

**Oracle® Ultra Search**

ユーザーズ・ガイド

10g (9.0.4)

**部品番号 : B12349-01**

2004 年 3 月

Oracle Ultra Search ユーザーズ・ガイド, 10g (9.0.4)

部品番号 : B12349-01

原本名 : Oracle Ultra Search User's Guide, 10g (9.0.4)

原本部品番号 : B10896-01

原著者 : Michele Cyran

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

Copyright © 2002, 2003 Oracle Corporation. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会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と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル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場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation, and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、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への用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

---

---

# 目次

<b>はじめに</b> .....	xi
対象読者 .....	xii
このマニュアルの構成 .....	xii
関連ドキュメント .....	xiv
表記規則 .....	xv
<b>Oracle Ultra Search の新機能</b> .....	xxi
Oracle Ultra Search リリース情報 .....	xxv
<b>1 Oracle Ultra Search の概要</b>	
Oracle Ultra Search の概要 .....	1-2
Oracle Ultra Search のコンポーネント .....	1-2
Oracle Ultra Search クローラ .....	1-2
Oracle Ultra Search のバックエンド .....	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 の機能 .....	1-4
Oracle Application Server との統合 .....	1-5
拡張可能なクローラおよびクローラ・エージェント .....	1-5
連携した検索 .....	1-5
セキュアな検索 .....	1-6
Oracle XML DB への依存 .....	1-7
サンプル検索アプリケーション .....	1-7
サンプル検索ポートレット .....	1-7
Query API .....	1-8

URL のリライト .....	1-8
ロボットの除外 .....	1-9
表示 URL サポート .....	1-9
ドキュメントおよび検索属性 .....	1-9
メタデータ・ローダー .....	1-10
ドキュメント検索オプション .....	1-10
データ収集モード .....	1-11
インスタンス・スナップショット・サポート .....	1-11
Oracle Internet Directory との統合 .....	1-11
OID の Oracle Ultra Search 管理グループ .....	1-11
管理権限の認可 .....	1-12
シングル・サインオン認証 .....	1-12
問合せ構文拡張 .....	1-13
Oracle Ultra Search のシステム構成 .....	1-13

## 2 Oracle Ultra Search のインストールと構成

Oracle Ultra Search の要件 .....	2-2
Oracle Ultra Search の表記規則 .....	2-2
Oracle Ultra Search のシステム要件 .....	2-3
ハードウェア要件 .....	2-3
ソフトウェア要件 .....	2-4
Oracle Ultra Search のバックエンドのインストール .....	2-4
Oracle Ultra Search のバックエンドのインストールと構成 .....	2-4
セキュアな Oracle Ultra Search インストールの構成 .....	2-6
Oracle Ultra Search のバックエンド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 .....	2-9
データベース要件 .....	2-9
OPCA を使用した Oracle Ultra Search のバックエンドの既存の Oracle9i Database へのインストール .....	2-10
Oracle Ultra Search のデフォルト・インスタンスの構成 .....	2-11
Oracle Ultra Search Middle-Tier の Web サーバー・ホストへのインストール .....	2-12
Web アプリケーションの概要 .....	2-12
ブラウザの要件 .....	2-13
Oracle Database による Oracle Ultra Search Middle-Tier のインストール .....	2-14
Oracle Application Server による Oracle Ultra Search Middle-Tier のインストール .....	2-14

Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier を構成する 場合 .....	2-15
Single Sign-On Server とともに Administration ツールを構成する場合 .....	2-18
Oracle Ultra Search EAR ファイルのサード・パーティ Middle-Tier への配置 .....	2-19
data-sources.xml ファイルの編集 .....	2-22
ultrasearch.properties ファイルの編集 .....	2-24
Web サーバーの起動 .....	2-25
Oracle Ultra Search Administration ツールのテスト .....	2-25
Oracle Ultra Search のサンプル検索アプリケーションのテスト .....	2-26
<b>Oracle Ultra Search のバックエンドのリモート・クローラ・ホストへのインストール</b> .....	2-27
Oracle Ultra Search のバックエンドのリモート・クローラ・ホストへのインストール .....	2-27
Oracle Ultra Search のバックエンドをリモート・クローラ・ホストで構成する場合 .....	2-28
リモート・クローラの登録解除 .....	2-29
<b>ホスティングされた環境での Oracle Ultra Search の構成</b> .....	2-30
ホスティングされた環境用の構成前の作業 .....	2-30
サブスクリイバ・コンテキストでの Oracle Ultra Search の構成 .....	2-30

### 3 インストール後の情報

Oracle Ultra Search スキーマのパスワードの変更 .....	3-2
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 ...	3-6
手順 5: 索引設定の変更 .....	3-7
<b>ストップリストの管理</b> .....	3-7
Oracle Ultra Search のデフォルトのストップリスト .....	3-7
インスタンスのストップリストの変更 .....	3-8
初回のクローラ前にインスタンスのストップリストを変更する場合 .....	3-8
初回のクローラ後にインスタンスのストップリストを変更する場合 .....	3-9
<b>Oracle Ultra Search のアップグレード</b> .....	3-10
アップグレード前の手順 .....	3-10
Oracle Database 付属の Oracle Ultra Search のアップグレード .....	3-10
Oracle Application Server 付属の Oracle Ultra Search のアップグレード .....	3-11
Oracle Collaboration Suite 付属の Oracle Ultra Search のアップグレード .....	3-11
Oracle Collaboration Suite リリース 1 への Oracle Ultra Search のアップグレード .....	3-12

Oracle Ultra Search リリース 1.0.3 からリリース 9.0.3 へのアップグレード .....	3-12
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 .....	3-15
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 .....	3-15
<b>検索アプリケーションの構成</b> .....	<b>3-16</b>
手順 1: data-sources.xml ファイルの編集 .....	3-16
手順 2: 複数インスタンスに対する複数の検索アプリケーションの配置 .....	3-16

## 4 チューニングおよびパフォーマンス

<b>Web クロール・プロセスのチューニング</b> .....	<b>4-2</b>
Web クロールの方針 .....	4-2
クロール・プロセスの監視 .....	4-2
URL ルーピング .....	4-2
<b>問合せパフォーマンスのチューニング</b> .....	<b>4-3</b>
<b>リモート・クローラの使用法</b> .....	<b>4-6</b>
スケーラビリティおよびロード・バランシング .....	4-7
インストールと構成の順序 .....	4-7
<b>Oracle Ultra Search の Real Application Clusters</b> .....	<b>4-10</b>
記憶域アクセスの構成 .....	4-10
リモート・クローラのファイル・キャッシュ .....	4-11
Oracle インスタンスへのログオン .....	4-12
Real Application Clusters の検索アプリケーション .....	4-12
Java クローラ .....	4-12
JDBC ドライバの選択 .....	4-12
<b>表データ・ソースの同期化</b> .....	<b>4-13</b>
Oracle データベースのクロールの同期化 .....	4-13
ログ表の作成 .....	4-13
ログ・トリガーの作成 .....	4-15
Oracle 以外のデータベースのクロールの同期化 .....	4-16

## 5 Oracle Ultra Search のセキュリティ

<b>Oracle Ultra Search のセキュリティについて</b> .....	<b>5-2</b>
Oracle Ultra Search セキュリティ・モデル .....	5-2
ユーザーのクラスとその権限 .....	5-3
Oracle Ultra Search のデフォルト・ユーザー .....	5-3

ホスティングされた環境での Oracle Ultra Search の管理者権限モデル .....	5-4
管理者権限モデル .....	5-5
Oracle Ultra Search で保護されるリソース .....	5-7
認可およびアクセス規定 .....	5-8
Oracle Ultra Search でセキュリティ・サービスが活用される仕組み .....	5-8
Oracle Ultra Search で Oracle Identity Management インフラストラクチャが活用される仕組み ...	5-8
Oracle Ultra Search の拡張性とセキュリティ .....	5-9
<b>Oracle Ultra Search 用のセキュリティ・フレームワークの構成 .....</b>	<b>5-9</b>
Oracle Ultra Search 用のセキュリティ・フレームワーク・オプションの構成 .....	5-9
Oracle Ultra Search 用の Oracle Identity Management オプションの構成 .....	5-9
<b>Oracle Ultra Search セキュリティの構成 .....</b>	<b>5-10</b>

## 6 Oracle Ultra Search クローラとデータ・ソースについて

Oracle Ultra Search クローラの概要 .....	6-2
クローラ設定 .....	6-2
データ・ソース .....	6-2
クローラ・エージェントの使用 .....	6-3
データ・ソースの同期 .....	6-3
表示 URL とアクセス URL .....	6-3
ドキュメント属性 .....	6-4
スケジュールのクロール・プロセス .....	6-5
ドキュメントのキューイングおよびキャッシュ .....	6-5
ドキュメントの索引付け .....	6-8
データ同期 .....	6-9
Oracle Ultra Search のリモート・クローラ .....	6-10

## 7 Oracle Ultra Search Administration ツールについて

Oracle Ultra Search Administration ツール .....	7-2
クローラ・パラメータの設定 .....	7-2
検索オプションの設定 .....	7-3
属性 .....	7-3
データ・グループ .....	7-3
異なる言語でのオンライン・ヘルプ .....	7-3
Oracle Ultra Search へのログオン .....	7-4

<b>SSO ユーザーとしてのインスタンスへのログオンと管理</b> .....	7-5
Oracle Ultra Search へのログオン .....	7-5
SSO ユーザーへの権限の付与 .....	7-6
<b>「インスタンス」タブ</b> .....	7-7
インスタンスの作成 .....	7-7
標準インスタンスの作成 .....	7-7
スナップショット・インスタンスの作成 .....	7-8
インスタンスの選択 .....	7-10
インスタンスの削除 .....	7-11
インスタンスの編集 .....	7-11
インスタンス・モード .....	7-11
スキーマ・パスワード .....	7-11
<b>「クローラ」タブ</b> .....	7-12
設定の構成 .....	7-12
リモート・クローラ・プロファイル .....	7-15
クローラ統計 .....	7-16
クローラ・アクティビティのサマリー .....	7-16
詳細なクローラ統計 .....	7-16
クローラの進行状況 .....	7-16
問題のある URL .....	7-16
<b>「Web アクセス」タブ</b> .....	7-17
プロキシ .....	7-17
認証 .....	7-17
HTTP 認証 .....	7-17
HTML フォーム .....	7-17
<b>「属性」タブ</b> .....	7-18
検索属性 .....	7-18
マッピング .....	7-19
<b>「ソース」タブ</b> .....	7-20
Web ソース .....	7-20
Web ソースの作成 .....	7-20
表ソース .....	7-23
表ソースの作成 .....	7-23
表ソースの編集 .....	7-24
複数の表で構成された表ソース .....	7-24
データベース・リンクの制限事項 .....	7-25



電子メール・ソース .....	7-25
電子メール・ソースの作成 .....	7-26
ファイル・ソース .....	7-26
ファイル・ソースの作成 .....	7-26
Oracle ソース .....	7-27
Oracle Portal ソース .....	7-27
連携したソース .....	7-28
ユーザー定義ソース .....	7-30
ユーザー定義データ・ソース・タイプの作成 .....	7-30
ユーザー定義ソースの作成 .....	7-31
<b>「スケジュール」タブ</b> .....	7-32
データ同期 .....	7-32
同期スケジュールの作成 .....	7-32
スケジュールの更新 .....	7-32
同期スケジュールの編集 .....	7-32
同期スケジュールの起動 .....	7-34
ステータスとクローラの進行状況との同期 .....	7-35
索引最適化 .....	7-36
<b>「問合せ」タブ</b> .....	7-37
データ・グループ .....	7-37
URL 発行 .....	7-37
検索オプション .....	7-38
問合せ統計 .....	7-39
構成 .....	7-40
<b>「ユーザー」タブ</b> .....	7-41
設定 .....	7-41
スーパー・ユーザー .....	7-41
権限 .....	7-42
<b>「グローバル化」タブ</b> .....	7-43
検索属性名 .....	7-43
LOV 表示名 .....	7-44
データ・グループ名 .....	7-44

## 8 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 .....	8-2
Oracle Ultra Search の Query API .....	8-2
問合せ構文拡張のカスタマイズ .....	8-4
デフォルトの問合せ構文拡張の実装 .....	8-4
エンド・ユーザーの問合せ構文 .....	8-5
クラスのスコア付け .....	8-6
拡張ルール .....	8-7
ルールの適用例 .....	8-8
ルールのカスタマイズ .....	8-9
Oracle Ultra Search の Query タグ・ライブラリ .....	8-10
Query タグの説明 .....	8-12
<instance> タグ: Oracle Ultra Search インスタンスとの接続 .....	8-12
<iterAttributes> タグ: すべての検索属性の表示 .....	8-13
<iterGroups> タグ: すべての検索グループの表示 .....	8-14
<iterLanguages> タグ: すべての検索言語の表示 .....	8-14
<iterLOV> タグ: 検索属性で定義されているすべての値の表示 .....	8-15
<getResult> タグ: 検索の実行 .....	8-16
<fetchAttribute> タグ: メタデータの選択 .....	8-17
<showHitCount> タグ: 見積りヒット数の表示 .....	8-17
<iterResult> タグ: 結果のレンダリング .....	8-18
<showAttributeValue> タグ: ドキュメント属性のレンダリング .....	8-18
Oracle Ultra Search クローラ・エージェント API .....	8-19
クローラ・エージェントの概要 .....	8-20
標準エージェント .....	8-21
スマート・エージェント .....	8-21
ドキュメントの属性とプロパティ .....	8-21
クローラ・エージェントの機能 .....	8-22
データ・ソース・タイプの登録 .....	8-22
データ・ソースの登録 .....	8-22
データ・ソース属性の登録 .....	8-23
ユーザー実装クローラ・エージェント .....	8-23
クローラとクローラ・エージェントの相互作用 .....	8-23
クローラ・エージェント API およびクラス .....	8-24
サンプル・エージェント・ファイル .....	8-24

サンプル・クローラ・エージェントの設定 .....	8-24
エージェント jar ファイルのコンパイルおよび作成 .....	8-24
データ・ソース・タイプの作成 .....	8-25
データ・ソース・パラメータの定義 .....	8-25
このタイプのデータ・ソースの定義 .....	8-26
<b>Oracle Ultra Search の Java Email API .....</b>	<b>8-27</b>
JavaMail の実装 .....	8-28
Java Email API .....	8-28
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 .....	8-28
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 .....	8-29
<b>Oracle Ultra Search の URL リライタ API .....</b>	<b>8-29</b>
URL リンクのフィルタ処理 .....	8-30
URL リンクのリライト .....	8-31
URL リライタの作成および使用 .....	8-32
<b>Oracle Ultra Search のサンプル検索アプリケーション .....</b>	<b>8-33</b>
サンプル検索アプリケーション .....	8-34
JSP の概要 .....	8-34

## **A   メタデータの Oracle Ultra Search へのロード**

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

## **B   クローラの Java CLASSPATH の変更**

クローラの Java CLASSPATH を変更するケース .....	B-2
クローラ CLASSPATH とリモート・クローラ CLASSPATH の違い .....	B-2
Oracle Ultra Search Server ホストでのクローラの Java CLASSPATH 変更 .....	B-2
リモート・クローラ・ホストでのクローラの Java CLASSPATH 変更 .....	B-3

## C 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 .....	C-2
エンド・ユーザーの問合せ構文 .....	C-2
ルールของサマリー .....	C-2
スコア付け .....	C-4
拡張ルール .....	C-4
ルールのカスタマイズ .....	C-6
expand_main ファンクション .....	C-6
expand_attr ファンクション .....	C-6
値の組合せの例 .....	C-7

## 索引

---

# はじめに

このマニュアルでは、Oracle Ultra Search および Oracle Ultra Search API の構成および使用方法について説明します。

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](#)
- [関連ドキュメント](#)
- [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 のインストールと構成
- Oracle Ultra Search 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「チューニングおよびパフォーマンス」

Oracle Ultra Search をチューニングし、パフォーマンスを改善する様々な方法について説明します。Web クロール・プロセスのチューニング、問合せパフォーマンスのチューニング、リモート・クローラの使用、Real Application Clusters における Oracle Ultra Search の使用および表データ・ソースの同期化などの方法があります。

### 第 5 章「Oracle Ultra Search のセキュリティ」

Oracle Ultra Search のセキュリティのアーキテクチャおよび構成について説明します。

## **第 6 章 「Oracle Ultra Search クローラとデータ・ソースについて」**

クローラの機能について説明します。クローラの設定、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・クローラについても説明します。

## **第 7 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 クローラを構成およびスケジュールする方法を説明します。

## **第 8 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、クローラ・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 「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 B 「クローラの Java CLASSPATH の変更」**

クローラの 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 C 「問合せ構文拡張 9.0.1 のカスタマイズ」**

Oracle Ultra Search リリース 1 (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。

## 関連ドキュメント

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、インストール関連ドキュメント、ホワイト・ペーパーまたはその他の関連ドキュメントは、OTN-J (Oracle Technology Network Japan) から、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。登録は、次の Web サイトから無償で行えます。

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

すでに OTN-J のユーザー名およびパスワードを取得している場合は、次の URL で OTN-J Web サイトのドキュメントのセクションに直接接続できます。

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



# 表記規則

この項では、このマニュアルの本文およびコード例で使用されている表記規則について説明します。この項の内容は次のとおりです。

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

## 本文の表記規則

本文では、特定の項目が一目でわかるように、次の表記規則を使用します。次の表に、その規則と使用例を示します。

規則	意味	例
太字	太字は、本文中で定義されている用語および用語集に記載されている用語を示します。	この句を指定すると、 <b>索引構成表</b> が作成されます。
固定幅フォントの大文字	固定幅フォントの大文字は、システム指定の要素を示します。このような要素には、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、パッケージおよびメソッドがあります。また、システム指定の列名、データベース・オブジェクト、データベース構造、ユーザー名およびロールも含まれます。	NUMBER 列に対してのみ、この句を指定できます。 BACKUP コマンドを使用して、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・ビュー内の 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。
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ユーザーが指定する要素のサンプルを示します。このような要素には、コンピュータ名およびデータベース名、ネット・サービス名および接続識別子があります。また、ユーザーが指定するデータベース・オブジェクトとデータベース構造、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も含まれます。 <b>注意:</b> プログラム要素には、大文字と小文字を組み合わせて使用するものもあります。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 ファイルで指定します。 /disk1/oracle/dbs ディレクトリ内のデータ・ファイルおよび制御ファイルのバックアップを作成します。 hr.departments 表には、department_id、department_name および location_id 列があります。 QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。 oe ユーザーとして接続します。 JRepUtil クラスが次のメソッドを実装します。

規則	意味	例
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、プレースホルダまたは変数を示します。	<code>parallel_clause</code> を指定できます。  <code>old_release.SQL</code> を実行します。ここで、 <code>old_release</code> とはアップグレード前にインストールしたリリースを示します。

## コード例の表記規則

コード例は、SQL、PL/SQL、SQL\*Plus または他のコマンドライン文の例です。次のように固定幅フォントで表示され、通常のテキストとは区別されます。

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

次の表に、コード例で使用される表記規則とその使用例を示します。

規則	意味	例
[ ]	大カッコは、カッコ内の項目を任意に選択することを示します。大カッコは、入力しないでください。	<code>DECIMAL (digits [ , precision ])</code>
{ }	中カッコは、カッコ内の項目のうち、1つが必須であることを表します。中カッコは、入力しないでください。	<code>{ENABLE   DISABLE}</code>
	縦線は、大カッコまたは中カッコ内の複数の選択項目の区切りに使用します。項目のうち、1つを入力します。縦線は、入力しないでください。	<code>{ENABLE   DISABLE}</code> <code>[COMPRESS   NOCOMPRESS]</code>
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none"> <li>■ 例に直接関連しないコードの一部が省略されている。</li> <li>■ コードの一部を繰り返すことができる。</li> </ul>	<code>CREATE TABLE ... AS subquery;</code>  <code>SELECT col1, col2, ... , coln FROM employees;</code>
.	垂直の省略記号は、例に直接関連しない複数の行が省略されていることを示します。	<code>SQL&gt; SELECT NAME FROM V\$DATAFILE;</code> <code>NAME</code> ----- <code>/fs1/dbs/tbs_01.dbf</code> <code>/fs1/dbs/tbs_02.dbf</code> <code>.</code> <code>.</code> <code>.</code> <code>/fs1/dbs/tbs_09.dbf</code> <code>9 rows selected.</code>

