データベース・マシンとキャッシュ、ログおよびメール・アーカイブ・ディレクトリを共有します。

リモート Crawler を構成するには、データベース・ホスト以外のマシンに、Oracle Ultra Search Middle-Tier コンポーネントのモジュールを最初にインストールする必要があります。インストール中、リモート Crawler は Oracle Ultra Search システムに登録され、リモート Crawler 用のプロファイルが作成されます。Oracle Ultra Search Middle-Tier コンポーネントのモジュールのインストール後、Oracle Ultra Search Administration ツールにログインし、リモート Crawler プロファイルを編集する必要があります。これによって、クロール・スケジュールへのリモート Crawler の割当てが可能になります。リモート Crawler プロファイルを編集するには、Administration ツールの「Crawler」タブを使用します。

注意： リモート Crawler を起動するとき、Oracle Ultra Search のバックエンド・データベースは、Java Remote Method Invocation (RMI) を使用してリモート・マシンと通信します。デフォルトでは、RMI はデータを暗号化しないでネットワークに送信します。リモート Crawler を使用したクロールには、潜在的なセキュリティ上のリスクがあります。企業内の悪意のある人物がネットワーク内で行き交うパケットをリスニングし、Oracle Ultra Search インスタンス・スキーマとパスワードが盗用される可能性があります。このセキュリティ上のリスクを避けるには、リモート Crawler 機能は使用しないでください。

Oracle Ultra Search Administration ツールについて

Oracle Ultra Search Administration ツールによって、Oracle Ultra Search Crawler を構成してスケジュールできます。この章では、次の項目について説明します。

- [Oracle Ultra Search Administration ツール](#)
- [Oracle Ultra Search へのログオン](#)
- [SSO ユーザーとしてのインスタンスへのログオンと管理](#)
- [「インスタンス」 タブ](#)
- [「Crawler」 タブ](#)
- [「Web アクセス」 タブ](#)
- [「属性」 タブ](#)
- [「ソース」 タブ](#)
- [「スケジュール」 タブ](#)
- [「問合せ」 タブ](#)
- [「ユーザー」 タブ](#)
- [「グローバリゼーション」 タブ](#)

関連項目： Oracle Ultra Search Administration ツールの使用方法の詳細は、Oracle Ultra Search オンライン・ヘルプを参照してください。

Oracle Ultra Search Administration ツール

Oracle Ultra Search Administration ツールは、Oracle Ultra Search Crawler を構成およびスケジュールするための Web アプリケーションです。このツールは通常、Web サーバーと同じマシンにインストールされています。このツールには、使用しているイントラネットのすべてのブラウザから、Oracle Ultra Search のデータベース・ユーザーとして直接アクセスするか、SSO Server を使用してシングル・サインオン (SSO) ・ユーザーとしてアクセスできます。

注意： Oracle Ultra Search Administration ツールおよび Oracle Ultra Search の検索アプリケーションは、Oracle Ultra Search Middle-Tier コンポーネント・モジュールの一部です。ただし、Oracle Ultra Search Administration ツールは、Oracle Ultra Search の検索アプリケーションからは独立しています。したがって、セキュリティまたは拡張性を強化するために、他のマシンで構築することができます。

Administration ツールを使用して、次のことを行います。

- Oracle Ultra Search へのログオン
- Oracle Ultra Search インスタンスの作成
- 管理ユーザーの管理
- データ・ソースの定義とデータ・グループへの割当て
- Oracle Ultra Search Crawler の構成およびスケジュール
- 検索オプションの設定
- 検索属性、値リスト (LOV) およびデータ・グループ表示名の異なる言語への変換

関連項目： Administration ツールの配置方法は、[第 2 章「Oracle Ultra Search のインストールと構成」](#)を参照してください。

Crawler パラメータの設定

Oracle Ultra Search Crawler を構成するには、次の処理を行う必要があります。

- Crawler スレッドの数などの Crawler パラメータを設定します。この作業を行うには、「[Crawler](#)」タブを使用します。
- 認証やプロキシ・サーバーなどの Web アクセス・パラメータを設定します。この作業を行うには、「[Web アクセス](#)」タブを使用します。
- Crawler データ・ソースを定義します。データ・ソースには、Web ページ、データベース表、ファイル、電子メール・メーリング・リスト、Oracle9i Application Server (9iAS) Portal またはユーザー定義のデータ・ソースがあります。Crawler スケジュー

ルには、1 つ以上のデータ・ソースを割り当てることができます。データ・ソースを定義するには、「[ソース](#)」タブを使用します。このタブでは、ソースに対するパラメータ (Web ソースに対するドメインの包含または除外など)、または表ソースに対する表示 URL のテンプレートまたは列も設定できます。

- **Crawler 同期スケジュール**を定義します。**Crawler** は、同期スケジュールを使用して、Oracle Ultra Search 索引と現行のデータ・ソースの内容を調整します。クロール・スケジュールを定義するには、「[スケジュール](#)」タブを使用します。

検索オプションの設定

検索オプションを使用して、ユーザーによる検索を制限します。検索をドキュメントの属性およびデータ・グループに制限できます。

属性

検索属性は、HTML メタ・タグ、列、ドキュメント属性および電子メールのヘッダーにマップできます。作成者や説明などの一部の属性は事前に定義されているため、構成する必要はありません。ただし、独自の属性をカスタマイズすることは可能です。問合せユーザーに公開するカスタム・ドキュメントの属性を設定するには、「[属性](#)」タブを使用します。

データ・グループ

データ・グループは、検索エンジンのユーザーに公開された論理エンティティです。問合せを入力すると、検索エンジンのユーザーは、検索対象となる 1 つ以上のデータ・グループを選択するように求められます。データ・グループは 1 つ以上のデータ・ソースで構成されます。データ・グループを定義するには、「[問合せ](#)」タブを使用します。

異なる言語でのオンライン・ヘルプ

Oracle Ultra Search では、「[ユーザー](#)」タブの言語設定に基づく状況依存のオンライン・ヘルプが提供されます。翻訳されたヘルプ・ファイルがローカル・マシンにインストールされていない場合は、英語のオンライン・ヘルプ・ファイルが使用されます。

最新のオンライン・ヘルプ・ファイルをダウンロードするには、OTN-J (Oracle Technology Network Japan) に接続してください。OTN-J を使用するには、オンラインでの登録が必要です。<http://otn.oracle.co.jp/membership/> の URL で無料で登録できます。

OTN-J のユーザー名とパスワードを取得済みの場合は、<http://otn.oracle.co.jp/document/> の OTN-J Web サイトの文書セクションに直接接続できます。

Oracle Ultra Search へのログオン

Oracle Ultra Search Administration ツールにログオンできるのは、次のユーザーです。

- Oracle Internet Directory (OID) で管理されているユーザー。これらのユーザーは、シングル・サインオン (SSO) サーバーで認証されます。
- Oracle Ultra Search が稼動しているデータベースのユーザー (non-SSO モード) [Oracle9iAS の場合]
- Enterprise Manager の IAS_ADMIN ユーザー [Oracle9iAS の場合]

Administration ツールにログオンするには、Web ブラウザで次の URL の 1 つを指定してください。

- non-SSO モード : `http://<hostname>:<port>/ultrasearch/admin/index.jsp`
- SSO モード : `http://<hostname>:<port>/ultrasearch/admin_sso/index.jsp`

インストール直後にインスタンスの作成と管理ができるのは、次のユーザーのみです。

- データベース・ユーザー : WKSYS
- Enterprise Manager のユーザー : IAS_ADMIN [Oracle9iAS の場合]
- デフォルトの SSO ユーザー : PORTAL [Oracle9iAS の場合]
- デフォルトの SSO ユーザー : DRCLADMIN [Oracle9iAS の場合]

これらの特別なユーザーの一人としてログオンした後は、他のユーザーに権限を付与し、Oracle Ultra Search インスタンスを作成および管理できるように指定できます。Oracle Ultra Search Administration ツールを使用すると、Oracle Ultra Search に関連する権限を既存のユーザーに付与したり、取り消すことができます。ユーザーの追加または削除には、シングル・サインオン・ユーザーの場合は OID、ローカル・データベース・ユーザーの場合は Oracle Enterprise Manager を使用します。

注意： Oracle Ultra Search 製品のデータベース・ディクショナリは、WKSYS スキーマにインストールされています。

関連項目：

- [第 2 章「Oracle Ultra Search のインストールと構成」](#)
- Oracle Ultra Search インスタンスの作成方法の詳細は、5-6 ページの「[「インスタンス」タブ](#)」を参照してください。
- 権限を他のユーザーに付与する方法は、5-34 ページの「[「ユーザー」タブ](#)」を参照してください。
- Oracle Ultra Search での SSO ユーザーの取扱方法は、5-5 ページの「[SSO ユーザーとしてのインスタンスへのログオンと管理](#)」を参照してください。

SSO ユーザーとしてのインスタンスへのログオンと管理

注意： Single Sign-On (SSO) は、Oracle9i Application Server (9iAS) のみ使用可能です。Oracle9i データベースでは使用できません。

Oracle9iAS Portal による Oracle Ultra Search へのログオン

シングル・サインオン (SSO) ・ユーザーが、Oracle9iAS Portal 管理ページを介して SSO で保護された Oracle Ultra Search Administration ツールにログインすると、次のいずれかの状況になります。

- SSO ユーザーが Oracle Ultra Search でスーパー・ユーザー権限を付与されている場合、Oracle Ultra Search Administration ツールにより、SSO ユーザーが管理するインスタンスを選択できるプルダウン・リストが表示されます。
- SSO ユーザーがスーパー・ユーザー権限を持っていない場合、1 つ以上の Oracle Ultra Search インスタンスを管理する権限が明示的に付与されている場合、Oracle Ultra Search Administration ツールにより、SSO ユーザーが管理するインスタンスを選択できるプルダウン・リストが表示されます。
- SSO ユーザーがスーパー・ユーザー権限を持っておらず、Oracle Ultra Search インスタンスを管理する権限も持っていない場合、Oracle Ultra Search Administration ツールにより、ユーザーが権限を持っていないことと、適切な認可レベルに連絡して権限を付与される必要があることを示すエラー・メッセージが表示されます。

「インスタンス」タブ

Oracle Ultra Search Administration ツールに正常にログオンした後は、「インスタンス」タブでユーザー自身を検索できます。このページでローカル・データベース内のすべての Oracle Ultra Search インスタンスを管理します。ページの左上部には、インスタンスを作成、選択、削除および編集するためのタブがあります。

クロールおよび索引付けの構成に Administration ツールを使用する前に、Oracle Ultra Search インスタンスを作成する必要があります。Oracle Ultra Search インスタンスは名前で識別され、独自のクロール・スケジュールおよび索引を持ちます。Oracle Ultra Search インスタンスを作成できるのは、WKADMIN ロールを付与されたユーザーのみです。

インスタンスの作成

インスタンスを作成するには、「インスタンス」タブで「作成」サブタブを選択します。標準インスタンス（マスター・インスタンス）および読取り専用のスナップショット・インスタンスを作成するためのリンクを持つ別のページが表示されます。新しいインスタンスを作成できるのは、Oracle Ultra Search スーパー・ユーザーのみです。

注意： Oracle Ultra Search インスタンスの検索ドメインがオーバーラップしている場合は、ロギング可能な表データ・ソース、電子メール・データ・ソースおよび一部のユーザー定義データ・ソースでクロールの競合が発生する可能性があります。

標準インスタンスの作成

インスタンスを作成する手順は、次のとおりです。

1. データベース・ユーザーを準備します。

すべての Oracle Ultra Search インスタンスは、1 つのデータベース・ユーザー / スキーマにのみ存在します。新しい Oracle Ultra Search インスタンスを作成するには、Oracle Ultra Search 用に構成され、かつ Oracle Ultra Search インスタンスが含まれていないデータベース・ユーザーが必要です。

Oracle Ultra Search インスタンスの格納用に作成するデータベース・ユーザーには、専用の自己完結型表領域を割り当てる必要があります。このインスタンスのスナップショット・インスタンスを作成する計画がある場合、これは重要です。これを行うには、新しい表領域を作成します。次に、デフォルト表領域が今作成した表領域である新規データベース・ユーザーを作成します。

関連項目：

- Oracle Ultra Search のデータベース・ユーザーの構成に関する情報と手順は、3-2 ページの「[Oracle Ultra Search 用の Oracle データベース・サーバーの構成](#)」を参照してください。
- 5-8 ページ「[スナップショット・インスタンスの作成](#)」
- トランスポータブル表領域の使用の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

2. Oracle Ultra Search Administration ツールでのインスタンスの作成方法に従います。

インスタンスを作成するメイン・ページから「インスタンスの作成」リンクを選択して、次の情報を指定します。

- インスタンス名
- データベース・スキーマ：これはステップ 1 のユーザー名です。
- スキーマ・パスワード

次のオプション索引設定も入力できます。

- レクサー

索引付けに使用するレクサー名を指定します。デフォルト・レクサーは `wksys.wk_lexer` で、`wk0pref.sql` ファイルに定義されています。インスタンスの作成後にレクサーを変更することはできません。

- ストップリスト

索引付けで使用するストップリスト名を指定します。デフォルト・ストップリストは `wksys.wk_stoplist` で、`wk0pref.sql` ファイルに定義されています。インスタンスを作成した後、ストップリストは変更しないでください。

- 記憶域

インスタンスの索引に対する記憶域設定の名前を指定します。デフォルト記憶域設定は `wksys.wk_storage` で、`wk0pref.sql` ファイルに定義されています。インスタンスの作成後は、記憶域設定を変更できません。

関連項目：

- レクサー、ストップリストおよび記憶域を作成および変更方法は、『Oracle Text リファレンス』を参照してください。
- 3-8 ページ「[ストップリストの管理](#)」

スナップショット・インスタンスの作成

スナップショット・インスタンスは、インスタンスのコピーです。標準インスタンスとは異なり、スナップショット・インスタンスは読取り専用で、その索引は検索ドメインと同期化されません。また、マスター・インスタンスが検索ドメインと再同期されると、スナップショット・インスタンスは古くなります。その時点でスナップショットを削除し、新しいスナップショットを作成する必要があります。

注意： スナップショットとそのマスター・インスタンスを同じデータベース内に存在させることはできません。

スナップショット・インスタンスは、次の場合に役立ちます。

- 問合せ処理

2つの Oracle Ultra Search インスタンスで、同じ検索ドメインに関する問合せに回答できます。したがって、一定期間内では、1つのインスタンスがドメインに関する問合せに回答するより、2つのインスタンスの方がより多く回答できます。スナップショット・インスタンスはクロールと索引付けに関係しないため、スナップショット・インスタンスの作成は高速でコストがかかりません。したがって、スナップショット・インスタンスによって拡張性を改善することができます。

- バックアップ

マスター・インスタンスが破損した場合は、インスタンス・モードを更新可能に編集することによって、そのスナップショットを標準インスタンスに変換できます。スナップショットとそのマスター・インスタンスは同じデータベース上に存在させることができないため、スナップショット・インスタンスを更新可能に変更できるのは、破損したマスター・インスタンスを置換する場合のみです。

スナップショット・インスタンスは、マスター・インスタンスからの認証を継承していません。したがって、スナップショット・インスタンスを更新可能にする場合は、検索ドメインのクロールに必要な認証情報を再度入力する必要があります。

スナップショット・インスタンスを作成する手順は、次のとおりです。

1. データベース・ユーザーを準備します。

標準インスタンスと同様に、スナップショット・インスタンスには、Oracle Ultra Search 用に構成され、かつ Oracle Ultra Search インスタンスが含まれていないデータベース・ユーザーが必要です。

2. マスター・インスタンスからデータをコピーします。

このコピーは、表領域名の変更ができないトランスポートブル表領域メカニズムを使用して行います。したがって、スナップショット・インスタンスは、そのマスター・インスタンスと同じデータベース上には作成できません。

マスター・インスタンス・データすべてが含まれている表領域または一連の表領域を識別します。次に、その表領域をコピーし、ステップ 1 のデータベース・ユーザーに接続します。

3. Oracle Ultra Search Administration ツールでのスナップショット・インスタンスの作成方法に従います。

インスタンスを作成するメイン・ページから「読み込み専用スナップショット・インスタンスの作成」リンクを選択して、次の情報を指定します。

- スナップショット・インスタンス名
- スナップショット・スキーマ名：これはステップ 1 のデータベース・ユーザー名です。
- スナップショット・スキーマ・パスワード
- データベース・リンク：これはマスター・インスタンスが稼動しているデータベースへのデータベース・リンク名です。
- マスター・インスタンス名

この情報を指定した後、「適用」をクリックします。

関連項目：

- Oracle Ultra Search のデータベース・ユーザーの構成に関する情報と手順は、3-2 ページの「[Oracle Ultra Search 用の Oracle データベース・サーバーの構成](#)」を参照してください。
- トランスポートابل表領域の使用の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

インスタンスの選択

複数の Oracle Ultra Search インスタンスを保持できます。たとえば、マーケティング、人事管理および開発の各ポータル用に個別の Oracle Ultra Search インスタンスを保持できます。Administration ツールでは、インスタンス固有の変更を行う前に、インスタンスを指定する必要があります。

インスタンスを選択する手順は、次のとおりです。

1. 「インスタンス」タブで「選択」サブタブをクリックします。
2. プルダウン・メニューからインスタンスを選択します。
3. 「適用」ボタンをクリックします。

注意： インスタンスはデータを共有しません。データ・ソース、スケジュールおよび索引は、各インスタンスに対して固有です。

インスタンスの削除

インスタンスを削除する手順は、次のとおりです。

1. 「インスタンス」タブで「削除」サブタブをクリックします。
2. プルダウン・メニューからインスタンスを選択します。
3. 「適用」ボタンをクリックします。

注意： Oracle Ultra Search インスタンスを削除するには、WKADMIN ロールが必要です。

インスタンスの編集

インスタンスを編集するには、「インスタンス」タブで「編集」サブタブを選択します。インスタンス・モード（インスタンスを更新可能に）やインスタンス・パスワードを変更できます。

インスタンス・モード

インスタンス・モードは、更新可能または読取り専用に変更できます。更新可能なインスタンスは、それ自体で設定スケジュールの検索ドメインと同期を行いますが、読取り専用のインスタンス（スナップショット・インスタンス）は同期を行いません。インスタンス・モードを設定するには、対応するボックスを選択して「適用」をクリックします。

スキーマ・パスワード

Oracle Ultra Search インスタンスは、常駐しているデータベース・ユーザーのパスワードを認識する必要があります。インスタンスは、この情報をデータベースから直接取得できません。インスタンス作成時に、Oracle がデータベース・ユーザーのパスワードを指定し、この情報をインスタンスがキャッシュします。

このデータベース・ユーザー・パスワードを変更した場合は、インスタンスがキャッシュしているパスワードを更新する必要があります。これを行うには、新規パスワードを入力して「適用」をクリックします。データベースと照合して検証された新規パスワードは、キャッシュされたパスワードを置換します。

「Crawler」タブ

Oracle Ultra Search Crawler は、スレッドを起動して、Web サイトやデータベース表、電子メール・アーカイブなど、定義済みのデータ・ソースをクロールする Java アプリケーションです。クロールは、「[スケジュール](#)」タブに定義されているとおりに、スケジューリングされた間隔で定期的に行われます。

このタブでは、次の操作を実行できます。

設定

Crawler スレッド

実行時に起動される Crawler スレッドの数を指定します。

プロセッサ数

Oracle Ultra Search Crawler を実行するサーバーに存在する CPU の数を指定します。この設定によって、システムで使用するドキュメント変換スレッドの最適数が決まります。ドキュメント変換スレッドは、複数形式のドキュメントを HTML ドキュメントに変換し、適切な索引付けを行います。

自動言語検出

Oracle Ultra Search Crawler で検索したすべてのドキュメントに、言語が指定されているわけではありません。言語の指定がないドキュメントについては、Oracle Ultra Search Crawler が自動的に言語を検出します。この機能を有効にする場合は、「はい」をクリックします。

言語認識機能は、様々な言語（デンマーク語、オランダ語、英語、フランス語、ドイツ語、イタリア語、ポルトガル語およびスペイン語）によるドキュメントのトライグラム・データを使用して統計的に言語を判断します。指定のドキュメントはいずれの言語にも属さないという仮説から開始し、最終的には、特定言語に対するこの仮説の誤りを可能な範囲で証明します。この機能は、Latin-1 アルファベットおよび文字の Unicode 範囲が確定している言語（中国語、日本語、韓国語など）で作動します。

Crawler は、言語コードが表データ・ソースである場合は、HTTP ヘッダー・コンテンツ言語または LANGUAGE 列をチェックすることによって、言語コードを判断します。言語を判断できない場合は、次の措置が講じられます。

1. 言語認識機能が使用可能でない場合や言語コードを判断できない場合は、デフォルト言語コードが使用されます。
2. 言語認識機能が使用可能な場合は、その出力が使用されます。

この言語コードは、wk\$url 表と wk\$doc 表の LANG 列に移入されます。Oracle Ultra Search で使用されるのは、マルチレクサーのみです。すべてのドキュメント URL は、wk\$doc（索引付け関連）および wk\$url（クロール関連）に格納されています。

デフォルト言語

自動言語検出が使用できない場合や Web ドキュメントに言語の指定がない場合、Crawler は Web ページがデフォルトの言語で書かれているものと認識します。ドキュメントの索引付け方法は言語によって決定されるため、この設定は重要です。

注意： このデフォルト言語は、Crawler がドキュメント言語をクロール時に判断できなかった場合にのみ使用されます。言語設定は「[ユーザー](#)」[タブ](#)で行います。

Crawler またはデータ・ソースに対してデフォルト言語を選択できます。索引付けと問合せ用のデフォルト言語として、次の言語がサポートされています。

- 英語
- ポルトガル語（ブラジル）
- デンマーク語
- オランダ語
- フランス語
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- ポルトガル語
- 簡体字中国語
- スペイン語
- スウェーデン語
- 繁体字中国語

クロールの深さ

Web ドキュメントには、他の Web ドキュメントへのリンクが含まれている場合があります、他の Web ドキュメントには、さらに多くのリンクが含まれている場合があります。この設定によって、Crawler が追跡するネストされたリンクの最大数を指定できます。

関連項目： クロールの深さの重要性については、付録 A 「Web クロー
ル・プロセスのチューニング」を参照してください。

Crawler タイムアウトしきい値

Crawler のタイムアウトを秒単位で指定します。Crawler タイムアウトしきい値は、Crawler が Web ページにアクセスできないときに、強制的にタイムアウトするために使用されます。

デフォルト・キャラクタ・セット

デフォルト・キャラクタ・セットを指定します。HTML ドキュメントにキャラクタ・セットが指定されていない場合、Crawler はこの設定を使用します。

一時ディレクトリの位置とサイズ

一時ディレクトリおよびサイズを指定します。Crawler は、索引付けを行うときの中間記憶域として一時ディレクトリを使用します。一時ディレクトリの絶対パスを指定してください。サイズは、Crawler が使用する一時領域の最大値（MB 単位）です。

一時ディレクトリのサイズは索引の断片化に影響するため重要です。サイズが小さいと、索引はより細かく断片化されます。その結果、問合せが遅くなり、索引の最適化が頻繁に必要になります。ディレクトリのサイズを大きくすると、索引の断片化は少なくなりますが、クロールのスループット（1 時間ごとにクロールされるドキュメントの合計数）も少なくなります。これは、大きい一時ディレクトリの索引付けには時間がかかり、新しいドキュメントをディレクトリに書き込む前に、Crawler で索引付けが完了するのを待機する必要があるためです。