規則	意味	例
その他の記号	大カッコ、中カッコ、縦線および省略記号以外の記号は、記載されているとおりに入力する必要があります。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック体	イタリック体は、特定の値を指定する必要があるプレースホルダや変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システム指定の要素を示します。これらの要素は、ユーザー定義の要素と区別するために大文字で示されます。大カッコ内にかぎり、表示されているとおりの順序および綴りで入力します。ただし、大 / 小文字が区別されないため、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などです。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr
	<b>注意:</b> プログラム要素には、大文字と小文字を組み合わせるものもあります。この場合は、記載されているとおりに入力してください。	CREATE USER mjones IDENTIFIED BY ty3MU9;

## Microsoft Windows オペレーティング・システム環境での表記規則

次の表に、Microsoft Windows オペレーティング・システム環境での表記規則とその使用例を示します。

規則	意味	例
ファイル名およびディレクトリ名	ファイル名およびディレクトリ名は大 / 小文字が区別されません。特殊文字の左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、縦線 ( ) およびハイフン (-) は使用できません。円記号 (¥) は、引用符で囲まれている場合でも、要素のセパレータとして処理されます。Windows では、ファイル名が ¥¥ で始まる場合、汎用命名規則が使用していると解釈されます。	c:¥winnt"¥"system32 は、 C:¥WINNT¥SYSTEM32 と同じです。
Windows コマンド・プロンプト	Windows コマンド・プロンプトには、カレント・ディレクトリが表示されます。コマンド・プロンプト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 コマンド・プロンプトで二重引用符 (") 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。丸カッコおよび一重引用符 (') にはエスケープ文字は必要ありません。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・システムのドキュメント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 ホームの名前を表します。ホーム名には、英数字で 16 文字まで使用できます。ホーム名に使用可能な特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

規則	意味	例
ORACLE_HOME および ORACLE_BASE	<p>Oracle8i より前のリリースでは、Oracle 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が最上位の ORACLE_HOME の直下に置かれました。ORACLE_HOME ディレクトリの名前は、デフォルトでは次のいずれかです。</p> <ul style="list-style-type: none"> <li>■ C:¥orant (Windows NT の場合)</li> <li>■ C:¥orawin98 (Windows 98 の場合)</li> </ul> <p>このリリースは、Optimal Flexible Architecture (OFA) のガイドラインに準拠しています。ORACLE_HOME ディレクトリ下に配置されないサブディレクトリもあります。最上位のディレクトリは ORACLE_BASE と呼ばれ、デフォルトでは、C:¥oracle です。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した場合、Oracle ホーム・ディレクトリは、デフォルトで C:¥oracle¥orann (nn は最新のリリース番号) に設定されます。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下に配置されています。</p> <p>このマニュアルに示すディレクトリ・パスの例は、すべて OFA の表記規則に準拠しています。</p>	<p>%ORACLE_HOME%¥rdbms¥admin ディレクトリにアクセスします。</p>



---

---

# 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。また、Oracle Ultra Search の更新履歴も示します。

## セキュアなクローल

Oracle Ultra Search には、次のタイプの認証を行うセキュアなクロールが備わっています。

**Digest 認証** Oracle Ultra Search は、HTTP Digest 認証をサポートしており、Oracle Ultra Search クローラは HTTP Digest 認証方式を使用して、Web サーバーに対して自らを認証できます。これは簡易要求 / 応答パラダイムに基づいていますが、パスワードは暗号化されません。

**HTML フォーム認証** HTML フォームに基づいた認証は、Web 上で最も頻繁に使用されている認証方式です。Oracle Ultra Search では、Web クローल時に Oracle Ultra Search クローラにより自動入力する HTML フォームを登録できます。HTML フォーム認証では、HTTP Cookie 機能が有効になっている必要があります。

**関連項目：** 7-20 ページ [「Web ソースの作成」](#)

## 動的ページの索引付け

動的 URL は、クローलおよび索引付けできます。動的ページによっては、同じページに対する複数の検索ヒットとして表示されるため、すべてを索引付けする必要がない場合もあります。他の動的ページはそれぞれ異なるため、索引付けする必要があります。

**関連項目：** 7-20 ページ [「Web ソースの作成」](#)

## HTTPS

Oracle Ultra Search は、HTTPS (SSL を使用した HTTP) をサポートするようになりました。Oracle Ultra Search クローラでは、HTTPS URL (https://www.foo.com など) をクロールできます。

**関連項目：** 7-20 ページ [「Web ソースの作成」](#)

## セキュアな検索

Oracle Ultra Search ではセキュアな検索をサポートしています。セキュアな検索では、検索ユーザーが表示できるドキュメントのみ戻されます。

索引付けされた各ドキュメントは、アクセス制御リスト (ACL) により保護できます。検索時に ACL が評価されます。検索を実行しているユーザーが保護されたドキュメントの読取り権限を持っている場合、このドキュメントは Query API により戻されます。読取り権限がない場合、このドキュメントは戻されません。

Oracle Ultra Search では、ACL は Oracle XML DB リポジトリに格納されます。また、Oracle Ultra Search では ACL の評価に Oracle XML DB 機能が使用されます。

**関連項目：** 1-6 ページ [「セキュアな検索」](#)

## Oracle Internet Directory との統合

Oracle Internet Directory (OID) は、Oracle 固有の LDAP バージョン 3 準拠のディレクトリ・サービスであり、Oracle Database の上部にアプリケーションとして構築されています。Oracle Ultra Search は、次の面で OID と統合されています。

- Oracle Ultra Search 管理グループおよびグループ・メンバーシップは、OID に格納されています。
- ユーザーは、シングル・サインオン (SSO) サーバーおよび OID で認証されます。
- OID では、Oracle Ultra Search ユーザーの管理権限に対して認可が行われます。

**関連項目：** 1-11 ページ [「Oracle Internet Directory との統合」](#)

## Cookie のサポート

Cookie のサポートは、デフォルトで有効になっています。



## クローラ・キャッシュの削除制御

クローリング時に、ドキュメントはキャッシュ・ディレクトリに格納されます。事前に設定されたサイズに達するたびに、クローリングが停止し、索引付けが始まります。旧リリースでは、索引付けが完了すると必ずキャッシュ・ファイルが削除されていました。今回のリリースでは、索引付けが完了したときもキャッシュ・ファイルを削除しないよう指定できます。このオプションはすべてのデータ・ソースに適用されます。デフォルトでは、索引付けの後にキャッシュ・ファイルが削除されます。

**関連項目：** 7-12 ページ [「クローラ」タブ](#)

## URL 境界ルールに含まれるポート番号の包含または除外

URL 境界ルールを設定して、クローリング領域を絞り込むことができます。現在では、Web サイトを特定のポートとともに包含または除外できます。たとえば、`www.oracle.com` を包含し、`www.oracle.com:8080` を除外できます。デフォルトでは、すべてのポートがクローリングされます。

**関連項目：** 7-20 ページ [「Web ソースの作成」](#)

## Web データ・ソースの URL 境界仕様で許可されているホスト名接頭辞

旧リリースでは、指定できるのは接尾辞包含ルールのみでした。たとえば、「`oracle.com`」で終わる URL のみクローリングするなどです。今回のリリースでは、接頭辞ルールも指定できるようになりました。たとえば、「`oracle.com`」はクローリングするが、「`stores.oracle.com`」はクローリングしないなどです。

**関連項目：** 7-20 ページ [「Web ソースの作成」](#)

## デフォルトの Oracle Ultra Search インスタンスおよびスキーマ

Oracle Ultra Search では、デフォルトの Oracle Ultra Search テスト・ユーザーに基づいて、デフォルトの Oracle Ultra Search インスタンスが自動的に作成されます。したがって、Oracle Ultra Search の機能をインストール後にデフォルトのインスタンスに基づいてテストできます。

**関連項目：** 2-11 ページ [「Oracle Ultra Search のデフォルト・インスタンスの構成」](#)

## クローラ再クロール・ポリシー

再クロール・ポリシーを、「変更されたドキュメントの処理」または「全ドキュメントの処理」に更新できます。

旧リリースでは、「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でした。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、またはホスト包含ルール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した。

今回のリリースでは、新規にクロールされた URL のみではなく、クロールされた URL はすべてクローラ設定の規定の対象となります。

**関連項目：** 7-32 ページ [「同期スケジュールの編集」](#)

## 連携した検索

Oracle Ultra Search では従来より、一元化された検索を利用して定期的にデータを収集し、検索可能な全データをカタログに追加した1つの索引を更新してきました。これにより高速検索が可能でしたが、データ・ソースが検索前にクロールできる状態になっている必要がありました。現在、Oracle Ultra Search では連携した検索も用意されており、複数の索引で1つの検索を実行できます。各索引は個別に保持できます。検索時にデータ・ソースを問い合わせることで、検索結果は常に最新の結果が得られます。ユーザー資格証明をデータ・ソースに渡し、データ・ソースにより認証できます。問合せは、データの固有の書式を使用して効果的に処理できます。

連携した検索を使用するには、Oracle Ultra Search 検索アダプタ、またはサーチレットを配置し、Oracle ソースを作成する必要があります。サーチレットは、中間層 (OC4J 内) に配置される Java モジュールであり、ユーザーにかわり企業情報システム内のデータを検索します。ユーザーの問合せがサーチレットに委譲された場合、サーチレットはユーザーにかわり問合せを実行します。サーチレットはすべて、JCA 1.0 に準拠するリソース・アダプタです。

**関連項目：** 7-28 ページ [「連携したソース」](#)

## Oracle Ultra Search リリース情報

Oracle Ultra Search は、Oracle Database、Oracle Application Server および Oracle Collaboration Suite とともにリリースされています。これまでいくつかのリリース番号が発行されてきたため、Oracle Ultra Search のリリース番号は多少わかりにくくなっています。

- Oracle Ultra Search リリース 9.0.4 は、Oracle Application Server (OracleAS) リリース 10g (9.0.4) の一部です。
- 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、Oracle9iAS リリース 2 (9.0.2) の一部です。



---

---

# 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 の機能
- Oracle Ultra Search のシステム構成

## Oracle Ultra Search の概要

Oracle Ultra Search は、Oracle Database と Oracle Text のテクノロジーに基づいて構築され、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、クローラを使用してドキュメントを収集します。検索する Web サイトにあわせてクローラをスケジュールできます。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。

さらに、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 Database 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
- 異なるリポジトリ間の属性検索を容易にするための属性マッピング
- 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、完全なグローバル化ソリューション

## 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 クローラ](#)
- [Oracle Ultra Search のバックエンド](#)
- [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 クローラ

Oracle Ultra Search クローラは、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クローラは、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、OracleAS 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。

クローラは、リンクをマップして関連を分析します。クローラ・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。クローラは、クローラ時に 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 6 章「Oracle Ultra Search クローラとデータ・ソースについて」](#)

## Oracle Ultra Search のバックエンド

Oracle Ultra Search のバックエンド（サーバー・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。バックエンドはユーザーには表示されず、クローラ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 のバックエンドのインストール」](#)

## Oracle Ultra Search Administration ツール

Oracle Ultra Search Administration ツールは、J2EE に準拠した Web アプリケーションです。Oracle Ultra Search インスタンスの管理に使用でき、イントラネット内のすべて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、セキュリティやスケーラビリティを強化するために、異なるコンピュータでホスト管理できます。

**関連項目：** [第 7 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。
- クローラ・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ラ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。クローラはこれを使用して、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 8 章「Oracle Ultra Search 開発者ガイドおよび API リファレンス」](#)
- 『Oracle Ultra Search API Reference』

## Oracle Ultra Search の機能

ここでは、Oracle Ultra Search の一部の機能について説明します。次の項目が含まれます。

- [Oracle Application Server との統合](#)
- [拡張可能なクローラおよびクローラ・エージェント](#)
- [連携した検索](#)
- [セキュアな検索](#)
- [サンプル検索アプリケーション](#)
- [サンプル検索ポートレット](#)
- [Query API](#)
- [URL のリライト](#)
- [ロボットの除外](#)
- [表示 URL サポート](#)
- [ドキュメントおよび検索属性](#)
- [メタデータ・ローダー](#)
- [ドキュメント検索オプション](#)
- [データ収集モード](#)
- [インスタンス・スナップショット・サポート](#)
- [Oracle Internet Directory との統合](#)
- [シングル・サインオン認証](#)
- [問合せ構文拡張](#)



## Oracle Application Server との統合

Oracle Application Server (OracleAS) の Oracle Ultra Search は、Oracle Collaboration Suite での Oracle Ultra Search、および Oracle Database 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 Database は、OracleAS Portal と統合されていません。OracleAS および Oracle Collaboration Suite では、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・ページに追加します。OracleAS および Oracle Collaboration Suite では、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。
- OracleAS には、Single Sign-On (SSO) Server が含まれています。SSO ユーザーは OracleAS 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 の管理インターフェースによって、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 のログオ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。SSO は、Oracle Identity Management インフラストラクチャでのみ使用可能です。

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

## 拡張可能なクローラおよびクローラ・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。独自のクローラ・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ー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。クローラによるデータ・ソースへのアクセスを可能にするモジュールをクローラ・エージェントと呼びます。

関連項目：

- 8-19 ページ「[Oracle Ultra Search クローラ・エージェント API](#)」
- 『[Oracle Ultra Search API Reference](#)』

## 連携した検索

Oracle Ultra Search では従来より、一元化された検索を利用して定期的にデータを収集し、検索可能な全データをカタログに追加した 1 つの索引を更新してきました。これにより高速検索が可能でしたが、データ・ソースが検索前にクロールできる状態になっている必要がありました。現在、Oracle Ultra Search では連携した検索も用意されており、複数の索引で 1 つの検索を実行できます。各索引は個別に保持できます。検索時にデータ・ソースを問い合わせることで、検索結果は常に最新の結果が得られます。ユーザー資格証明をデータ・ソースに渡し、データ・ソースにより認証できます。問合せは、データの固有の書式を使用して効果的に処理できます。

連携した検索を使用するには、Oracle Ultra Search 検索アダプタ、またはサーチレットを配置し、Oracle ソースを作成する必要があります。サーチレットは、中間層 (OC4J 内) に配置される Java モジュールであり、ユーザーにかわり企業情報システム内のデータを検索します。ユーザーの問合せがサーチレットに委譲された場合、サーチレットはユーザーにかわり問合せを実行します。サーチレットはすべて、JCA 1.0 に準拠するリソース・アダプタです。

**関連項目：** 7-28 ページ「[連携したソース](#)」

## セキュアな検索

Oracle Ultra Search はセキュアな検索をサポートしており、検索ユーザーが表示できる検索基準を満たすドキュメントのみが戻されます。セキュアな検索では、索引付けされた各ドキュメントは、アクセス制御リスト (ACL) により保護されます。検索時に ACL が評価されます。検索を実行しているユーザーが保護されたドキュメントの読取り権限を持っている場合、このドキュメントは Query API により戻されます。読取り権限がない場合、このドキュメントは戻されません。

データ・ソースを保護するには次の 2 つの方法があります。

- データ・ソースの全ドキュメントを保護する単一 ACL を指定します。

管理者が、Oracle Ultra Search Administration ツールで単一 ACL のアクセス権を指定します。結果の ACL は、そのデータ・ソースに属する全ドキュメントの保護に使用されます。

- データ・ソースから ACL をクロールします。

データ・ソースでは、ドキュメントとともに ACL が提供されます。これによって、各ドキュメントはそれぞれの一意の ACL により保護されます。

Oracle Ultra Search では ACL 重複検出が実行されます。これは、クロールしたドキュメントの ACL が Oracle Ultra Search システムにすでに存在する場合、Oracle Ultra Search 内で新規の ACL を作成せずに、既存の ACL を使用してドキュメントを保護することを意味します。このポリシーにより、記憶領域が削減され、パフォーマンスが向上します。

Oracle Ultra Search は、シングル LDAP ドメインのみサポートしています。ACL に指定されている LDAP ユーザーおよびグループは、同じ LDAP ドメインに属する必要があります。

---

**注意：** データ・ソースから ACL をクロールする場合、クロールするデータ・ソースが同じ LDAP ドメインに属することを管理者が責任を持って確認する必要があります。そうしないと、検索ユーザーがアクセスしてはならないドキュメントへのアクセス権限が、誤って検索ユーザーに付与される可能性があります。

---

セキュアな検索対応の Oracle Ultra Search インスタンスに対して検索が実行される場合、セキュアな検索非対応のインスタンスに対して実行される場合より遅くなります。これは、

ヒットした各候補に ACL 評価が必要となるためです。ACL は元々、パフォーマンスを最大にするために Oracle データベース・サーバーにより評価されます。ただし、これは限られた時間で行われます。したがって、セキュアな検索でヒットを戻すための所要時間は、必要となる ACL 評価の数に応じて異なります。

## Oracle XML DB への依存

Oracle Ultra Search では、ACL は Oracle XML DB リポジトリに格納されます。また、Oracle Ultra Search では ACL の評価に Oracle XML DB 機能が使用されます。

ACL は Oracle Ultra Search により管理されます。ACL は、単一の Oracle Ultra Search インスタンスからドキュメントにより一意に参照されます。ACL は、複数の Oracle Ultra Search インスタンスで共有されません。許容できるパフォーマンスを得るため、ACL キャッシュ・サイズは、実行時に評価されるすべての ACL に対応できる十分な大きさがが必要です。

XML DB リポジトリ内の ACL は、他の ACL (「プロテクタ ACL」と呼ばれる) により保護されます。Oracle Ultra Search は、Oracle Ultra Search が XML DB ACL 評価機能を起動するための適切な権限がプロテクタ ACL により付与されていることを確認します。評価のパフォーマンスは主に、すべての XML DB クライアント・アプリケーションが使用する ACL の総数に影響を受けます。XML DB クライアント・アプリケーションは、ACL 評価機能も使用します。この一連のアプリケーションには、Oracle Ultra Search が含まれます。

**関連項目：**『Oracle9i XML データベース開発者ガイド - Oracle XML DB』  
および 2-6 ページの「セキュアな Oracle Ultra Search インストールの構成」

## 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、OracleAS 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、OracleAS 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：** 8-2 ページ「Oracle Ultra Search の Query API」

## サンプル検索ポートレット

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

Oracle Ultra Search の検索ポートレットは、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：**

- ポートレットの詳細は、OracleAS 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。Query API には次の機能があります。

- 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：**

- 8-2 ページ [「Oracle Ultra Search の Query API」](#)
- 『Oracle Ultra Search API Reference』

## 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。クローラは、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を実行します。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：**

- 7-20 ページ [「Web ソース」](#)
- 8-29 ページ [「Oracle Ultra Search の URL リライタ API」](#)
- 『Oracle Ultra Search API Reference』

## 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 クローラは、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、クローラ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：** 7-20 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。これは、データベース表内のデータに対応する Web アプリケーションの画面によりレンダリングされます。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：** 6-3 ページ「[クローラ・エージェントの使用](#)」

## 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、クローラは、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：** 6-3 ページ「[データ・ソースの同期](#)」

## 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 A 「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、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。

検索オプションでは、検索ユーザーが入力した特定の問合せについて、ドキュメントにスコアが割り当てられます。

---

---

**注意：** 検索された問合せのいずれかではない問合せを入力した場合、ドキュメントには Oracle Text により計算されたスコアが割り当てられたままとなります。

---

---

検索オプションには次の制限事項があります。

- ユーザーの問合せと検索される問合せの比較には、文字列の完全一致が使用されます。完全一致の比較では、大 / 小文字が区別され、スペースが認識されます。したがって、「ultrasearch」と入力した場合、「Ultra Search」についての検索オプション・スコアが割り当てられたドキュメントは検索されません。
- 検索オプションでは、検索アプリケーションにより検索条件が Query API `getResult()` メソッド・コールに渡される必要があります。基本検索条件を検索条件として渡すためのサンプル・アプリケーションが設計されています。検索属性に基づいた拡張検索基準は無視されます。

**関連項目：** [7-37 ページ 「「問合せ」タブ」](#)

## データ収集モード

初期の計画の目的として、索引付けを行わず、クローラで URL の収集のみを行う場合があります。クローラの終了後に、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。クローラ・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： 7-32 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： 7-7 ページ「[「インスタンス」タブ](#)」

## Oracle Internet Directory との統合

Oracle Internet Directory (OID) は、Oracle 固有の LDAP バージョン 3 準拠のディレクトリ・サービスであり、Oracle Database の上部にアプリケーションとして構築されています。OID では Oracle の共通 ID がホスティングされます。Oracle の Web ベース製品はすべて、シングル・サインオンをサポートするために SSO Server と統合しています。

### OID の Oracle Ultra Search 管理グループ

Oracle Ultra Search 管理グループには、ユーザーのセットが含まれます。各ユーザーは、1 つまたは複数のグループに属することができます。すべてのグループは、groupOfUniqueNames および orclGroup オブジェクト・クラスを使用して作成されません。

ユーザー管理権限を付与する唯一の方法は、この権限を管理グループに割り当てる方法です。Oracle Ultra Search では、ユーザーが属する管理グループに基づいてユーザー管理権限が許可されます。各 Oracle Ultra Search インスタンスごとに次のグループが作成されます。

- スーパー・ユーザー：このグループのユーザーは、Oracle Ultra Search インスタンスを作成または削除し、インストール内の Oracle Ultra Search インスタンスを管理できます。スーパー・ユーザーは、ドキュメント検索オプションのルールと、Oracle Ultra Search インスタンスに関連付けられた各ドキュメントに定義されている ACL に従う必要があります。たとえば、あるドキュメント ACL でスーパー・ユーザーまたはグルー

プにアクセス権限が付与されていない場合、スーパー・ユーザーはこのドキュメントを検索および表示できません。

- インスタンス管理者: このグループのユーザーは、Oracle Ultra Search インスタンスを管理できます。インスタンスを削除できるのは、インスタンス・データベース・スキーマ・ユーザーおよびスーパー・ユーザー・グループのメンバーのみです。

## 管理権限の認可

管理ユーザーの認可は、次の手順で実行されます。

1. 管理ユーザーが SSO Server または Oracle Ultra Search データベースにより正常に認証された後、Oracle Ultra Search GUI により、Oracle Ultra Search インスタンスを選択するための初期画面が表示されます。
2. Oracle Ultra Search GUI では、OID サーバーまたは Oracle Ultra Search リポジトリが参照され、管理ユーザーが管理権限を持つインストール内のすべての Oracle Ultra Search インスタンスが検索されます。
3. 管理ユーザーは、リストから Oracle Ultra Search インスタンスを選択します。

**関連項目:** 『Oracle Internet Directory 管理者ガイド』

## 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)
- Portal ユーザー
- Enterprise Manager ユーザー

---

---

**注意:** シングル・サインオン (SSO) は、Oracle Identity Management インフラストラクチャでのみ使用可能です。

---

---

**関連項目:** 7-4 ページ [「Oracle Ultra Search へのログオン」](#)



## 問合せ構文拡張

Oracle Ultra Search は、各ユーザー問合せをデータベース問合せに変換します。この処理を「問合せ構文拡張」と呼びます。拡張ロジックにより、検索オプション、検索結果の呼出しが決定されます。Oracle Ultra Search のデフォルトの拡張では、ユーザーの問合せがタイトルの一部に一致するドキュメントは、検索結果が高くランク付けされます。

問合せ構文拡張は、問合せ API によってカスタマイズできます。

**関連項目：** 8-4 ページ「[問合せ構文拡張のカスタマイズ](#)」

## Oracle Ultra Search のシステム構成

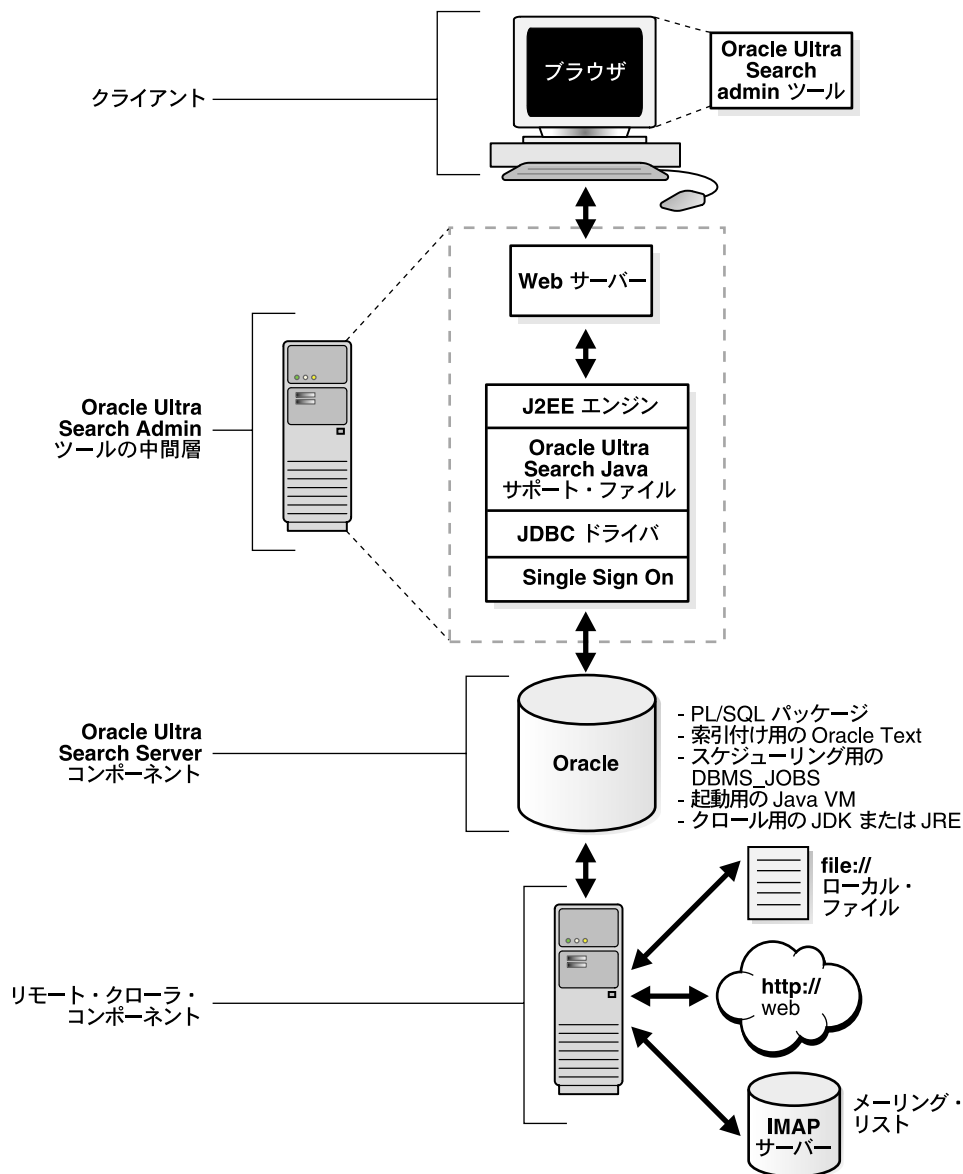
Oracle Ultra Search は、Oracle データベース・サーバーに対して実行されるクライアント・プログラムです。バックエンドと中間層の 2 つの構成で配置できます。

Oracle Ultra Search の問合せインタフェースと Administration ツールは、任意の HTML ブラウザ・クライアントからアクセスできます。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コンピュータと同じか、Oracle Application Server を実行している別のコンピュータの場合があります。Oracle Ultra Search データベースのバックエンド（サーバー・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、クローラ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。クローラ自体は、データベース・サーバー・コンピュータ上で実行するか、別のコンピュータ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 のバックエンドのインストール
- Oracle Ultra Search のバックエンド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 のデフォルト・インスタンスの構成
- 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 のバックエンドのリモート・クローラ・ホストへのインストール
- ホスティングされた環境での Oracle Ultra Search の構成

## 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 のバックエンド（サーバー・コンポーネント）
  - Oracle Ultra Search のリポジトリ
  - Oracle Ultra Search クローラ
- Oracle Ultra Search Middle-Tier
  - Oracle Ultra Search Administration ツール
  - Oracle Ultra Search の Query API

---

**注意：** デフォルトで、Oracle Ultra Search クローラは、Oracle Ultra Search のリポジトリが含まれているデータベースの ORACLE\_HOME に常駐します。Oracle Ultra Search クローラコンポーネントを、他の Oracle ホーム（通常はリモート・ホスト）に追加でインストールできます。これらはリモート・クローラと呼ばれ、システムのスケラビリティに対処する有効な方法です。

---

Oracle Ultra Search のバックエンド（サーバー・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 (9.0.1.4) 以上）にインストールできます。

**関連項目：** 2-9 ページ [「Oracle Ultra Search のバックエンド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 の表記規則

ORACLE\_HOME は、Oracle Ultra Search のバックエンド（サーバー・コンポーネント）または Middle-Tier ビットが含まれている Oracle ホーム・ディレクトリを示します。

REMOTE\_ORACLE\_HOME は、Oracle Ultra Search のリモート・クローラ・ビットが含まれている 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

Oracle Ultra Search のハードウェア要件は、Oracle Ultra Search を使用して処理するデータの量に基づいて異なります。Oracle Ultra Search は、Oracle Text を索引付けエンジンとして使用し、Oracle データベースをリポジトリとして使用します。

**関連項目：**『Oracle Text アプリケーション開発者ガイド』および『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』

**十分な RAM** データベースおよび Oracle Text 索引付けエンジンのリソース要件に加え、Oracle Ultra Search クローラのメモリー要件も考慮する必要があります。Oracle Ultra Search クローラは Pure Java プログラムです。インストール直後、クローラを起動した際、JVM は 25MB で開始し、256MB まで増加するよう構成されています。著しく大量のデータをクロールする場合、これらの値を調整する必要があります。

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 のバックエンド（サーバー・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 のバックエンド（サーバー・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勧めします。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。

- OracleAS Infrastructure またはデータベース、および Oracle Ultra Search のバックエンド（サーバー・コンポーネント）用のディスク領域は、約 3GB。
- Web サーバーのディスク領域に加え、Oracle Ultra Search Middle-Tier 用に 15MB のディスク領域。
- 各リモート・クローラ・ホストについては、Oracle Ultra Search のバックエンド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、クローラ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。OracleAS の使用をお勧めします。

## Oracle Ultra Search のバックエンドのインストール

Oracle Ultra Search のバックエンド（サーバー・コンポーネント）は、Oracle Database の一部です。Oracle Database のインストール時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 のバックエンドは、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 クローラの Java クラス
- Oracle Ultra Search のリモート・クローラ
- Oracle Ultra Search の製品ライブラリ

## Oracle Ultra Search のバックエンドのインストールと構成

### 手順 1: OracleAS Infrastructure および Oracle Ultra Search のバックエンドのインストール

**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 のバックエンド（サーバー・コンポーネント）が Oracle Database とともにインストールされます。

**OracleAS Infrastructure** 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、「OracleAS Infrastructure 10g」というオプションを選択します。デフォルトで、Oracle Ultra Search のバックエンド（サーバー・コンポーネント）が OracleAS Infrastructure とともにインストールされます。

OracleAS 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 のバックエンド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 のバックエンド・ファイルが、**ultrash** 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` (パスワードはランダム化) 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：** `WKSYS` パスワードの変更方法の詳細は、3-2 ページの「[Oracle Ultra Search スキーマのパスワードの変更](#)」を参照してください。

必要な環境変数の設定については、インストレーション・ガイドを参照してください。

## 手順 2: INSO フィルタを使用するための環境設定

Oracle Ultra Search クローラでは、Oracle Text INSO フィルタ `ctxhx` が使用されます。このフィルタでは、共有ライブラリ・パス環境変数に `$ORACLE_HOME/ctx/lib` パスが含まれる必要があります。このパスがないと、バイナリ・ドキュメントのフィルタ処理に失敗します。

インストール後、Oracle Universal Installer により、この変数に `$ORACLE_HOME/ctx/lib` が含まれるよう自動的に設定されます。ただし、インストール後にデータベースを再起動した場合は、Oracle プロセスを開始する前に、共有ライブラリ・パス環境変数に `$ORACLE_HOME/ctx/lib` が含まれるよう、手動で設定する必要があります。フィルタ処理を機能させるには、データベースを再起動して新しい値を選択する必要があります。

たとえば、UNIX では `$LD_LIBRARY_PATH` 環境変数に `$ORACLE_HOME/ctx/lib` が含まれるよう設定し、Windows では `$PATH` 環境変数に `$ORACLE_HOME/bin` が含まれるよう設定します。

## 手順 3: Oracle Ultra Search 用の Oracle Database 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、オプションで Oracle Database 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](#)」

## セキュアな Oracle Ultra Search インストールの構成

### 手順 1: データベースのバージョン要件の確認および Oracle Identity Management の構成

セキュアな Oracle Ultra Search インストールを設定する前に、次の手順を実行する必要があります。

- Oracle Database 9.2.0.4 以上をインストール、またはこれにアップグレード。  
Middle-Tier および IM (Oracle Identity Management) のバージョンは 9.0.4 以上である必要があります。RepCA を使用して、9.2.0.4 データベースを iAS 9.0.4 メタデータ・リポジトリに変換できます。
- Oracle Internet Directory (OID) のインストールおよび構成
- Oracle-OID リンクの構成

セキュアな検索機能では、Oracle Ultra Search データベースが Oracle バージョン 10.1.0 以上であり、Oracle Ultra Search データベースが OID の互換性のあるインスタンスにリンクされている必要があります。これは、Oracle Ultra Search で、特定のバージョンの Oracle を必要とする XML DB 機能が利用されるためです。XML DB では、OID への有効なリンクが必要です。このリンクを通じて、LDAP の重要な情報がすべて取得されます。セキュアな検索が動作するには、Oracle-OID リンクが常に機能している必要があります。このリンクを設定するには、Oracle Identity Management を使用するよう Oracle Database を構成します。

**関連項目：** Oracle Identity Management および OID を使用するためにデータベースを構成する方法の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

### 手順 2: Oracle リスナーの再起動

前の手順で、Oracle Identity Management を使用するよう Oracle Database を構成しました。この処理には、ディレクトリを使用するための Oracle ホームの構成が含まれていました。Oracle ホームに変更を加えたため、これらの変更を継承するよう、Oracle リスナーを必ず再起動します。まだリスナーを再起動していない場合は再起動します。

### 手順 3: Oracle Ultra Search のインストールまたはアップグレード (必要な場合)

Oracle Ultra Search データベースを OID とともに動作するよう構成した後は、Oracle Ultra Search のバックエンド (サーバー・コンポーネント) を Oracle サーバーにインストールまたはアップグレードできます (済んでいない場合)。

### 手順 4: /sys/apps/ultrasearch フォルダの作成

インストールまたはアップグレードの直後、XML DB リポジトリ内に /sys/apps/ultrasearch フォルダを作成するために、SQL スクリプトを実行する必要があります。このフォルダは、すべての Oracle Ultra Search ACL を XML DB で格納します。

/sys/apps/ultrasearch フォルダを作成するには、次の手順を実行します。



1. \$ORACLE\_HOME/ultrasearch/admin ディレクトリに移動します。
2. SQL\*Plus を使用して、ユーザー WKSYS として Oracle Ultra Search データベースにログインします。
3. @wk0prepxdb.sql SQL スクリプトを起動します。

**関連項目：** WKSYS パスワードの変更方法の詳細は、3-2 ページの「[Oracle Ultra Search スキーマのパスワードの変更](#)」を参照してください。

wk0prepxdb.sql スクリプトは、次の SQL を実行することにより、Oracle Ultra Search に関連のあるすべての XML DB リソースを終了時にリスト表示します。

```
SELECT any_path FROM resource_view WHERE any_path LIKE '%ultrasearch%';
```

この SQL 文を実行すると、次の 2 行が表示されます。

```
/sys/apps/ultrasearch  
/sys/apps/ultrasearch_acl.xml
```

この確認メッセージが表示されない場合、この手順は失敗しており、この先の手順には進めません。前述の手順がすべて正しく実行されていることを再確認してください。

#### 手順 5: Oracle Ultra Search のセキュアな検索機能をオンにする

現在、Oracle-OID が正しくインストールされていることをプログラムによって確認する方法はないため、Oracle Ultra Search のセキュアな検索機能はデフォルトでオフになっています。前述の手順をすべて完了した後、この機能を明示的にオンに設定する必要があります。

#### 手順 6: 検索アプリケーションのセキュアな検索機能をオンにする

Oracle Ultra Search のセキュアな検索機能をオンにする手順は、次のとおりです。

1. SQL\*Plus を使用して、ユーザー WKSYS として Oracle Ultra Search データベースにログインします。
2. PL/SQL API: `exec WK_ADM.SET_SECURE_MODE(1)` を起動します。

引数 (1) は、セキュアな検索をオンにしていることを示します。

セキュアな検索機能をオンにした後、セキュアな検索対応の Oracle Ultra Search インスタンスを作成できます。

---

**注意：** 後で、`WK_ADM.SET_SECURE_MODE(0)` を起動してセキュリティをオフにすることもできます。オフにすると、その後に作成されたインスタンスはすべてセキュアな検索をサポートしないよう指定されます。ただし、既存のセキュアな検索対応インスタンスは変更されません。したがって、Oracle-OID リンクが機能しなくなった場合、保護されているクローラ済ドキュメントには検索を実行できません。

---

Oracle Ultra Search はセキュアな検索をサポートしており、検索ユーザーが表示できる検索基準を満たすドキュメントのみが戻されます。

検索アプリケーションのセキュアな検索機能をオンにする手順は、次のとおりです。

1. Oracle Ultra Search の問合せ (`sample.ear`) を配置します。
2. OID に接続できるよう、OC4J の `jazn.xml` ファイルを編集します。次に例を示します。

```
<jazn provider="LDAP" default-realm="us" location="ldap://localhost:3060">
<property name="ldap.user" value="orcladmin"/>
<property name="ldap.password" value="!welcome"/>
</jazn>
```

3. OC4J を再起動します。
4. `applications/ultrasearch_query/META-INF/orion-application.xml` を編集し、JAZN LDAP をオンにします。
5. `applications/ultrasearch_query/query/WEB-INF/web.xml` を編集し、`usearch.jsp` のログイン機能を有効にします。次に例を示します。

```
<init-param>
<param-name>login enabled</param-name>
<param-value>>true</param-value>
</init-param>
```

6. Apache で `mod_osso` を有効にします。
7. `http://<hostname>:<port>/ultrasearch/query/usearch.jsp` にアクセスしてログイン機能を確認し、セキュアな検索をテストします。

**関連項目：** 1-6 ページ [「セキュアな検索」](#)

## Oracle Ultra Search のバックエンド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 のバックエンド（サーバー・コンポーネント）は、既存の Oracle9i 以降のデータベースにインストールできます。これには、2つの方法があります。

- Oracle Portal Configuration Assistant（OPCA）を使用して Oracle Portal をデータベースにインストールします。これにより、Oracle Ultra Search もインストールされます。
- repCA を使用して Oracle 10g Application Metadata Repository を作成します。この過程で、Oracle Ultra Search がインストールされます。これが最も簡単な方法ですが、すべての OracleAS コンポーネント・スキーマもターゲット・データベースにインストールされるというオーバーヘッドがあります。

---

**注意：** この機能は、OracleAS でのみサポートされています。

repCA を使用した方法については『Oracle Application Server 10g インストール・ガイド』の OracleAS Metadata Repository の既存データベースの使用に関する項で説明されているため、この項では OPCA を使用したデータベースのインストール方法のみ説明します。

---

### データベース要件

データベースの要件は次のとおりです。

- 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（9.0.1.4）以上であること
- 初期化パラメータ・ファイルで JOB\_QUEUE\_PROCESSES が 2 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、クローラを実行するために、Java 1.2 準拠の Java Runtime Environment（JRE）がデータベース・コンピュータにインストールされている必要があります。

---

## OPCA を使用した Oracle Ultra Search のバックエンドの既存の Oracle9i Database へのインストール

Oracle Ultra Search のバックエンド（サーバー・コンポーネント）を既存の Oracle9i Database にインストールするには、次の手順を実行します。

1. Oracle Portal Configuration Assistant (OPCA) を起動し、Oracle Portal をカスタム・データベースにインストールします。OPCA は、Oracle Ultra Search スキーマおよびデータベース・オブジェクトもインストールします。Oracle Ultra Search のシステム・スキーマ 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 が最新のリリースでない場合、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 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:** セキュリティ上の問題が発生しないように、WKSYS パスワードをただちに変更する必要があります。詳細は、3-2 ページの [「Oracle Ultra Search スキーマのパスワードの変更」](#) を参照してください。

OPCA は OracleAS 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、2-12 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 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 Database 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。

**UNIX 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 のデフォルト・インスタンスの構成

Oracle Ultra Search インストーラでは、デフォルトの Oracle Ultra Search テスト・ユーザーに基づいて、デフォルトの独自の Oracle Ultra Search インスタンスが作成されます。したがって、Oracle Ultra Search の機能をインストール後にデフォルトのインスタンスに基づいてテストできます。

デフォルトのインスタンス名は、`WK_INST` です。これは、データベース・ユーザー `WK_TEST` に基づいて作成されます。デフォルトのユーザー・パスワードは、`WK_TEST` です。

セキュリティ上の理由から、`WK_TEST` はインストール後にロックされます。管理者は、`DBA` ロールとしてデータベースにログインし、`WK_TEST` ユーザーをロック解除してから、パスワードを `WK_TEST` に変更する必要があります。(パスワードはインストール後に失効します。)パスワードが `WK_TEST` 以外に変更されている場合、データベースでパスワードを変更した後、Administration ツールの「**インスタンスの編集**」ページを使用して、キャッシュされたスキーマ・パスワードを更新する必要があります。