Crawler のロギング

次の項目を指定します。

- 詳細のレベル:すべて、またはサマリーのみ
- Crawler ログ・ファイル・ディレクトリ
- Crawler ログ・ファイル言語

ログ・ファイル・ディレクトリには、Crawler ログ・ファイルが格納されます。ログ・ファイルには、特定のスケジュールに対応した Crawler のアクティビティ、警告およびエラー・メッセージがすべて記録されます。ログ・ファイルには、起動時、実行時およびシャットダウン時に記録されたメッセージが含まれます。多数のドキュメントをクロールする場合は、すべてを記録するとログ・ファイルのサイズが非常に大きくなります。ただし、詳細なアクティビティを各スケジュールのログ・ファイルに出力するように Crawler を構成することが有効な場合があります。Crawler ログ・ファイル言語とは、Crawler がログ・ファイルの生成に使用する言語です。

データベース接続文字列

データベース接続文字列は、データベースに接続する場合に、リモート Crawler で使用される標準 JDBC 接続文字列です。接続文字列は、[hostname]:[port]:[sid] の形式または TNS キーワード ー 値の構文の形式で指定できます。次に例を示します。

```
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=...)(PORT=5521)...))"
```

関連項目：『Oracle9i JDBC 開発者ガイドおよびリファレンス』

Real Application Clusters 環境では、TNS キーワード ー 値の構文を使用する必要があります。この構文により、システムのすべてのノードに接続できます。次に例を示します。

```
"(DESCRIPTION=(LOAD_BALANCE=yes)(ADDRESS=(PROTOCOL=TCP)(HOST=c1s02a)(PORT=3001))
(ADDRESS=(PROTOCOL=TCP)(HOST=c1s02b)(PORT=3001)))(CONNECT_DATA=(SERVICE_
NAME=sales.us.acme.com))"
```

リモート Crawler プロファイル

このページを使用して、リモート Crawler プロファイルを表示および編集します。リモート Crawler プロファイルには、Oracle Ultra Search データベース以外のリモート・マシンで Oracle Ultra Search Crawler を実行するために必要なすべてのパラメータが含まれます。リモート Crawler プロファイルは、ホスト名で識別されます。プロファイルには、リモート Crawler がデータベース・マシンと共有するキャッシュ、ログおよびメール・ディレクトリが含まれます。

これらのパラメータを設定するには、「編集」をクリックします。リモート Crawler から使用できる共有ディレクトリ・パスを入力します。これらのディレクトリが、適切に共有またはマウントされていることを確認する必要があります。

Crawler 統計

このページを使用して、次の Crawler 統計を参照します。

Crawler アクティビティのサマリー

Crawler アクティビティに関する一般的なサマリーが提供されます。

- Crawler 統計の集計
- 索引付けされたドキュメントの総数
- データ・ソース・タイプによる Crawler 統計

詳細な Crawler 統計

次の統計が含まれます。

- クロールと索引付けが行われたホストのリスト
- 深さに基づくドキュメントの配布
- ドキュメント・タイプに基づくドキュメントの配布
- データ・ソース・タイプに基づくドキュメントの配布

Crawler の進行状況

過去の週について Crawler の進行状況が表示されます。現在の時刻までの 1 週間に索引付けされたドキュメント総数が表示されます。時間列は現在の時刻に最も近い時間に丸められます。

問題のある URL

クロール・プロセス中に発生したエラーがリストされます。エラーが発生した URL の数も表示されます。

「Web アクセス」タブ

このタブを使用して、基本認証とプロキシを設定します。

URL 認証

Oracle Ultra Search では、基本認証情報を、それを必要とするホストに対して提供します。基本認証は、クライアントが各領域でユーザー名とパスワードによって自らを認証する必要があるというモデルに基づいています。領域とは、Web サーバー上の保護された URL のセットを識別する文字列です。ホスト、領域、ユーザー名およびパスワードを入力し、「追加」をクリックします。

プロキシ

組織のファイアウォール外にある Web ページが検索領域に含まれている場合は、プロキシ・サーバーを指定します。プロキシ・サーバーの指定は任意です。現在は、HTTP プロトコルのみがサポートされています。

注意： Crawler では、プロキシ認証が必要なプロキシ・サーバーは使用できません。

ドメイン例外を設定することもできます。

「属性」タブ

索引付けされたドキュメントに、作成者や日付情報などのメタデータが含まれる場合、問合せユーザーはこれらの情報に基づいて検索を絞り込むことができます。たとえば、作成者属性に特定の値を持つすべてのドキュメントを検索できます。

ドキュメント属性の値リスト (LOV) は、検索の問合せを指定する際に役立ちます。属性値には表示名を指定できます。たとえば、国属性には属性値として国コードを使用しますが、ユーザーには国名が表示されます。属性の表示名には複数の翻訳がある場合があります。

検索属性を定義するには、「検索属性」サブタブを使用します。Oracle Ultra Search では、「作成者」や「説明」など、システム定義の属性がいくつか提供されます。独自の属性を定義することもできます。

検索属性を定義した後は、データ・ソースに対するドキュメント属性とグローバル検索属性との間でマッピングを行う必要があります。この作業を行うには、「マッピング」サブタブを使用します。

注意： Oracle Ultra Search では、検索属性 LOV や表示名などのメタデータを Oracle Ultra Search データベースにロードするためのコマンドライン・ツールが提供されます。データが大量にある場合は、このツールによって HTML ベースの管理ツールより早くロードできます。詳細は、[付録 E「メタデータの Oracle Ultra Search へのロード」](#)を参照してください。

検索属性

検索属性は、問合せユーザーに公開されています。Oracle Ultra Search では、「作成者」や「説明」など、システム定義の属性が提供されます。Oracle Ultra Search は検索属性のグローバルなリストを維持します。検索属性は、追加、編集または削除できます。「LOV の管理」をクリックして、検索属性の値リストを変更することもできます。属性 LOV には、すべてのデータ・ソースを対象とするグローバル属性 LOV と、データ・ソース固有の属性 LOV の 2 つのカテゴリがあります。

独自の属性を定義するには、テキスト・ボックスに属性名を入力して、文字列、日付または数値を選択し、「追加」をクリックします。

属性の LOV エントリと表示名は、追加または削除できます。表示名はオプションです。表示名が指定されていない場合は、問合せ画面で LOV エントリが使用されます。

注意： LOV は文字列タイプとしてのみ表示されます。LOV が日付書式の場合は、「DD-MM-YYYY」書式を使用して入力する必要があります。

ポリシーのアップデート

ポリシー値を更新するには、該当する属性の「LOV の管理」アイコンをクリックします。

データ・ソース指定の LOV は次の 3 つのうちいずれかの方法で更新できます。

1. 手動による更新
2. Crawler エージェントによる、クロール・プロセス中の LOV 自動更新
3. 入力ドキュメントの属性値を調べ、新規 LOV エントリを自動的に追加

注意： ポリシーのアップデートが Crawler エージェントによって制御されている場合、LOV およびすべての変換された値は次のクロールで削除されます。

マッピング

ユーザー定義ソースのマッピング情報について説明します。マッピングはエージェント・レベルで実行され、ドキュメント属性はまず同名の検索属性に自動的にマップされます。ドキュメント属性と検索属性は 1 対 1 でマップされます。各ユーザー定義データ・ソースについて、ドキュメント属性のマップ先のグローバル検索属性を編集できます。

Web または表データ・ソースのマッピングは、データ・ソースの作成時に手動で作成します。ユーザー定義データ・ソースのマッピングは、後続のクロールで自動的に作成されます。

このマッピングを変更するには、「マッピングの編集」をクリックしてください。

既存マッピングの編集は、このデータ・ソースのすべてのドキュメントを再クロールする必要があります。そのため、コストがかかります。既存マッピングの編集は、必要な時以外は避けてください。

注意： 一度定義した検索属性のマッピングは削除できません。

注意： 電子メール・ソースには、ユーザーが管理するマッピングはありません。電子メールには、事前に定義されたマッピングが 2 つあります。電子メールの「From (差出人)」は、Oracle Ultra Search の「Author」属性にマップされています。同様に、電子メールの「Subject (件名)」は、Oracle Ultra Search の「Subject」属性にマップされています。電子メール・メッセージの摘要は、「説明」属性にマップされています。

「ソース」タブ

ドキュメントのコレクションをソースと呼びます。データ・ソースは、Web サイトや電子メール受信ボックスなど、位置に関するプロパティを持つ特性があります。Oracle Ultra Search Crawler は、1 つ以上のデータ・ソースからデータを取得します。

次のタイプのソースがあります。

- [Web ソース](#)
- [表ソース](#)
- [電子メール・ソース](#)
- [ファイル・ソース](#)
- [ユーザー定義ソース](#) (Crawler エージェントが必要)
- [Oracle9iAS Portal ソース](#)

関連項目：

- 1 つ以上のデータ・ソースを同期スケジュールに割り当てるには、5-26 ページの「[スケジュール](#)」タブを参照してください。
- データ・ソースをデータ・グループに割り当て、制限的な問合せを可能にするには、5-30 ページの「[問合せ](#)」タブを参照してください。

データ・ソースは、必要な数だけ作成できます。ここでは、データ・ソースの作成方法と編集方法について説明します。

Web ソース

Web ソースは、特定の Web サイトの HTML コンテンツを表示します。Web ソースは他のタイプのデータ・ソースとは異なり、特定の Web サイトのクロールを容易にメンテナンスするために使用します。

Web ソースの作成

新しい Web ソースを作成する手順は、次のとおりです。

1. Web ソースの名前を指定します。
2. 各 Web ソースに関するデフォルトの Crawler 設定を上書きします。このステップはオプションです。クロールの深さ、キャラクタ・セット、言語、Crawler スレッド数および Crawler のタイムアウトしきい値を変更できます。また、ロボットの除外、言語検出および URL リライタを使用可能または使用禁止にできます。「Web ソースの編集」を使用して、これらを編集することもできます。デフォルト設定を変更する場合は、「更新」をクリックします。

ロボットの除外によって、ロボットで参照できるサイトを制御できます。ロボットの除外が有効（デフォルト）な場合、Web Crawler は、Web サーバーの robots.txt ファイルに指定されているアクセス・ポリシーに基づいてページ間を横断します。たとえば、ロボットは、`http://www.foobar.com/` を参照する場合 `http://www.foobar.com/robots.txt` をチェックします。このファイルが見つかった場合、Crawler はその内容を分析して、ドキュメントを取り出せるかどうかを確認します。Web サイトを所有している場合は、ロボットの除外を無効にすることができます。ただし、他の Web サイトをクロールするときは、ロボットの除外を有効にして robots.txt に従う必要があります。

URL リライタは、Ultra Search UrlRewriter インタフェースを実装するためにユーザーが提供する Java モジュールです。Crawler は URL リライタを使用して、URL キューに挿入される前に抽出した URL リンクをフィルタ処理またはリライトします。URL のフィルタ処理によって不要なリンクが削除され、URL のリライトによって URL リンクが変換されます。この変換は、アクセス URL を使用する場合に必要です。

URL リライタは、リンクについて考えられる次の結果を提供します。

- リンクの変更はありません。Crawler はリンクをそのまま挿入します。
- リンクを廃棄します。挿入はありません。
- 新規表示 URL が戻されました。挿入のために URL リンクが置換されます。
- 表示 URL およびアクセス URL が戻されました。表示 URL は、URL リンクと同一でない可能性があります。

生成された新しい URL リンクは、すべての既存のホスト、パス、MIME タイプすべての包含ルールと除外ルールの対象となります。

実装したリライタ・クラスを jar ファイルに格納し、クラス名と jar ファイル名をここで指定する必要があります。

関連項目：

- 6-13 ページ [「Oracle Ultra Search の URL リライタ API」](#)
- 4-3 ページ [「表示 URL とアクセス URL」](#)
- デフォルト言語の詳細は、5-11 ページの [「Crawler」タブ](#) を参照してください。

3. 開始アドレスを入力します。これは、Crawler がクロールを開始する URL です。
4. URL 境界ルールを設定して、クロール領域を絞り込みます。ホストおよび URL パスは、包含または除外できます。たとえば、包含ドメインが `oracle.com` の場合、Oracle Ultra Search Crawler は全世界のオラクル社に属するホストのみをクロールします（接尾辞も包含されるため、`oracle.com` で終わるものはすべてクロールされます。ただし、`http://www.oracle.com.tw` はクロールされません）。除外ドメインが `uk.oracle.com` の場合、Crawler はイギリスにある Oracle ホストをクロールしません。

ポート番号は包含または除外できません。つまり、`www.acme.com:7777` はクロールできませんが、`www.acme.com:8888` はクロールできます。これには、URL リライタを使用します。除外ルールは常に包含ルールをオーバーライドします。

5. Oracle Ultra Search Crawler がこのソースについて処理するドキュメントのタイプを指定します。HTML とプレーン・テキストは、Crawler が常に処理するデフォルトのドキュメント・タイプです。
6. Web ソースのメタ・タグのマッピングを定義、編集または削除します。メタ・タグは、HTML ドキュメントのヘッダーに表示される説明タグです。1 つのメタ・タグは、1 つの検索属性にのみマップできます。

表ソース

表ソースは、データベース表またはビューのコンテンツを表します。データベース表またはビューは、Oracle Ultra Search データベースのインスタンスまたはリモート・データベースに存在するものが使用できます。Oracle Ultra Search はデータベース・リンクを使用して、リモート・データベースにアクセスします。

関連項目： 5-21 ページ [「データベース・リンクの制限事項」](#)

表ソースの作成

表ソースを作成するには、「新規表ソースの作成」をクリックし、次の手順を実行します。

1. 表ソース名を指定し、データベース・リンク、スキーマおよび表の名前を指定します。「表の割当て」をクリックします。
2. デフォルト言語や主キー列など、表ソースに関する設定を指定します。最終的なコンテンツが渡される列、およびその列に格納されるデータのタイプ（HTML、プレーン・テキストまたはバイナリ）も指定できます。デフォルト言語の詳細は、5-11 ページの「[Crawler タブ](#)」を参照してください。
3. 表ソースに関する情報を検証します。
4. 表データ・ソースのクロールを最適化するために、Oracle Ultra Search のロギング機能を使用するかどうかを決定します。このロギング機能を使用すると、新たに更新されたドキュメントのみがクロール・プロセスで再検査されます。ロギング機能は、Oracle 表および Oracle 以外の表に対して使用可能または使用禁止にできます。ロギング機能を使用可能にする場合は、ログ表とログ・トリガーを作成するように求められます。Oracle 表には、Oracle SQL 文が提供されています。Oracle 以外の表を使用する場合は、ログ表とログ・トリガーを手動で作成する必要があります。作成例に従って、ログ表とログ・トリガーを作成します。表を作成した後、「ログ表名」に表名を入力します。
5. 列をドキュメント属性にマップします。各列は、1 つのドキュメント属性のみにマップできます。これによって、検索エンジンは表ソースからデータを継続して検索できます。

6. 表ソースに対して表示 URL のテンプレートまたは列を指定します。このステップはオプションです。Oracle Ultra Search は、表データ・ソース用にデフォルトのテキスト・ビューアを使用します。表示 URL を指定すると、Oracle Ultra Search は定義済みの Web URL を使用して、取得した表データを表示します。表示 URL 列が使用可能な場合、Oracle Ultra Search はその列を使用して URL を取得し、表データ・ソースのコンテンツを表示します。表示 URL のテンプレートは、次のように指定してください。
`http://[hostname]:[port]/[path]?[parameter_name]=$ (key1)` ここで key1 は該当する表の主キー列です。

「列のキーへのマッピング」セクションでは、マッピング情報を提供します。Oracle Ultra Search では、STRING 型、NUMBER 型または DATE 型の表キーをサポートしています。key1 が NUMBER 型または DATE 型の場合は、Web サイトで使用する書式モデルを指定する必要があります。これによって、Oracle は文字列の解析方法を識別できます。たとえば、文字列「11-Nov-1999」の日付書式モデルは「DD-Mon-YYYY」です。他の列を Oracle Ultra Search 属性にマップすることもできます。テキスト列はマップしないでください。

関連項目： 書式モデルの詳細は、『Oracle9i SQL リファレンス』を参照してください。

表ソースの編集

「編集」をクリックして、表ソース名の変更、列と検索属性とのマッピングの変更、追加または削除、表示 URL のテンプレートまたは列の変更、および表ソースの設定値の表示を行います。

複数の表で構成された表ソース

表ソースに複数の表がある場合は、関連する表を結合するビューを作成する必要があります。Oracle Ultra Search は、このビューを表ソースとして使用します。たとえば、マスター/ディテール関係にある 2 つの表は、マスター表の SELECT 文およびディテール表の行を連結するユーザー実装の PL/SQL ファンクションを使用して連結できます。

データベース・リンクの制限事項

次の制限事項は、Crawler がデータベース・リンクを介してアクセスするリモート・データベース上の実表やビューに適用されます。

- 実表またはビューのテキスト列が BLOB 型または CLOB 型の場合は、表に ROWID 列が必要です。次の場合は、表またはビューに ROWID 列がない場合があります。
 - 1 つ以上の表の結合で構成されたビュー
 - GROUP BY 句を使用している単一表に基づいたビュー

リモート表やビューが Oracle Ultra Search で問題なくクロールされるかどうかを確認する最善の方法は、ROWID 列の存在をチェックすることです。それには、SQL*Plus を使用している表やビューに対して、次の SQL 文を実行します。

```
SELECT MIN(ROWID) FROM <table or view name>;
```

- 実表やビューに BFILE 型のテキスト列を含めることはできません。

電子メール・ソース

電子メール・ソースは、特定の電子メール・アドレスに送信された電子メールから内容を導出します。Oracle Ultra Search Crawler が電子メール・ソースを検索する場合、Crawler は電子メールのヘッダー・フィールドの「宛先:」または「Cc:」に特定の電子メール・アドレスがあるすべての電子メールを収集します。

最も一般的な電子メール・ソースのアプリケーションでは、電子メール・ソースは、メーリング・リストに送信されたすべての電子メールを示します。この場合は、複数の電子メール・ソースが定義され、各電子メール・ソースは1つの電子メール・リストに対応します。

電子メール・ソースをクロールするには、IMAP アカウントが必要です。現在の Oracle Ultra Search Crawler でクロールできるのは、1つの IMAP アカウントのみです。このため、クロール対象のすべての電子メールは IMAP アカウントの受信ボックスに存在する必要があります。たとえば、メーリング・リストの場合、IMAP アカウントは、クロールするすべてのメーリング・リストにサブスクライブされている必要があります。メーリング・リストに新しく投稿されたすべてのメールは、IMAP の電子メール・アカウントに送信されてからクロールされます。Oracle Ultra Search Crawler は、IMAP4 に準拠しています。

電子メール・メッセージを取得した Oracle Ultra Search Crawler は、IMAP サーバーからそのメッセージを削除します。次に、Crawler は電子メール・メッセージの内容を HTML に変換し、その HTML を索引付け用のキャッシュ・ディレクトリに一時的に格納します。さらに、取得したすべてのメッセージをアーカイブ・ディレクトリというディレクトリに格納します。このディレクトリに格納された電子メール・ファイルは、問合せ結果として、検索したエンド・ユーザーに表示されます。

電子メール・ソースをクロールするには、IMAP サーバーの電子メール・アカウントのユーザー名とパスワードを指定する必要があります。また、IMAP サーバーのホスト名およびアーカイブ・ディレクトリも指定する必要があります。

電子メール・ソースの作成

電子メール・ソースを作成するには、電子メール・アドレスおよび説明を入力する必要があります。この説明は検索を実行したすべてのエンド・ユーザーによって参照されるため、簡潔でわかりやすい名前を指定する必要があります。電子メール・ソースを作成（登録）する場合、使用する名前はメーリング・リストの電子メールです。電子メールが登録済みのメーリング・リストの1つに送信されない場合、その電子メールはクロールされません。

電子メール・ソースの電子メール・アドレス別名を指定できます。電子メール・ソースに別名を指定すると、メインの電子メール・アドレスに送信されたすべての電子メールはアドレスの別名にも送信され、Crawler によって収集されます。

ファイル・ソース

ファイル・ソースは、Oracle Ultra Search データベース・マシンまたはリモート Crawler マシンで、FILE プロトコルを使用してアクセスできるドキュメントのセットです。

ファイル・ソースの名前を編集するには、「編集」をクリックします。

ファイル・ソースの作成

新しいファイル・ソースを作成する手順は、次のとおりです。

1. ファイル・ソースの名前を指定します。
2. クロールするファイルまたはディレクトリを指定します。URL が 1 つのファイルを表す場合、Oracle Ultra Search Crawler ではそのファイルのみを検索します。URL が 1 つのディレクトリを表す場合、Crawler ではそのディレクトリ内のすべてのファイルとサブディレクトリを再帰的にクロールします。
3. このファイル・ソースに関連したクロール領域を変更するには、包含パスと除外パスを指定します。このステップはオプションです。包含パスはクロール領域を制限します。除外パスを使用すると、クロール領域をさらに制限できます。どちらのパスも指定されていない場合、クロールはファイル・システムの基本的なアクセス権限に制限されます。
4. Oracle Ultra Search Crawler がこのファイル・ソースについて処理するドキュメントのタイプを指定します。HTML とプレーン・テキストは、Crawler が常に処理するデフォルトのドキュメント・タイプです。

デフォルトでは、Oracle Ultra Search は、ファイル・データ・ソースをテキスト・フォーマットで表示します。ただし、ファイル・データ・ソースに対して表示 URL を指定した場合、Oracle Ultra Search はその URL を使用してファイル・データ・ソースを表示します。

ファイル・データ・ソースに対して表示 URL を使用すると、その URL では、HTTP や HTTPS などのネットワーク・プロトコルを使用してファイル・データ・ソースにアクセスします。ファイル・データ・ソースに対する表示 URL を生成するには、元のファイル URL の接頭辞と表示 URL の接頭辞を指定します。Oracle Ultra Search は、ファイル URL の接頭辞を表示 URL の接頭辞で置き換えます。

たとえば、ファイル URL が `file:///home/archive/<sub_dir_name>/<file_name>` で、表示 URL が `https://host:7777/private/<sub_dir_name>/<file_name>` の場合は、ファイル URL の接頭辞を `file:///home/archive、` 表示 URL の接頭辞を `https://host:7777/private` に指定できます。

ユーザー定義ソース

Oracle Ultra Search では、提供されているデータ・ソースとタイプ以外に、独自のデータ・ソースとタイプを定義、編集または削除できます。独自のドキュメント・リポジトリをクロールして索引付けするために独自の **Crawler** エージェントを実装したり、**Lotus Notes** や **Documentum** など独自のデータベースやインタフェースを持つ管理システムを実装できます。

新規の各データ・ソース・タイプについて、**Crawler** エージェントを **Java** クラスとして実装する必要があります。エージェントは、独自のドキュメント・ソースからドキュメント URL および関連付けられたメタデータを収集し、**Oracle Ultra Search Crawler** に情報を戻します。**Oracle Ultra Search Crawler** では、後でクロールするためにその情報をエンキューします。