デフォルト・インスタンスは、Oracle Ultra Search サンプル検索アプリケーションにも使用されます。`data-sources.xml` ファイルを必ず更新してください。

### 関連項目:

- 7-11 ページ [「スキーマ・パスワード」](#)
- 2-22 ページ [「data-sources.xml ファイルの編集」](#)

## 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 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 のサンプル検索アプリケーション

Oracle Application Server 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「OracleAS 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 をインストールして構成する必要があります。

Oracle Database の場合、Oracle Ultra Search Middle-Tier は、データベース・サーバーのインストール時に Oracle Ultra Search のバックエンド（サーバー・コンポーネント）とともにインストールされます。これは、データベース・クライアントの一部でもあります。Oracle Ultra Search Middle-Tier は、Oracle J2EE コンテナ (OC4J) とともにインストールされ、構成されます。

**関連項目：** Oracle Ultra Search Middle-Tier で使用される Infrastructure サービス（たとえば別の Oracle Internet Directory またはメタデータ・リポジトリなど）の変更方法の詳細は、『Oracle Application Server 10g 管理者ガイド』を参照してください。

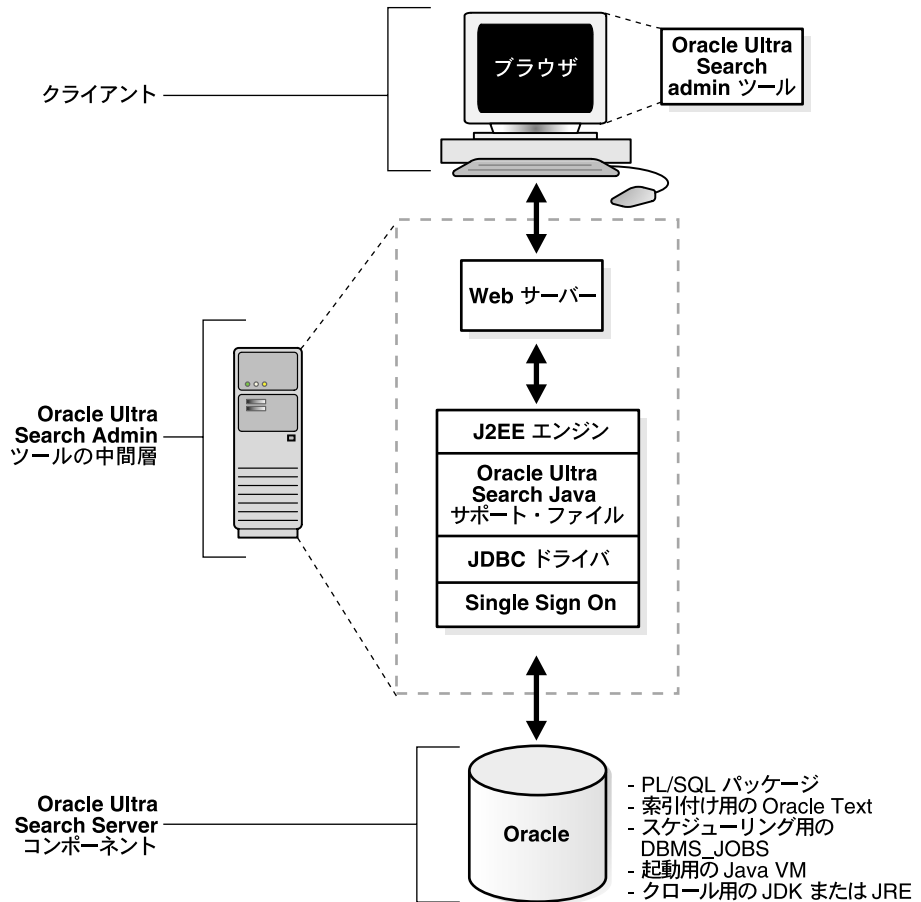
### Web アプリケーションの概要

Oracle Ultra Search Administration ツールおよび Oracle Ultra Search の検索アプリケーションは、J2EE に準拠した 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（第 1 層）、Web サーバーとサーブレット・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。

Web サーバーはブラウザからリクエストを受け取り、リクエストをサーブレット・エンジンに転送して処理します。Oracle Ultra Search Middle-Tier は、次に、図 2-1 に示すように、JDBC を介して Oracle データベースと通信します。

Oracle Ultra Search Administration ツールまたは Oracle Ultra Search のサンプル検索アプリケーションには、すべてのブラウザからアクセスできます。URL については後述します。

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 Database による Oracle Ultra Search Middle-Tier のインストール

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 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。

OUI では、Oracle J2EE コンテナ (OC4J) により、Oracle Ultra Search Middle-Tier が自動的に構成されます。2-24 ページの「[ultrasearch.properties ファイルの編集](#)」に進んでください。

## Oracle Application Server による Oracle Ultra Search Middle-Tier のインストール

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. 「Oracle Application Server 10g」オプションを選択して、「次へ」をクリックします。
2. オプション「Portal and Wireless」を選択して、「次へ」をクリックします。
3. 「構成オプションの選択」画面で、「OracleAS 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 (OPCA) は、Oracle Ultra Search を使用して、Oracle HTTP Server と OC4J を構成できます。このオプションの選択を解除する場合は、「[Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier 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。
4. OracleAS 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「構成オプションの選択」画面で「OracleAS Portal」オプションのチェックを解除し、2-19 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 のすべてのファイルが `$ORACLE_HOME` にコピーされます。

---

---

Oracle Universal Installer の「構成オプションの選択」画面で「OracleAS Portal」オプションを選択した場合は、後述の構成が Oracle Portal Configuration Assistant (OPCA) によって自動的に実行されます。2-22 ページの「[data-sources.xml ファイルの編集](#)」に進んでください。

このオプションを選択しなかった場合は、2-15 ページの「[Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier 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。
- OracleAS Configuration Assistant の詳細は、『Oracle Application Server 10g インストール・ガイド』のトラブルシューティングの付録を参照してください。

## Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier を構成する場合

---

**注意：** Oracle Database では、Oracle Containers for J2EE (OC4J) はデフォルトで構成されています。Oracle HTTP Server および OC4J を手動で構成することもできますが、異なる Oracle ホームに格納されます。

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 OC4J の使用をお勧めします。

---

**関連項目：** サード・パーティの J2EE コンテナまたはサーブレット・エンジンを使用する場合は、2-19 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、OracleAS Containers for J2EE 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="UltrasearchAdmin" path="$ORACLE_HOME/ultrasearch/webapp/ultrasearch_admin.ear" />
```

```
<application name="UltrasearchQuery" path="$ORACLE_HOME/ultrasearch/sample.ear" />
```

```
<application name="UltrasearchPortlet" 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、2-26 ページの「[Oracle Ultra Search の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、2-24 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> タグの下に、次の行を追加します。

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

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

<web-app application="UltrasearchPortlet" 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` について、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:port/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/           OC4J_Portal
Oc4jMount /ultrasearch/*          OC4J_Portal
Oc4jMount /ultrasearch/query      OC4J_Portal
Oc4jMount /ultrasearch/query/*    OC4J_Portal
Oc4jMount /ultrasearch/ohw        OC4J_Portal
Oc4jMount /ultrasearch/ohw/*      OC4J_Portal
Oc4jMount /ultrasearch/admin_sso  OC4J_Portal
Oc4jMount /ultrasearch/admin_sso/* OC4J_Portal
Oc4jMount /ultrasearch/admin      OC4J_Portal
Oc4jMount /ultrasearch/admin/*    OC4J_Portal
```

3. 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。Oracle9i リリースでの 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。すべてのサンプル JSP ソース・コードは `OC4J` アプリケーション・ディレクトリ内にあります。

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。

- 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：** シングル・サインオンは、Oracle Identity Management インフラストラクチャでのみ使用可能です。

---

---

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 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="UltrasearchAdmin" 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/* OC4J_Portal
```

### 3. 次の点を確認します。

- \$ORACLE\_HOME/Apache/Apache/conf/httpd.conf に oracle\_apache.conf が含まれていること
- \$ORACLE\_HOME/Apache/Apache/conf/oracle\_apache.conf に ultrasearch.conf が含まれていること
- \$ORACLE\_HOME/ultrasearch/webapp/config/ultrasearch.conf に次の内容が含まれていること

```
# add alias for ultra search online help and welcome page
```

```
Alias /ultrasearch/doc/ "/private/nli/ora9ias/ultrasearch/doc/"
```

```
Alias /ultrasearch/ "/private/nli/ora9ias/ultrasearch/sample/"
```

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

## 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 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。
- 2-15 ページ「[Oracle HTTP Server および OC4J とともに Oracle Ultra Search Middle-Tier を構成する場合](#)」

**Administration ツールの配置** Oracle Ultra Search Administration ツールは、J2EE に準拠した Web アプリケーションです (\$ORACLE\_HOME/ultrasearch/webapp/ultrasearch\_admin.ear)。Oracle Enterprise Manager を使用してこの Web アプリケーションを配置または配置解除することもできます。

ultrasearch\_admin のファイル構造を参照するには、次のコマンドを実行します。

```
jar -tvf ultrasearch_admin.ear
META-INF/
META-INF/application.xml
META-INF/orion-application.xml
admin.war
admin_sso.war
ohw.war
```

**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、次のコマンドを実行します。

```
jar tf sample.ear

META-INF/application.xml
META-INF/orion-application.xml
query.war
welcome.war
rewriter/SampleRewriter.java
agent/SampleAgent.java
agent/README.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/ldap/jlib/ldapclnt9.jar
$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: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、2-22 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、次のコマンドを実行します。

```
jar -xvf ultrasearch_portlet.ear

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 接続に戻す必要があります。このデータ・ソースには、`oracle.jdbc.pool.OracleConnectionFactoryImpl` 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、2-22 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：** `data-sources.xml` にクリア・テキストのパスワードを格納すると、セキュリティ上のリスクが発生します。これを防ぐには、パスワードの間接化を使用してパスワードを指定します。これにより、自動的に暗号化される `jazn-data.xml` にパスワードを入力し、`data-sources.xml` からこのパスワードを指し示すことができます。詳細は、『Oracle Application Server Containers for J2EE セキュリティ・ガイド』の「間接パスワードの作成」を参照してください。

---

---

### Oracle Application Server Infrastructure

Oracle Application Server 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・コンピュータのホスト名、`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。

デフォルトの Oracle Ultra Search インスタンス・ユーザー `wk_test` にデータソースを追加する場合、まず `wk_test` のロックを解除する必要があります。



**関連項目：** 2-11 ページ「[Oracle Ultra Search のデフォルト・インスタンスの構成](#)」

```
<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：** 『Oracle Application Server Containers for J2EE セキュリティ・ガイド』

## データベース・リリース

データベース・リリースについては、前述と同じ指示に従ってください。ただし、`data-sources.xml` ファイルは若干異なります。

```
<data-source
  class="oracle.jdbc.pool.OracleDataSource"
  name="UltraSearchDS"
  location="jdbc/UltraSearchPooledDS"
  username="wk_test"
```

```
password="wk_test"  
url="jdbc:oracle:thin:@dlsun1517.us.oracle.com:5521:dczade4"  
connectionCachingEnabled="true"/>
```

## ultrasearch.properties ファイルの編集

\$ORACLE\_HOME/ultrasearch/webapp/config/ultrasearch.properties ファイルには、Oracle Ultra Search Middle-Tier で使用される構成情報が含まれています。このファイルは、Oracle Installer によって自動的に構成されるため、編集する必要はありません。

次に、ultrasearch.properties ファイルの例を示します。

```
connection.driver=oracle.jdbc.driver.OracleDriver  
connection.url=jdbc:oracle:thin:@ldap://dlsun8888.cn.oracle.com:3060/iasdb,cn=oracle  
context  
oracle.net.encryption_client=REQUESTED  
oracle.net.encryption_types_client=(RC4_56,DES56C,RC4_40,DES40C)  
oracle.net.crypto_checksum_client=REQUESTED  
oracle.net.crypto_checksum_types_client=(MD5)  
oid.app_entity_cn=m16bi.sgtcnsun03.cn.oracle.com  
domain=us.oracle.com
```

ここで、

- connection.driver は、使用している JDBC ドライバを指定します。
- connection.url は、Middle-Tier の接続先データベースを指定します。Oracle Ultra Search では、次の形式をサポートしています。
  - *host:port:SID* (*host* は Oracle Ultra Search を実行している Oracle Database インスタンスのホスト名のフルネーム、*port* は Oracle Database インスタンスのリスナー・ポート番号、*SID* は Oracle Database インスタンスの ID)
  - HA が認識される文字列 (たとえば TNS キーワード - 値の構文)

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.net.encryption\_client、oracle.net.encryption\_types\_client、oracle.net.crypto\_checksum\_client および oracle.net.crypto\_checksum\_types\_client は、データベースへのセキュアな JDBC 接続のプロパティを制御します。詳細は、『Oracle9i JDBC 開発者ガイドおよびリファレンス』を参照してください。
- oid.app\_entity\_cn は、Ultra Search Middle-Tier アプリケーションのエンティティ名を指定します。

- domain は、IM (Oracle Identity Management) のマシンと Oracle Ultra Search Middle-Tier のマシンの共通ドメインを指定します。これにより、Delegated Administration Service (DAS) の値リストを Internet Explorer で使用できます。たとえば、Oracle Ultra Search Middle-Tier が us.oracle.com であり、Oracle Identity Management マシンが uk.oracle.com である場合、共通ドメインは oracle.com です。ultrasearch.properties に次の行を追加します。domain=oracle.com

## Web サーバーの起動

OracleAS の場合、Oracle Enterprise Manager Application Server Control を使用して Web サーバーを起動します。

**関連項目：** Application Server Control の詳細は、『Oracle Application Server 10g 管理者ガイド』を参照してください。

Oracle Database の場合、次の手順を実行します。

```
java -jar $ORACLE_HOME/j2ee/home/oc4j.jar -config  
$ORACLE_HOME/j2ee/OC4J_SEARCH/config/server.xml
```

## 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:port/ultrasearch/admin/index.jsp` を表示します。  
*hostname.domainname* は Oracle Ultra Search Middle-Tier をインストールしているホストのフルネーム、*port* はデフォルトの Web サーバー・ポートです。
- Oracle Ultra Search のバックエンド (サーバー・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。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:port/ultrasearch/query/search.jsp` を表示します。
- 「Oracle Ultra Search へようこそ」 ページ (`http://hostname.domainname:port/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 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:port/ultrasearch/query/` です。

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

OracleAS 問合せの URL ルートは、`http://hostname.domainname:port/ultrasearch/query/` にあります。

(たとえば、`search.jsp` にアクセスするには、`http://hostname.domainname:port/ultrasearch/query/search.jsp` を使用します。)

Oracle Database の問合せ (Oracle9i の Query API と、`gsearch.jsp` を含む問合せ JSP) は、`$ORACLE_HOME/ultrasearch/sample/query/9i/` にあります。

Oracle Database 問合せの URL ルートは、`http://hostname.domainname:port/ultrasearch/query/9i/` にあります。

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:port/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:port/ultrasearch/query/tag/` です。

## Oracle Ultra Search のバックエンドのリモート・クローラ・ホストへのインストール

Oracle Ultra Search のリモート・クローラでは、異なるホスト上で複数のクローラを並列に実行できます。ただし、すべてのリモート・クローラ・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 のバックエンドのリモート・クローラ・ホストへのインストール

Oracle Ultra Search のリモート・クローラは、Oracle Ultra Search のバックエンド（サーバー・コンポーネント）の一部です。そのため、リモート・クローラのインストール手順は、Oracle Ultra Search のバックエンドのインストール手順と同じです。

Oracle Ultra Search のバックエンドは、各リモート・クローラ・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。

OracleAS 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、リモートのクローラ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 `$REMOTE_ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/unix/define_env` を指定し、Windows システムの場合は `$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 のバックエンドをリモート・クローラ・ホストで構成する場合

Oracle Ultra Search のリモート・クローラ・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・クローラのホスト上で 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 OracleAS Middle-Tier に接続し、リモート・クローラ・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 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 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 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・ユーザーのパスワードを指定します。

WKSYS スーパー・ユーザーまたはスーパー・ユーザー権限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。

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_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 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・クローラ・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 ホストでは /u01/oracle9i、Windows ホストでは D:/u01/oracle9i です。(Windows ホストでは、スラッシュを使用していることに注意してください。)

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

## リモート・クローラの登録解除

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・クローラ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・クローラ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。

## ホスティングされた環境での Oracle Ultra Search の構成

Oracle Ultra Search は、デフォルト・インストールでは、ホスティングされていない環境用に構成されます。ホスティングされた環境に変更するには、次の手順を実行し、Oracle Ultra Search をホスティングされた環境で構成します。

### ホスティングされた環境用の構成前の作業

ホスティング・モードが有効になっていることを確認します。また、OID サーバーでサブスクライバが作成されていることを確認します。

**関連項目：** ホスティング・モードを有効にする手順は『Oracle Application Server Portal 構成ガイド』の **Out-of-Box Portal** での **ホスティングの有効化**に関する項、SSO/OID へのサブスクライバの追加手順は **サブスクライバの追加**に関する項を参照してください。

### サブスクライバ・コンテキストでの Oracle Ultra Search の構成

各サブスクライバについて、次のスクリプトを実行し、Oracle Ultra Search を OID のサブスクライバ・コンテキストで構成します。スクリプトにより、次のことが実行されます。

- サブスクライバ・コンテキストに参照オブジェクトが作成されます。
- サブスクライバ・コンテキストにデフォルトの権限グループ・エントリが作成されません。
- Oracle Ultra Search のメタデータ・リポジトリでサブスクライバ情報が更新されます。

次のようにスクリプトを使用します。

```
ORACLE_HOME/ultrasearch/setup/usca.sh -action add_subscriber -user <the OID user DN> -password <password of the user 'orcladmin'> -subscriber <the DN of the subscriber>
```

OID ユーザーは iASAdmins 権限を所有している必要があります。スクリプトを実行する前に、スクリプトの実行権限があることを確認し、ORACLE\_HOME 環境変数を設定します。

次の例では、サブスクライバ 'dc=us, dc=oracle, dc=com' で Oracle Ultra Search を構成します。

```
ORACLE_HOME/ultrasearch/setup/usca.sh -action add_subscriber -user 'cn=orcladmin' -password welcome1 -subscriber 'dc=us,dc=oracle,dc=com'
```

サブスクライバを削除するには、まず次のスクリプトを実行して OID のサブスクライバ・コンテキストから Oracle Ultra Search のエントリを削除します。

```
ORACLE_HOME/ultrasearch/setup/usca.sh -action remove_subscriber -user <the OID user DN> -password <password of the user 'orcladmin'> -subscriber <the DN of the subscriber>
```



---

## インストール後の情報

この章では、次の項目について説明します。

- Oracle Ultra Search スキーマのパスワードの変更
- 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
- Oracle Ultra Search のアップグレード
- 検索アプリケーションの構成

## Oracle Ultra Search スキーマのパスワードの変更

インストール時に、WKSYS および WKPROXY という 2 つの Oracle Ultra Search システム・スキーマが作成されます。スキーマのパスワードを更新するには、次の手順を実行します。

### Oracle Database 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（または WKPROXY）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（または WKPROXY）をロック解除します。

```
ALTER USER WKSYS ACCOUNT UNLOCK;
```

### Oracle Application Server または Oracle Collaboration Suite 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（または WKPROXY）としてログインするには、Oracle Enterprise Manager の「**Infrastructure**」ページから「スキーマ・パスワードの変更」リンクをたどり、WKSYS（または WKPROXY）スキーマのパスワードを変更します。

## 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。

### 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 100MB 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 100m;
ALTER DATABASE ADD LOGFILE 'redo_log_directory/newredo2.log' size 100m;
ALTER DATABASE ADD LOGFILE 'redo_log_directory/newredo3.log' size 100m;
```

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合的に UNDO と呼ばれます。Oracle Universal Installer によって作成される UNDO 領域は、通常は小さすぎる場合があります。自動 UNDO 管理を使用し、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。

### Oracle 初期化パラメータのチューニング

初期化ファイルで次の値を設定します。

- PROCESSES: 50 以上に設定します。
- SORT\_AREA\_SIZE: 5MB 以上に設定します。
- SORT\_AREA\_RETAINED\_SIZE: 5MB 以上に設定します。
- JOB\_QUEUE\_PROCESSES: 3 以上に設定します。(少なくとも 1 に設定します。) Oracle Ultra Search クローラは、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 クローラが起動しません。この値を設定する場合、ジョブ・キュー・プロセスの他の要件についても考慮してください。