関連項目： 6-3 ページ [「Oracle Ultra Search Crawler エージェント API」](#)

新規データ・ソースを定義するには、最初に、そのデータ・ソースを表すデータ・ソース・タイプを定義します。タイプ名、**Crawler** エージェントの **Java** クラス・ファイルまたは **jar** ファイル、および使用するパラメータ（開始アドレスなど）を定義します。データ・タイプを定義した後は、パラメータ値を指定して、新規データ・ソースを定義します。

ユーザー定義データ・ソースの作成

新規のユーザー定義データ・ソースを作成するには、「新規ソースの作成」をクリックします。データ・ソース・タイプを作成、編集または削除するには、「タイプの管理」をクリックします。

ユーザー定義データ・ソース・タイプを作成する手順は、次のとおりです。

1. データ・ソース・タイプ名、説明、および **Crawler** エージェントの **Java** クラス・ファイル名または **jar** ファイル名を指定します。**Crawler** エージェントの **Java CLASSPATH** は、インストール時に事前定義されています。エージェントは、独自のドキュメント・ソースからドキュメント URL のリストおよび関連付けられたメタデータを収集し、**Oracle Ultra Search Crawler** に情報を戻します。**Oracle Ultra Search Crawler** では、後でクロールするためにその情報をエンキューします。エージェントのクラス・ファイルまたは **jar** ファイルは、`$ORACLE_HOME/ultrasearch/lib/agent/` にある必要があります。
2. このデータ・ソース・タイプのパラメータを指定します。パラメータを追加する場合は、パラメータ名と説明を入力する必要があります。パラメータ値を暗号化するかどうかも決定する必要があります。

データ・ソース・タイプに関する情報は、データ・ソース・タイプ名、説明、**Crawler** エージェントの **Java** クラス・ファイル名や **jar** ファイル名、またはパラメータを変更することによって編集します。

ユーザー定義データ・ソースを作成する手順は、次のとおりです。

1. データ・ソースの名前、データ・ソース・タイプおよびデフォルト言語を指定します。各データ・ソースは、データ・ソース・タイプの定義に基づいて作成されます。デフォルト言語の詳細は、5-11 ページの「[Crawler タブ](#)」を参照してください。
2. パラメータ値（開始ポイントなど）を入力します。
3. マッピングを指定します。このステップはオプションです。ドキュメント属性はクロール時に直接、同名の検索属性に自動的にマップされます。ドキュメント属性を他の検索属性にマップする場合は、ここで検索属性を指定します。Crawler は、Crawler エージェントから戻された属性、あるいはここで指定した属性を使用します。

ユーザー定義データ・ソースは、名前、タイプ、デフォルト言語または開始アドレスを変更することによって編集します。

Oracle9iAS Portal ソース

Oracle Ultra Search は、Oracle9i Application Server (9iAS) Portal のクロールと索引付けをサポートしています。これによって、複数のポータルを検索できます。Oracle9iAS Portal をクロールするには、最初に使用するポータルを Oracle Ultra Search に登録する必要があります。ポータルを登録する手順は、次のとおりです。

1. ポータル・ソースの名前およびポータル URL ベースを選択します。ポータル URL ベースは、一度作成すると更新できません。デフォルト言語の詳細は、5-11 ページの「[Crawler タブ](#)」を参照してください。
2. 「ポータルの登録」をクリックします。Oracle Ultra Search は、Oracle9iAS Portal インスタンスに接続し、このインスタンスに関する情報を取得します。

ポータルの登録後に、索引付けを行う Oracle9iAS Portal のページ・グループを選択します。選択した各ページ・グループは、Oracle9iAS Portal ソースとして作成されます。

Oracle Ultra Search Crawler がポータル・ソースで処理するドキュメントのタイプを編集できます。HTML とプレーン・テキストは、Crawler が常に処理するデフォルトのドキュメント・タイプです。ドキュメントが作成された後に、ポータル・ソースの「編集」アイコンをクリックして、ドキュメントのタイプを編集します。

関連項目： Oracle 9iAS Portal のマニュアルを参照してください。

「スケジュール」タブ

このタブを使用して、データ同期および索引最適化をスケジュールします。データ同期の目的は、すべてのデータ・ソースについて **Oracle Ultra Search** の索引を最新に保つことです。索引最適化の目的は、更新された索引を最適化して問合せのパフォーマンスを最大に保つことです。

関連項目： 4-3 ページ「データ・ソースの同期」

データ同期

このページの表には、同期スケジュールに関する情報が表示されます。同期スケジュールには1つ以上のデータ・ソースが割り当てられています。同期スケジュールの頻度によって、割り当てられたデータ・ソースの同期時期が指定されます。スケジュールは最初、名前でソートされています。同期スケジュールでは、個々のデータ・ソースをソース名またはソース・タイプでソートして表示できます。

同期スケジュールの作成

新しいスケジュールを作成するには、「新規スケジュールの作成」をクリックし、次の手順を実行します。

1. スケジュールの名前を入力します。
2. スケジュール頻度を選択し、索引付け用にスケジュールですべての URL を自動的に実行するか、あるいは索引付けの前に URL を調査するかを決定します。スケジュールをリモート **Crawler** プロファイルに関連付けることもできます。
3. データ・ソースをスケジュールに割り当てます。あるグループに割り当てたデータ・ソースを別のグループに割り当てることはできません。

スケジュールの更新

索引付けのオプションは、「スケジュールの更新」タブで更新します。スケジュールの索引付けの前に URL を調査する場合は、スケジュールの実行後、スケジュールのステータスが「索引付け要求」になります。

データ収集モードでは、最初にクロールを開始してください。クロールの完了後、「URL を調査」をクリックしてドキュメント URL およびステータスを調査し、不要なドキュメントを削除して、索引付けを開始します。「索引付けの開始」をクリックすると、スケジュールのステータスが「起動中」から「実行中」や「スケジュール」などに変わります。

特定ホストへのリンクをクリックすると、そのホスト用にクロールされたドキュメント URL のリストが表示されます。ドキュメント URL は、このセクションで削除できます。

同期スケジュールの編集

同期スケジュールを定義すると、同期スケジュール・リストで次の操作を実行できます。

- データベース・ホストで実行されている **Crawler**、または個別のホストで実行されているリモート **Crawler** のいずれかにスケジュールを割り当てるには、「ホスト名」をクリックします。
- スケジュールの頻度を変更するには、スケジュール間隔のテキスト部分をクリックします。
- スケジュールのステータスを変更するには、「ステータス」をクリックします。
- スケジュールを削除するには、「削除」をクリックします。
- スケジュール名、データ・ソース割当て、再クロール・ポリシーまたはクロール・モードを編集するには、「編集」をクリックします。**Crawler** では、ドキュメントを取得するとき、そのドキュメントが変更されているかどうかをチェックします。デフォルトでは、ドキュメントが変更されていない場合、**Crawler** はそのドキュメントを処理しません。特定の状況では、**Crawler** ですべてのドキュメントを強制的に再処理できます。「編集」をクリックし、次の手順でスケジュールを編集します。
 - スケジュール名を更新します。このステップはオプションです。スケジュール名を変更するには、スケジュールの名前を指定し、「スケジュール名の更新」をクリックします。
 - データ・ソースをスケジュールに割り当てます。データ・ソースを割り当てるには、1 つ以上の使用可能なソースを選択し、「>>」をクリックします。あるグループに割り当てたデータ・ソースを別のグループに割り当てることはできません。データ・ソースの割当てを取り消すには、1 つ以上のスケジュールされたソースを選択し、「<<」をクリックします。
 - **Crawler** 再クロール・ポリシーを更新します。再クロール・ポリシーを次のように更新できます。
 - * 変更されたドキュメントの処理 : これはクロールのメンテナンスです。変更されたドキュメントのみが再クロールされ、索引が付けられます。**Web** データ・ソースの場合は、更新されたドキュメントに新しいリンクがあると、それらのリンクが後に続きます。ファイル・データ・ソースの場合は、その親ディレクトリが変更されると新規ファイルが収集されます。
 - * 全ドキュメントの処理 : **Crawler** はデータ・ソースを再クロールします。たとえば、ユーザーは **Web** サイトでテキストと **HTML** のみをクロールすると仮定します。その後、**Microsoft Word** と **Adobe PDF** のドキュメントもクロールする必要性が生じたとします。この場合、ユーザーはソースのドキュメント・タイプを変更し、スケジュールを編集して「全ドキュメントの処理」を選択し、**Crawler** がこのデータ・ソースに対して **PDF** と **doc** のドキュメント・タイプを選択するようにスケジュールを再実行する必要があります。**Crawler** は、各ドキュメントを変更済みとして処理します。つまり、各ドキュメントは再度フェッチされて処理されます。たとえば、新しい **MIME** タイプ・ドキュメント

が追加されると、新しいホスト / パス包含ルールが追加されるか、またはクロールの深さが増加（たとえば、5 から 7 へ）します。各ドキュメントは、深さ制限のために以前に削除されたリンクを選択するために、再度解析する必要があります。

注意：「全ドキュメントの処理」は、クロール範囲が狭くなった場合は役に立ちません。たとえば、クロールの深さが 7 から 5 に削減された場合、PDF MIME タイプが削除された場合、またはホスト包含が削除された場合は、ユーザーが SQL*Plus セッションで影響を受けたドキュメントを手動で削除する必要があります。

- クロール・モードを更新します。クロール・モードを次のように更新できます。
 - * 索引付けのために全ての URL を実行：このモードではクロールと索引付けが行われます。
 - * 索引付けの前に URL を調査：このモードではクロールのみが行われます。
 - * 索引付けのみ：このモードでは索引付けのみが行われます。

Crawler の動作は、収集されるドキュメントによって異なります。

クロール・モードと再クロール・ポリシーは、異なる 6 つの組合せで一体化できます。たとえば、「全ドキュメントの処理」と「索引付けのみ」の組合せでは、このデータ・ソースの既存のドキュメントが強制的に再度索引付けされます。一方、「変更されたドキュメントの処理」と「索引付けのみ」では、変更されたドキュメントのみが再度索引付けされます。

同期スケジュールの起動

次のように、同期スケジュールを起動できます。

- スケジュールの頻度を設定し、事前に決めた起動時間まで待機します。
- すぐに実行します。「ステータス」をクリックし、「すぐに実行」をクリックします。

注意： 同期スケジュールの起動には、長時間かかる場合があります。以前にスケジュールを起動したことがある場合は、次のスケジュール起動時に、スケジュールでクロールされるデータ・ソースに属している全 URL がキュー表にコピーされます。そのデータ・ソースに関連付けられている URL の数によっては、コピー処理に長時間かかる可能性があります。その間、Oracle Ultra Search Administration ツールはスケジュールのステータスを「起動中」と表示します。

ステータスと Crawler の進行状況との同期

ステータス列のリンクをクリックして、同期スケジュール・ステータスを確認します。このスケジュールに関連付けられているデータ・ソースのクロール進行状況を確認するには、「統計」アイコンをクリックします。

クロール進行状況には、次の情報が含まれます。

- データ・ソース・タイプ
- データ・ソース名
- 開始時間
- 完了時間
- 経過時間
- 合計索引作成時間
- 収集されたドキュメント・データの合計サイズ
- 平均ドキュメント・サイズ
- 平均フェッチ・スループット

次の統計も含まれます。

- フェッチするドキュメント
- フェッチされたドキュメント：「索引付けできないドキュメント」、「ドキュメント変換の失敗」および「ドキュメントが索引付けされました」の合計です。
- ドキュメント・フェッチの失敗：HTTP サーバー・タイムアウトまたは別の HTTP サーバー・エラーの可能性があります。
- 拒否されたドキュメント：ドキュメントが URL 境界規則の範囲外です。
- 検出したドキュメント：「フェッチするドキュメント」、「ドキュメントがフェッチされました」、「ドキュメント・フェッチの失敗」および「ドキュメントが拒否されました」の合計です。
- 索引付けされたドキュメント
- 索引付けできないドキュメント：ファイル・ディレクトリ、検出ノードである Oracle9iAS Portal のページ、または索引がないロボット・メタ・タグの可能性があります。
- ドキュメント変換の失敗：バイナリ・ファイルのフィルタ処理に失敗しました。

索引最適化

索引最適化

Oracle Ultra Search Crawler は、すべてのデータ・ソースにわたってクロールしたすべてのドキュメントのアクティブな索引を維持し、問合せ結果を高速に取得します。索引の最適化は、このページでスケジュールできます。索引は、その使用率が低い時間に最適化してください。

注意： Crawler の一時ディレクトリ・サイズを大きくすると、索引の断片化が少なくなります。

索引最適化スケジュール

索引の最適化頻度を指定できます。選択するオプションに必要なすべてのデータを確実に指定してください。索引をすぐに最適化するか、またはスケジュールを使用可能にできます。

最適化プロセスの長さ

索引最適化処理の最大の長さを指定します。最適化に要する実時間がこの制限を超えることはありませんが、この時間より短くなることはあります。長い最適化時間を指定するほど、効率的な最適化が行われます。または、最適化が完了するまで最適化を続行することもできます。

「問合せ」タブ

このセクションでは、データ・グループ、URL 発行、検索オプション、問合せ統計など、問合せに関連する設定を指定できます。

データ・グループ

データ・グループは、検索エンジンのユーザーに公開された論理エンティティです。問合せを入力するときに、ユーザーは検索対象となる 1 つ以上のデータ・グループを選択するように求められます。このページを使用して、これらのデータ・グループを定義してください。

データ・グループは 1 つ以上のデータ・ソースで構成されます。データ・ソースは複数のデータ・グループに割り当てることができます。データ・グループは最初、名前でソートされています。各データ・グループ内にリストされている個々のデータ・ソースは、ソースの名前またはタイプによってソートできます。

新規データ・グループを作成する手順は、次のとおりです。

1. グループの名前を指定します。
2. データ・ソースをグループに割り当てます。Web ソースまたは表データ・ソースをこのデータ・グループに割り当てるには、1 つ以上の使用可能な Web ソースまたは表ソースを選択し、「>>」をクリックします。あるグループに割り当てたデータ・ソースを別のグループに割り当てることはできません。Web ソースまたは表データ・ソースの割当てを解除するには、1 つ以上のスケジュールされたソースを選択し、「<<」をクリックします。
3. 「完了」をクリックします。

URL 発行

URL 発行

URL 発行によって、問合せユーザーは URL を発行できます。これらの URL は開始アドレス・リストに追加され、Oracle Ultra Search Crawler 検索領域に含まれます。ここでは、問合せユーザーによる URL の発行を許可または禁止できます。

URL 境界ルール・チェック

URL は特定の Web データ・ソースに対して発行されます。URL 境界ルール・チェックによって、発行済みの URL が Web データ・ソースの URL 境界ルールに準拠していることが確認されます。URL 境界ルール・チェックは許可または禁止できます。

検索オプション

検索オプションによって、管理者は検索結果をオーバーライドでき、問合せ結果リストにリンク付けされているドキュメントの順序を変更できます。これにより、重要なドキュメントを高くスコア付けできます。また、ドキュメントの検索が容易になります。

検索オプションに関する URL の指定方法には、検索による指定、手動による URL の入力の 2 種類があります。

検索による指定

URL を追加するには、まず URL を検索して指定してください。検索範囲を限定するためにホスト名を指定することもできます。URL の検索後は、「情報」をクリックし、問合せ文字列とドキュメントのスコアを編集します。

手動による URL の入力

クロールまたは索引付けされていないドキュメントは、検索対象になりません。ただし、URL を指定して検索オプション情報を入力することは可能です。それには、「作成」をクリックし、次の項目を入力します。

1. ドキュメント URL を指定します。この URL をデータ・ソースに割り当てる必要があります。このドキュメントには次のクロール時に索引が付けられます。
2. 1 つ以上の問合せ文字列に対し、1 から 100 の範囲でスコアを入力します。ユーザーが正確な問合せ文字列を使用して検索すると、スコアがこの URL に適用されます。

ドキュメントが問合せ条件に対してロードされると、ドキュメントは検索可能になります。また、このドキュメントには次のスケジュール実行時に索引が付けられます。

URL を手動で入力した場合は、Web データ・ソースにのみ URL を割り当てることができます。Web データ・ソースが定義されていない場合は、このページにエラー・メッセージが表示されます。

注意： Oracle Ultra Search では、メタデータを Oracle Ultra Search データベースにロードするためのコマンドライン・ツール（ドキュメント検索オプションなど）が提供されます。データが大量にある場合は、このツールによって HTML ベースの管理ツールより早くロードできます。詳細は、[付録 E「メタデータの Oracle Ultra Search へのロード」](#)を参照してください。

問合せ統計

問合せ統計の使用可能または不可

このセクションでは、問合せ統計の収集を使用可能または使用禁止にできます。問合せ統計をロギングすると、問合せのパフォーマンスが低下します。したがって、通常の処理では問合せ統計の収集を使用禁止にしておくことをお勧めします。

注意： 問合せ統計を使用可能にすると、統計データを格納する表は毎週日曜日の午前 1 時に切り捨てられます。

統計の参照

問合せ統計が使用可能になっている場合は、次のカテゴリのいずれかをクリックできます。

- [毎日の問合せ統計のサマリー](#)
- [トップ 50 の問合せ](#)
- [トップ 50 の無効な問合せ](#)

- **トップ 50 の失敗した問合せ**

毎日の問合せ統計のサマリー すべての問合せアクティビティが毎日集計されています。収集された統計は次のとおりです。

- 平均問合せ時間 : すべての問合せにかかる平均時間
- 問合せ数 : その日の問合せ総数
- ヒット数 : 各問合せにより戻される平均結果数

トップ 50 の問合せ 24 時間以内に最も頻度の高かった 50 の問合せが集計されています。

- 問合せ文字列 : 問合せ文字列
- 平均問合せ時間 : 結果が戻るまでの平均時間
- 問合せ数 : 24 時間以内に実行された問合せの総数
- ヒット数 : 各問合せにより戻される平均結果数
- 頻度 : すべての問合せ文字列について、問合せ総数で除算した問合せの数
- 無効な問合せ率 : すべての問合せ文字列について、問合せ総数で除算した無効な問合せの数

トップ 50 の無効な問合せ 24 時間以内に行われた最も効果のなかった 50 の問合せが集計されています。表の各行には、特定の問合せ文字列に対する統計の説明があります。

- 問合せ文字列 : 問合せ文字列
- 問合せ数 : 24 時間以内に実行された問合せの総数
- 無効な問合せ率 : この文字列について、問合せ総数で除算した無効な問合せの数

トップ 50 の失敗した問合せ 24 時間以内に失敗した 50 の問合せが集計されています。失敗した問合せは、検索エンジンのエンド・ユーザーが結果を検索しなかった問合せです。

次の列があります。

- 問合せ文字列 : 問合せ文字列
- 問合せ数 : 24 時間以内に実行された問合せの総数
- 頻度 : 失敗した問合せの発生率
- 累積頻度 : すべての失敗した問合せの累積発生率

関連項目 : 付録 B 「問合せパフォーマンスのチューニング」

「ユーザー」 タブ

このタブを使用して、Oracle Ultra Search の管理ユーザーを管理します。ユーザーを割り当てて、Oracle Ultra Search インスタンスを管理できます。言語を選択することもできます。

設定

このセクションでは、Oracle Ultra Search 管理者の設定オプションを設定できます。

日付書式と時刻書式を指定できます。次の言語がプルダウン・メニューにリストされます。

- 英語
- ポルトガル語（ブラジル）
- フランス語
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- 簡体字中国語
- スペイン語
- 繁体字中国語

各ページに表示する行数も選択できます。

スーパー・ユーザー

スーパー・ユーザー権限を持つユーザーは、インスタンスの作成、削除および権限の付与など、インスタンスに関するすべての管理機能を実行できます。このページにアクセスできるのは、スーパー・ユーザーのみです。

他のユーザーにスーパー・ユーザー権限を付与するには、ユーザー名とタイプを指定します。他のユーザーにスーパー・ユーザー権限を付与できるかどうかも指定します。その後、「追加」ボタンをクリックします。

権限

このページにアクセスできるのは、インスタンスの所有者、このインスタンスの一般管理権限を付与されたユーザーまたはスーパー・ユーザーのみです。インスタンス所有者には WKUSER ロールを付与する必要があります。

一般管理権限を付与されたユーザーは、このインスタンスの一般的な設定を変更できます。この権限を付与するには、ユーザー名とタイプを指定します。他のユーザーに管理権限を付与できるかどうかも指定します。その後、「追加」ボタンをクリックします。

このインスタンスの管理者リストから 1 つ以上のユーザーを削除するには、削除するユーザー名を現在の管理者リストから選択し、「削除」ボタンをクリックします。

一般管理権限に、次の権限は含まれません。

- インスタンスの作成
- インスタンスの削除

これらの権限はスーパー・ユーザーの権限です。

関連項目： 3-5 ページ「[手順 4: 各 Oracle Ultra Search インスタンス用の新規データベース・ユーザーの作成および構成](#)」

「グローバルゼーション」タブ

Oracle Ultra Search では、名前を異なる言語に翻訳できます。このタブでは、検索属性、LOV 表示名およびデータ・グループに対して複数の値を入力できます。

検索属性名

このセクションでは、属性表示名を異なる言語に翻訳できます。次の言語がプルダウン・メニューにリストされます。

- 英語
- アラビア語
- ポルトガル語（ブラジル）
- フランス語（カナダ）
- チェコ語
- デンマーク語
- オランダ語
- フィンランド語
- フランス語

- ドイツ語
- ギリシャ語
- ヘブライ語
- ハンガリー語
- イタリア語
- 日本語
- 韓国語
- スペイン語（ラテン・アメリカ）
- ノルウェー語
- ポーランド語
- ポルトガル語
- ルーマニア語
- ロシア語
- 簡体字中国語
- スロバキア語
- スペイン語
- スウェーデン語
- タイ語
- 繁体字中国語
- トルコ語

LOV 表示名

このセクションでは、属性 LOV 表示名を異なる言語に翻訳できます。プルダウン・メニューから検索属性（作者、説明、MIME タイプ、件名またはタイトル）を選択します。LOV タイプを選択し、プルダウン・メニューから言語を選択します。言語オプションがプルダウン・メニューにリストされます。

データ・グループ名

このセクションでは、データ・グループの表示名を異なる言語に翻訳できます。言語オプションがプルダウン・メニューにリストされます。

Oracle Ultra Search 開発者ガイドおよび API リファレンス

この章では、Oracle Ultra Search の API および関連情報について説明します。この章では、次の項目について説明します。

- [Oracle Ultra Search API の概要](#)
- [Oracle Ultra Search の Query API](#)
- [Oracle Ultra Search Crawler エージェント API](#)
- [Oracle Ultra Search の Java Email API](#)
- [Oracle Ultra Search の URL リライター API](#)
- [Oracle Ultra Search のサンプル検索アプリケーション](#)
- [Oracle Ultra Search の Query タグ・ライブラリ](#)
- [問合せ構文拡張のカスタマイズ](#)

関連項目： [Oracle Ultra Search Javadoc](#)