Oracle Ultra Search に関連する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、小さすぎることがあります。Oracle Ultra Search は、Oracle Text エンジンを集散的に使用します。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 デバイス上に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。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。

次に、新規表領域の作成方法の例を示します。

```
CREATE TABLESPACE lmtbsb DATAFILE '/u02/oracle/data/lmtbsb01.dbf' SIZE 150M;
```

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場合があります。この場合、より多くの記憶領域が必要になるたびに、新しいエクステントを割り当てることとなります。extent management 句で新しいエクステントが現在のエクステント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トが、使用可能なエクステントより大きくなる場合があります。このような場合は、表領域に新しいエクステント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、初期エクステント・サイズ、次のエクステント・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$I  
    (initial extent size 5M, next extent size 50M, PCTINCEASE 1)  
DR$WK$DOC_PATH_IDX$K  
    (initial extent size 5M, next extent size 50M, PCTINCEASE 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 デバイス上に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 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。

---

---

**注意：** Oracle Ultra Search 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

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)。

---

---

新規 Oracle Ultra Search インスタンスをホストするデータベース・ユーザーに対し、WKUSER ロールを付与する必要があります。

**関連項目：** 7-41 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、Administration ツールの「ユーザー」ページをクリックして進み、該当する権限を付与します。

## 手順 5: 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：** クローラは、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）。

## インスタンスのストップリストの変更

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。

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。最初に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 クローラ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。

### 初回のクロール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場合

たとえば、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');
```

**関連項目：** WKSYS パスワードの変更方法の詳細は、3-2 ページの「[Oracle Ultra Search スキーマのパスワードの変更](#)」を参照してください。

## 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、クロール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。

たとえば、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 クローラ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。

## Oracle Ultra Search のアップグレード

Oracle Ultra Search は、Oracle Database、Oracle Application Server および Oracle Collaboration Suite に付属しています。Oracle Ultra Search を以前のリリースから最新のリリースにアップグレードするには、使用している製品により、異なる手順を実行する必要があります。

この項では、次の項目について説明します。

- [アップグレード前の手順](#)
- [Oracle Database 付属の Oracle Ultra Search のアップグレード](#)
- [Oracle Application Server 付属の Oracle Ultra Search のアップグレード](#)
- [Oracle Collaboration Suite 付属の Oracle Ultra Search のアップグレード](#)
- [Oracle Collaboration Suite リリース 1 への Oracle Ultra Search のアップグレード](#)

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

### アップグレード前の手順

アップグレードを開始する前に、Oracle Ultra Search Administration ツールにログオンします。すべての Oracle Ultra Search インスタンスのすべてのクローラの同期スケジュールを停止し、無効にします。アップグレード後、すべてのクローラの同期スケジュールを有効にできます。同期スケジュールを停止し、無効にする方法の詳細は、7-32 ページの「[スケジュール](#)」タブを参照してください。

### Oracle Database 付属の Oracle Ultra Search のアップグレード

Oracle Database 付属の Oracle Ultra Search をアップグレードするには、次の手順を実行します。

1. Oracle Ultra Search のバックエンド（サーバー・コンポーネント）のアップグレードを実行します。これには、Oracle Ultra Search のデータベース・スキーマおよびサーバー・ファイルのアップグレードが含まれます。新しい Oracle ソフトウェアをインストールし、Oracle Database Upgrade Assistant を実行してデータベースと Oracle Ultra Search コンポーネントを新規リリースにアップグレードします。詳細は、『Oracle Database アップグレード・ガイド』を参照してください。
2. 2-12 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](#)」の手順に従い、新規 Oracle Ultra Search Middle-Tier をインストールします。

## Oracle Application Server 付属の Oracle Ultra Search のアップグレード

Oracle Application Server 付属の Oracle Ultra Search をアップグレードするには、次の手順を実行します。

1. 新規 Oracle Application Server をインストールし、Oracle Application Server Upgrade Assistant を使用して Middle-Tier をアップグレードします。詳細は、『Oracle Application Server 10g アップグレード』の Middle-Tier のアップグレードに関する項を参照してください。
2. Oracle Application Server Metadata Repository の Oracle Ultra Search スキーマをアップグレードします。詳細は、『Oracle Application Server 10g アップグレード』の Metadata Repository のアップグレードおよび Oracle Ultra Search スキーマのアップグレード・スクリプトに関する項を参照してください。

## Oracle Collaboration Suite 付属の Oracle Ultra Search のアップグレード

Oracle Collaboration Suite リリース 1 を使用しており、最新の Oracle Collaboration リリースにアップグレードする場合、最新の Oracle Collaboration Suite リリースをインストールし、Oracle Collaboration Suite Upgrade Assistant を使用して Oracle Ultra Search Middle-Tier およびバックエンド（サーバー・コンポーネント）の両方をアップグレードします。詳細は、『Oracle Collaboration Suite インストールおよび構成ガイド』を参照してください。

Oracle Ultra Search 9.0.2（Oracle Application Server 付属）または Oracle Ultra Search 1.0.3 または 9.2（Oracle Database 付属）を使用しており、最新の Oracle Collaboration リリースにアップグレードする場合、次のアップグレード手順を実行します。

1. まず、Oracle Collaboration Suite リリース 1 ソフトウェアを入手し、Oracle Ultra Search を Oracle Collaboration Suite リリース 1 にアップグレードします。詳細は、3-12 ページの「[Oracle Collaboration Suite リリース 1 への Oracle Ultra Search のアップグレード](#)」を参照してください。
2. 最新の Oracle Collaboration Suite リリースをインストールし、Oracle Collaboration Suite Upgrade Assistant を使用して Oracle Ultra Search Middle-Tier およびバックエンドをアップグレードします。詳細は、『Oracle Collaboration Suite インストールおよび構成ガイド』を参照してください。

## Oracle Collaboration Suite リリース 1 への Oracle Ultra Search 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 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 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 のバックエンド（サーバー・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。

### Oracle Ultra Search リリース 1.0.3 からリリース 9.0.3 へのアップグレード

Oracle Ultra Search 1.0.3（Oracle9i Database リリース 1（9.0.1））から 9.0.3 にアップグレードするには、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、クローラ・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クローラ・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。移行後も、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問い合わせることができます。

**関連項目：** 2-9 ページ「[Oracle Ultra Search のバックエンド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。

- `SYSPW`: ユーザー `SYS` のパスワード
- `WKSYSPW`: ユーザー `WKSYS` のパスワード
- `HOST`: データベース・ホスト・コンピュータ
- `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 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ल・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、同じコンピュータまたは異なるコンピュータの別の 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 システムと同じコンピュータ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. 後に 9.0.2 にダウングレードする場合に備え、OracleAS 9.0.2 Infrastructure 層 \$ORACLE\_HOME/ultrasearch/ 内のすべての Oracle Ultra Search 9.0.2 ファイルを再帰的に別のディレクトリにコピーします。
2. Oracle Ultra Search 9.0.2 Administration ツールにログオンします。すべての Oracle Ultra Search インスタンスのすべてのクローラの同期スケジュールを停止し、無効にします。
3. Oracle Collaboration Suite リリース 1 のインストーラを起動し、Infrastructure をインストールします。
4. Oracle ホームには OracleAS 9.0.2 Infrastructure のディレクトリを指定します。
5. これにより、Oracle Universal Installer は以前にインストール済のデータベースを検出し、Infrastructure データベースおよび Oracle Ultra Search のバックエンド（サーバー・コンポーネント）を自動的にアップグレードします。

## 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：** 3-15 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」](#)

## 検索アプリケーションの構成

Oracle Ultra Search 検索アプリケーションは、Oracle Ultra Search のインストール時に自動的に配置されます。ただし、Oracle Ultra Search では複数のインスタンスによる異なるスキーマ・ユーザーの使用が許可されているため、検索アプリケーションは自動的にデータベースに接続できるよう構成されていません。データベース接続は、OC4J にデータソースを作成することによって構成されます（Oracle Ultra Search のデータソースではありません）。これを行うには、`data-sources.xml` ファイルを編集します。

### 手順 1: data-sources.xml ファイルの編集

`data-sources.xml` ファイルは、OC4J の接続管理機能です。Oracle Ultra Search 検索アプリケーションは、OC4J を使用してデータベースに接続します。この点は Administration ツールと異なります。問合せユーザーはデータベース・ユーザーではないので、データベースのログイン・パスワードを知らないためです。

`data-sources.xml` を編集することにより、データベース・ユーザーとパスワード情報が OC4J で構成されます。Oracle Ultra Search 検索アプリケーションは、データソースの位置 `jdbc/UltraSearchPooledDS` を使用することにより、データソースを検出します。

**関連項目：** 2-22 ページ [「data-sources.xml ファイルの編集」](#)

### 手順 2: 複数インスタンスに対する複数の検索アプリケーションの配置

Oracle Ultra Search では複数のインスタンスによる異なるスキーマ・ユーザーの使用が許可されているため、複数の検索アプリケーションが同じデータベースに共存できます。

各検索アプリケーションは、データベースへの接続情報が `data-sources.xml` で定義されている必要があります。`jdbc/UltraSearchPooledDS1`、`jdbc/UltraSearchPooledDS2` などのように、それぞれ異なる位置の値を定義する必要があります。これに対応して、検索アプリケーションは複数回 OC4J で配置される必要があります。

**関連項目：** 2-20 ページ [「サンプル検索アプリケーションの配置」](#)

最後に、各アプリケーションの配置が、`data-sources.xml` 内の正しいエントリを使用するよう構成する必要があります。これを行うには、問合せの JSP ソースを編集します。検索アプリケーション全体については、`common_customize_instance.jsp` を編集し、次の行を編集して正しい位置の値を使用するように設定します。

```
String m_datasource_name = "jdbc/UltraSearchPooledDS";
```



---

---

## チューニングおよびパフォーマンス

この章では、次の項目について説明します。

- [Web クロール・プロセスのチューニング](#)
- [問合せパフォーマンスのチューニング](#)
- [リモート・クローラの使用法](#)
- [Oracle Ultra Search の Real Application Clusters](#)
- [表データ・ソースの同期化](#)

## Web クロール・プロセスのチューニング

Oracle Ultra Search クローラは、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（表または電子メール・データ・ソースなど）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を、クローラ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かず。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、クローラがある 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、クローラ統計を監視します。
- 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、クローラは、最終的にはそのサイト内のすべてのドキュメントを分析するため、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 のパラメータはリンクごとに異なります。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 ツールでクローラ統計を監視し、最も多くクロールされている URL および Web サーバーを判断します。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クローラによる 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には制限できません)。
- クロールの深さの制限: クローラ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。

「クローラ」タブでパラメータを変更した後は、必ずクローラを再起動してください。設定の変更は、クローラ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

この項では、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。

- DB\_CACHE\_SIZE 初期化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。

- 索引の最適化

クローラ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。そのためには、索引の最適化を定期的にスケジューリングします。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:** 7-36 ページ「索引最適化」

- 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。

- 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：**

- 8-4 ページ「問合せ構文拡張のカスタマイズ」
- oracle.ultrasearch.query.Query のインタフェースについては、『Oracle Ultra Search API Reference』を参照してください。

- 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。

- 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。Oracle Application Server 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>
```

- 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・クローラの使用法

Oracle Ultra Search のリモート・クローラがない場合は、Oracle Database と同じホストで Oracle Ultra Search クローラ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、Oracle Database とは別の 1 つ以上のホストで Oracle Ultra Search クローラを実行すると、パフォーマンスが向上します。Oracle Ultra Search クローラは Pure Java アプリケーションで、JDBC を介して Oracle Database と通信します。

Oracle Ultra Search のリモート・クローラのインスタンスは、常に Oracle Database によって起動されます。リモート・クローラを Oracle Database から起動することによって、Oracle Database 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 Oracle Database 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。

Oracle Database は、Java Remote Method Invocation (RMI) を使用してリモート・クローラ・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。

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・クローラ・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 クローラを起動します。Oracle Ultra Search クローラは、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。

---

---

**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・クローラ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を盗用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・クローラ機能は使用しないでください。

---

---

## スケーラビリティおよびロード・バランシング

Oracle Ultra Search スケジュールは、それぞれ1つのクローラに関連付けられます。クローラは、Oracle 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・クローラ・ホストの数に制限はありません。ただし、各リモート・クローラ・ホストに、Oracle Ultra Search Middle-Tier 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・クローラ・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、クローラ・プロセス全体のスケーラビリティおよびロード・バランシングが実現します。

## インストールと構成の順序

1. Oracle Database に Oracle Ultra Search のバックエンド（サーバー・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 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・クローラ・ホストに Oracle Ultra Search Middle-Tier 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・クローラ・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・クローラ・ホストと共有します。

リモート・クローラ・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るとします。各ホストはそれぞれ、Oracle Ultra Search のバックエンド（サーバー・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 上で実行されているリモート・クローラ・ホスト（ホスト Y1）、Solaris 上で実行されているリモート・クローラ・ホスト（ホスト Y2）、そして Linux 上で実行されているリモート・クローラ・ホスト（ホスト Y3）であるとしてします。

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 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。ホスト X が Linux サーバーの場合

合は、Samba 共有ディレクトリを作成して Windows 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・クローラを構成します。

定義した共有クローラ・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・クローラ・プロファイルを編集します。リモート・クローラ・プロファイルを編集するには、「クローラ」タブの「リモート・クローラ・プロファイル」サブタブに進み、編集するリモート・クローラ・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・クローラ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・クローラ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・クローラ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、クローラを始める前に、次のクローラ・パラメータを指定する必要があります。

- リモート・クローラがドキュメント収集用に使用するクローラ・スレッドの数
- リモート・クローラ・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、クローラ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・クローラまたはローカルクローラに割り当てる必要があります。(ローカルクローラは、Oracle のローカル・データベース・ホスト上で実行されるクローラです。) スケジュールをリモート・クローラ・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」ページ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・クローラ機能を停止し、スケジュールを使用して特定のリモート・クローラ・ホストのかわりにローカル・データベース・ホスト上のクローラを起動することもできます。リモート・クローラ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・クローラ・ホストが選択されている場合は、リモート・クローラを使用可能または使用禁止にできます。

**関連項目：** [第7章「Oracle Ultra Search Administration ツールについて」](#)



5. 各リモート・クローラ・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・クローラが 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・クローラが Windows 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・クローラを起動し、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・クローラ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・リストのクローラ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・クローラは起動できません。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・クローラの起動後は、次のいずれかの方法で、リモート・クローラが実行されていることを確認してください。

- リモート・クローラ・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・クローラがリモート・クローラ・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・クローラ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。

## Oracle Ultra Search の Real Application Clusters

Oracle Ultra Search は、Real Application Clusters システムの記憶域アクセスの構成によって、固定ノードまたは任意のノードをクロールできます。必要に応じて、クローラを実行するノードを指定するための PL/SQL API が用意されています。Oracle Ultra Search Administration および Oracle Ultra Search の検索アプリケーションの場合、Real Application Clusters の任意のノードに接続するように、接続文字列を構成できます。

**関連項目：** Oracle Database Real Application Clusters のマニュアル

### 記憶域アクセスの構成

Real Application Clusters システムの任意のノードのディスクは、共有（クラスタ・ファイル・システム）または非共有（RAW ディスク）にできます。クラスタ・ファイル・システム（CFS）上の Real Application Clusters の場合、任意のノード上のクローラによって生成されたキャッシュ・ファイルは任意の Oracle インスタンスに表示され、索引同期を実行する Oracle インスタンスによって索引付けできます。ディスクを共有しない場合はクローラを特定の Oracle インスタンスで実行し、すべてのキャッシュ・ファイルが索引付けされるようにする必要があります。

これは、Oracle Text の索引付けの性質によるものです。Oracle Text の索引付けでは、異なるセッションによって 1 つの表に挿入された行は同じペンディング・キューに入り、索引同期を開始したユーザーが、挿入されたすべての行を索引付けします。この制限により CFS では、Oracle Ultra Search は任意のデータベース・インスタンスでクローラを起動するように構成されます。CFS 以外の場合、Oracle Ultra Search は、`INSTANCE_NUMBER = 1` のデータベース・インスタンスでクローラを起動します。

Oracle Ultra Search 管理者は、次の PL/SQL API を使用してクローラを実行するインスタンスを構成できます。

```
WK_ADM.SET_LAUNCH_INSTANCE(instance_name, connect_url);
```

`instance_name` は起動インスタンスの名前（任意のノードで起動される場合はデータベース名）、`connect_url` は接続記述子です。

単一のデータベース・インスタンスに接続する場合、記述子は略式の「*host:port:SID*」、または接続記述子（Oracle Net のキーワード値ペア）にできます。次に例を示します。

```
(DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=cls02a)(PORT=3999)))(CONNECT_DATA=(
SERVICE_NAME=acme.us.com)))
```

任意のデータベース・インスタンスに接続するには、完全なデータベース接続記述子を使用する必要があります。次に例を示します。

```
(DESCRIPTION=(LOAD_BALANCE=yes)(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=cls02a)(PORT=3999
))(ADDRESS=(PROTOCOL=TCP)(HOST=cls02b)(PORT=3999)))(CONNECT_DATA=(SERVICE_
NAME=acme.us.com)))
```

**関連項目：** 構成の詳細は、『Oracle9i JDBC 開発者ガイドおよびリファレンス』を参照してください。

クラスタ・ファイル・システム以外のノードでクローラを起動するように、Oracle Ultra Search を構成することはできません。

既存の起動インスタンス構成を問い合わせるには、次の PL/SQL API を使用します。

```
WK_ADM.GET_LAUNCH_INSTANCE RETURN VARCHAR2;
```

これにより起動インスタンスの名前、またはデータベース名（任意のノードがクローラを起動する場合）が戻されます。

## リモート・クローラのファイル・キャッシュ

Oracle Ultra Search のリモート・クローラでは、索引付けのためにリモート・ファイル・システムを Oracle インスタンスにマウントする必要があります。

クラスタ・ファイル・システムの Real Application Clusters の場合、リモート・コンピュータのファイル・システムは、システムの全ノードに NFS でマウントされる必要があります。

クラスタ・ファイル・システム以外の Real Application Clusters の場合、NFS マウントは、Oracle インスタンスがリモート・クローラを提供している特定のノードに制限されます。すべてのノードにリモート・ファイル・システムをマウントしても効果はありません。ノードが停止したときに、古い NFS ハンドルになる可能性があります。別の Oracle インスタンスに移動するように構成が変更された場合、リモート・ファイル・システムはそれに従って、新しいノードに NFS でマウントする必要があります。

## Oracle インスタンスへのログオン

Oracle Ultra Search のすべてのコンポーネントは、JDBC Thin Driver、および「hostname:port:SID」で構成される接続文字列、または tnsnames.ora で参照される完全な接続記述子を使用します。

管理中間層は、ultrasearch.properties ファイルに指定された JDBC 接続を使用して、Oracle データベースに接続します。クライアントが提供するノードが停止した場合は、別の Oracle インスタンスに接続するように、ultrasearch.properties ファイルを手動で編集する必要があります。

### Real Application Clusters の検索アプリケーション

問合せコンポーネントは、Real Application Clusters を十分に活用する必要があります。Real Application Clusters 内の任意の Oracle インスタンスに接続できるように、JDBC 接続文字列をデータベース接続記述子として指定できます。次に例を示します。

```
"jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=yes) (ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=cls02a) (PORT=3999
)) (ADDRESS=(PROTOCOL=TCP) (HOST=cls02b) (PORT=3999))) (CONNECT_DATA=(SERVICE_
NAME=acme.us.com)))"
```

**関連項目：**『Oracle9i JDBC 開発者ガイドおよびリファレンス』

### Java クローラ

Oracle Ultra Search クローラで使用される接続文字列は、インストール時に初期化され、WK\_ADM.SET\_LAUNCH\_INSTANCE API を使用して変更できます。ノードの追加や削除など、システム構成が変更されると、接続文字列も自動的に変更されます。

### JDBC ドライバの選択

Oracle Ultra Search 管理者は、JDBC OCI ドライバを使用してデータベースにログオンするように、ローカルクローラをオプションで構成できます。これは、次の PL/SQL API を使用して実行されます。

```
WK_ADM.SET_JDBC_DRIVER(driver_type)
```

ここで、

- Thin Driver (デフォルト) `driver_type = 0`
- OCI ドライバ `driver_type = 1`

この API には、スーパー・ユーザー権限が必要です。変更はすべての Oracle Ultra Search インスタンスに影響します。

---

---

**注意：** OCI ドライバでは、LD\_LIBRARY\_PATH および NLS\_LANG などの環境変数を、起動データベース・インスタンスに正しく設定する必要があります。クローラは、Oracle プロセスから環境設定を継承します。したがって、Oracle を起動する前に環境変数を正しく構成する必要があります。

---

---

**関連項目：** OCI ドライバを使用する構成の詳細は、『Oracle9i JDBC 開発者ガイドおよびリファレンス』を参照してください。

次の PL/SQL API により、現在使用されている JDBC ドライバの種類を判別します。

```
WK_ADM.GET_JDBC_DRIVER RETURN NUMBER;
```

## 表データ・ソースの同期化

Oracle Ultra Search クローラは、Oracle Ultra Search がインストールされている Oracle のローカル・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 Oracle 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 Oracle 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 クローラでは、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ます。

ログ表の構造のルールは、次のとおりです。

- 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。
- 実表は、主キー列を最大 8 列持つことができます。
- 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。
- 主キー列に対応するログ表の各列の型は、VARCHAR2(1000) です。
- 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。
- mark という列のデフォルト値は F です。

たとえば、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 (表名は自動的に生成されます) 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クローラに示されます。

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クローラに示されます。ログ表の 2 番目の行には、`k1` 列と `k2` 列に、`id` と `name` の新しい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クローラに示されます。

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。



---

---

## Oracle Ultra Search のセキュリティ

Web コンテンツへのユーザー・アクセスを制御する機能は重要です。Oracle Ultra Search のセキュリティのアーキテクチャおよび構成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 のセキュリティについて](#)
- [Oracle Ultra Search 用のセキュリティ・フレームワークの構成](#)
- [Oracle Ultra Search セキュリティの構成](#)

### 関連項目：

- Oracle Application Server のセキュリティとそのコア機能の概要は、『Oracle Application Server 10g セキュリティ・ガイド』を参照してください。
- Oracle セキュリティ・インフラストラクチャのガイダンスは、『Oracle Identity Management 概要および配置プランニング・ガイド』を参照してください。

## Oracle Ultra Search のセキュリティについて

ここでは、Oracle Ultra Search セキュリティ・モデルについて説明します。ここでは、次の項目について説明します。

- [Oracle Ultra Search セキュリティ・モデル](#)
- ユーザーのクラスとその権限
- ホスティングされた環境での Oracle Ultra Search の管理者権限モデル
- Oracle Ultra Search で保護されるリソース
- 認可およびアクセス規定
- Oracle Ultra Search でセキュリティ・サービスが活用される仕組み
- Oracle Ultra Search で Oracle Identity Management インフラストラクチャが活用される仕組み
- Oracle Ultra Search の拡張性とセキュリティ

## Oracle Ultra Search セキュリティ・モデル

情報への無許可のアクセスなどのセキュリティ上の問題により、生産性が失われる可能性があります。Oracle Ultra Search のような検索エンジンでは、単一ゲートウェイで多種多様なコンテンツ・リポジトリにアクセスできます。これらの各リポジトリに独自のセキュリティ・モデルがあり、これによって特定のエンド・ユーザーが特定のドキュメントにアクセスできるかどうかが決まります。Oracle Ultra Search では複数のリポジトリのデータにアクセスできるため、無許可のアクセスを防ぐには、各リポジトリ内の既存のセキュリティ情報がサポートされる必要があります。

ここでは、Oracle Ultra Search のセキュリティ・アーキテクチャについて説明します。セキュリティは次のレベルで実装されます。

- ユーザー認証  
これは、LDAP および Oracle Internet Directory により Oracle Ultra Search フロントエンド・インタフェースで行われるユーザーの識別です。
- ユーザーの資格  
これにより、ユーザーが結果リストの特定の項目に関する情報にアクセスできるかどうかが決まります。これはアクセス制御リスト (ACL) により実装されます。Oracle Ultra Search では、ドキュメントの索引付けおよび Oracle Ultra Search への格納時に各ドキュメントのアクセス制御リストを取得することで、マップされたセキュリティをサードパーティ・リポジトリに提供します。Oracle Ultra Search は、アクセス権限を検証する際にリポジトリ自体に接続する必要はありません。

- セキュアな通信  
コンテンツのクローल、索引付けおよび問合せはすべて、HTTP プロトコルを使用した暗号化の世界標準 (HTTPS) である Secure Sockets Layer (SSL) を使用して暗号化されます。
- Oracle Ultra Search のセキュリティ  
実際の Oracle Ultra Search セキュリティは、Oracle Ultra Search データベース、管理ユーザーおよびパスワード・データで、ディクショナリ・データにより処理されます。

## ユーザーのクラスとその権限

Oracle Ultra Search ユーザーに管理権限を付与するには、ユーザーを管理グループに割り当てる必要があります。各ユーザーは、1 つ以上のグループに属することができます。各 Oracle Ultra Search インスタンスごとに次のグループが作成されます。

- インスタンス管理者: このグループのユーザーは、権限を持つインスタンスのみ管理できます。
- スーパー・ユーザー: このグループのユーザーは、インスタンスの作成、削除および権限の付与を含め、すべてのインスタンスを管理できます。

また、Oracle Ultra Search には次の 2 つのユーザー・クラスがあります。

- シングル・サインオン (SSO) ユーザー: これらのユーザーは、Oracle Internet Directory (OID) により管理され、SSO Server により認証されます。Oracle Ultra Search Administration ツールにより、SSO ユーザーがアクセスできるすべての Oracle Ultra Search インスタンスが識別されます。これは、Oracle Identity Management インフラストラクチャがインストールされている場合のみ使用可能です。
- データベース・ユーザー (non-SSO): これらのユーザーは、Oracle Ultra Search が動作するデータベースに存在します。

## Oracle Ultra Search のデフォルト・ユーザー

Oracle Ultra Search の新規インスタンスには、次のユーザーが含まれます。

- WK\_TEST: これは、WK\_INST と呼ばれるデフォルト・インスタンスをホスティングするインスタンス管理ユーザーです。つまり、WK\_TEST は WK\_INST のインスタンス管理者です。セキュリティ上の理由から、WK\_TEST はインストール後にロックされます。管理者は、DBA ロールとしてデータベースにログインし、WK\_TEST ユーザーをロック解除してから、パスワードを WK\_TEST に変更する必要があります。(パスワードはインストール後に失効します。) パスワードが WK\_TEST 以外に変更されている場合、データベースでパスワードを変更した後、Administration ツールの「**インスタンスの編集**」ページを使用して、キャッシュされたスキーマ・パスワードを更新する必要があります。

- **WKSYS:** これはデータベースのスーパー・ユーザーです。WKSYS は、WK\_TEST などの他のユーザーにスーパー・ユーザー権限を付与できます。すべての Oracle Ultra Search データベース・オブジェクトは、WKSYS スキーマにインストールされます。

---

**注意:** インスタンスのホスティングには、WKUSER ロールが必要です。

---

## ホスティングされた環境での Oracle Ultra Search の管理者権限モデル

ホスティングされた環境では、1つの企業（たとえばアプリケーション・サービス・プロバイダ）が Oracle Ultra Search を他の企業に提供し、その企業の情報を保存します。ホスティングを提供している企業は**デフォルト・サブスクリイバ**と呼ばれ、ホスティングされている企業は**サブスクリイバ**と呼ばれます。

---

**注意:** この機能は、Oracle Application Server リリースおよび Oracle Collaboration Suite リリースで使用できます。Oracle Database では使用できません。

---

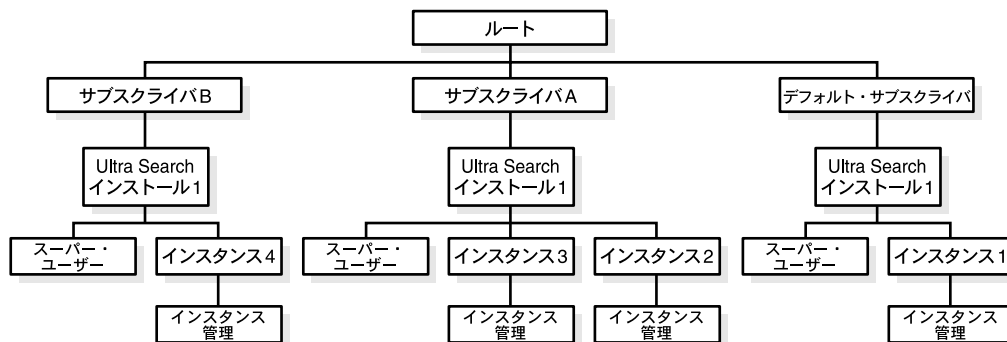
デフォルト・サブスクリイバおよびその検索ベースは、OID エントリ "cn=Common,cn=Products,cn=OracleContext" の次の属性で指定されます。

- orclDefaultSubscriber
- orclSubscriberSearchBase

サブスクリイバが存在しないホスティングされていない環境では、Oracle Ultra Search をインストールしている企業がデフォルト・サブスクリイバです。すべての Oracle Ultra Search 管理グループ（スーパー・ユーザーおよびインスタンス管理者グループ）は、OID のディレクトリ情報ツリー（DIT）のデフォルト・サブスクリイバ Oracle コンテキスト（たとえば、cn=OracleContext,dc=us,dc=oracle,dc=com）下に作成されます。

図 5-1 に、ホスティングされた環境の OID トポロジの例を示します。2つのサブスクリイバ（A および B）とデフォルト・サブスクリイバが存在します。各サブスクリイバには、関連付けられている固有のスーパー・ユーザー権限グループが存在します。Oracle Ultra Search バックエンドの「インストール 1」内には、4つの Oracle Ultra Search のインスタンスが作成されています。「インスタンス 1」はデフォルト・サブスクリイバに関連付けられています。「インスタンス 2」および「インスタンス 3」は「サブスクリイバ A」に関連付けられています。「インスタンス 4」は「サブスクリイバ B」に関連付けられています。各 Oracle Ultra Search インスタンスには、関連付けられているインスタンス管理グループがあります。

図 5-1 ホスティングされた環境の OID トポロジ



## 管理者権限モデル

この項では、ホスティングされた環境での Oracle Ultra Search Administration ツールの権限モデルについて説明します。このモデルは、SSO ログイン・モードと非 SSO ログイン・モードの両方に適用されます。

**Oracle Ultra Search インスタンスを管理するための権限** 非 SSO モードでは、データベース・ユーザーのみ Administration ツールにログインできます。

- ログイン・データベース・ユーザーがスーパー・ユーザー権限を所有している場合、そのユーザーは、デフォルト・サブスクリバとその他すべてのサブスクリバの全 Oracle Ultra Search インスタンスを管理できます。
- ログイン・データベース・ユーザーが特定の Oracle Ultra Search インスタンスの管理者権限のみ所有している場合、そのユーザーは、インスタンスがデフォルト・サブスクリバまたはその他のサブスクリバに関連付けられているかにかかわらず、そのインスタンスを管理できます。

SSO モードでは、SSO ユーザーのみ Administration ツールにログインできます。

- SSO ユーザーがデフォルト・サブスクリバに属している場合、次のことが言えます。
  - SSO ユーザーがスーパー・ユーザー権限を所有している場合、そのユーザーは、デフォルト・サブスクリバとその他すべてのサブスクリバの全 Oracle Ultra Search インスタンスを管理できます（たとえばインスタンス 1、2、3、4）。
  - SSO ユーザーがデフォルト・サブスクリバ内の特定の Oracle Ultra Search インスタンス（たとえばインスタンス 1）の管理者権限を所有している場合、そのユーザーは、デフォルト・サブスクリバに関連付けられているインスタンス（インスタンス 1）を管理できます。

- SSO ユーザーがサブスクリイバに属している場合、次のことが言えます。
  - SSO ユーザーがスーパー・ユーザー権限を所有している場合、そのユーザーは、属しているサブスクリイバの Oracle Ultra Search インスタンスのみ管理できます。(たとえば、サブスクリイバ A のユーザーがスーパー・ユーザー権限を所有している場合、そのユーザーはインスタンス 2 と 3 のみ管理できます。)
  - SSO ユーザーが特定の Oracle Ultra Search インスタンス (たとえばインスタンス 2) の管理者権限を所有している場合、そのユーザーは、サブスクリイバ (サブスクリイバ A) に関連付けられているインスタンス (インスタンス 2) を管理できます。

**Oracle Ultra Search インスタンスの作成および削除権限** Oracle Ultra Search インスタンスを作成または削除するには、ユーザー (データベースまたは SSO ユーザー) はスーパー・ユーザー権限が必要です。

非 SSO モードでは、データベース・ユーザーは、インスタンスを作成または削除したり、デフォルト・サブスクリイバを含めたすべてのサブスクリイバにインスタンスを関連付けることができます。

SSO モードの場合、次のようになります。

- ログイン SSO ユーザーがデフォルト・サブスクリイバに属する場合、そのユーザーは、インスタンスを作成または削除したり、デフォルト・サブスクリイバを含めたすべてのサブスクリイバにインスタンスを関連付けることができます。
- ログイン SSO ユーザーが特定のサブスクリイバに属する場合、そのユーザーがインスタンスを作成すると、ログイン・ユーザーが属するサブスクリイバにインスタンスが作成され、関連付けられます。ユーザーはインスタンス・スキーマ作成のためにデータベースにアクセスできない場合があるため、インスタンスをホスティングするためのデータベース・スキーマを作成するよう、ホスティング会社 (デフォルト・サブスクリイバ) に連絡する必要があります。

**スーパー・ユーザー権限の付与または取消し権限** スーパー・ユーザー権限の付与または取消しを行うには、スーパー・ユーザーとして Administration ツールにログインします。

非 SSO モード (データベース・ユーザー・ログイン) では、スーパー・ユーザーのみが、他のデータベース・ユーザーのスーパー・ユーザー権限を付与または取り消すことができます。

SSO モードの場合、次のようになります。

- ログイン SSO ユーザーがデフォルト・サブスクリイバに属している場合、次のことを実行できます。
  - デフォルト・サブスクリイバに属する SSO ユーザーのスーパー・ユーザー権限の付与または取消し。
  - 特定のサブスクリイバに属する SSO ユーザーのスーパー・ユーザー権限の付与または取消し。

- ログイン SSO ユーザーが特定のサブスクリバに属する場合、ログイン・ユーザーが属するサブスクリバ内のユーザーのスーパー・ユーザー権限を付与または取り消すことができます。

**インスタンス管理者権限の付与または取消し権限** インスタンス管理者権限の付与または取消しを行うには、スーパー・ユーザーまたはインスタンス管理者として **Administration** ツールにログインします。

非 SSO モード (データベース・ユーザー・ログイン) では、スーパー・ユーザーまたはインスタンス管理者のみが、他のデータベース・ユーザーのインスタンス管理者権限を付与または取り消すことができます。

SSO モードの場合、次のようになります。

- ログイン SSO ユーザーは、インスタンスが関連付けられているサブスクリバ内の SSO ユーザーにのみ、インスタンス管理者権限を付与または取り消すことができます。たとえば、サブスクリバ A 内の SSO ユーザーには、「インスタンス 2」または「インスタンス 3」の管理者権限を付与できます。
- ログイン SSO ユーザーは、異なるサブスクリバ内の SSO ユーザーには、インスタンス管理者権限を付与または取り消すことができません。たとえば、サブスクリバ B 内の SSO ユーザーには、「インスタンス 2」または「インスタンス 3」の管理者権限を付与できません。

## Oracle Ultra Search で保護されるリソース

パブリックにクロールされるデータはすべて、パブリックにアクセスできます。

次のリソースは、Oracle Ultra Search で保護されています。

- アクセス制御リスト (ACL) が認識されるクロール・データは保護されます。つまり、ACL で指定されたユーザー専用です。
- パスワードはすべて保護されます。
- ユーザー定義のデータ・ソース・パラメータは保護されます。

## 認可およびアクセス規定

Oracle Ultra Search へのエントリ・ポイントには次の 3 つがあります。

- データベース：これにはすべてのデータが含まれます。データおよびメタデータはすべて、行レベルのセキュリティにより保護されます。パスワードはすべて暗号化されません。
- Oracle Ultra Search Administration ツール：これにはクロールされたデータは含まれません。SSO またはデータベース認証により認証する必要があります。
- Oracle Ultra Search 問合せツール：これにはクロールされたデータが含まれます。認証されていないユーザーは、公開データのみ参照できます。認証済のユーザーは、公開データおよび ACL で保護された情報を参照できます。ユーザーは、プライベート情報を参照するには自分自身を認証する必要があります。

## Oracle Ultra Search でセキュリティ・サービスが活用される仕組み

Oracle Ultra Search では、次の機能によりセキュリティ・サービスが活用されます。

- Oracle Ultra Search では Secure Sockets Layer (SSL) が使用されます。これは、インターネット上のメッセージ送信のセキュリティを管理する業界標準プロトコルです。これは、RMI 接続、HTTPS クロールおよびセキュアな JDBC の保護に使用されます。
- JAZN: OracleAS Containers for J2EE (OC4J) は、JAZN と呼ばれる Java Authentication and Authorization Service (JAAS) プロバイダを実装しています。これにより、アプリケーション開発者にはユーザー認証、認可および委譲サービスが提供され、これらをアプリケーション環境に統合できます。

**関連項目：** 2-6 ページ「[セキュアな Oracle Ultra Search インストールの構成](#)」

## Oracle Ultra Search で Oracle Identity Management インフラストラクチャが活用される仕組み

Oracle Ultra Search では、SSO Server および OID により、Oracle Identity Management インフラストラクチャが活用されます。

SSO Server により、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 のログオ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。Oracle Ultra Search Administration ツールおよび問合せツールでは、SSO が使用されます。

Oracle Internet Directory (OID) は、Oracle 固有の LDAP バージョン 3 準拠のディレクトリ・サービスであり、Oracle Database の上部にアプリケーションとして構築されています。



OID では Oracle の共通 ID がホスティングされます。Oracle Ultra Search インスタンスはすべて OID に登録されます。

**関連項目：** 1-11 ページ [「Oracle Internet Directory との統合」](#)

Oracle Ultra Search には固有の Oracle Identity Management があります。したがって、Oracle Identity Management インフラストラクチャが存在しない場合、Oracle Ultra Search では Oracle データベースで使用可能な固有のユーザー管理が使用されます。

## Oracle Ultra Search の拡張性とセキュリティ

Oracle Ultra Search は拡張可能（クローラ・エージェントなど）ですが、余分なセキュリティ上の考慮事項は発生しません。

## Oracle Ultra Search 用のセキュリティ・フレームワークの構成

ここでは、Oracle Ultra Search 内での特別なセキュリティ構成手順について説明します。

## Oracle Ultra Search 用のセキュリティ・フレームワーク・オプションの構成

`data-sources.xml` にクリア・テキストのパスワードを格納すると、セキュリティ上のリスクが発生します。これを防ぐには、パスワードの間接化を使用してパスワードを指定します。これにより、自動的に暗号化される `jazn-data.xml` にパスワードを入力し、`data-sources.xml` からこのパスワードを指し示すことができます。

**関連項目：**

- 2-22 ページ [「data-sources.xml ファイルの編集」](#)
- 『Oracle Application Server Containers for J2EE サービス・ガイド』

## Oracle Ultra Search 用の Oracle Identity Management オプションの構成

Oracle Ultra Search Administration ツールを SSO Server とともに構成するには、特定の手順を実行する必要があります。

**関連項目：** 2-18 ページ [「Single Sign-On Server とともに Administration ツールを構成する場合」](#)

## Oracle Ultra Search セキュリティの構成

Oracle Ultra Search には特定のセキュリティ・パスワードはありません。

**関連項目：** セキュリティを活用するための Oracle Ultra Search の構成上の問題の詳細は、5-9 ページの「[Oracle Ultra Search 用のセキュリティ・フレームワーク・オプションの構成](#)」を参照してください。

---

---

# Oracle Ultra Search クローラと 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 クローラの概要](#)
- [クローラ設定](#)
- [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・クローラ](#)

**関連項目：** [4-3 ページ「問合せパフォーマンスのチューニング」](#)

## Oracle Ultra Search クローラの概要

Oracle Ultra Search クローラは、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。クローラ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、クローラは 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。

---

---

## クローラ設定

クローラを使用する前に、クローラ・スレッド数、クローラ・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。Administration ツールの「クローラ」タブを使用して設定します。

**関連項目：** 7-12 ページ「[「クローラ」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
- OracleAS Portal のページ・グループ
- ユーザー定義データ・ソース（クローラ・エージェントが必要）

## クローラ・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、クローラ・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 クローラに情報を戻します。Oracle Ultra Search クローラでは、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「ソース」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：** 7-20 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
- クローラが作成した検索属性（クローラ・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、クローラは、クローラ・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、クローラは、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

クローラを最初に実行する場合は、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

6-6 ページの [図 6-1](#) および 6-7 ページの [図 6-2](#) 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、クローラでは他のデータ・ソース・タイプをクロールすることもできます。