Oracle Ultra Search API の概要

Oracle Ultra Search には、次の API が用意されています。

- **Query API**。索引付けされたデータで使います。Java API は、HTML レンダリング要素を適用しません。HTML インタフェースは、アプリケーションで完全にカスタマイズできます。
- **Crawler エージェント API**。独自のドキュメント・リポジトリをクロールして索引付けします。
- **Email API** は、Oracle Ultra Search の検索アプリケーションが電子メールを表示する際に使用されます。この API は、独自のカスタム検索アプリケーションを作成するときにも使用できます。
- **URL リライター API**。Crawler はこれを使用して、抽出した URL リンクが URL キューに挿入される前にフィルタ処理およびリライトします。

Oracle Ultra Search には、検索結果を問い合わせるための、高度な機能を備えた検索アプリケーションも含まれています。検索アプリケーションは Java Server Pages (JSP) に基づいており、JSP1.1 準拠のエンジンで動作します。

Oracle Ultra Search の Query API

Oracle Ultra Search は、索引付けされたデータの問合せ用に Java API を提供します。この API のメソッドによって、問合せ結果を検索および表示します。この API は Java で記述されているため、JSP (バージョン 1.1 以上) などの Java ベースのテクノロジーをサポートする様々な Web アプリケーション・サーバーと互換性があります。Java API は、拡張性を得るために JDBC 接続プーリングを使用します。

Java API は、HTML レンダリング要素を必要としません。アプリケーションでは、HTML インタフェースを完全にカスタマイズできます。次に例を示します。

- 基本検索フォーム
- 拡張検索フォーム
- 問合せ結果の表示
- ヘルプ・ページ
- フィードバック・ページ
- URL 登録

Oracle Ultra Search Java Query API を使用して、Web アプリケーションに Oracle Ultra Search の問合せ機能を埋め込みます。この API では、次の 2 種類のメソッドをサポートします。

1. 問合せ結果データのみを取得するメソッド
2. 問合せ結果データを含む HTML コードを取得するメソッド

HTML コードを取得するメソッドでは、問合せ入力ボックスおよび結果リストを Web アプリケーションに埋め込む機能をサポートしています。データのためのメソッドは、HTML を戻しません。これは、HTML コードを完全に制御してレンダリングする必要がある場合に使用します。

Oracle Ultra Search Java Query API の機能により、次のことが可能です。

- 問合せ結果の取得。
- 戻される合計ヒット数など、問合せのプロパティの設定。
- 問合せセッションの言語の設定。
- Oracle Ultra Search のディクショナリ・データの取得。Oracle Ultra Search の表にアクセスし、定義済みのすべてのデータ・グループや属性などを取得します。
- Query インタフェースおよび検索結果画面のカスタマイズおよび生成。Web アプリケーションへの埋込みが可能な HTML コードのブロックを戻すプロシージャにより可能。
- 検索を行うエンド・ユーザーに対する、シード URL リストへの URL 発行の許可。
- Oracle Ultra Search Java Query API は、`oracle.ultrasearch.query` パッケージ内にカプセル化されています。

関連項目： [付録 B「問合せパフォーマンスのチューニング」](#)

Oracle Ultra Search Crawler エージェント API

Crawler エージェントを実装すると、Lotus Notes や Documentum などの独自のドキュメント・リポジトリをクロールし、索引付けできます。Oracle Ultra Search では、独自のリポジトリを、ユーザー定義データ・ソースと呼びます。Crawler によるデータ・ソースへのアクセスを可能にするモジュールを Crawler エージェントと呼びます。

エージェントは、ユーザー定義データ・ソースからドキュメント URL および関連付けられたメタデータを収集し、Oracle Ultra Search Crawler に情報を戻します。Oracle Ultra Search Crawler では、後でクロールするためにその情報をエンキューします。Crawler エージェントは、Oracle Ultra Search の Crawler エージェント API を使用して、Java で実装する必要があります。

Oracle Ultra Search には、Oracle Ultra Search エージェント API を使用したユーザー定義 Crawler エージェントの実装サンプルが用意されています。起動時に、このサンプル・エー

ジェントは指定の **Oracle** データベースに接続し、**Crawler** が収集して索引付けを行う表の内容を取得します。

サンプル・エージェントは完全な機能を備えており、他のデータベース・ベースのデータ・ソースに適合するようにカスタマイズできます。このエージェントは、次のタスクを実行します。

- データ・ソース・パラメータの読み込み
- データ・ソースが含まれるデータベースとの接続
- データ・ソースからフェッチするドキュメント URL および属性の初期化
- ドキュメント URL および属性のデータ・ソースからのフェッチ
- データ・ソースからの切断

Crawler エージェントの概要

Crawler エージェントは次の処理を行います。

- データ・ソースにアクセスするための **Crawler** の認証
- HTTP URL（表示 URL）を使用したデータ・ソース・ドキュメントへのアクセスの提供
- ドキュメント属性の形式でドキュメントのメタデータを提供
- ドキュメント属性を、エンド・ユーザーが使用する一般的な属性名にマップ
- ドキュメントがストリーム方式で 1 つずつ取り出されるように、データ・ソースのフラット化されたビューを提供
- 必要に応じて、作成者やタイトルなどの標準メタデータに関する URL ドキュメントを解析するように **Crawler** に指示
- （オプション）指定したタイム・スタンプ以降に変更された URL のリストを提供
- （オプション）ドキュメントの処理について、表示 URL のみでなくアクセス URL を提供

エージェントは、ターゲット・データ・ソースから URL のリストを取り出し、クロール処理の前にそのリストを **Crawler** キューに保存します。

注意： なんらかの理由で **Crawler** が中断すると、元のクロールの最新タイム・スタンプを使用してエージェント起動プロセスが繰り返されます。エージェントからフェッチされた URL のエンキューがすでに終了し、クロール中の場合、**Crawler** はエージェントを開始するだけで、エージェントからの URL のフェッチは行いません。かわりに、すでにエンキューした URL のクロールを終了します。

2 種類の Crawler エージェントがあります。

- [標準エージェント](#)
- [スマート・エージェント](#)

標準エージェント

標準エージェントは、現在データ・ソースに存在している URL のリストを戻します。URL が以前にクロールされたかどうかは認識されず、ターゲット・データ・ソースに対する更新の検出は Crawler に依存します。標準エージェントと Crawler の相互作用は次のとおりです。

- Crawler は、このデータ・ソースのすべての既存 URL がターゲット・データ・ソースにすでに存在しない場合、これらの URL にガベージ・コレクション用のマークを付けます。
- エージェントをコールして、更新された URL リストを取得します。すべての既存 URL にクロール用のマークを付けます。新しい URL は、URL 表およびキューに挿入されます。
- 前述の処理後、まだガベージ・コレクション用のマークが付いている URL を削除します。
- クロール用のマークが付いているすべての URL を参照し、更新されているかどうかをチェックします。

スマート・エージェント

スマート・エージェントは、Crawler が提供する最終更新タイム・スタンプを使用して、更新、挿入または削除された URL のリストを戻します。Crawler は、エージェントから戻された URL のみをクロールし、既存の URL は再クロールしません。削除された URL は、Crawler によって URL 表から削除されます。スマート・エージェントが更新または挿入された URL のみを戻し、削除された URL を戻すことができない場合、Crawler 側では削除された URL は検出されません。この場合は、スケジュールの Crawler 再クロール・ポリシーを変更し、強制再クロール・モードでスケジュールを定期的に行う必要があります。強制再クロール・モードでは、エージェントに対し、データ・ソースのすべての URL を戻すように指示が出されます。

`isDeltaCrawlingCapable()` は、起動したエージェントが標準エージェントであるか、スマート・エージェントであるかを Crawler に通知します。`startCrawling(boolean forceRecrawl, Date lastCrawlTime)` によって、Crawler は、強制再クロール・モードで実行中かどうか、および最後にクロールした時間をエージェントに通知できます。

ドキュメントの属性とプロパティ

ドキュメント属性（メタデータ）は、ドキュメントのプロパティを示します。一部の属性は、ユーザー・アプリケーションと関連を持たない場合があります。Crawler エージェントの作成者は、抽出および保存するドキュメント属性を決定する必要があります。収集した属

性のリストが構成可能になるように、エージェントを作成することもできます。Oracle Ultra Search は、エージェントから戻された属性を自動的に登録します。あるドキュメントに関してどの属性を戻すかをエージェントで決定できます。

Crawler エージェントの機能

データ・ソース・タイプの登録

データ・ソース・タイプとは、データ・ソースを抽象化したものです。次の属性を使用して、新しいデータ・ソース・タイプを定義できます。

- データ・ソース・タイプの名前 : Lotus Notes など。最大長は 100 バイトです。
- データ・ソース・タイプの ID: 自動的に割り当てられます。
- データ・ソース・タイプの説明 : 最大長は 4000 バイトです。
- エージェント Java クラス名 : WebDbAgent など。このクラスの位置は、Oracle Ultra Search によって `$ORACLE_HOME/ultrasearch/lib/agent/` に事前定義されており、変更できません。
- エージェント Java jar ファイル名 : エージェント・クラスは Java の jar ファイルに格納できます。この jar ファイルは、`$ORACLE_HOME/ultrasearch/lib/agent/` に存在している必要があります。この `$ORACLE_HOME` は、Middle-Tier コンポーネントではなく、Oracle Ultra Search Server コンポーネントがインストールされている Oracle ホーム・ディレクトリです。
- パラメータ : データ・ソースのプロパティです。たとえば、Web データ・ソースのシード URL、包含パターンおよびロボットの除外などです。パラメータは、パラメータ名（最大 100 バイト）と説明（最大 4000 バイト）を指定して定義します。デフォルトでは、パラメータは暗号化されません。
- 暗号化 : 格納時にこのパラメータの値を暗号化するかどうかを指定します。

Oracle Ultra Search ではパラメータの出現回数は規定されません。特定のパラメータが 0 回以上、少なくとも 1 回または 1 回のみ出現するよう指定できません。

データ・ソースの登録

データ・ソース・タイプを定義した後は、次の属性を使用して、そのデータ・ソース・タイプのインスタンスを定義できます。

- データ・ソース名。
- データ・ソースの説明 : 最大長は 4000 バイトです。
- データ・ソース・タイプの ID。

- デフォルト言語 : デフォルトは「en」(英語) です。
- パラメータ値 : seed - <http://www.oracle.com/depth-8> など。

データ・ソース属性の登録

Oracle Ultra Search に新規属性を追加するには、属性名と属性のデータ型を指定します。データ型は、文字列、数値または日付です。名前が同じでデータ型が異なる属性を追加することもできます。エージェントから戻された属性が定義されていない場合は、自動的に登録されます。

ユーザー実装 Crawler エージェント

Crawler エージェントの要件は次のとおりです。

- Java で実装すること。
- Oracle Ultra Search によって定義された Java エージェント API をサポートしていること。
- URL の属性およびプロパティを戻すこと。
- (オプション) Crawler によるデータ・ソースへのアクセスを認証できること。
- ドキュメントがストリーム方式で 1 つずつ取り出されるように、データ・ソースをフラット化すること。これは、特定のデータ・ソースのクロール・ロジックをエージェントにカプセル化することです。
- Oracle Ultra Search で保持する必要のあるドキュメント属性を決定すること。Oracle Ultra Search で定義されていない属性は自動的に登録されます。
- 属性をデータ・ソースのプロパティにマップできること。たとえば、属性「ID」がドキュメントの一意の ID で、「ID」がプロパティ「document_key」にマップされており、この特定のドキュメントに対する ID の値が 4 の場合、エージェントは (document_key, 4) を戻す必要があります。
- リクエストに応じて、使用可能な属性 LOV を戻すこと。

Crawler と Crawler エージェントの相互作用

Crawler は、ユーザーが提供したエージェントを起動することによって、ユーザーが定義したデータ・ソースをクロールします。Crawler は次の処理を実行できます。

- 定義済みデータ・ソースの Crawler エージェントの起動
- データ・ソース・パラメータ情報のエージェントへの提供
- エージェントを使用した自己認証 (必要な場合)
- URL リストの取出し、およびクロールする必要がある属性 / プロパティの関連付け
- エージェントから提供された URL を使用したドキュメントの取出し

- データ・ソースに対する挿入、更新および削除の検出
- 属性 LOV データの取出し（使用可能な場合）

Crawler エージェント API およびクラス

Crawler エージェント API は、Crawler エージェントの実装に使用するメソッドの集合です。Crawler エージェントの実装例 `SampleAgent.java` は、`$ORACLE_HOME/ultrasearch/sample/` にあります。

UrlData: Crawler エージェントは、このインタフェースを使用して、ドキュメントのプロパティと属性値を移入します。Oracle Ultra Search は、このインタフェースの基本実装を提供します。エージェントは、このインタフェースを直接使用したり、必要に応じて拡張できます。クラスは、引数のないコンストラクタを持つ `DocAttributes` です。エージェントは、`UrlData` オブジェクトのプールを作成し、クロール時にそのプールを繰り返し参照することもできます。最も単純な実装の場合、エージェントは `DocAttributes` オブジェクトを1つ作成してデータのリセットと移入を繰り返し、このオブジェクトを戻します。

LovInfo: Crawler エージェントは、このインタフェースを使用して、属性の LOV 定義を送信します。

DataSourceParams: Crawler エージェントは、このインタフェースを使用して、データ・ソース・パラメータの読取りおよび書込みを行います。

AgentException: Crawler エージェントは、エラー発生時にこの例外クラスを使用します。

CrawlerAgent: このインタフェースによって、Crawler はユーザー定義のデータ・ソースと通信できます。Crawler エージェントはこのインタフェースを実装する必要があります。

サンプル・エージェント・ファイル

サンプル・エージェント・ファイルは、`$ORACLE_HOME/ultrasearch/sample` ディレクトリに含まれています。テキスト・エディタを使用して、サンプル・エージェントのソース・コードを直接表示できます。

`sample_agent_readme.htm` ファイルと `SampleAgent.java` ファイルがあります。これは、エージェント API を使用するサンプル Crawler エージェント実装用です。

サンプル Crawler エージェントの設定

エージェント jar ファイルのコンパイルおよび作成

サンプル・エージェントの Java ソース・コードは、最初にクラス・ファイルにコンパイルしてから、`$ORACLE_HOME/ultrasearch/lib/agent/` ディレクトリの jar ファイルに格納する必要があります。この `$ORACLE_HOME` は、Middle-Tier コンポーネントではなく、Oracle Ultra Search Server コンポーネントがインストールされている Oracle ホーム・ディレクトリです。

コンパイルに必要なクラスは、JDK クラス (classes.zip)、Oracle JDBC Thin ドライバ (classes12.zip) および ultraserach.jar です。次に例を示します。

```
javac -J-ms16m -J-mx96m -O -classpath /jdk1.2.2
05/lib/classes.zip:/lib/classes12.zip:
$ORACLE_HOME/ultrasearch/lib/ultrasearch.jar SampleAgent.java
```

sampleAgent.jar ファイルは次のように作成します。

```
/jdk1.2.2_05/bin/jar cv0f /oracle/ultrasearch/lib/agent/sampleAgent.jar
SampleAgent.class 'SampleAgent$DocNode.class'
```

データ・ソース・タイプの作成

最初に、サンプル・エージェントを使用するデータ・ソース・タイプを作成します。

- 名前: URL 表タイプ
- 説明: URL の行がある表
- エージェント名: SampleAgent
- エージェント jar ファイル: sampleAgent.jar

データ・ソース・パラメータの定義

データ・ソース・タイプのパラメータを定義します。

- データベース接続文字列 (DB 接続)
- ユーザー名 (URL 表のスキーマ所有者)
- パスワード (スキーマ所有者の暗号化されたパスワード)
- 表名 (URL 表名)
- URL 列 (ドキュメント URL を保持する列)
- 無視フラグ列 (無視する場合は 1、それ以外は 0)
- 言語列 (ドキュメント言語)
- 属性リスト (属性の列のリスト)
- [column name/attribute name] <data type> [column name/attribute name] <data type> ... の書式で指定します。<data type> は、0 が数値、1 が文字列、2 が日付です。たとえば、ドキュメントに Company Name、Category、Revenue および S&P Rating の 4 つの属性がある場合は、[Company Name/Company/1][Category/Classification/1][Revenue/Revenue/0][Rating/Analyst Rating/1] のように指定します。

- ログ・ファイル名 (ログ・ファイル)
- ログ・ディレクトリ (ログ・ファイルの位置)

このタイプのデータ・ソースの定義

データ・ソースを定義して、データ・ソース・パラメータを初期化します。たとえば、値を指定して、次のスキーマの表にアクセスします。

```
TABLE NEWS (  
  ARTICLE_NO    NUMBER,  
  NEWS_URL      VARCHAR2(740),  
  TITLE         VARCHAR2(200),  
  AUTHOR        VARCHAR2(100),  
  PUB_DATE      DATE default SYSDATE,  
  PUBLISHER     VARCHAR2(100),  
  PRICE         NUMBER,  
  LANG          VARCHAR2(10),  
  IGNORE        NUMBER DEFAULT 0,  
  PRIMARY KEY (NEWS_URL)  
);
```

- データベース接続文字列 dlsun1710:5521:search
- ユーザー名 : SCOTT
- パスワード : TIGER
- 表名 : NEWS
- URL 列 : NEWS_URL
- 無視フラグ列 : IGNORE
- 言語列 : LANG
- 属性リスト : [ARTICLE_NO/Article Number/0][TITLE/Article Title/1][AUTHOR/Author/1][PUB_DATE/Report Date/2][PUBLISHER/Newspaper/1][PRICE/Download Cost/0]
- ログ・ファイル名 : testagent.log
- ログ・ディレクトリ : /tmp/ultrasearch/

Oracle Ultra Search の Java Email API

Oracle Ultra Search は、アーカイブされた電子メールにアクセスするための Java API を提供します。Oracle Ultra Search 検索アプリケーションは、この API を使用して、Oracle Ultra Search システムで索引付けされたメーリング・リストにアドレス指定されている電子メールを表示します。この API は、独自のカスタム検索アプリケーションを作成するためにも使用できます。

アプリケーションのユーザー・インタフェース・ロジックは完全に JSP で制御されているため、必要に応じて外観と操作性をカスタマイズできます。

電子メール・ドキュメントには有効な情報が含まれていますが、関連情報の検索が容易にできる構成にはなっていません。Oracle Ultra Search では、IMAP4 プロトコルをサポートするサーバー上の電子メールを取得および索引付けすることができます。

電子メール・ソースは、特定の電子メール・アドレスに送信された電子メールから内容を導出するデータ・ソースです。Oracle Ultra Search Crawler が電子メール・ソースを検索する場合、Crawler は電子メールのヘッダー・フィールド「宛先:」または「Cc:」に特定の電子メール・アドレスがあるすべての電子メールを収集します。

注意： Oracle Ultra Search では、検索されたすべての電子メールのコピーを Oracle Ultra Search Server がインストールされているローカル・ファイル・システムに格納します。

電子メール・ソースのアプリケーションは、メーリング・リストに送信されたすべての電子メールの電子メール・ソースです。この場合は、複数の電子メール・ソースが定義され、各電子メール・ソースは 1 つの電子メール・リストに対応します。

Oracle Ultra Search の電子メールのクロールおよびレンダリングは、Sun 社の JavaMail のリファレンス実装を使用して、JavaMail API の最上位に構築されます。これによって、Oracle Ultra Search は索引付けされた電子メールにアクセスするための Java API を提供できます。この API が、Oracle Ultra Search の Java Email API です。この API によって、電子メールのヘッダー情報、内容、添付ファイルなどの情報を取得できます。

この API を使用して、JSP またはサーブレット・ベースの Web アプリケーションに Oracle Ultra Search の電子メールの表示機能を埋め込みます。出荷時の Oracle Ultra Search には、完全な機能を持つ JSP の Web アプリケーションが含まれています。このアプリケーションは、API を直接使用して、索引付けされた電子メールをレンダリングします。ソース・コードは表示可能なため、電子メール・ブラウザをカスタマイズするときに使用できます。

JavaMail の実装

Oracle Ultra Search には、JavaMail 1.1 に準拠した実装が必要です。Sun 社のリファレンス実装は、JavaMail バージョン 1.2 です。このリファレンス実装は、出荷時の Oracle Ultra Search に含まれています。

Java Email API

Oracle Ultra Search の Java Email API は、oracle.ultrasearch.query パッケージ内にカプセル化されています。

サンプル・メーリング・リスト・ブラウザ・アプリケーション・ファイル

サンプル・メーリング・リスト・ブラウザ・アプリケーション・ファイルは、`$ORACLE_HOME/ultrasearch/sample/query` ディレクトリに含まれています。テキスト・エディタを使用すると、サンプル・メーリング・リスト・ブラウザ・アプリケーションのソース・コードを直接表示できます。

次の表に、すべてのサンプル・メーリング・リスト・ブラウザ・アプリケーション・ファイルを示します。

README ファイルおよびスタイルシート：

ファイル	説明
README.html	このファイル
mail.css	サンプル電子メール Web アプリケーション用のスタイルシート

JSP のサンプル・メーリング・リスト・ブラウザ・アプリケーション・ファイル：

ファイル	説明
mail.jsp	エンド・ユーザーが表示するコードに応じて、他の JSP ファイルから戻された HTML コードが含まれているメーリング・リスト・ブラウザ・アプリケーション
mailindex.jsp	Oracle Ultra Search インスタンスのすべての電子メール・ソース（メーリング・リスト）が表示される JSP ページ
mailmsgs.jsp	電子メール・ソース（メーリング・リスト）のすべての電子メールが表示される JSP ページ

ファイル	説明
mailreader.jsp	電子メールが表示される JSP ページ
mailutil.jsp	mailreader.jsp によって使用される様々な機能を定義する JSP ページ

すべてのアプリケーションの画像ファイル:

ファイル	説明
images/ultra_mediumbanner.gif	Oracle Ultra Search のバナー
images/wsd.gif	サンプル検索アプリケーションで使用される背景イメージ

サンプル・メーリング・リスト・ブラウザ・アプリケーションの設定

JSP のサンプル・メーリング・リスト・ブラウザ・アプリケーションの詳細は、2-9 ページの「[Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール](#)」を参照してください。

Oracle Ultra Search の URL リライタ API

URL リライタは、Ultra Search UrlRewriter Java インタフェースを実装するためにユーザーが提供する Java モジュールです。Crawler は、アクティブ化された URL リライタを使用して、抽出した URL リンクを URL キューに挿入される前にフィルタ処理およびリライトします。

Web クロールは通常、次のステップで構成されます。

1. URL キューから次の URL を取得します (キューが空の場合、Web クロールは停止します)。
2. URL の内容をフェッチします。
3. 内容から URL リンクを抽出します。
4. リンクを URL キューに挿入します。

生成された新しい URL リンクは、既存のホスト、パスおよび MIME タイプすべての包含ルールと除外ルールの影響を受けます。

抽出した URL リンクで実行できる操作は 2 つあります。

- フィルタ処理: 不要な URL リンクを取り除きます。
- リライト: URL リンクを変換します。

URL リンクのフィルタ処理

ユーザーは、Oracle Ultra Search Crawler がサポートする次のメカニズムを使用して、キューに挿入できる URL リンクのタイプを制御します。

- ターゲット Web サイトの robots.txt ファイル（たとえば、/cgi ディレクトリからの URL を禁止する）
- ホストの包含ルールと除外ルール（たとえば、www.acme.com の URL のみを許可する）
- ファイル・パスの包含ルールと除外ルール（たとえば、/archive ディレクトリ以下の URL のみを許可する）
- MIME タイプの包含ルール（たとえば、HTML ファイルと PDF ファイルのみを許可する）
- ロボット・メタ・タグ NOFOLLOW（たとえば、そのページのいかなるリンクも抽出しない）
- ブラック・リスト URL（たとえば、明示的に指定した URL をクロールしない）

これらのメカニズムによって、フィルタ処理基準に一致する URL リンクのみが処理されます。ただし、ユーザーが URL リンクのフィルタ処理に使用できる他の基準があります。次に例を示します。

- 特定のファイル名拡張子を持つ URL の許可
- 特定のポート番号からの URL のみ許可
- 特定ディレクトリの PDF ファイルの禁止

基準は、Crawler が抽出した URL リンクの評価時に使用する、ユーザー実装モジュールであるため、考えられる基準は無限です。

URL リンクのリライト

一部のアプリケーションについては、セキュリティ上の理由から、クロールされる URL とエンド・ユーザーが参照する URL が異なる場合があります。たとえば、ファイアウォール内にある内部 Web サイトでのクロールはセキュリティ・チェックなしで行われますが、エンド・ユーザーによる問合せがあると、ファイアウォール外の対応するミラー URL を使用する必要があります。

表示 URL は、検索ヒット表示に使用される URL 文字列です。これは、ユーザーが検索ヒット・リンクをクリックするために使用される URL です。アクセス URL は、Crawler がクロールと索引付けを行うために使用する URL 文字列です。アクセス URL はオプションです。アクセス URL がない場合、Crawler は表示 URL を使用してクロールと索引付けを行います。アクセス URL が指定されている場合、Crawler は表示 URL を使用せずに、このアクセス URL を使用してクロールします。

標準的な Web クロールで使用できるのは、表示 URL のみです。しかし、状況によっては、Crawler は外部使用の目的で表示 URL を保持しながら、内部サイトをクロールするために

アクセス URL を必要とします。すべての内部 URL に対して、ミラー化された外部 URL があります。

次に例を示します。

```
http://www.acme-ga.us.com:9393/index.html  
http://www.acme.com/index.html
```

URL リンク `http://www.acme-ga.us.com:9393/index.html` が抽出されると、キューに挿入される前に、**Crawler** によって、この URL リンクに対する新規の表示 URL とアクセス URL が生成されます。

```
Access URL - http://www.acme-ga.us.com:9393/index.html  
Display URL - http://www.acme.com/index.html
```

抽出された URL リンクはリライトされ、**Crawler** は、エンド・ユーザーに公開せずに内部 Web サイトをクロールします。

別の例として、**Crawler** が選択するリンクが動的に生成され、それらのリンクがすべて同じページを指し示すときも、(参照ページまたはその他の要因に従って) 異なるリンクとなる場合があります。次に例を示します。

```
http://compete3.acme.com/rt/rt.wvw_media.show?p_type=text&p_id=4424&p_  
currcornerid=281&p_textid=4423&p_language=us  
http://compete3.acme.com/rt/rt.wvw_media.show?p_type=text&p_id=4424&p_  
currcornerid=498&p_textid=4423&p_language=us
```

Crawler が同じ内容の異なる URL を検出するのは、重複が相当数存在する場合のみです。このため、URL キューは膨大な数の URL となる可能性があり、その結果、過度の URL リンク生成を引き起こします。この場合は、同じページを指し示す URL が同じ URL となるように、抽出したリンクを正規化することができます。これらの URL をリライトするアルゴリズムはアプリケーションごとに異なり、**Crawler** による一般的な方法では処理できません。

URL リンクがリライトされると、次のいずれかの結果となります。

- リンクが未変更のまま挿入されます。
- リンクが取り消され、挿入されません。
- 新規表示 URL が戻されます。挿入のために URL リンクが置換されます。
- 表示 URL およびアクセス URL が戻されます。表示 URL は、URL リンクと同一でない可能性があります。

URL リライタの作成および使用

URL リライターを作成して使用するには、次の手順を実行します。

1. `UrlRewriter` インタフェースの `open()`、`close()` および `rewrite()` の各メソッドを実装する新規 Java ファイルを作成します。サンプル・リライター `SampleRewriter.java` は、`$ORACLE_HOME/ultrasearch/sample/` 以下の参照に使用できます。
2. リライター Java ファイルをクラス・ファイルにコンパイルします。次に例を示します。

```
/jdk1.3.1/bin/javac -O -classpath $ORACLE_HOME/ultrasearch/lib/ultrasearch.jar  
SampleRewriter.java
```
3. リライター・クラス・ファイルを `$ORACLE_HOME/ultrasearch/lib/agent/` ディレクトリ下の `jar` ファイルにパッケージ化します。次に例を示します。

```
/jdk1.3.1/bin/jar cv0f $ORACLE_HOME/ultrasearch/lib/agent/sample.jar  
SampleRewriter.class
```
4. 5-18 ページの「[Web ソースの作成](#)」のステップ 2、または既存の Web データ・ソースの `Crawler` パラメータ・ページの `Administration` ツールで、リライター・クラス名と `jar` ファイル名（たとえば、`SampleRewriter` と `sample.jar`）を指定します。
5. `Administration` ツールの [Web ソース](#) タブで、URL リライター・オプションを有効にします。
6. 対応するスケジュールを起動して、ターゲット Web データ・ソースをクロールします。`Crawler` ログ・ファイルは、「URL リライター `SampleRewriter` のロード」メッセージによって URL リライタの使用を確認します。

注意： URL リライトは、Web データ・ソースにのみ使用可能です。

関連項目：

- API (`oracle.ultrasearch.crawler` パッケージ) の詳細は、Oracle Ultra Search Javadoc を参照してください。
- サンプル URL リライター `SampleRewriter.java` は、`$ORACLE_HOME/ultrasearch/sample/` にあります。
- 5-18 ページ「[Web ソース](#)」

Oracle Ultra Search のサンプル検索アプリケーション

Oracle Ultra Search には、複数のサンプル検索アプリケーションおよびサンプル Crawler エージェントが含まれています。サンプル検索アプリケーションは、独自の検索アプリケーションを作成するための例として使用します。検索アプリケーションは、JSP アプリケーションとして記述されています。検索アプリケーションは、Oracle Ultra Search の Query API を使用します。サンプル Crawler エージェントを使用して、独自の Crawler エージェントを作成することもできます。

注意： サンプル検索アプリケーションとサンプル Crawler エージェントの Java ソース・コード、およびそれに対応する README へのポインタは、「Oracle Ultra Search へようこそ」ページ <http://<hostname.domainname>:<HTTPport>/ultrasearch/index.html> にあります。

サンプル検索アプリケーションは、配置済みの J2EE Web アプリケーション (sample.ear) として提供されています。このコンポーネントは、Web ページ、JDBC ドライバ、および電子メール結果を表示するための Java Email API を管理する J2EE コンテナに依存しています。Oracle Containers for J2EE (OC4J) によって sample.ear ファイルが配置されると、Query API の使用方法を示す一連の JSP ファイルが表示されます。

サンプル検索アプリケーションには、サンプル検索ポートレットが含まれています。Oracle Ultra Search のサンプル・ポートレットは、Oracle9iAS Portal で使用する検索ポートレットの作成方法を説明します。

ユーザーが任意の検索アプリケーションで問合せを発行すると、問合せ結果を含むヒット・リストが戻されます。ユーザーは、ヒット・リストから参照するドキュメントを選択できます。ヒット・リストには、HTML ドキュメント、ファイル、データベース表のコンテンツ、アーカイブされた電子メール、または Oracle9iAS の項目を含めることができます。Oracle Ultra Search のサンプル検索アプリケーションには、電子メールの読み込みおよび表示を行うための電子メール・ブラウザも組み込まれています。

Oracle Ultra Search Administration ツールおよび Oracle Ultra Search のサンプル検索アプリケーションは、Oracle Ultra Search Middle-Tier コンポーネントのモジュールの一部です。ただし、Oracle Ultra Search Administration ツールは、Oracle Ultra Search のサンプル検索アプリケーションからは独立しています。したがって、セキュリティまたは拡張性を強化するために、他のマシンで構築することができます。

サンプル検索アプリケーションを使用しない場合でも、独自の検索アプリケーションを構築して、Oracle Ultra Search Java Query API を直接起動することができます。この API は Java でコード化されているため、Java サーブレットや JSP など、あらゆる Java ベースのアプリケーションから API のメソッドを起動できます (提供されているサンプル検索アプリケーションは、これに該当します)。また、クロールおよび索引付けの完了した電子メールのレンダリングには、Oracle Ultra Search の Java Email API のメソッドを直接起動することもできます。

JSP のサンプル検索アプリケーション

JSP サンプル検索アプリケーションは、`$ORACLE_HOME/ultrasearch/sample` ディレクトリに含まれています。

JSP の概要

前述のとおり、JSP コードおよび Java API を使用して、Web アプリケーションを作成できます。通常、この Web アプリケーションは、Oracle9iAS などのアプリケーション・サーバーで実行されます。アプリケーション・サーバーは、パフォーマンスおよび拡張性を向上させるために、Oracle データベース・サーバーとは別のマシンで実行されます。Oracle データベース・サーバーは、Oracle Ultra Search の索引を保持します。

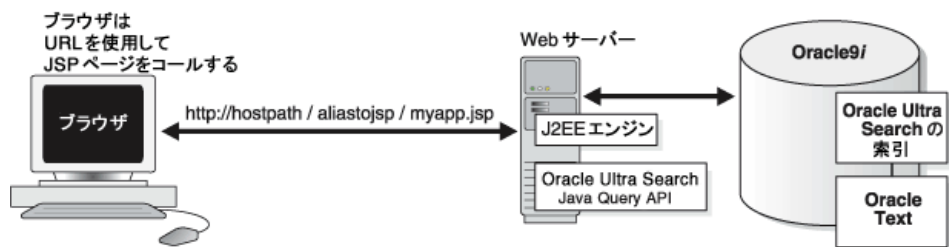
JSP アプリケーションは、実行時に Java サブレットにコンパイルされます。コンパイルされたサブレットは、1 つ以上の Java Virtual Machine (Java VM) で実行されます。JSP アプリケーションは、Oracle JDBC ドライバを使用して Oracle データベース・サーバーと通信します。

すべての Java アプリケーションと同様に、Java Query API および Email API を使用するには、Servlet Engine の CLASSPATH に次のファイルを追加してください。

1. `$ORACLE_HOME/ultrasearch/lib/ultrasearch_query.jar`
2. `$ORACLE_HOME/lib/mail.jar`
3. `$ORACLE_HOME/lib/activation.jar`

図 6-1 に、Web 検索アプリケーションで Oracle Ultra Search Java Query API をコールする方法を示します。

図 6-1 JSP のコール



Oracle Ultra Search の Query タグ・ライブラリ

Oracle Ultra Search では、Java Query API に加えて、検索アプリケーションの開発に代わる JSP タグ・ライブラリが提供されています。Oracle Ultra Search のタグ・ライブラリは、Sun 社の JSP バージョン 1.1 の仕様に基づいて、動的な Java 開発作業と静的な HTML 開発作業を区別します。このため、Java に慣れていない Web 開発者でもアプリケーションに検索機能を組み込むことができます。

Oracle Ultra Search のタグ・ライブラリでは、Java Query API 機能のサブセットを提供しています。カスタム問合せ拡張や URL 発行などの拡張機能は、タグとしては使用できません。タグ・ライブラリの主な機能は次のとおりです。

- 拡張検索フォームをレンダリングするための検索属性、グループ、言語、LOV を取得する機能
- 結果のヒットのセットを反復処理し、結果ページをレンダリングするためのドキュメント属性とプロパティを取得する機能

次の表に、タグ・ライブラリのサマリーを示します。

タグ	説明	属性
instance	Oracle Ultra Search インスタンスとの接続を確立する。	instanceId username password URL dataSourceName tablePagePath emailPagePath filePagePath
showAttributes	拡張検索でこのタグを使用し、使用可能な属性のリストを表示する。	instance locale
showGroups	拡張検索でこのタグを使用し、グループのリストを表示する。	instance locale
showLanguages	拡張検索でこのタグを使用し、インスタンスに定義されている言語のリストを表示する。	instance
showLOV	検索属性に定義されているすべての値を表示する。	instance locale attributeName attributeType

タグ	説明	属性
getResult	検索を実行する。	resultId instance query queryLocale documentLanguage from to boostTerm withCount
fetchAttribute	getResult 内にネストされているタグ。問合せ結果とともにフェッチされる各ドキュメントの属性を指定する。fetchAttribute タグは必要な数に応じてネストできる。	attributeName attributeType
showHitCount	getResult タグで withCount="true" を設定すると、結果にヒット総数が含まれ、showHitCount でその総数を表示できる。	result
showResults	検索結果をレンダリングする。	result instance
showAttributeValue	ドキュメント属性をレンダリングする。	attributeName attributeType

これらのタグの詳細は、後述します。Oracle Ultra Search のタグを使用するためには、次の要件を満たしてください。

- ultrasearch_query.jar ファイルをインストールし、CLASSPATH または Web アプリケーションの WEB-INF/lib ディレクトリに格納します。このファイルは、Oracle Ultra Search をインストールすると提供され、ultrasearch/lib ディレクトリに格納されています。
- タグ・ライブラリ説明ファイルの ultrasearch-taglib.tld がアプリケーションとともに配置され、次のように、JSP ページの taglib ディレクティブの指定場所に存在していることを確認してください。

```
<%@taglib uri="/WEB-INF/ultrasearch-taglib.tld" prefix="US" %>
```

Oracle Ultra Search のタグ・ライブラリ定義（TLD）ファイルは、sample.ear が配置されている \$ORACLE_HOME/ultrasearch/sample/query/WEB-INF/ultrasearch-taglib.tld にあります。META-INF/taglib.tld という名前で ultrasearch_query.jar にも格納されています。

Query タグの説明

ここでは、Oracle Ultra Search の各タグおよびその属性と操作について説明します。ここで示す例には静的な HTML は含まれていませんが、静的な HTML を挿入して出力をフォーマットすることはできます。

<instance> タグ : Oracle Ultra Search インスタンスとの接続

Oracle Ultra Search インスタンスとの接続を確立する。このタグが機能するには、JDBC 接続文字列、スキーマのユーザー名とパスワード、Oracle Ultra Search インスタンス名などの基本パラメータを設定する必要があります。

属性名	説明
instanceId=" 名前 "	このタグによって定義されたインスタンスに名前を付けます。この名前は、他の Oracle Ultra Search タグで検索対象のインスタンスを指定するときに使用します。
username	データベース接続を作成します。
password	データベース接続を作成します。
url	JDBC 接続の作成に使用する URL を取得します。 dataSourceName が指定されている場合、この属性はオプションです。
dataSourceName	JDBC データ・ソースを識別する JNDI 名です。ユーザーは、URL または dataSourceName のいずれかのプロパティを設定する必要があります。URL が指定されている場合、これはオプションです。
instanceName	スキーマ・ユーザーが所有している Oracle Ultra Search インスタンスの名前です。スキーマ・ユーザーが 1 つの Oracle Ultra Search インスタンスのみ所有している場合、これはオプションです。
tablePagePath	データベース表の内容を渡す Web アプリケーションの URL パスです。
emailPagePath	電子メールの内容を渡す Web アプリケーションの URL パスです。
filePath	ファイルの内容を渡す Web アプリケーションの URL パスです。

このタグによって、instanceId プロパティで設定される名前のスクリプト変数を定義します。他のすべてのタグ・プロパティは、oracle.ultrashsearch.query.QueryInstance クラスのプロパティに対応しています。URL または dataSourceName 属性を設定する必要があります。この 2 つは同時には指定できません。

次に、データベースに接続するための URL プロパティの例を示します。

```
<US:instance
  instanceId="mybookstore"
  url="oracle:jdbc:thin:@dbhost:1521:inst1"
  username="scott"
  password="tiger"
  tablePage="../../display.jsp"
  emailPage="../../mail.jsp"
  filePage="../../display.jsp"
/>
```

<iterAttributes> タグ : すべての検索属性の表示

ユーザーが拡張検索を実行するとき、アプリケーションでは、使用可能な属性のリスト、グループのリスト、およびインスタンスで定義されている言語のリストを表示する必要があります。これらを表示するには、ページのレンダリングで使用するスクリプト変数を定義するいくつかの反復タグを使用します。

Oracle Ultra Search の各属性には名前、タイプおよび表示名があり、QueryInstance タグに設定されているロケールに基づいて翻訳されます。属性タイプは、この属性で使用できる演算子、およびユーザー入力の解析方法を判断するために使用します。

属性名	説明
instance=" 名前 "	instance タグで定義されたオブジェクトを参照するための必須属性です。
locale=" ロケール "	このタグを使用してフェッチされる表示名を決定します。

このタグは反復タグです。instance タグ属性で参照されるインスタンス内のすべての検索属性をループ処理します。このタグは、ループ処理ごとに "attribute" という名前のスクリプト変数 (oracle.ultrasearch.query.Attribute オブジェクト) を定義します。また、"displayname" という名前の文字列変数も定義します。この名前はローカライズされた属性名です。

次の例は、"mybookstore" インスタンス内のすべての属性を示します。ここでは英語の表示名を使用しています。

```
<US:iterAttributes instance="mybookstore" locale="<%=Locale.ENGLISH%>" >
  <%= attribute %>
  <%= displayname %>
</US:iterAttributes>
```

<iterGroups> タグ : すべての検索グループの表示

showAttributes タグと同様に、showGroups タグはインスタンス内のすべてのグループを反復処理します。

属性名	説明
instance=" 名前 "	instance タグで定義されたオブジェクトを参照するための必須属性です。
locale=" ロケール "	このタグを使用してフェッチされる表示名を決定します。

このタグは、instance タグ属性で参照されるインスタンス内のすべての検索グループをループ処理します。このタグは、ループ処理ごとに "group" という名前のスクリプト変数 (oracle.ultrashsearch.query.Group オブジェクト) を定義します。また、"displayname" という名前の文字列変数も定義します。この名前はローカライズされたグループ名です。

次の例は、"mybookstore" インスタンス内のすべてのグループを示します。ここでは英語の表示名を使用しています。

```
<US:iterGroups instance="mybookstore" locale="<%=Locale.ENGLISH%" >
<%= group %>
<%= displayname %>
</US:iterGroups >
```

<iterLanguages> タグ : すべての検索言語の表示

showAttributes タグと同様に、showLanguages タグはインスタンス内のすべての言語を反復処理します。各言語は java.util.Locale オブジェクトで定義されるため、その表示名は Oracle Ultra Search では処理されません。したがって、このタグでは表示名のスクリプト変数は定義されません。

属性名	説明
instance=" 名前 "	instance タグで定義されたオブジェクトを参照するための必須属性です。

このタグは反復タグです。instance タグ属性で参照されるインスタンス内のすべての検索言語をループ処理します。このタグは、ループ処理ごとに "language" という名前のスクリプト変数 (java.util.Locale オブジェクト) を定義します。言語の表示名は、Java の getDisplayName() メソッドを使用して、オブジェクト自体のプロパティとして提供されます。

次の例は、"mybookstore" インスタンス内のすべての言語を示します。ここでは英語の表示名を使用しています。

```
<US:iterLanguages instance="mybookstore">
  <%= language %>
  <%= language.getDisplayName (Locale.ENGLISH) %>
</US:iterLanguages >
```

<iterLOV> タグ：検索属性で定義されているすべての値の表示

属性名	説明
instance=" 名前 "	instance タグで定義されたオブジェクトを参照するための必須属性です。
locale=" ロケール "	このタグを使用してフェッチされる表示名を決定します。
attributeName=" 属性名 "	属性の名前で、属性の LOV はこの LOV にフェッチされます。
attributeType="string number date"	属性のタイプで、属性の LOV はこの LOV にフェッチされます。属性名ではインスタンス内の属性を一意に識別できないため、属性のタイプが必要です。

このタグは反復タグです。検索属性 LOV 内のすべての値をループ処理します。このタグは、ループ処理ごとに "value" という名前のスクリプト変数を定義します。これは、属性タイプに応じて java.lang.String、java.util.Date または java.math.BigDecimal のいずれかのオブジェクトになります。また、"displayname" という名前の文字列変数も定義します。この名前はローカライズされた値の表示名です。

次の例は、"mybookstore" インスタンス内にある "Dept" という文字列属性のすべての値を示します。ここでは英語の表示名を使用しています。

```
<US:iterLOV instance="mybookstore" attribute_name="Dept" attribute_type="String" >
  <%= value %>
  <%= displayname %>
</US:iterLOV >
```

問合せの明示

Oracle Ultra Search では、問合せを作成するための一連のクラスをサポートしています。現在、これらのクラスに相当するタグはありません。

<getResult> タグ : 検索の実行

このタグは、oracle.ultrasearch.query.Result タイプのスクリプト変数を定義することで、検索を実行し結果を戻します。

属性名	説明
resultId=" 名前 "	このタグによって生成された結果に名前を付けます。この名前は、他のタグで結果をページにレンダリングするために使用します。
instance=" 名前 "	instance タグで定義されたオブジェクトを参照するための必須属性です。
query="<%= expression %>"	検索対象の問合せオブジェクトを指定します。
queryLocale=" ロケール "	問合せオブジェクトのロケールを指定します。
documentLanguage=" ロケール "	検索対象のドキュメントの言語を指定します。この属性はオプションです。この属性を指定しない場合は、すべての言語が検索対象になります。
from=" 数値 "	最初のヒットの索引を指定します。
to=" 数値 "	最後のヒットの索引を指定します。
boostTerm=" 文字列 "	検索オプションで使用する検索条件を指定するために使用します。この属性はオプションです。
withCount="true false"	結果に合計ヒット数の見積りを含めるかどうかを指定します。この属性はオプションです。この属性を指定しない場合は、withCount=false を指定した場合と同じ動作になります。

<getResult> タグは、oracle.ultrasearch.query.Instance クラスの getResult() メソッドに対応しています。getResult() メソッドがフェッチする属性を指定できることを除き、タグの属性はメソッドのパラメータに直接マップされます。<getResult> タグは、メタデータを選択するために、ネストされた <fetchAttribute> タグを使用する必要があります。

次の例は、フランス語のドキュメントの中の、最初の 20 の英語による問合せドキュメントの検索を示します。

```
<US:getResult
  resultId="searchresult"
  instance="mybookstore"
  query=""
  queryLocale=""
  documentLanguage=""
  from="1" to="20">
</US:getResult>
```

<fetchAttribute> タグ : メタデータの選択

このタグは、<getResult> 内でネストされたタグとして使用します。このタグは、各ドキュメントの属性が問合せ結果とともにフェッチされるかどうかを指定します。各 <getResult> には、ネストした <fetchAttribute> タグを必要数に応じて設定できます。

属性名	説明
attributeName=" 属性名 "	属性の名前で、属性の LOV はこの LOV にフェッチされます。
attributeType="string number date"	属性のタイプで、属性の LOV はこの LOV にフェッチされます。属性名ではインスタンス内の属性を一意に識別できないため、属性のタイプが必要です。

<fetchAttribute> タグごとに、<getResult> タグで起動した getResult() メソッドに渡す属性リストが追加されます。

次の例では、前述の <getResult> タグの検索に加えて、各ドキュメントの title 属性と publication-date 属性をフェッチします。

```
<US:getResult
  resultId="searchresult"
  instance="mybookstore"
  query=""
  queryLocale=""
  documentLanguage=""
  from="1" to="20">
<US:fetchAttribute
  attributeName="title"
  attributeType="string" />
<US:fetchAttribute
  attributeName="publication-date"
  attributeType="date" />
</US:getResult>
```