[図 6-1](#) では、クローラおよびクローラ・スレッドをアクティブにする方法を示します。また、クローラ・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 6-2](#) では、クローラが 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、クローラを起動します。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 6-1](#))。
2. クローラはマルチ・クローラ・スレッドを初期化します。
3. クローラ・スレッドは、キュー内の次の URL を削除します。
4. クローラ・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. クローラ・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. クローラは、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 (6-7 ページの [図 6-2](#))。
7. クローラは、ドキュメント表に URL を登録します。
8. クローラ・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クローラで検出されると、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る場合でも削除されます）。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 6-1 URL のキューイング

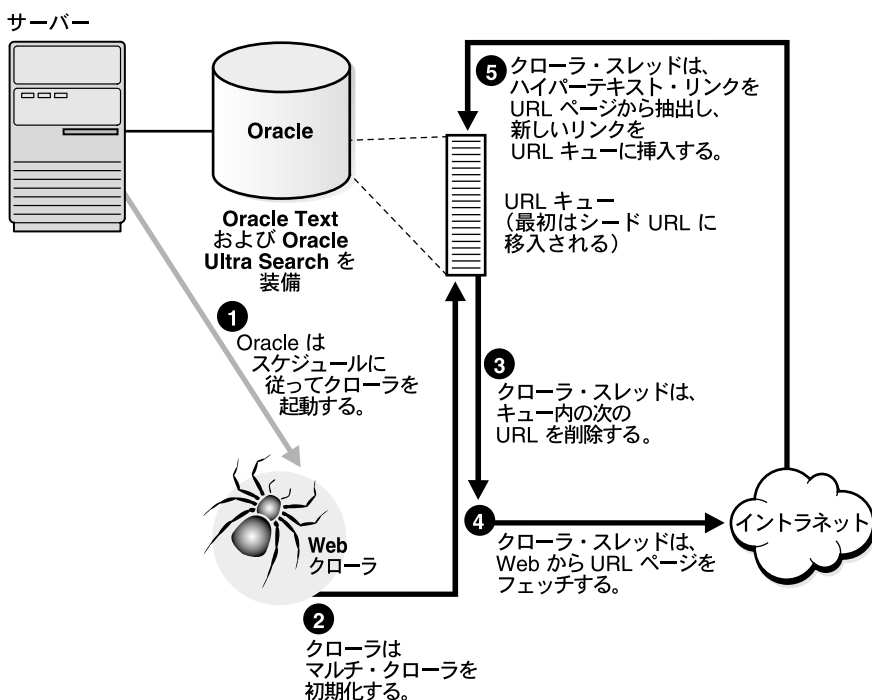
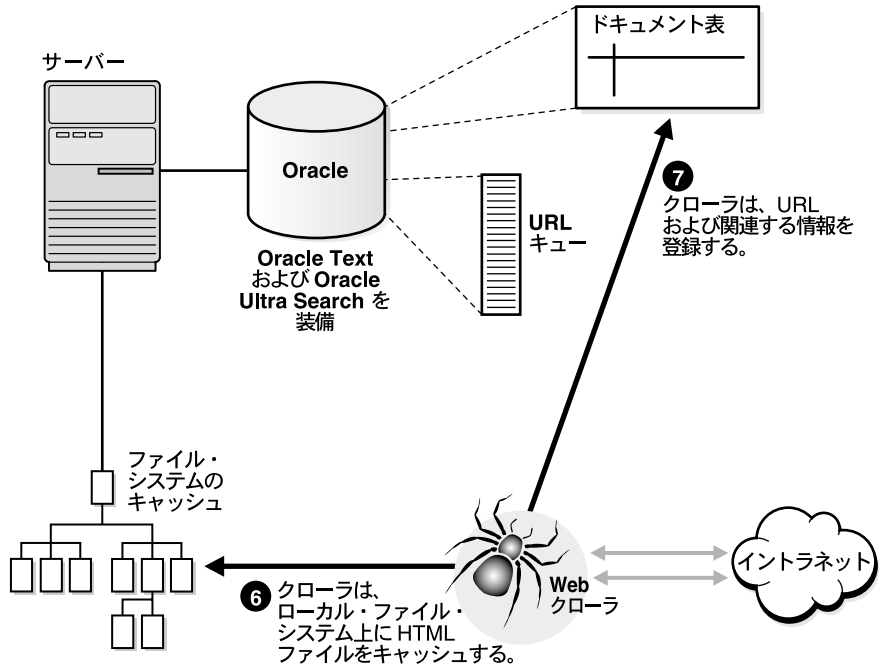




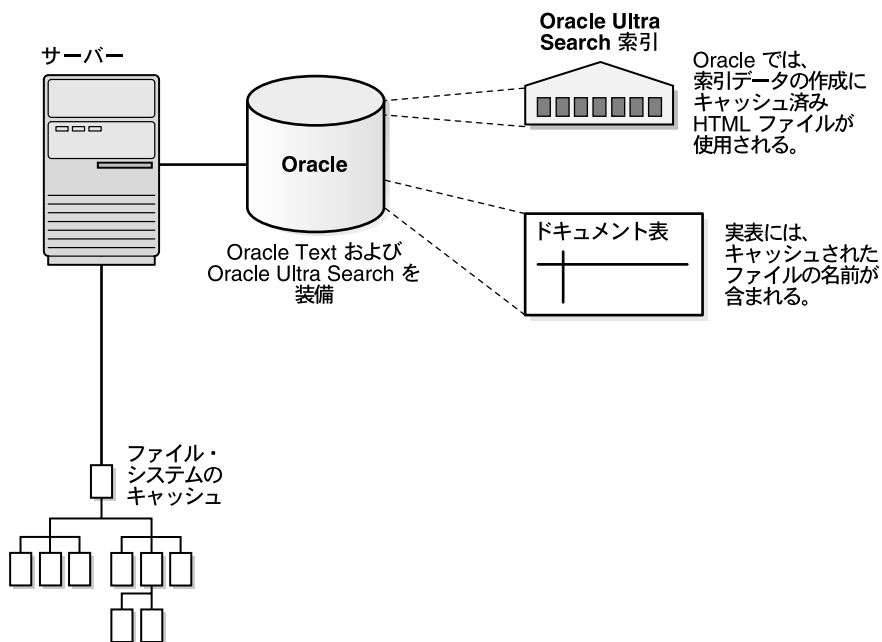
図 6-2 URL のキャッシュ



## 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 6-3 を参照してください。

図 6-3 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。クローラは、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、クローラ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、クローラを起動します。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. クローラはマルチ・クローラ・スレッドを初期化します。
3. 各クローラ・スレッドは、キュー内の次の URL を削除します。
4. 各クローラ・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. 各クローラ・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、クローラは手順 3 に戻ります。それ以外の場合は、次の手順に進みます。
6. 各クローラ・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。ドキュメント表にすでにあるリンクは、廃棄されます。
7. クローラは、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。☒ [6-2](#) を参照してください。
8. クローラは、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、クローラ・スレッドは、手順 3 から繰り返します。

## Oracle Ultra Search のリモート・クローラ

クローラ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つ以上のコンピュータ上で Oracle Ultra Search クローラを設定します。これらのコンピュータは、リモート・クローラと呼ばれます。ただし、各コンピュータは、データベース・コンピュータとキャッシュ、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・クローラを構成するには、データベース・ホスト以外のコンピュータに、Oracle Ultra Search Middle-Tier 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・クローラは Oracle Ultra Search システムに登録され、リモート・クローラ用のプロファイルが作成されます。Oracle Ultra Search Middle-Tier のインストール後、Oracle Ultra Search Administration ツールにログオンし、リモート・クローラ・プロファイルを編集する必要があります。これによって、クローラ・スケジュールへのリモート・クローラの割当てが可能になります。リモート・クローラ・プロファイルを編集するには、Administration ツールの「クローラ」タブを使用します。

---

---

**注意：** リモート・クローラを起動するとき、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・コンピュータと通信します。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・クローラを使用したクローラには、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を盗用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・クローラ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 ツールを使用して、Oracle Ultra Search インスタンスを管理できます。この章では、Oracle Ultra Search Administration ツールの画面の概略を示します。次の項目が含まれます。

- [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](#)
- [「クローラ」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](#)

## Oracle Ultra Search Administration ツール

Oracle Ultra Search Administration ツールは、J2EE に準拠した Web アプリケーションです。このツールを使用して、Oracle Ultra Search インスタンスを管理できます。Oracle Ultra Search Administration ツールを使用するには、データベース・ユーザー、Enterprise Manager スーパー・ユーザー、Portal ユーザーまたは SSO ユーザーとして任意のブラウザからログオンします。

---

---

**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 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スケーラビリティを強化するために、他のコンピュータで構築することができます。

---

---

Administration ツールを使用して、次のことを実行できます。

- 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 クローラ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

## クローラ・パラメータの設定

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

- クローラのログ・ファイル・ディレクトリなどのクローラ・パラメータを設定します。この作業を行うには、「[クローラ](#)」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。
- 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、Oracle ソース（たとえば OracleAS Portal や連携したソース）またはユーザー定義のデータ・ソースがあります。クローラ・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。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。

- 同期スケジュールを定義します。クローラは、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。クローラ・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検索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・ソースで構成されず、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、様々な言語で表示可能な状況依存のオンライン・ヘルプが提供されます。言語設定は「[ユーザー](#)」タブで変更できます。

## 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。

- シングル・サインオン (SSO) ユーザー：これらのユーザーは、Oracle Internet Directory (OID) により管理され、SSO Server により認証されます。Oracle Ultra Search Administration ツールにより、SSO ユーザーがアクセスできるすべての Oracle Ultra Search インスタンスが識別されます。これは、Oracle Identity Management インフラストラクチャがインストールされている場合のみ使用可能です。
- データベース・ユーザー (non-SSO)：これらのユーザーは、Oracle Ultra Search が動作するデータベースに存在します。
- Oracle Enterprise Manager のユーザー

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。

- 非 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
- Oracle Enterprise Manager のユーザー
- デフォルトの SSO ユーザー：PORTAL (Oracle Database ではサポートされていません)
- デフォルトの SSO ユーザー：ORCLADMIN (Oracle Identity Management インフラストラクチャがインストールされている場合のみ使用できます)

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」](#)
- WKSYS パスワードの変更方法の詳細は、3-2 ページの「[Oracle Ultra Search スキーマのパスワードの変更](#)」を参照してください。
- Oracle Ultra Search インスタンスの作成方法の詳細は、7-7 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、7-41 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、7-5 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：** シングル・サインオン (SSO) は、Oracle Identity Management インフラストラクチャがインストールされている場合のみ使用できます。

---

---

### Oracle Ultra Search へのログオン

シングル・サインオン (SSO) ・ユーザーが、SSO で保護された Oracle Ultra Search Administration ツールにログインすると、まず SSO ログイン画面が表示されます。

SSO ユーザー名とパスワードを入力します。SSO Server によってユーザーが認証されると、管理権限のある Oracle Ultra Search インスタンスのリストがユーザーに表示されます。

ユーザーにより、URL が異なります。次に例を示します。

- SSO ユーザー : `http://<host>:<http port>/ultrasearch/admin_sso/index.jsp`
- Portal ユーザー : `http://<host>:<http port>/pls/portal`
- Oracle Enterprise Manager のユーザー : `http://<host>:<em port>/`

## SSO ユーザーへの権限の付与

場合によっては、スーパー・ユーザー権限、または Oracle Ultra Search インスタンスを管理する権限を SSO ユーザーに付与する必要があります。このプロセスは、後述するように、OracleAS Portal がホスト・モードまたは非ホスト・モードで実行されているかどうかによって、若干異なります。

---

---

**注意：** SSO ユーザーは、SSO ニックネーム / サブスクリバ・ニックネームの組合せを持つ Oracle Ultra Search によって一意に識別されます。

---

---

- 非ホスト・モードで、SSO ユーザーに権限を付与する場合、サブスクリバ・ニックネームは必要ありません。これは、非ホスト・モードの OracleAS Portal にはサブスクリバが 1 つしかないためです。
- ホスト・モードでは、SSO ユーザーに権限を付与する場合、サブスクリバ・ニックネームが必要です。これは、OracleAS Portal に 1 つ以上のサブスクリバが存在する可能性があるためです。同じ SSO ニックネーム (PORTAL など) を持つ 2 人以上のユーザーは、サブスクリバ・ニックネームで区別される、個別の SSO ユーザーである可能性があります。ホスト・モードで実行する場合は、次の点にも注意してください。
  - デフォルトのサブスクリバ・ユーザーに権限を付与する場合は、実際のニックネームが異なる場合 (「ORACLE」など) でも、サブスクリバ・ニックネームには常に「DEFAULT COMPANY」を指定してください。実際のニックネームは、Oracle Ultra Search では認識されません。
  - デフォルトのサブスクリバ・ユーザーとして SSO にログインする場合は、サブスクリバ・ニックネームを空白にしておきます。実際のサブスクリバ・ニックネームのかわりに「DEFAULT COMPANY」と入力することもできます。たとえば、「ORACLE」と入力すると、Oracle Ultra Search で認識されます。

---

---

**注意：** インストール後はいつでも、OracleAS Portal スクリプトを実行して、実行モードを非ホスト・モードからホスト・モードに変更できます。この変更を行うと、OracleAS Portal スクリプトは Oracle Ultra Search スクリプトを呼び出して、Oracle Ultra Search に非ホスト・モードからホスト・モードへの変更を通知します。

---

---

**関連項目：** 「Hosting Developer's Guide」 (<http://otn.oracle.com/>)

## 「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参照できます。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、編集および削除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、スーパー・ユーザー権限を付与されているユーザーのみです。

## インスタンスの作成

インスタンスを作成するには、「作成」をクリックします。標準インスタンスまたは読取り専用のスナップショット・インスタンスを作成できます。新しいインスタンスを作成できるのは、スーパー・ユーザー権限を持つユーザーのみです。

---

---

**注意：**異なる Oracle Ultra Search インスタンス内で同じデータ・ソースを定義すると、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、WKUSER ロールを持つデータベース・ユーザー / スキーマに基づいています。

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。
- 7-8 ページ「[スナップショット・インスタンスの作成](#)」

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**」をクリックして、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。レクサーは、言語にあわせてテキストをトークンに分割します。このトークンは、通常は単語です。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する方法は、『Oracle Text リファレンス』を参照してください。
- 3-7 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。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スケーラビリティ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・インスタンスにはデータベース・ユーザーが必要です。このユーザーには WKUSER ロールを付与する必要があります。

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ابل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**」をクリックして、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
4. セキュアな検索を実行するために、スナップショットを有効にします。

スナップショットのマスター・インスタンスがセキュアな検索に対応しており、スナップショットを作成している宛先のデータベースが、セキュアな検索に対応したインスタンスをサポートしている場合は、スナップショットを作成している宛先のデータベースで、PL/SQL プロシージャも実行する必要があります。

このプロシージャを実行すると、宛先データベースのアクセス制御リスト (ACL) の ID が変換され、ID が有効になります。WKSYS ユーザーとしてデータベースにログオンします。次のようにして、プロシージャを起動します。

```
exec WK_ADM.USE_INSTANCE('instance_name');  
exec WK_ADM.TRANSLATE_ACL_IDS();
```

*instance\_name* は、スナップショット・インスタンスの名前です。

この文が、エラーなしで正常に完了することを確認します。

### 関連項目：

- WKSYS パスワードの変更方法および Oracle Ultra Search のデータベース・ユーザーの構成手順は、[第 3 章「インストール後の情報」](#)を参照してください。
- 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。

## 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」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」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、ユーザーにスーパー・ユーザー権限が付与されている必要があります。

---

---

## 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」を選択します。

(インスタンスを更新可能にする) インスタンス・モードやインスタンス・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。

## 「クローラ」タブ

Oracle Ultra Search クローラは、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。

## 設定の構成

### クローラ・スレッド

実行時に起動されるクローラ・スレッドの数を指定します。

### プロセッサ数

Oracle Ultra Search クローラ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な数が決まります。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 クローラ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 クローラ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。

クローラは、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 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 で使用されるレクサーは、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、クローラは 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、クローラ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。

---

---

クローラまたはデータ・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。

- ポーランド語
- 中国語
- ハンガリー語
- ノルウェー語
- ルーマニア語
- フィンランド語
- 日本語
- スペイン語
- スロバキア語
- 英語
- トルコ語
- デンマーク語
- スウェーデン語
- ロシア語
- ドイツ語
- 韓国語
- オランダ語
- イタリア語
- ギリシャ語
- ポルトガル語
- チェコ語

- ヘブライ語
- フランス語
- アラビア語

### クローラの深さ

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、クローラ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：** クローラの深さの重要性については、4-2 ページの「[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。

### クローラ・タイムアウトしきい値

クローラのタイムアウトを秒単位で指定します。クローラ・タイムアウトしきい値は、クローラが 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、クローラはこの設定を使用します。

### キャッシュ・ディレクトリ

キャッシュ・ディレクトリの絶対パスを指定してください。クロール時に、ドキュメントはキャッシュ・ディレクトリに格納されます。事前に設定されたサイズに達するたびに、クローラが停止し、索引付けが始まります。

機密情報をクロールしている場合は、キャッシュ・ディレクトリに適切なファイル・システムの読取り権限を設定してください。

索引付けが完了した後に、キャッシュを消去するかどうかを選択できます。

### クローラのロギング

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
- クローラ・ログ・ファイル・ディレクトリ
- クローラ・ログ・ファイル言語

ログ・ファイル・ディレクトリには、クローラ・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クローラのアクティビティ、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クローラを構成することが有効な場合があります。クローラ・ログ・ファイル言語とは、クローラがログ・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・クローラで使われる標準 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』

JDBC 接続文字列を別の形式（LDAP 形式など）に更新できます。ただし、JDBC 接続文字列が別のデータベースを指すように変更することはできません。JDBC 接続文字列は、中間層が指すデータベースに設定する必要があります。つまり、中間層と JDBC は同じデータベースを指す必要があります。

Real Application Clusters 環境では、TNS キーワード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・クローラ・プロファイル

このページを使用して、リモート・クローラ・プロファイルを表示および編集します。リモート・クローラ・プロファイルには、Oracle Ultra Search データベース以外のリモート・コンピュータで Oracle Ultra Search クローラ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・クローラ・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・クローラがデータベース・コンピュータ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・クローラ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。

## クローラ統計

このページを使用して、次のクローラ統計を参照します。

### クローラ・アクティビティのサマリー

クローラ・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。

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

### 詳細なクローラ統計

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

### クローラの進行状況

過去の週についてクローラ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、認証とプロキシを設定します。

### 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：** クローラでは、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。

### 認証

すべてのデータ・ソースに関するグローバル認証情報を入力します。

---

---

**注意：** データ・ソース固有の認証は、このグローバル認証よりも優先されます。

---

---

#### HTTP 認証

HTTP 認証が必要なホストおよび領域のユーザー名およびパスワードを指定します。Oracle Ultra Search では、Basic および Digest 認証の両方をサポートしています。

#### HTML フォーム

Web クロール時に Oracle Ultra Search クローラにより自動入力する HTML フォームを登録します。HTML フォームをサポートするには、HTTP Cookie 機能を有効にする必要があります。

HTML フォームは、手動で、またはフォーム登録ウィザードを使用して登録できます。HTML フォームに JavaScript が含まれている場合、ウィザードが失敗し、手動で登録する必要があります。

## 「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 A 「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。

---

---

ポリシー値を更新するには、該当する属性の「**LOV の管理**」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。

- LOV を手動で更新します。
- クローラ・エージェントは、クロール・プロセス中に LOV を自動的に更新できます。
- 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できます。

---

---

**注意：** ポリシーのアップデートがクローラ・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

すべてのデータ・ソースのマッピング情報について説明します。ユーザー定義のデータ・ソースでは、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、ファイルまたは表データ・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るために、コストがかかります。必要なとき以外は、既存マッピングの編集をしないでください。

---

---

**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 の作成者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 の件名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 クローラは、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](#)
- [Oracle ソース](#)
- [ユーザー定義ソース](#) (クローラ・エージェントが必要)

### 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、7-32 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、7-37 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 サイトのコンテンツを表示します。Web ソースは、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 ソースの名前および開始アドレスを指定します。これは、クローラがクロールを開始する URL です。開始アドレスには、HTTP または HTTPS を指定できます。
2. URL 境界ルールを設定して、クロール領域を絞り込みます。固有の名前で始まるか固有の名前で終わる、または固有の名前と同じホストあるいはドメインを包含または除外できます。

たとえば、包含ドメインが `oracle.com` で終わる場合、Oracle Ultra Search クローラは全世界のオラクル社に属するホストのみをクロールするように制限します `oracle.com` で終わるものはすべてクロールされますが、`http://www.oracle.com.tw` はクロールさ