<showHitCount> タグ : 見積りヒット数の表示

検索の実行後に、その結果をレンダリングする必要があります。<US:getResult> タグで withCount=true を設定すると、結果にヒット総数が含まれ、<showHitCount> タグでその総数を表示できます。

属性名	説明
result=" 名前 "	<US:getResult> タグで指定する resultId を参照します。

このタグは、ヒット数をページに出力します。

次の例では、検索結果のヒット数を表示します。

```
<US:showHitCount result="searchresult" />
```

<iterResult> タグ : 結果のレンダリング

このタグは反復タグです。検索結果内のすべてのドキュメントをループ処理します。

属性名	説明
result=" 名前 "	<US:getResult> タグで指定する resultId を参照します。
instance=" 名前 "	<US:instance> タグで指定する instanceId を参照します。

このタグは、検索結果のすべてのドキュメントをループ処理し、"doc" というスクリプト変数 (oracle.ultrashsearch.query.Document オブジェクト) を定義します。さらに、ネストされた <showAttributeValue> タグを設定して、ドキュメントの属性をレンダリングできます。指定した結果が、指定のインスタンスを検索して取得した結果でない場合は、エラーになります。つまり、結果は、インスタンスを検索した結果であることが必要です。

次の例では、検索結果に含まれるすべてのドキュメントの URL を表示します。

```
<US:iterResult  
  result="searchresult"  
  instance="mybookstore">  
</US:iterResult>
```

<showAttributeValue> タグ : ドキュメント属性のレンダリング

このタグは、<US:iterResult> タグ内のドキュメントの属性を表示します。

属性名	説明
attributeName=" 属性名 "	ドキュメント属性の名前です。
attributeType="string number date"	ドキュメント属性のタイプです。属性名ではインスタンス内の属性を一意に識別できないため、属性のタイプが必要です。
default=" デフォルト文字列 "	この属性に対してドキュメントで値を指定しなかった場合に出力される値です。これは、ドキュメントにタイトルが指定されていない場合に便利です。文字列 "No Title" は、デフォルト値として表示されます。

このタグは、ドキュメント属性値を参照し、その値をページにレンダリングします。検索結果の一部として属性がフェッチされていない場合、ページには何も出力されません。

次の例では、検索結果に含まれるすべてのドキュメントのタイトルと発行日を表示します。

```
<US:iterResult
result="searchresult"
instance="mybookstore">
<US:showAttributeValue attributeName="title" attributeType="string" default="No
Title" />
<US:showAttributeValue attributeName="publication-date" attributeType="date" />
</US:iterResult>
```

問合せ構文拡張のカスタマイズ

Oracle Ultra Search は、Oracle Text エンジンを使用して、ドキュメントの索引付けおよび検索を行います。エンド・ユーザーが問合せ文字列を指定すると、Oracle Ultra Search は、その文字列を Oracle Text の問合せ式に変換します。この処理を「問合せ構文拡張」と呼びます。

Oracle Ultra Search では、問合せ構文拡張を独自に実装して使用するようカスタマイズできます。以前のリリースでは、デフォルトの問合せ構文拡張の実装が WK_QUERYEXP PL/SQL パッケージに含まれていました。

Contains 問合せによって、インターネット検索エンジンに類似した問合せ構文を指定できます。この構文は、'title' (タイトル) StringAttribute でユーザーの問合せと一致するドキュメントを高くスコア付けします。Contains の構文は、ドキュメントのコンテンツに対して使用される場合も StringAttributes に対して使用される場合も同じです。

この構文は、Contains 問合せをサブクラス化し、expand() メソッドを独自の実装でオーバーライドすることで、カスタマイズできます。実際に、Query インタフェースを実装すると、提供されている Contains 問合せを無視できます。これは、Query API が、Query インタフェースを実装するすべてのオブジェクトを受け入れるためです。

ここでは、問合せ構文拡張の実装を、使用している環境向けにカスタマイズする方法を説明します。

デフォルトの問合せ構文拡張の実装

デフォルトの問合せ構文拡張の実装は、次の方法に直接影響します。

- エンド・ユーザーによる問合せ文字列の入力方法（「エンド・ユーザーの問合せ構文」を参照）。
- 問合せに一致するドキュメントへのスコア付けの方法（「スコア付け」を参照）。
- エンド・ユーザーが入力した問合せ文字列から Oracle Text の問合せ文字列への変換方法（「拡張ルール」を参照）。

エンド・ユーザーの問合せ構文

デフォルトの問合せ構文拡張の実装で定義されるエンド・ユーザーの問合せ構文は、Web の検索エンジンで使用されている標準テキスト問合せ構文に似ています。

- トークン: トークンは、二重引用符 (") で囲まれた文字列です。1 つの単語または句です。
- 演算子: デフォルトの実装では、3 つの演算子が定義されます。[+]、[-] および [*] です。これらの演算子は、デフォルトの実装で定義されています。カスタム実装によって、これらの演算子を変更します。

プラス演算子 [+]。その後に続くトークンが、検索結果に含まれるすべてのドキュメントに存在することを指定します。

マイナス演算子 [-]。その後に続くトークンが、検索結果に含まれるすべてのドキュメントに存在しないことを指定します。

アスタリスク [*]。検索のワイルド・カードを指定します。0 (ゼロ) または 1 つ以上の文字に相当します。アスタリスクで始まるトークンは、無視されます。アスタリスクは、トークンの末尾 (右側) または途中のみに指定できます。たとえば、「hel*o」および「hell*」では正しくアスタリスクを使用していますが、「*ello」は受け入れられません。

次に、Oracle Ultra Search のエンド・ユーザーの問合せ構文に関するルールを示します。

注意: エンド・ユーザーのすべての問合せ文字列を、大カッコで囲んで示します。たとえば、エンド・ユーザーの問合せ文字列 Oracle Applications は、[Oracle Applications] と記述します。

ルール	説明
単一の単語の検索	1 つの単語を入力すると、その単語を含むドキュメントを検索する。 たとえば、[Oracle] は、「Oracle」という単語を含むすべてのドキュメントを検索する。
複数の単語の検索	2 つ以上の単語を入力すると、順序に関係なく、それらの単語のいずれかを含むドキュメントを検索する。 たとえば、[Oracle Applications] の場合は、「Oracle」、「Applications」または「Oracle Applications」を含むドキュメントを検索する。

ルール	説明
強制的な包含 [+]	<p>単語の前に [+] を付けると、その単語は、一致したすべてのドキュメントに存在することを示す。</p> <p>たとえば、[Oracle+Applications] の場合は、「Applications」という単語を含むドキュメントのみを検索する。</p> <p>注意: 複数の単語の検索では、一番最初のトークンを含むすべてのトークンの前に [+] を付けることができる。</p>
強制的な排除 [-]	<p>単語の前に [-] を付けると、その単語は、一致したすべてのドキュメントに存在しないことを示す。</p> <p>たとえば、[Oracle-Applications] の場合は、「Applications」という単語を含まないドキュメントのみを検索する。</p> <p>注意: 複数の単語の検索では、一番最初のトークン以外のすべてのトークンの前に [-] を付けることができる。</p>
句の一致 ["..."]	<p>一連の単語を二重引用符で囲むと、句を正確に含むドキュメントのみを検索する。</p> <p>たとえば、["Oracle Applications"] の場合は、「Oracle Applications」という文字列を含むドキュメントのみを検索する。</p>
ワイルドカードの一致 [*]	<p>単語の右側に [*] を付けると、アスタリスクの左側が一致する結果を戻す。</p> <p>たとえば、[Ora*] という文字列の場合は、「Oracle」や「Orator」など、「Ora」で始まるすべての単語を含むドキュメントを検索する。単語の途中にアスタリスクを挿入することもできる。たとえば、[A*e] という文字列の場合、「Apple」、「Ate」、「Ape」などの単語を含むドキュメントを検索する。ワイルドカードの一致では、より多くの計算処理が必要なため、他の問合せより低速である。</p>

クラスのスコア付け

エンド・ユーザーの問合せ文字列に応じてドキュメントを一致させる方法には、3通りあります。これらの3つの方法を「クラスのスコア付け」といいます。ドキュメントが高度なクラス要件を満たす場合、そのドキュメントは高くスコア付けされ、高い位置にランク付けされます。クラスごとに、ドキュメントがクラスのスコア付け条件にどの程度一致するかによってランク付けが異なります。

クラス1は、最も高く重み付けされたクラスです。ドキュメント内で完全に一致した文字列の数によって、スコアが決まります。一致した句が多いドキュメントほど、一致した句が少ないドキュメントより高くスコア付けされます。

クラス2は、その次に高く重み付けされたクラスです。このクラスでは、ドキュメントに現れる各トークンが接近しているほど、高くスコア付けされます。たとえば、エンド・ユーザーの問合せ文字列が [Oracle Applications Financials] で、3つのドキュメントが検出さ

れたとします。3つのドキュメントとも、「Oracle Applications Financials」という文字列は含みません。ただし、ドキュメント X には、同じ文にあって他の単語で分離される「Oracle」、「Applications」および「Financials」の3つのトークンが含まれます。ドキュメント Y には、同じ段落で異なる文にそれぞれのトークンが含まれます。ドキュメント Z には3つのトークンが含まれますが、各トークンは異なる段落にあります。この場合は、各トークンが最も接近しているドキュメント X が最も高くスコア付けされます。同様に、ドキュメント Y はドキュメント Z より高くスコア付けされます。

クラス 3 は、重み付けが最も低いクラスです。トークンを多く含むドキュメントは、高くスコア付けされます。たとえば、エンド・ユーザーの問合せ文字列が [Oracle Applications Financials] で、3つのドキュメントが検出されたとします。ドキュメント X には、3つのトークンすべてが含まれます。ドキュメント Y には、「Oracle」および「Applications」のみが含まれます。ドキュメント Z には、「Oracle」のみが含まれます。この場合は、ドキュメント X はドキュメント Y より高くスコア付けされます。同様に、ドキュメント Y はドキュメント Z より高くスコア付けされます。

拡張ルール

前述のとおり、エンド・ユーザーの問合せは Oracle Text の問合せに拡張されます。拡張された問合せ文字列のルールは、BNF 表記に準拠します。また、これらのルールは、デフォルトの問合せ構文拡張の実装として Oracle Ultra Search で使用されます。

次に、拡張検索を定義するルールを示します。

<expanded query> ::= (<expression> within <title section>)*2, <expression>

<expression> ::= <generic query expression> | <simple query expression>

<generic query expression> ::= (([<plus expression>*100 &]) (<main expression>)) [<minus expression>]

<simple query expression> ::= (<phrase expression>)*2, (<main expression>)

<main expression> ::= (<near expression>)*2, (<accum expression>)

前述のルールで使用されている項目およびその意味は、次のとおりです。

<plus expression> は、プラス・トークンすべての AND 式です。

<minus expression> は、すべてのマイナス・トークンの NOT 式です。

<phrase expression> は、<main expression> 内のすべてのトークンで構成された PHRASE です。

<near expression> は、マイナス・トークンを除くすべてのトークンの NEAR 式です。

<accum expression> は、マイナス・トークンを除くすべてのトークンの ACCUMULATE 式です。

<simple query expression> は、エンド・ユーザーの問合せに複数のトークンが含まれ、演算子や二重引用符が含まれない場合にのみ使用されます。

それ以外の場合は、<generic query expression> が使用されます。

プラス・トークン、マイナス・トークン以外のトークンがない場合、<plus expression> および <accum expression> は排除されます。

ルールの適用例

次に、デフォルトの問合せ構文拡張の実装によって、エンド・ユーザーの問合せ文字列を Oracle Text 互換の問合せ文字列に変換する方法を示します。

エンド・ユーザーの 問合せ文字列	Oracle Text で認識可能な拡張検索文字列
[Oracle]	((({Oracle})) within TITLE__31)*2,(({Oracle}))
[Oracle + Applications]	(((((Applications))*10)*10&(({Oracle});Applications))*2,({Oracle},Applications)) within TITLE__31)*2,(((Applications))*10)*10&(({Oracle};Applications))*2,({Oracle},Applications)))
[Oracle - Applications]	(((({Oracle})~Applications)) within TITLE__31)*2,(({Oracle})~Applications))
["Oracle Applications"]	((({Oracle Applications}) within TITLE__31)*2,({Oracle Applications}))
[Ora*]	((((Ora%)) within TITLE__31)*2,((Ora%)))
[Oracle Applications]	(((((Oracle Applications))*2,(({Oracle};Applications))*2,({Oracle},Application s))) within TITLE__31)*2,(({Oracle Applications))*2,(({Oracle};Applications))*2,({Oracle},Applications)))

ルールのカスタマイズ

独自の問合せ構文拡張の定義および実装によって、企業の目的に合うように拡張をカスタマイズします。そのためには、**Oracle Text** の問合せ要件を理解する必要があります。**Oracle Text** の問合せについては、このドキュメントでは詳しく説明していません。

関連項目：

- 『Oracle Text アプリケーション開発者ガイド』
- 『Oracle Text リファレンス』

Oracle Ultra Search をカスタマイズして独自の問合せ構文拡張の実装を使用するには、**Contains** 問合せを使用します。この問合せは、そのコンテンツまたはその文字列属性内にいくつかのテキストを含むドキュメントを検索します。**Contains** 問合せは、日付または数値属性には適用されません。属性が指定されていない場合、**Contains** は、ドキュメントの属性ではなく、ドキュメントのコンテンツ上で動作します。ドキュメントのタイトル属性内の一致は、ドキュメントのコンテンツ内の一致より高くスコア付けされます。

コンストラクタ

- **Contains(StringAttribute, String, InstanceMetaData)**

```
public Contains(StringAttribute att, java.lang.String val, InstanceMetaData instmd)
```

このコンストラクタには、文字列属性に関する問合せが含まれます。

- **Contains(String, InstanceMetaData)**

```
public Contains(java.lang.String val, InstanceMetaData instmd)
```

このコンストラクタには、ドキュメントのコンテンツに関する問合せが含まれます。

メソッド

- **compile()**

```
public java.lang.String compile()
```

このメソッドは、問合せ文字列にコンパイルされます。

指定方法：

Query インタフェースに **compile()** を指定

戻り値：

この問合せを表す問合せ文字列

■ `expand(StringAttribute, String, InstanceMetaData)`

```
public java.lang.String expand(StringAttribute att, java.lang.String str,  
    InstanceMetaData instmd)
```

このメソッドは、ユーザーの属性 `Contains` 問合せ文字列をテキスト問合せに変換します。

パラメータ：

`att` - 文字列属性

`str` - メイン問合せ文字列

`instmd` - `InstanceMetaData` オブジェクト

戻り値：

変換された Oracle Text 問合せ文字列 (`Contains` 句)

■ `expand(String, InstanceMetaData)`

```
public java.lang.String expand(java.lang.String str, InstanceMetaData instmd)
```

このメソッドは、ユーザーの問合せ文字列をテキスト問合せに変換します。

パラメータ：

`str` - メイン問合せ文字列

`instmd` - `InstanceMetaData` オブジェクト

戻り値：

変換された Oracle Text 問合せ文字列 (`Contains` 句)

Web クロール・プロセスのチューニング

Oracle Ultra Search Crawler は、組織のイントラネット内の Web サイトに関する情報を検出する強力なツールです。この機能は、特に Web クロールに適しています。その他のデータ・ソースについては、ユーザーが認識していない可能性のある他のドキュメントへのリンクは、Crawler で追跡しないように定義されています。

Web クロールの方針

Web クロールの方針は、組織内の他のイントラネット・サイトへのリンクを含む主要なサイトのみを認識するだけという簡単な方針にできます。この方針は、これらのサイトを索引付けせずにクロールするとテストできます。初回のクロール後、イントラネットにあるホストについて、どう対処すればよいか見当がついてきます。これによって、各 Web ソースを定義でき、各サイトのクロールおよび索引付けが簡単になります。

ただし、実際には、イントラネットの検出とクロール・プロセスはインタラクティブで、クロール結果の定期的な分析と、クロール・プロセスをある程度指示するクロール・パラメータの変更を特徴としています。たとえば、**Crawler** がある Web ホストのクロールに数日を費やしていた場合、そのホストのクロールを除外するか、クロールの深さを制限できます。

クロール・プロセスの監視

次の方法を組み合わせて、クロール・プロセスを監視します。

- Administration ツールを使用して、スケジュール・ステータスを監視します。
- Administration ツールを使用して、スケジュールの進行状況をリアルタイムに監視します。
- Administration ツールを使用して、**Crawler** 統計を監視します。
- 現行スケジュールのログ・ファイルを監視します。

URL ルーピング

URL ルーピングとは、同一ドキュメントを指す一意の URL が、なんらかの理由により多数存在する場合のことです。1つのサイトに大量のページがあり、それぞれが同一サイト内にある他のページへリンクされていたとしても、**Crawler** は、最終的にはそのサイト内のすべてのドキュメントの分析を完了するため、このような状況が問題になることはありません。

ただし、Web サーバーには、生成された URL にパラメータを付加することで、リクエストされた情報を追跡するものもあります。このような Web サーバーは、同一ドキュメントを指す一意の URL を多数生成することがあります。たとえば、`http://mycompany.com/somedocument.html?p_origin_page=10` は、同一ドキュメントを `http://mycompany.com/somedocument.html?p_origin_page=13` として参照する場合がありますが、これらは参照するページが異なるため、`p_origin_page` のパラメータが異なります。多数のパラメータが指定されており、参照するリンクも多数ある場合、1つの一意のドキュメントに対して参照元のリンクが無数になります。このような場合に、URL ルーピングが発生する可能性があります。

Oracle Ultra Search Administration ツールで **Crawler** 統計を監視し、最も頻繁にクロールされている URL および Web サーバーの見当をつけます。特定のサイトや URL に、膨大な数の URL アクセスがあることを発見した場合は、次のいずれかを実行してください。

- **Web サーバーの除外:** 該当するホストでの **Crawler** による **URL** のクロールを停止します (ホストの固有ポートを除外するようには制限できません)。
- **クロールの深さの制限:** **Crawler** が追跡する参照リンクのレベル数を制限します。特定のホストで **URL** ルーピングが影響していることを発見した場合は、画面から調査を行い、そのサイトのリーフ・ページの深さを見積ります。リーフ・ページとは、他のページへのリンクがないページのことです。一般的なガイドラインとしては、リーフ・ページの深さに 3 を加えた値をクロールの深さに設定します。

「**Crawler**」タブでパラメータを変更した後は、必ず **Crawler** を再起動してください。設定の変更は、**Crawler** を再起動するまで反映されません。

問合せパフォーマンスのチューニング

この付録では、Oracle Ultra Search の問合せパフォーマンスを改善する方法をいくつか説明します。問合せのパフォーマンスは通常、応答時間とスループットの影響を受けます。

1. DB_CACHE_SIZE パラメータのチューニング

データベースのバッファ・キャッシュには、データ・ファイルから頻繁にアクセスするデータが保存されています。このバッファ・キャッシュを効率よく使用することによって、Oracle Ultra Search の問合せパフォーマンスを改善できます。キャッシュ・サイズは DB_CACHE_SIZE 初期化パラメータによって制御されます。

関連項目： このパラメータのチューニング方法については、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

2. 索引の最適化

Crawler による大規模な更新の後には、Oracle Ultra Search の索引を最適化してください。この最適化は、索引の最適化を定期的にスケジューリングすることで実行します。索引の最適化中は、問合せパフォーマンスが大幅に低下するため、最適化はピーク時以外にスケジュールしてください。

関連項目： 5-30 ページ「索引最適化」

3. トークンに基づく索引の最適化

Oracle Ultra Search の索引は、頻繁に検索されるトークンに基づいて最適化してください。問合せをログに記録するには、Administration ツールを使用して問合せ統計の収集をオンにします。頻繁に検索されるトークンをトークン・モードで CTX_DDL.OPTIMIZE_INDEX に渡すことができます。Oracle Ultra Search の索引名は WK\$DOC_PATH_IDX です。

関連項目： OPTIMIZE_INDEX の詳細は、『Oracle Text リファレンス』を参照してください。

4. 問合せ拡張の簡素化

検索の応答時間は、使用された Oracle Text 問合せ文字列によって直接影響を受けます。Oracle Ultra Search には、ユーザー入力を Oracle Text の問合せに拡張するデフォルトの機能がありますが、拡張が簡単であれば検索時間を大幅に削減できます。

関連項目：

- 6-28 ページ「[問合せ構文拡張のカスタマイズ](#)」
- oracle.ultrashsearch.query の Query インタフェースについては、Oracle Ultra Search Javadoc を参照してください。

5. 共有プールのサイズ指定

共有プールには、ライブラリ・キャッシュとディクショナリ・キャッシュが格納されています。ライブラリ・キャッシュには、最近実行された SQL と PL/SQL コードが格納されています。データ・ディクショナリ・キャッシュやライブラリ・キャッシュでのキャッシュ・ミスは、バッファ・キャッシュでのミスに比較してはるかにコストがかかります。このため、共有プールには、頻繁に使用されるデータが確実にキャッシュされるように、サイズを指定する必要があります。共有プールのサイズは、SHARED_POOL_SIZE 初期化パラメータによって制御されます。

関連項目： このパラメータのチューニング方法については、『Oracle9i データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』を参照してください。

6. JDBC 接続プーリングの定義

Oracle Ultra Search Middle-Tier は、JDBC を介してデータベースに接続しています。JDBC での接続の作成には時間がかかるため、問合せの応答時間を削減するためにオープン接続のプールが使用されます。Oracle9iAS では、OC4J によってアプリケーション用の接続プールを管理できます。

最小サイズ、最大サイズおよびプールの割当アルゴリズムは、OC4J の data-sources.xml 構成ファイルで指定できます。

次に、最小 2、最大 30 のオープン接続数を持つデータ・ソース定義の例を示します。各接続は 30 秒の非アクティブ後にクローズし、新規の接続が負荷に従って動的に作成されます。その他のキャッシュ・スキームには、FIXED_WAIT_SCHEME および FIXED_RETURN_NULL_SCHEME があります。

注意： DYNAMIC_SCHEME = 1、FIXED_WAIT_SCHEME = 2、および
FIXED_RETURN_NULL_SCHEME = 3

```
<data-source
  class="oracle.jdbc.pool.OracleConnectionCacheImpl"
  name="UltraSearchDS"
  location="jdbc/UltraSearchPooledDS"
  username="user"
  password="pass"
  url="jdbc:oracle:thin:@hostname:1521:oracle_sid"
  min-connections="2"
  max-connections="30"
  inactivity-timeout="30" >
  <property name="cacheScheme" value="1" />
</data-source>
```

7. 問合せパッケージのメモリー内への確保

頻繁に使用するパッケージは、共有メモリー・プール内に確保してください。確保されたパッケージは、プールの使用状況やアクセス頻度に関係なく、メモリー内に残ります。パッケージの確保には、提供されているパッケージの DBMS_SHARED_POOL を使用できます。

Oracle Ultra Search の問合せに使用する PL/SQL パッケージは、WKSYS.WK_QRY です。

関連項目：『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』

リモート Crawler の使用方法

Oracle Ultra Search のリモート Crawler がない場合は、Oracle9i データベースと同じホストで Oracle Ultra Search Crawler を実行する必要があります。大規模なデータ・セットの場合は、Oracle9i データベースとは別の 1 つ以上のホストで Oracle Ultra Search Crawler を実行すると、パフォーマンスが向上します。Oracle Ultra Search Crawler は Pure Java アプリケーションで、JDBC を介して Oracle9i データベースと通信します。

Oracle Ultra Search のリモート Crawler のインスタンスは、常に Oracle9i データベースによって起動されます。Oracle9i データベースは、Java Remote Method Invocation (RMI) を使用してリモート Crawler ホストと通信します。そのため、各リモート・ホストでは、RMI レジストリおよび RMI デーモンを実行する必要があります。

リモート Crawler を Oracle9i データベースから起動することによって、Oracle9i データベースの高可用性を利用することができます。Oracle Ultra Search のスケジューリング・メカニズムは Oracle9i データベース内で動作するため、データベースの高可用性が自動的に反映されます。

注意： デフォルトでは、RMI はデータを暗号化しないでネットワークに送信します。リモート Crawler を使用したクロールには、潜在的なセキュリティ上のリスクがあります。企業内の悪意のあるエンティティがネットワーク内で行き交うパケットをリスニングし、Oracle Ultra Search インスタンス・スキーマとパスワードをスティーリングする可能性があります。このセキュリティ上のリスクを避けるには、リモート Crawler 機能は使用しないでください。

拡張性およびロード・バランシング

Oracle Ultra Search スケジュールは、それぞれ 1 つの Crawler に関連付けられます。Crawler は、Oracle9i データベース・ホストまたはリモート・ホストでローカルに実行できます。実行できるスケジュールの数に制限はありません。同様に、実行できるリモート Crawler ホストの数に制限はありません。ただし、各リモート Crawler ホストに、Oracle Ultra Search Middle-Tier コンポーネントがインストールされている必要があります。

いくつかのリモート Crawler ホストを使用し、スケジュールを特定のホストに割り当てることによって、クロール・プロセス全体の拡張性およびロード・バランシングが実現します。

起動方法

起動には、Java RMI テクノLOGYを使用します。

1. クロール・スケジュールがアクティブになると、Oracle Ultra Search スケジューラは、データベース・ホスト上の個別のプロセスとして Java プログラムを起動します。この Java プログラムは、ActivationClient と呼ばれます。
2. このプログラムは、ポート 1098 および 1099 の標準 RMI レジストリおよび RMI デーモンを介して、リモート Crawler ホストに接続します。正常に接続すると、ActivationClient は、リモート・ホストで実行されている Java オブジェクトのリモート参照を受信します。このリモート Java オブジェクトは、ActivatableCrawlerLauncher と呼ばれます。
3. ActivationClient は、ActivatableCrawlerLauncher を使用してリモート・ホスト上の Oracle Ultra Search Crawler を起動します。Oracle Ultra Search Crawler は、ActivatableCrawlerLauncher によってリモート・ホスト上の個別の Java プロセスとして起動します。

インストールと構成の順序

1. Oracle9i データベースに Oracle Ultra Search Server コンポーネントがインストールされていること、1 つ以上の Web サーバー・ホストに Oracle Ultra Search Middle-Tier コンポーネントがインストールされていること、そして、すべてのリモート Crawler ホストに Oracle Ultra Search Middle-Tier コンポーネントがインストールされていることを確認します。

関連項目： 第 2 章「Oracle Ultra Search のインストールと構成」

2. データベース・ホスト上の次の共通リソースをエクスポートします。
 - 一時ディレクトリ
 - ログ・ディレクトリ
 - メール・アーカイブ・ディレクトリ (Oracle Ultra Search メーリング・リスト機能を使用している場合)

これらのリソースは、ネットワーク上のすべてのリモート Crawler インスタンスからアクセスできるディレクトリである必要があります。任意の機能を使用して、これらのリソースをリモート Crawler ホストと共有します。

リモート Crawler コードは Pure Java です。そのため、プラットフォームから独立しています。たとえば、Oracle Ultra Search インストールが 4 つのホストで構成されています。各ホストがそれぞれ、Oracle Ultra Search Server コンポーネントがインストールされ、Solaris を実行しているデータベース・サーバー (ホスト X)、Windows NT 上で実行されているリモート Crawler ホスト (ホスト Y1)、Solaris 上で実行されているリモート Crawler ホスト (ホスト Y2)、そして Linux 上で実行されているリモート Crawler ホスト (ホスト Y3) であるとしします。

この例では、UNIX の `export` コマンドを使用して、ホスト X 上の共有ディレクトリをエクスポートします。次に、ホスト Y2 および Y3 上で UNIX の `mount` コマンドを使用して、エクスポートされたディレクトリをマウントします。ホスト Y1 の場合は、Windows NT 用のサード・パーティの NFS クライアントを購入し、そのクライアントを使用して共有ディレクトリをマウントする必要があります (**注意**: ホスト X が Linux サーバーの場合は、Samba 共有ディレクトリを作成して Windows NT 上にマウントできるため、サード・パーティのソフトウェアは不要です)。

3. Oracle Ultra Search Administration ツールを使用して、リモート Crawler を構成します。

定義した共有 Crawler リソース用のマウント・ポイントをすべて手動で入力して、リモート Crawler プロファイルを編集します。リモート Crawler プロファイルを編集するには、「Crawler」タブの「リモート Crawler プロファイル」ページに進み、編集するリモート Crawler プロファイルの「編集」アイコンをクリックします。次のパラメータの値を指定します。

- リモート Crawler によって参照される一時ディレクトリ・パス用のマウント・ポイント
- リモート Crawler によって参照されるログ・ディレクトリ・パス用のマウント・ポイント
- リモート Crawler によって参照されるメール・アーカイブ・パス用のマウント・ポイント（Oracle Ultra Search メーリング・リスト機能を使用している場合）

さらに、クロールを始める前に、次の Crawler パラメータを指定する必要があります。

- リモート Crawler がドキュメント収集用に使用する Crawler スレッドの数
 - リモート Crawler ホスト上のプロセッサの数
4. Oracle Ultra Search Administration ツールを使用して、Crawler の構成を完了します。

構成が必要なパラメータの最小セットは、次のとおりです。

- シード URL
- Web プロキシ
- スケジュール

各スケジュールは、リモート Crawler またはローカル Crawler に割り当てる必要があります（ローカル Crawler は、Oracle9i のローカル・データベース・ホスト上で実行される Crawler です）。スケジュールをリモート Crawler ホストまたはローカル・データベース・ホストに割り当てるには、「[スケジュール](#)」タブにあるスケジュールのホスト名をクリックします。

各スケジュールに対するリモート Crawler 機能を停止し、スケジュールを使用して特定のリモート Crawler ホストのかわりにローカル・データベース・ホスト上の Crawler を起動することもできます。リモート Crawler 機能を停止するには、「同期スケジュール」ページのスケジュールのホスト名をクリックします。リモート Crawler ホストが選択されている場合は、リモート Crawler を使用可能または使用禁止にできます。

関連項目： 第 5 章「Oracle Ultra Search Administration ツールについて」

5. 各リモート Crawler ホスト上で、RMI レジストリおよび RMI デーモンを起動します。

起動するには、`$ORACLE_HOME/tools/remotecrawler/scripts/<operating system>` のスクリプトを使用します。

- リモート Crawler が UNIX プラットフォーム上で実行されている場合は、Bourne シェル・スクリプト
`$ORACLE_HOME/tools/remotecrawler/scripts/unix/runall.sh` を使用します。
- リモート Crawler が Windows NT ホスト上で実行されている場合は、
`%ORACLE_HOME%\tools\remotecrawler\scripts\winnt\runall.bat` ファイルを実行します。

runall.sh スクリプトと runall.bat スクリプトによって、次のタスクが順に実行されます。

- define_env が起動され、必要な環境変数が定義されます。
- runregistry が起動され、RMI レジストリが起動されます。
- runrmid が起動され、RMI デーモンが起動されます。
- register_stub が起動され、必要な Java クラスが RMI サブシステムに登録されます。

runregistry、runrmid および register_stub は、個別に起動できます。ただし、最初に define_env を起動して必要な環境変数を定義する必要があります。

6. Administration ツールからリモート Crawler を起動し、実行されていることを確認します。

スケジュールのステータスは、「[スケジュール](#)」タブにリストされます。リモート Crawler の起動プロセスで障害が発生した場合は、ステータスが LAUNCHING から FAILED に変更されるまで最大 90 秒かかります。

スケジュールのステータスを表示するには、スケジュール・リストの Crawler のステータスをクリックします。障害が発生したイベントの詳細を表示するには、スケジュールのステータスをクリックします。詳細なスケジュールのステータスが表示されます。

要件のいずれかが満たされていない場合、リモート Crawler は起動できません。次に例を示します。

- RMI レジストリが、各リモート・ホストのポート 1099 で実行またはリスニングされていない場合
- RMI デーモンが、各リモート・ホストのポート 1098 で実行またはリスニングされていない場合
- 必要な Java オブジェクトが、各 RMI レジストリに正常に登録されていない場合

リモート Crawler の起動後は、次のいずれかの方法で、リモート Crawler が実行されていることを確認してください。

- リモート Crawler ホスト上のアクティブな Java プロセスを確認します。
 リモート Crawler がリモート Crawler ホスト上で実行されていることを確認する簡単な方法は、UNIX システム上で ps などのオペレーティング・システムのコマンドを使用することです。アクティブな Java プロセスを検索する必要があります。
- スケジュール・ログ・ファイルの内容を監視します。
 リモート Crawler が正常に実行されている場合は、スケジュール・ログ・ファイルの変更内容を定期的に確認する必要があります。スケジュール・ログ・ファイルは共有ログ・ディレクトリ内にあります。

表データ・ソースの同期化

Oracle Ultra Search Crawler は、Oracle Ultra Search がインストールされている Oracle9i のメイン・データベース・インスタンスにあるデータベース表をクロールします。また、リモート・データベースが Oracle9i のメイン・データベースにリンクされている場合は、そのリモート・データベースをクロールすることもできます。リモート・データベースは、データベース・リンクを経由して、Oracle9i のメイン・インスタンスにリンクされます。

関連項目： データベース・リンクの作成方法については、『Oracle9i データベース管理者ガイド』を参照してください。

Oracle Ultra Search には、表ソースのクロールを最適化するためのロギング機能があります。このロギング機能を使用すると、新たに更新されたドキュメントのみがクロール・プロセスで再検査されます。ソース・データベースが Oracle データベースではない場合、この機能を使用するには、一連の処理を行う必要があります。

Oracle データベースのクロールの同期化

ログ表とログ・トリガーを作成する前に、Oracle Ultra Search インスタンス・スキーマに 'CREATE ANY TABLE' と 'CREATE ANY TRIGGER' のシステム権限があることを確認してください。Oracle データベースにある表の場合は、データ定義言語（DDL）文を指定して次のものを作成します。

ログ表の作成

ログ表には、実表に対して行われた変更が保存されます。Oracle Ultra Search Crawler では、変更情報を使用して、再クロールする必要がある行を特定します。たとえば、Oracle Ultra Search が生成したログ表には、「WK\$LOG」などの名前が付きます。

ログ表の構造のルールは、次のとおりです。

- 1. 実表のすべての主キー列について、ログ表に列を作成します。
- 2. 実表は、主キー列を最大 8 列持つことができます。
- 3. 主キー列に対応するログ表の各列の名前には、「Kx」（x は、1 から 8 の数字）を指定する必要があります。
- 4. 主キー列に対応するログ表の各列の型は、VARCHAR2（1000）です。
- 5. CHAR（1）型を持つ「mark」という名前の列は、1 つである必要があります。
- 6. 「mark」という列のデフォルト値は「F」です。

例

実表 employees の構造は、次のとおりです。

列名	列型
ID	NUMBER
NAME	VARCHAR2（200）
ADDRESS	VARCHAR2（400）
TELEPHONE	VARCHAR2（10）
USERNAME	VARCHAR2（24）

employees 表の主キーが ID 列と NAME 列で構成されている場合、ログ表 WK\$LOG（表名は作成中に生成されます）の構造は、次のようになります。

列名	列型
K1	NUMBER
K2	VARCHAR2(200)

ログ表を作成する SQL 文は、次のとおりです。

```
CREATE TABLE WK$LOG(  
  K1 VARCHAR2(1000),  
  K2 VARCHAR2(1000),  
  MARK CHAR(1) default 'F')
```

ログ・トリガーの作成

INSERT トリガー、UPDATE トリガーおよび DELETE トリガーが作成されます。Oracle トリガーは、次のように定義されます。

INSERT トリガー文

実表 employees に行が挿入されるたびに、INSERT トリガーによって、1 行がログ表に挿入されます。ログ表の行には、K1 列と K2 列に、id と name の新しい値がそれぞれ記録されます。「F」が mark 列に挿入され、この行には処理が必要であることが Crawler に示されます。

次に例を示します。

```
CREATE OR REPLACE TRIGGER wk$ins  
  AFTER INSERT ON employees  
  FOR EACH ROW  
  
BEGIN  
  INSERT INTO WK$LOG(k1,k2,mark)  
    VALUES(:new.id,:new.name,'F');  
END;
```

UPDATE トリガー文

実表 `employees` で行が更新されるたびに、UPDATE トリガーによって、2 行がログ表に挿入されます。ログ表の最初の行には、K1 列と K2 列に、`id` と `name` の更新前の値がそれぞれ記録されます。「F」が `mark` 列に挿入され、この行には処理が必要であることが **Crawler** に示されます。ログ表の 2 番目の行には、K1 列と K2 列に、`id` と `name` の更新後の値がそれぞれ記録されます。

次に例を示します。

```
CREATE OR REPLACE TRIGGER wk$upd
AFTER UPDATE ON employees
FOR EACH ROW

BEGIN
    INSERT INTO WK$LOG(k1,k2,mark)
        VALUES(:old.id,:old.name,'F');
    INSERT INTO WK$LOG(k1,mark)
        VALUES(:new.id,:new.name,'F');
END;
```

DELETE トリガー

実表 `employees` から行が削除されるたびに、DELETE トリガーによって、1 行がログ表に挿入されます。ログ表の行には、K1 列と K2 列に、`id` と `name` の削除前の値がそれぞれ記録されます。「F」が `mark` 列に挿入され、この行には処理が必要であることが **Crawler** に示されます。

次に例を示します。

```
CREATE OR REPLACE TRIGGER wk$del
AFTER DELETE ON employees
FOR EACH ROW

BEGIN
    INSERT INTO WK$LOG(k1,k2,mark)
        VALUES(:old.id,:old.name,'F');
END;
```

Oracle 以外のデータベースのクロールの同期化

Oracle 以外のリモート・データベースにある表の場合は、次の手順を実行する必要があります。

1. ログ表を手動で作成します。このログ表は、前述のログ表のルールに従う必要があります。また、実表と同じスキーマおよびデータベース・インスタンスにあることが必要です。
2. 実表に挿入、更新および削除を記録する 3 つのトリガーを作成します。これらのトリガーは、Oracle 表の場合に示したトリガーと同様に動作する必要があります。
3. ログ表を関連付けます。これらの処理が終了した後で、Oracle Ultra Search の表データ・ソースの作成時に、「ロギング機能使用可能 (Oracle 以外の表)」オプションを選択します。このオプションを選択すると、Oracle Ultra Search Administration ツールによって、リモート・データベースのログ表の名前を入力するプロンプトが表示されます。Oracle Ultra Search は、このログ表と実表を関連付けます。Oracle Ultra Search では、手順 1 と手順 2 が正しく実行されていることを想定しています。

メタデータの Oracle Ultra Search へのロード

Oracle Ultra Search では、メタデータを Oracle Ultra Search データベースにロードするためのコマンドライン・ツールが提供されます。データが大量にある場合は、このツールによって HTML ベースの管理ツールより早くロードできます。

このローダー・ツールは次の種類のメタデータをサポートします。

- 検索属性の値リスト (LOV) および表示名
- ドキュメント検索オプションおよびドキュメント・ロード

メタデータ・ローダーは Java アプリケーションです。このプログラムを使用するには、後述する XML Schema の書式に準拠した XML ファイルにメタデータを格納する必要があります。これによって、XML ファイル名、データベース関連パラメータおよびローダー・タイプ・パラメータを指定して Java プログラムを起動できます。プログラムでは XML ファイルを解析し、メタデータをアップロードします。ステータスとエラーのメッセージは、端末のコンソールに表示されます。

ロード・ツールの起動

ローダー・プログラムのバイナリ・ファイルは、次のディレクトリにあります。

```
%ULTRASEARCH_HOME%/bin/MetaLoader.class
```

使用するマシンには、Java 1.2 に準拠した Java Runtime が必要です。システムの Java CLASSPATH には、次の Java ライブラリが含まれている必要があります。

- Oracle JDBC Thin ドライバのバージョン 1.2。ファイル名は classes12.zip です。
- Oracle XML Parser for Java のバージョン 2。ファイル名は xmlparserv2.jar です。
- Oracle XML Schema Processor for Java。ファイル名は xmlschema.jar です。
- Oracle Ultra Search Java ライブラリ。ファイル名は ultrasearch.jar です。
- Oracle JDBC グローバリゼーション・サポートのバージョン 1.2。ファイル名は nls_charset12.zip です。

ファイルを起動するには、次のように入力します。

```
% java MetaLoader -db <database_connection_string> -u <user_name> -p <password> -i  
<instance_name> -type <loader_type> -f <input_file><>
```

ここで、

- -db はデータベース接続文字列です。
- -u はデータベース・スキーマ・ユーザー名です。
- -p はデータベース・スキーマ・パスワードです。
- -i は Oracle Ultra Search インスタンス名です。
- -type はローダーのメタデータ・タイプで、lov または doc です。
- -f は入力メタデータの XML ファイル名です。

たとえば、このツールを使用して XML ファイル test.xml に指定されている属性 LOV をロードする際、次の引数を指定するとします。

- データベース接続文字列 : dlsun576:5521:isearch
- スキーマ・ユーザー名 : wk_test
- スキーマ・パスワード : welcome
- Oracle Ultra Search インスタンス名 : wk_inst

次の文によってローダー・プログラムを起動します。

```
% java MetaLoader -db dlsun576:5521:isearch -u wk_test -p welcome -i wk_inst -type  
lov -f test.xml
```


ドキュメントおよび関連性スコアのロード

このローダー・ツールを使用してドキュメントとその検索オプション・スコアを Oracle Ultra Search に追加するには、パラメータ `-type` の値を `doc` にする必要があります。

入力 XML ファイル

ドキュメント URL および検索オプション・スコアは、XML ファイルに定義されています。検索対象のドキュメントは 1 つ以上定義できます。各ドキュメントに、1 つ以上の検索オプション・スコアのペアを指定できます。XML ファイルの定義は、XML Schema に格納されます。

関連項目： E-5 ページ「ドキュメント検索オプション用の XML Schema」

ドキュメント検索オプション用の XML ファイルの例

```
<?xml version = "1.0" encoding = "UTF-8"?>
<doc_list>
  <doc url="http://www.oracle.com" data_source_name="Data Source A">
    <term score="100">database</term>
    <term score="90">internet</term>
    <term score="80">software</term>
  </doc>
  <doc url="http://www-st.us.oracle.com" data_source_name="Data Source B">
    <term score="100">Sever Technology</term>
    <term score="100">ST Website</term>
    <term score="95">st</term>
  </doc>
</doc_list>
```

この例では、ドキュメント URL 「`http://www.oracle.com`」 がデータ・ソース「Data Source A」にロードされます。これは、検索オプション項目に「`database`」（スコア「100」）、「`internet`」（スコア「90」）、「`software`」（スコア「80」）を指定して、Oracle Ultra Search で定義されます。

注意： データ・ソース名は、データ・ソースの表示名ではなく、元のデータ・ソース名です。

検索属性 LOV および LOV 表示名のロード

ローダー・ツールを使用して LOV エントリおよび表示名を Oracle Ultra Search に追加するには、パラメータ `-type` の値を `lov` にする必要があります。

LOV XML ファイル

LOV エントリおよび表示名は、XML ファイルに定義されています。XML ファイルには、検索属性 LOV を 1 つ以上定義できます。デフォルト LOV およびデータ・ソース固有の LOV は、両方とも XML ファイルに格納されます。XML ファイルの定義は、XML Schema に格納されます。

関連項目： E-6 ページ [「LOV および LOV 表示名用の XML Schema」](#)

LOV XML ファイルの例

```
<?xml version = "1.0" encoding = "UTF-8"?>
<lov_list>
  <lov search_attr_name="Department" search_attr_type="string">
    <default>
      <lov_values>
        <entry value="100"></entry>
        <entry value="200"></entry>
      </lov_values>
      <lov_display_names lang="en-US">
        <entry value="100" display_name="Human Resource"></entry>
        <entry value="200" display_name="Finance"></entry>
      </lov_display_names>
    </default>
  <data_source name ="data source a">
    <lov_values>
      <entry value="300"></entry>
      <entry value="400"></entry>
    </lov_values>
    <lov_display_names lang="en-US">
      <entry value="300" display_name="Sales"></entry>
      <entry value="400" display_name="Marketing"></entry>
    </lov_display_names>
  </data_source>
  <data_source name ="data source b">
    <lov_values>
      <entry value="500"></entry>
      <entry value="600"></entry>
    </lov_values>
    <lov_display_names lang="en-US">
      <entry value="500" display_name="Production"></entry>
      <entry value="600" display_name="Research"></entry>
    </lov_display_names>
  </data_source>
</lov_list>
```

```

        </lov_display_names>
      </data_source>
    </lov>
  </lov_list>

```

前述の例では、文字列タイプの検索属性「Department」に関する複数の LOV が Oracle Ultra Search にロードされます。次の LOV がロードされます。

- 検索属性「Department」に関するデフォルトの LOV エントリ
- データ・ソース「data source a」用の検索属性「Department」の LOV
- データ・ソース「data source b」用の検索属性「Department」の LOV

ドキュメント検索オプション用の XML Schema

ドキュメント検索オプション項目およびスコア用の XML Schema について説明します。

```

<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  <xsd:element name = "doc_list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "doc" maxOccurs = "unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "term" maxOccurs = "unbounded">
                <xsd:complexType>
                  <xsd:simpleContent>
                    <xsd:extension base = "xsd:string">
                      <xsd:attribute name = "score" use = "required" type = "xsd:integer"/>
                    </xsd:extension>
                  </xsd:simpleContent>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:element>
          <xsd:attribute name = "url" use = "required" type = "xsd:string"/>
          <xsd:attribute name = "data_source_name" use = "required" type = "xsd:string"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

LOV および LOV 表示名用の XML Schema

LOV エントリおよび表示名用の XML Schema について説明します。

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  <xsd:element name = "lov_list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "lov" maxOccurs = "unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "default" minOccurs = "0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name = "lov_values" minOccurs = "0">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name = "entry" maxOccurs = "unbounded">
                            <xsd:complexType>
                              <xsd:attribute name = "value" use = "required" type = "xsd:string"/>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            <xsd:element name = "lov_display_names" minOccurs = "0" maxOccurs = "unbounded">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name = "entry" maxOccurs = "unbounded">
                    <xsd:complexType>
                      <xsd:attribute name = "value" use = "required" type = "xsd:string"/>
                      <xsd:attribute name = "display_name" use = "required" type =
"xsd:string"/>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
              <xsd:attribute name = "lang" use = "required">
                <xsd:simpleType>
                  <xsd:restriction base = "xsd:string">
                    <xsd:length value = "5"/>
                    <xsd:pattern value = "[a-zA-Z]{2}\-[a-zA-Z]{2}"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:attribute>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      <xsd:attribute name = "lang" use = "required">
        <xsd:simpleType>
          <xsd:restriction base = "xsd:string">
            <xsd:length value = "5"/>
            <xsd:pattern value = "[a-zA-Z]{2}\-[a-zA-Z]{2}"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "data_source" minOccurs = "0" maxOccurs = "unbounded">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "lov_values" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "entry" maxOccurs = "unbounded">
                <xsd:complexType>
                  <xsd:attribute name = "value" use = "required" type = "xsd:string"/>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name = "lov_display_names" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "entry" maxOccurs = "unbounded">
                <xsd:complexType>
                  <xsd:attribute name = "value" use = "required" type = "xsd:string"/>
                  <xsd:attribute name = "display_name" use = "required" type =
"xsd:string"/>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name = "lang" use = "required">
              <xsd:simpleType>
                <xsd:restriction base = "xsd:string">
                  <xsd:length value = "5"/>
                  <xsd:pattern value = "[a-zA-Z]{2}\-[a-zA-Z]{2}"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:attribute>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name = "name" use = "required" type = "xsd:string"/>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
<xsd:attribute name = "search_attr_name" use = "required" type = "xsd:string"/>
<xsd:attribute name = "search_attr_type" use = "required">
  <xsd:simpleType>

```

```
        <xsd:restriction base = "xsd:string">
            <xsd:enumeration value = "string"/>
            <xsd:enumeration value = "number"/>
            <xsd:enumeration value = "date"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Crawler の Java CLASSPATH の変更

Oracle Ultra Search Crawler は、Java Virtual Machine (Java VM) で動作する Pure Java アプリケーションです。Java Virtual Machine (Java VM) は Java CLASSPATH を使用して、実行時にクラスを検索します。Oracle Ultra Search のインストール時に、デフォルトの Crawler CLASSPATH がデータベースに格納されます。新しく Oracle Ultra Search インスタンスが作成されると、常にこのデフォルトの CLASSPATH がコピーされ、Crawler CLASSPATH として特定のインスタンスで使用されます。

Crawler の Java CLASSPATH を変更するケース

通常は、Crawler の Java CLASSPATH を変更する必要はありません。ただし、特定の理由から変更を行う場合があります。JavaMail の参照実装をサード・パーティの JavaMail に置き換える場合などは、変更を行います。

Crawler CLASSPATH とリモート Crawler CLASSPATH の違い

Crawler CLASSPATH とは、Oracle Ultra Search Server コンポーネントがインストールされているホストで実行する Crawler の CLASSPATH です。ただし、Oracle Ultra Search には、拡張性を得るために他のホストで実行できるリモート Crawler があります

リモート Crawler の起動には、Java Remote Method Invocation (RMI) テクノロジーを使用します。その結果、リモート Crawler の CLASSPATH 設定は、RMI レジストリおよび RMI デーモンの CLASSPATH 設定から継承されます。

関連項目： [付録 C「リモート Crawler の使用方法」](#)

Oracle Ultra Search Server ホストでの Crawler の Java CLASSPATH 変更

1. Oracle Ultra Search Server コンポーネントがインストールされているホストにログオンします。`$ORACLE_HOME/ultrasearch/admin/wk0cpath.sql` ファイルの位置を特定します。
2. SQL*Plus を使用して、WKSYS スーパー・ユーザーまたは WKADMIN ロールを付与されたデータベース・ユーザーとして `wk0cpath.sql` スクリプトを実行します。
3. プロンプトに従って、変更が必要なデフォルトの CLASSPATH またはインスタンス固有の CLASSPATH を指定します。デフォルトの CLASSPATH を変更すると、その後作成されるすべてのインスタンスでその CLASSPATH が使用されます。既存のインスタンスが変更されることはありません。
4. インスタンス固有の CLASSPATH を変更する場合は、プロンプトに従って Oracle Ultra Search インスタンスの名前を入力します。デフォルトの CLASSPATH を変更する場合、ここでの入力是不要です。
5. プロンプトに従って、CLASSPATH 全体の更新か、または CLASSPATH の追加のいずれかを指定します。CLASSPATH の追加では、エントリが CLASSPATH の先頭に追加されます。複製のクラスの場合は通常、後続のエントリは CLASSPATH の元のエントリによってオーバーライドされます。
6. CLASSPATH 全体を更新する場合は、プロンプトに従って新しい CLASSPATH を入力します。1 つ以上のディレクトリまたはライブラリ・ファイルを CLASSPATH に追加する場合は、Oracle Ultra Search Server コンポーネントがインストールされているプラットフォームに対応した CLASSPATH セパレーター（プラットフォームが UNIX の場合はコ

ロン、Windows の場合はセミコロン) で区切られたディレクトリやライブラリ・ファイルを入力します。

リモート Crawler ホストでの Crawler の Java CLASSPATH 変更

1. Oracle Ultra Search Middle-Tier コンポーネントがインストールされているリモート Crawler ホストにログオンします。そのホストが UNIX マシンの場合は、次のファイルの位置を特定します。
\$ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/unix/define_env
ホストが Windows NT マシンの場合には、次のファイルの位置を特定します。
\$ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/winnt/define_env.bat
2. define_env ファイルでは、RMI サブシステムで使用されるすべての環境設定を指定します。CLASSPATH を変更するには、テキスト・エディタを使用して変数 APPLICATION_CLASSPATH を変更します。
3. これらの変更が反映されるように、RMI サブシステムを再起動してください。

関連項目： RMI サブシステムの起動の詳細は、[付録 C「リモート Crawler の使用方法」](#) を参照してください。

問合せ構文拡張 9.0.1 のカスタマイズ

Oracle Ultra Search は、Oracle Text エンジンを使用して、ドキュメントの索引付けおよび検索を行います。エンド・ユーザーが問合せ文字列を指定すると、Oracle Ultra Search は、その文字列を Oracle Text の問合せ式に変換します。この処理を「問合せ構文拡張」と呼びます。

Oracle Ultra Search には、デフォルトの問合せ構文拡張の実装が用意されています。この実装のコードは、PL/SQL パッケージ WK_QUERYEXP に含まれています。コードは、Oracle データベース・サーバーホスト上の

\$ORACLE_HOME/ultrasearch/admin/wk0queryexp.pkb ファイルで参照できます。

ここでは、問合せ構文拡張の実装を、使用している環境向けにカスタマイズする方法を説明します。

デフォルトの問合せ構文拡張の実装

デフォルトの問合せ構文拡張の実装は、次の方法に直接影響します。

1. エンド・ユーザーによる問合せ文字列の入力方法（「エンド・ユーザーの問合せ構文」を参照）。
2. 問合せに一致するドキュメントへのスコア付けの方法（「スコア付け」を参照）。
3. エンド・ユーザーが入力した問合せ文字列から Oracle Text の問合せ文字列への変換方法（「拡張ルール」を参照）。

エンド・ユーザーの問合せ構文

デフォルトの問合せ構文拡張の実装で定義されるエンド・ユーザーの問合せ構文は、Web の検索エンジンで使用されている標準テキスト問合せ構文に似ています。

トークン

トークンは、二重引用符 (") で囲まれた文字列です。1 つの単語または句です。

演算子

デフォルトの実装では、3 つの演算子が定義されます。[+]、[-] および [*] です。これらの演算子は、デフォルトの実装で定義されています。カスタム実装によって、これらの演算子を変更します。

プラス演算子 [+]。その後に続くトークンが、検索結果に含まれるすべてのドキュメントに存在することを指定します。

マイナス演算子 [-]。その後に続くトークンが、検索結果に含まれるすべてのドキュメントに存在しないことを指定します。

アスタリスク [*]。検索のワイルド・カードを指定します。0（ゼロ）または1 つ以上の文字に相当します。アスタリスクで始まるトークンは、無視されます。アスタリスクは、トークンの末尾（右側）または途中のみに指定できます。たとえば、「hel*o」および「hell*」では正しくアスタリスクを使用していますが、「*ello」は受け入れられません。

サマリー

次に、Oracle Ultra Search のエンド・ユーザーの問合せ構文に関するルールを示します。

注意： エンド・ユーザーのすべての問合せ文字列を、大カッコで囲んで示します。たとえば、エンド・ユーザーの問合せ文字列 Oracle Applications は、[Oracle Applications] と記述します。

ルール	説明
単一の単語の検索	<p>1つの単語を入力すると、その単語を含むドキュメントを検索する。</p> <p>たとえば、[Oracle] は、「Oracle」という単語を含むすべてのドキュメントを検索する。</p>
複数の単語の検索	<p>2つ以上の単語を入力すると、順序に関係なく、それらの単語のいずれかを含むドキュメントを検索する。</p> <p>たとえば、[Oracle Applications] の場合は、「Oracle」、「Applications」または「Oracle Applications」を含むドキュメントを検索する。</p>
強制的な包含 [+]	<p>単語の前に [+] を付けると、その単語は、一致したすべてのドキュメントに存在することを示す。</p> <p>たとえば、[Oracle+Applications] の場合は、「Applications」という単語を含むドキュメントのみを検索する。</p> <p>注意: 複数の単語の検索では、一番最初のトークンを含むすべてのトークンの前に [+] を付けることができる。</p>
強制的な排除 [-]	<p>単語の前に [-] を付けると、その単語は、一致したすべてのドキュメントに存在しないことを示す。</p> <p>たとえば、[Oracle-Applications] の場合は、「Applications」という単語を含まないドキュメントのみを検索する。</p> <p>注意: 複数の単語の検索では、一番最初のトークン以外のすべてのトークンの前に [-] を付けることができる。</p>
句の一致 ["..."]	<p>一連の単語を二重引用符で囲むと、句を正確に含むドキュメントのみを検索する。</p> <p>たとえば、["Oracle Applications"] の場合は、「Oracle Applications」という文字列を含むドキュメントのみを検索する。</p>
ワイルドカードの一致 [*]	<p>単語の右側に [*] を付けると、アスタリスクの左側が一致する結果を戻す。</p> <p>たとえば、[Ora*] という文字列の場合は、「Oracle」や「Orator」など、「Ora」で始まるすべての単語を含むドキュメントを検索する。単語の途中にアスタリスクを挿入することもできる。たとえば、[A*e] という文字列の場合、「Apple」、「Ate」、「Ape」などの単語を含むドキュメントを検索する。ワイルドカードの一致では、より多くの計算処理が必要なため、他の問合せより低速である。</p>

スコア付け

エンド・ユーザーの問合せ文字列に応じてドキュメントを一致させる方法には、3通りあります。これらの3つの方法を「クラスのスコア付け」といいます。ドキュメントが高度なクラス要件を満たす場合、そのドキュメントは高くスコア付けされ、高い位置にランク付けされます。クラスごとに、ドキュメントがクラスのスコア付け条件にどの程度一致するかによってランク付けが異なります。

クラス1は、最も高く重み付けされたクラスです。ドキュメント内で完全に一致した文字列の数によって、スコアが決まります。一致した句が多いドキュメントほど、一致した句が少ないドキュメントより高くスコア付けされます。

クラス2は、その次に高く重み付けされたクラスです。このクラスでは、ドキュメントに現れる各トークンが接近しているほど、高くスコア付けされます。たとえば、エンド・ユーザーの問合せ文字列が [Oracle Applications Financials] で、3つのドキュメントが検出されたとします。3つのドキュメントとも、「Oracle Applications Financials」という文字列は含みません。ただし、ドキュメントXには、同じ文にあって他の単語で分離される「Oracle」、「Applications」および「Financials」の3つのトークンが含まれます。ドキュメントYには、同じ段落で異なる文にそれぞれのトークンが含まれます。ドキュメントZには3つのトークンが含まれますが、各トークンは異なる段落にあります。この場合は、各トークンが最も接近しているドキュメントXが最も高くスコア付けされます。同様に、ドキュメントYはドキュメントZより高くスコア付けされます。

クラス3は、重み付けが最も低いクラスです。トークンを多く含むドキュメントは、高くスコア付けされます。たとえば、エンド・ユーザーの問合せ文字列が [Oracle Applications Financials] で、3つのドキュメントが検出されたとします。ドキュメントXには、3つのトークンすべてが含まれます。ドキュメントYには、「Oracle」および「Applications」のみが含まれます。ドキュメントZには、「Oracle」のみが含まれます。この場合は、ドキュメントXはドキュメントYより高くスコア付けされます。同様に、ドキュメントYはドキュメントZより高くスコア付けされます。

拡張ルール

前述のとおり、エンド・ユーザーの問合せは、Oracle Text の問合せに拡張されます。拡張された問合せ文字列のルールは、BNF 表記に準拠します。また、これらのルールは、デフォルトの問合せ構文拡張の実装として Oracle Ultra Search で使用されます。

ルール

次に、拡張検索を定義するルールを示します。

```
<expanded query> ::= (<expression> within <title section>)*2, <expression>
<expression> ::= <generic query expression> | <simple query expression>
<generic query expression> ::= (([ <plus expression>*100 & ]) (<main expression>)) [
<minus expression> ]
<simple query expression> ::= (<phrase expression>)*2, (<main expression>)
<main expression> ::= (<near expression>)*2, (<accum expression>)
```

前述のルールで使用されている項目およびその意味は、次のとおりです。

<plus expression> は、すべてのプラス・トークンの AND 式です。

<minus expression> は、すべてのマイナス・トークンの NOT 式です。

<phrase expression> は、<main expression> 内のすべてのトークンで構成された PHRASE 式です。

<near expression> は、マイナス・トークンを除くすべてのトークンの NEAR 式です。

<accum expression> は、マイナス・トークンを除くすべてのトークンの ACCUMULATE 式です。

<simple query expression> は、エンド・ユーザーの問合せに複数のトークンが含まれ、演算子や二重引用符が含まれない場合にのみ使用されます。

それ以外の場合は、<generic query expression> が使用されます。

プラス・トークン、マイナス・トークン以外のトークンがない場合、<plus expression> および <accum expression> は排除されます。

ルールの適用例

次に、デフォルトの問合せ構文拡張の実装によって、エンド・ユーザーの問合せ文字列を Oracle Text 互換の問合せ文字列に変換する方法を示します。

エンド・ユーザーの 問合せ文字列	Oracle Text で認識可能な拡張検索文字列
[Oracle]	((({Oracle})) within TITLE__31)*2,(({Oracle}))
[Oracle + Applications]	(((((({Applications})*10)*10&(({Oracle};{Applications}))*2,({Oracle},{Applications}))) within TITLE__31)*2,(((({Applications})*10)*10&(({Oracle};{Applications}))*2,({Oracle},{Applications}))))
[Oracle - Applications]	(((((({Oracle})~{Applications}) within TITLE__31)*2,(({Oracle})~{Applications})))
["Oracle Applications"]	(((({Oracle Applications}) within TITLE__31)*2,(({Oracle Applications}))
[Ora*]	(((((Ora%)) within TITLE__31)*2,((Ora%)))
[Oracle Applications]	(((((({Oracle Applications})*2,(({Oracle};{Applications}))*2,(({Oracle},{Applications}))) within TITLE__31)*2,(({Oracle Applications})*2,(({Oracle};{Applications}))*2,(({Oracle},{Applications}))))

ルールのカスタマイズ

独自の問合せ構文拡張の定義および実装によって、組織の目的に合うように拡張をカスタマイズします。そのためには、Oracle Text の問合せ要件を理解する必要があります。Oracle Text の問合せについては、このドキュメントでは詳しく説明していません。

関連項目：

- 『Oracle Text アプリケーション開発者ガイド』
- 『Oracle Text リファレンス』

Oracle Ultra Search をカスタマイズして独自の問合せ構文拡張の実装を使用するには、WK_QUERYEXP パッケージを変更します。このパッケージには、編集が必要な 2 つの PL/SQL ファンクションがあります。expand_main および expand_attr です。Expand_main は、エンド・ユーザーが入力した問合せ文字列に適用されます。Expand_attr は、拡張検索で指定した各検索属性に適用されます。各 expand_attr ファンクションの戻り値は、expand_main ファンクションの戻り値に追加されます。この結果の問合せ文字列は、問合せを実行する Oracle Text に対して指定される内容です。

expand_main ファンクション

このファンクションは、基本検索ボックスまたは拡張検索ボックスに入力された問合せ文字列を取得し、カスタム問合せ構文拡張の実装ルールに従って、Oracle Text の問合せ文字列に変換します。

```
CREATE OR REPLACE FUNCTION expand_main(query varchar2)
RETURN varchar2
AS
    newqry varchar2(4000);
BEGIN
    newqry := <Convert the input query string into an
              Oracle Text query string according to
              your custom rules>
    return newqry;
END;
```

expand_attr ファンクション

このファンクションは、拡張検索の各検索属性に適用されます。各属性を取得し、カスタム問合せ構文拡張の実装ルールに従って、Oracle Text の問合せ文字列に変換します。

```
CREATE OR REPLACE FUNCTION expand_attr(query varchar2)
RETURN varchar2
AS
    newqry varchar2(4000);
BEGIN
    newqry := <Convert a search attribute into an
```



```
Oracle Text query string according to
your custom rules>
return newqry;
END;
```

注意： カスタマイズされたファンクションはすべてインスタンス固有であり、Oracle Ultra Search インスタンス・ユーザーのスキーマで定義される必要があります。

ファンクションは、定義者権限を使用して実行する必要があります。

expand_main の戻り値と expand_attr の値の組合せの例

次に、デフォルトの問合せ構文拡張の実装によって、エンド・ユーザーの問合せ文字列 Oracle Applications を Oracle Text 互換の問合せ文字列に変換する方法を示します。2 つの検索属性で追加される句は、太字でハイライトされています。

エンド・ユーザーの 問合せ文字列	Oracle Text で認識可能な拡張検索文字列
[Oracle Applications]	(((((Oracle Applications))*2,(({Oracle};{Applications}))*2,({Oracle}, {Applications}))) within TITLE__31)*2,(({Oracle Applications})*2,(({Oracle};{Applications})* 2,({Oracle},{Applications}))))
タイトル属性が「MyTitle」 に、作成者属性が 「MyAuthor」に限定された [Oracle Applications]	(((((Oracle Applications))*2,(({Oracle};{Applications}))*2,({Oracle}, {Applications}))) within TITLE__31)*2,(({Oracle Applications})*2,(({Oracle};{Applications}))*2,({Oracle}, {Applications}))))&(((MyTitle)) WITHIN TITLE__ 31)&(((MyAuthor)) WITHIN AUTHOR__32))*10)*10

索引

C

Crawler

- CLASSPATH, F-1
- Crawler エージェント, 4-3
 - 概要, 4-2
 - クロール・プロセス, 4-3
 - 設定, 4-2, 5-11
 - データ・ソース, 4-2
 - 統計, 5-14
 - パラメータ, 5-2, 5-34
 - リモート Crawler, 5-14

Crawler エージェント

- API, 6-3
- 機能, 6-6
- サンプル・エージェント・ファイル, 6-8
- スマート・エージェント, 6-5
- 設定, 6-8
- 標準エージェント, 6-5

CTXSYS ユーザー, 3-4

D

- data-sources.xml ファイル, 2-24
- DB_CACHE_SIZE パラメータ, B-1
- DR\$WK\$DOC_PATH_IDX\$I 表, 3-6

E

- Email API, 6-11
- Enterprise Manager, 5-4

J

- JDBC, C-1
- JOB_QUEUE_PROCESSES 初期化パラメータ, 3-3

O

- Oracle Enterprise Manager, 2-13, 2-26, 3-2
- Oracle Internet Directory, 2-22, 5-4
- Oracle Text, 1-2, 2-2, 2-5, 3-4, 4-2, 4-5, 4-8, 6-28, G-1
- Oracle Ultra Search
 - Administration ツール, 5-2
 - Middle-Tier コンポーネント, 2-9
 - Oracle Ultra Search Server コンポーネント, 2-4
 - アップグレード, 3-10, 3-11
 - 方法, 3-11
 - インスタンス, 5-6, 5-9
 - 管理権限, 5-35
 - グローバリゼーション, 5-35
 - 言語, 5-12, 5-35
 - 構成, 3-2
 - システム要件, 2-2
 - スナップショット・インスタンス, 5-8
 - チューニング, 3-2, A-1
 - メタデータ・ローダー, E-1
 - ユーザー, 5-34
 - ユーザーの管理, 5-34

P

- PROCESSES 初期化パラメータ, 3-3

Q

Query API, 6-2
Query タグ・ライブラリ, 6-19

R

REDO ログ・ファイル
 サイズ変更, 3-2
robots.txt ファイル, xviii, 5-19, 6-14

S

SORT_AREA_RETAINED_SIZE 初期化パラメータ, 3-3
SORT_AREA_SIZE 初期化パラメータ, 3-3

U

ultrasearch.properties ファイル, 2-10, 2-15, 2-25
UNDO 領域
 サイズ変更, 3-3
URL 認証, 5-15
URL 発行, 5-31
URL リライター, 5-18, 5-20, 6-13
 作成, 6-16
 使用, 6-16
URL リンクのフィルタ処理, 6-14
URL リンクのリライト, 6-14
URL ルーピング, A-2

W

Web クロール, 6-13
WK\$DOC 表, 3-6
WK\$URL 表, 3-6
wk0migrate.sql スクリプト, 3-13
wk0pref.sql ファイル, 3-7, 3-8, 4-2
wk0upgrade.sql スクリプト, 3-11, 3-13
WKSYS.WK_QRY パッケージ, B-3
WKSYS データベース・ユーザー, 2-4, 2-5, 2-30, 3-6, 3-9, 3-13, 5-4, F-2
WKUSER ロール, 3-5, 5-35

あ

アクセス URL, 4-3, 6-4, 6-14
値リスト (LOV), xvii, xviii, xix, 4-4, 5-16, 5-35, 5-36, 6-8, 6-19, E-1

い

移行ログ, 3-13

け

検索オプション, xix, 5-31
検索属性, 5-16

さ

索引
 最適化, 5-30
 変更, 3-7, 4-2
サンプル検索アプリケーション, 6-17

し

シングル・サインオン, 5-2

す

スケジュール
 索引最適化, 5-30
 データの同期, 5-26
ストップリスト, 3-8
 デフォルト, 3-8
 変更, 3-8

て

データ・グループ, 5-3, 5-30
データ・ソース, 5-18
 Oracle9iAS Portal, 5-25
 Web, 5-18
 電子メール, 5-22
 同期, 4-3
 表, 5-20
 同期, D-1
 ファイル, 5-23
 ユーザー定義, 4-3, 5-24

と

問合せ構文拡張, 6-28

問合せ統計, 5-32

ドキュメントのキャッシュ, 4-5

ドキュメントのキューイング, 4-4

ドキュメントの索引付け, 4-8

トリガー, D-3

に

認証, xx

 シングル・サインオン, xx, 5-4

ひ

表示 URL, 4-3, 5-19, 5-21, 5-23, 6-4, 6-14

ふ

プロキシ・サーバー, 5-15

め

メタデータ, 4-3

 ロード, E-1

り

リモート Crawler, 4-10, C-1

 プロファイル, 5-14

ろ

ロボットの除外, xviii, 5-19