れません。新しいシード "http://www.yahoo.com" を使用して包含ドメインを yahoo.com に変更すると、すべての oracle.com URL がクローラによって削除されます。

除外ドメインが uk.oracle.com の場合、クローラはイギリスにある Oracle ホストをクロールしません。特定のポートを使用して、Web サイトを包含または除外することもできます (デフォルトでは、すべてのポートがクロールされます)。特定のホストに対し、ポート包含ルールまたはポート除外ルールを指定できますが、両方は使用できません。除外ルールは常に包含ルールをオーバーライドします。

3. Oracle Ultra Search クローラ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、クローラが常に処理するデフォルトのドキュメント・タイプです。
4. 認証設定を指定します。このステップはオプションです。「HTTP 認証」で、認証が必要なホスト領域のユーザー名およびパスワードを指定します。「HTML フォーム」で、Web クロール中に Oracle Ultra Search クローラによって自動的に入力される HTML フォームを登録できます。HTML フォームをサポートするには、HTTP Cookie 機能を有効にする必要があります。データ・ソースを保護する認証フォームを登録するには、「HTML フォームの登録」をクリックします。注意: クロールするフォーム URL について、URL が robots.txt ファイルで除外されていないことを確認してください。除外されている場合、このデータ・ソースのロボットの除外を無効にする必要があります。(デフォルトでは、Oracle Ultra Search はロボットの除外を有効にします。)
5. データ・ソースに対して「ACL を使用しない」または「Ultra Search ACL」を選択します。ユーザーが検索を実行すると、ACL (アクセス制御リスト) により、ユーザーがアクセスできるドキュメントが制御されます。デフォルトは「ACL を使用しない」で、すべてのドキュメントが検索可能および表示可能とみなされます。データ・ソースの ACL に、2 つ以上のグループおよびユーザーを追加できます。インスタンスでセキュリティが有効になっている場合のみ、選択可能です。
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。
7. 各 Web ソースに関するデフォルトのクローラ設定を上書きします。このステップはオプションです。オーバーライドできるパラメータは、クロールの深さ、クローラ・スレッド数、言語、クローラ・タイムアウトしきい値、キャラクタ・セット、最大 Cookie サイズ、最大 Cookie 数および各ホストの最大 Cookie 数です。ロボット除外、言語検出、URL リライター、動的ページの索引付け、HTTP Cookie、および Cookie ログ・ファイルの内容を表示するかどうかを有効または無効にすることもできます。「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。

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 クローラは、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、クローラ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

認めます。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せん。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 UriRewriter 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。クローラは、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を実行します。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。クローラ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。

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。

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。

「動的ページの索引付け」を「はい」に設定すると、動的 URL がクロールおよび索引付けされます。このオプションを使用してすでにクロールされたデータ・ソースについて、「動的ページの索引付け」を「いいえ」に設定し、データ・ソースを再クロールすると、索引から動的 URL がすべて削除されます。

動的ページによっては、同じページに対する複数の検索ヒットとして表示されるため、すべてを索引付けする必要がない場合もあります。他の動的ページはそれぞれ異なるため、索引付けする必要があります。この 2 種類の動的ページを区別してください。一般的に、メニューの拡張時にメニューの内容に影響を与えることなく変更のみを行う動的ページは、索引付けする必要はありません。次の 3 つの URL を考慮してください。

[http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming\\_convention.html](http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html)

[http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming\\_convention.html?nsdnv=14z1](http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html?nsdnv=14z1)

[http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming\\_convention.html?nsdnv=14](http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html?nsdnv=14)

URL 内の疑問符 (?) は、残りの文字列が入力パラメータであることを示します。重複するヒットは基本的に、異なる側にメニュー拡張がある同一ページです。同じ問合せにヒットは 1 つのみというのが理想的です。

[http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming\\_convention.html](http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html)

動的ページ索引の制御は、データ・ソース全体に適用されます。したがって、Web サイトに2種類の動的ページがある場合は、その動的ページの索引付けを制御するために、2つのデータ・ソースとして個別に定義する必要があります。

**関連項目：**

- 8-29 ページ「[Oracle Ultra Search の URL リライタ API](#)」
- 6-3 ページ「[クロウラ・エージェントの使用](#)」
- デフォルト言語の詳細は、7-12 ページの「[「クロウラ」タブ](#)」を参照してください。

## 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：** 7-25 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、「[表ソース作成](#)」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、7-12 ページの「[「クロウラ」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。クロールを有効にすると、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. 列を検索属性にマップします。各列は、1つの検索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=$(key1)` ここで key1 は該当する表の主キー列です。たとえば、URL の  
`http://bug:7777/pls/bug?rptno=1234567` を使用して、バグ番号 1234567 (バグ番号は表の主キー) を問い合わせるとします。表ソースの表示 URL テンプレートを `http://bug:7777/pls/bug?rptno=$(key1)` に設定できます。  
**「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。
7. データ・ソースの ACL (アクセス制御リスト) ポリシーを指定します。ユーザーが検索を実行すると、ACL により、ユーザーがアクセスできるドキュメントが制御されます。デフォルトは「ACL を使用しない」で、すべてのドキュメントがパブリックおよび表示可能とみなされます。Ultra Search ACL を使用するよう指定することもできます。データ・ソースの ACL に、2 つ以上のグループおよびユーザーを追加できます。インスタンスでセキュリティが有効になっている場合のみ、選択可能です。

**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

表ソースの名前を変更するには、「表ソース」メイン・ページで「編集」をクリックします。表の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、クローラがデータベース・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_name/view_name;
```

- 実表やビューに BFILE、RAW 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 クローラが電子メール・ソースを検索する場合、クローラ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 クローラ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 クローラは、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 クローラは、IMAP サーバーからそのメッセージを削除します。次に、クローラは電子メール・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。オプションで、電子メールの別名および ACL ポリシーを指定できます。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、クローラによって収集されます。別名は、2 つ以上の電子メール・アドレスが論理的に同じ場合に便利です。たとえば、配布リストを表す電子メール・ソース `list@company.com` は、`list@my.company.com` という代替アドレスを持つことができます。別名リストに `list@my.company.com` が追加されると、そのアドレスに送信された電子メールは、`list@company.com` に送信されたかのように処理されます。

データ・ソースの ACL（アクセス制御リスト）ポリシーを指定します。ユーザーが検索を実行すると、ACL により、ユーザーがアクセスできるドキュメントが制御されます。デフォルトは「ACL を使用しない」で、すべてのドキュメントが検索可能および表示可能とみなされます。データ・ソースの ACL に、2 つ以上のグループおよびユーザーを追加できます。

## ファイル・ソース

ファイル・ソースは、ローカル・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・ソースの名前およびデフォルトの言語を指定します。
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 クローラ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、クローラ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。パス・ルールはホストに固有ですが、ホストごとに 2 つ以上のパス・ルールを指定できます。たとえば、同じホスト上に `path files://host/doc` および除外パス `files://host/doc/unwanted` を含めることができます。

4. Oracle Ultra Search クローラがこのファイル・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、クローラが常に処理するデフォルトのドキュメント・タイプです。
5. 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/operation/doc/file.doc` で、表示 URL が `https://webhost/client/doc/file.doc` の場合は、ファイル URL の接頭辞を `file:///home/operation`、表示 URL の接頭辞を `https://webhost/client` に指定できます。
6. データ・ソースの ACL (アクセス制御リスト) ポリシーを指定します。ユーザーが検索を実行すると、ACL により、ユーザーがアクセスできるドキュメントが制御されます。デフォルトは「ACL を使用しない」で、すべてのドキュメントが検索可能および表示可能とみなされます。Ultra Search ACL を使用するように指定することもできます。データ・ソースの ACL に、2 つ以上のグループおよびユーザーを追加できます。インスタンスでセキュリティが有効になっている場合のみ、選択可能です。

## Oracle ソース

Oracle ソースは、作成、編集または削除できます。連携したデータ・ソースまたは OracleAS Portal (クロール可能) データ・ソースを選択できます。連携したソースは、独自の索引を保持するリポジトリです。Oracle Ultra Search は問合せを発行し、リポジトリは問合せ結果を返します。Oracle Ultra Search は、OracleAS Portal のクロールと索引付けも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。

### Oracle Portal ソース

Portal ソースを作成するには、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. 名前およびポータル URL ベースを指定します。ポータル名は、「Oracle Portal リスト」ページでこのポータル・エントリを識別するために使用されます。URL ベースは、ポータル・ホームページの先頭部分です。これには、ホスト名、ポート番号および DAD が含まれます。ポータル URL ベースは、一度作成すると更新できません。「**ポータルの登録**」をクリックします。Oracle Ultra Search は、OracleAS 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。

2. 索引付けに使用する 1 つ以上のページ・グループを選択します。ページ・グループごとにポータル・データ・ソースが作成されます。既存のポータル・データ・ソースを削除するには、「**削除**」をクリックします。

Oracle Ultra Search クローラがポータル・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、クローラが常に処理するデフォルトのドキュメント・タイプです。ドキュメントのタイプを編集するには、ドキュメントが作成された後に、ポータル・ソースの「**編集**」をクリックします。

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

## 連携したソース

連携したソースを作成するには、新しいデータ・ソースの名前および JNDI を指定します。デフォルトでは、使用可能なリソース・アダプタはありません。

連携したソースを作成するには、Oracle Ultra Search リソース・アダプタ、またはサーチレットを配置する必要があります。サーチレットは、中間層 (OC4J 内) に配置される Java モジュールであり、ユーザーにかわり企業情報システム内のデータを検索します。サーチレットは、中間層 (OC4J 内) に配置される Java モジュールであり、ユーザーにかわり企業情報システム内のデータを検索します。ユーザーの問合せがサーチレットに委譲された場合、サーチレットはユーザーにかわり問合せを実行します。サーチレットはすべて、JCA 1.0 に準拠するリソース・アダプタです。

**関連項目：** リソース・アダプタおよび Java Connector Architecture の詳細は、Javasoft による JCA 1.0 仕様を参照してください。

**Oracle Ultra Search サーチレットの配置およびバインド** Oracle Ultra Search サーチレットでは、1 つの Oracle Ultra Search インスタンスに対して問合せが可能です。Oracle Ultra Search サーチレットは `ultrasearch.rar` としてパッケージされ、`$ORACLE_HOME/ultrasearch/adapter/` ディレクトリに格納されて出荷されます。

Oracle Ultra Search サーチレットをスタンドアロンの OC4J に配置するには、`admin.jar` を次のように使用します。

```
java -jar admin.jar ormi://<hostname> <admin> <welcome> -deployconnector -file ultrasearch.rar -name UltraSearchSearchlet
```

この時点で、`ultrasearch.rar` は OC4J に配置されています。ただし、Oracle Ultra Search インスタンスに接続するようインスタンス化されていません。次の手順を繰り返すことで、Oracle Ultra Search サーチレットを複数回インスタンス化し、複数の Oracle Ultra Search インスタンスに接続できます。

サーチレットをインスタンス化するには、構成パラメータ値を指定し、サーチレット・インスタンスのバインド先となる JNDI ロケーションを指定する必要があります。そのためには、手動で `oc4j-ra.xml` を編集する必要があります。通常、このファイルは `$J2EE_HOME/application-deployments/default/UltraSearchSearchlet/` ディレクトリの下にあります。Oracle Ultra Search サーチレットには、`connectionURL`、`userName`、



password および instanceName の 4 つの構成プロパティが必要です。たとえば、「eis/UltraSearch」の下にサーチレットをバインドして、マシン 'dbhost' のデフォルト・インスタンス 'wk\_test' に接続するには、次のエントリを使用できます。

```
<connector-factory location="eis/UltraSearch" connector-name="Ultra Search Adapter">
  <config-property name="connectionURL" value="jdbc:oracle:thin:@dbhost:1521:sid"/>
  <config-property name=:userName" value="wk_test"/>
  <config-property name="password" value="wk_test"/>
  <config-property name="instanceName" value="wk_test"/>
</connector-factory>
```

oc4j-ra.xml を編集した後、OC4J インスタンスを再起動します。再起動時にエラーが表示されない場合は、サーチレットが正常にインスタンス化され、JNDI にバインドされています。

**Federator サーチレットの配置およびバインド** Federator サーチレットは、他のサーチレットと対話することで、複数のリポジトリに対するシングル・ポイント検索が可能になります。たとえば、Federator サーチレットは複数の Oracle Ultra Search サーチレットを起動して、複数の Oracle Ultra Search インスタンスに対して同時に問合せできます。同様に、Federator サーチレットは、Oracle Files、電子メールなどについてサーチレットを起動できます。

Federator サーチレットは、**連携したソース・タブ**で、Oracle Ultra Search Administration ツールにより構成および管理されます。

Federator サーチレットは federator.rar としてパッケージされ、\$ORACLE\_HOME/ultrasearch/adapter/ ディレクトリに格納されて出荷されます。

federator.rar の配置手順は、Oracle Ultra Search サーチレットの配置と同様です。Federator サーチレットをスタンドアロンの OC4J に配置するには、admin.jar を次のように使用します。

```
java -jar admin.jar ormi://<hostname> <admin> <welcome> -deployment -file
federator.rar -name FederatorSearchlet
```

サーチレットをインスタンス化するため、Federator サーチレットでは、oc4j-ra.xml ファイルに、connectionURL、userName、password および instanceName の 4 つの構成プロパティが必要です。通常、このファイルは \$J2EE\_HOME/application-deployments/default/FederatorSearchlet/ ディレクトリの下にあります。次に例を示します。

```
<connector-factory location="eis/Federator" connector-name="Federator Adapter">
  <config-property name="connectionURL" value="jdbc:oracle:thin:@dbhost:1521:sid"/>
  <config-property name="userName" value="wk_test"/>
  <config-property name="password" value="wk_test"/>
  <config-property name=InstanceName" value="wk_test"/>
</connector-factory>
```

oc4j-ra.xml を編集した後、OC4J インスタンスを再起動します。再起動時にエラーが表示されない場合は、サーチレットが正常にインスタンス化され、JNDI にバインドされています。

## 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クローラ・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、クローラ・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 クローラに情報を戻します。Oracle Ultra Search クローラでは、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：** 8-19 ページ [「Oracle Ultra Search クローラ・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。

### ユーザー定義データ・ソース・タイプの作成

データ・ソース・タイプを作成、編集または削除するには、「[ソース・タイプの管理](#)」をクリックします。新しいタイプを作成するには、「[新規タイプの作成](#)」をクリックします。

1. データ・ソース・タイプ名、説明、およびクローラ・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。クローラ・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 クローラに情報を戻します。Oracle Ultra Search クローラでは、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/ にあることが必要です。
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、クローラ・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。

## ユーザー定義ソースの作成

ユーザー定義のデータ・ソースを作成するには、タイプを選択して、「実行」をクリックします。

1. データ・ソースの名前、デフォルト言語およびパラメータ値を指定します。デフォルト言語の詳細は、「[クローラ](#)」タブを参照してください。
2. 認証設定を指定します。このステップはオプションです。「**HTTP 認証**」で、認証が必要なホストおよび領域のユーザー名およびパスワードを指定します。「**HTML フォーム**」で、Web クロール中に Oracle Ultra Search クローラによって自動的に入力される HTML フォームを登録できます。HTML フォームをサポートするには、HTTP Cookie 機能を有効にする必要があります。データ・ソースを保護する認証フォームを登録するには、「**HTML フォームの登録**」をクリックします。
3. データ・ソースの ACL (アクセス制御リスト) ポリシーを、ACL を使用しない、リボジトリが生成した ACL、または Ultra Search ACL の中から指定します。ユーザーが検索を実行すると、ACL により、ユーザーがアクセスできるドキュメントが制御されます。デフォルトは「ACL を使用しない」で、すべてのドキュメントが検索可能および表示可能とみなされます。Ultra Search ACL の場合、データ・ソースの ACL に、2 つ以上のグループおよびユーザーを追加できます。
4. 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。クローラは、クローラ・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。
5. クロール・パラメータを編集します。
6. クローラがこのデータ・ソースについて処理するドキュメント・タイプを指定します。デフォルトでは、HTML およびプレーン・テキストが常に処理されます。

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できます。

## 「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：** 6-3 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・クローラ・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。

### 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クローラ、または個別のホストで実行されているリモート・クローラ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。クローラでは、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、クローラはそのドキュメントを処理しません。特定の状況では、クローラ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。
  - クローラ再クロール・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**:クローラはデータ・ソースを再クロールします。たとえば、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、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、クローラがこのデータ・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。クローラは、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。

スケジュールを再起動するとき、次のルールにより、再クロールされる URL が決定します。

- \* 前のクロールが完了していない場合（クロールを停止したり、データベースの表領域が満杯の場合など）、クローラは URL キュー内に残っている URL のみをクロールします。すでにクロールされた URL は、再クロールには関係しません。
- \* URL キューが空で最後のクロール以降に新しいシードが追加された場合、クローラは新しいシードのみをクロールします。
- \* URL キューが空で新しいシードがない場合、クローラはクロールされたすべての URL を再クロールします。

したがって、クローラを停止して「動的ページの索引付け」を「いいえ」に設定した場合、まだクロールされていないキュー内の URL のみに影響します。すでにクロールされた動的ページは、キューが空の場合、3 回目の再クロール時に索引から削除されます。

---

**注意：** 新しくクロールされた URL のみではなく、クロールされたすべての URL がクローラ設定の対象です。

---

- 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。
  - \* 索引付けのためにすべての 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。

クローラの動作は、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

スケジュールの同期の頻度は、別のスケジュールの同期の頻度と同じです。これにより、データ・ソースの同期を管理するときに、最大限の柔軟性を得ることができます。

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。
- スケジュールを手動で開始します。

---

**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 が更新され、キューに挿入されます。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、エンキュ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。

---

URL キューが空ではない場合、または最後のクロール以降に新しいシードが追加された場合は、スケジュールを起動しても、エンキューは実行されません。たとえば、ユーザーが早めにクロウラを停止した場合、または Oracle 表領域が不十分なためにクロウラが終了した場合には、URL キューは空になりません。したがって、次の起動時には、クロウラはエンキューを行わずに、URL キューが空になるまで既存の URL キューを処理します。つまり、起動時にキューが空の場合にのみ、エンキューが実行されます。

## ステータスとクロウラ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」をクリックします。

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。

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」、「変換に失敗したドキュメント」および「索引付けされたドキュメント」の合計です。
- フェッチに失敗したドキュメント：Oracle HTTP サーバー・タイムアウトまたは別の HTTP サーバー・エラーの可能性がります。
- 拒否されたドキュメント：ドキュメントが URL 境界ルールの範囲外です。

- 検出したドキュメント: 「フェッチするドキュメント」、「フェッチされたドキュメント」、「フェッチに失敗したドキュメント」および「拒否されたドキュメント」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 OracleAS Portal のページ、または索引を指定しないロボット・メタ・タグの可能性があります。
- 変換に失敗したドキュメント: 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 クローラは、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、これにより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:** クローラの一時的ディレクトリ・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場合があります。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。

Oracle Ultra Search インスタンスがセキュアな検索に対応している場合は、索引最適化処理により、未使用のアクセス制御リスト (ACL) のガベージ・コレクションもトリガーされます。



## 「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。

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 クローラ検索領域に含められます。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。

**関連項目：** 1-10 ページ [「ドキュメント検索オプション」](#)

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 A 「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 時間以内に行われた最も使用された 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 時間以内に行われた問合せの総数
- 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使用された 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 :** 4-3 ページ [「問合せパフォーマンスのチューニング」](#)

## 構成

検索アプリケーションおよび Federation エンジンでは、ヒットの最大数および検索オプションの有効化など、いくつかのパラメータを構成できます。

## 「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。

シングル・サインオン (SSO) ユーザーは、Delegated Administrative Service (DAS) の値リストを使用して、他の SSO ユーザーをスーパー・ユーザーとして追加できます。これらのユーザーは、アクセスを許可される前に、SSO サーバーで認証されます。データベース・ユーザーは、他のデータベース・ユーザーをスーパー・ユーザーとして追加できます。

他のユーザーにスーパー・ユーザー権限を付与するには、ユーザーのユーザー名を入力します。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。

シングル・サインオン (SSO) ユーザーは、Delegated Administrative Service (DAS) の値リストを使用して、他の SSO ユーザーに権限を追加できます。これらのユーザーは、アクセスを許可される前に、SSO サーバーで認証されます。データベース・ユーザーは、他のデータベース・ユーザーに権限を追加できます。

---

---

**注意：** データベース・ユーザーは、SSO ユーザーに権限を付与することはできません。また、SSO ユーザーは、データベース・ユーザーに権限を付与することはできません。DAS の値リストには、SSO ユーザーのみが表示されます。

---

---

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。これを行うには、ユーザー名を入力して、そのユーザーが他のユーザーに管理権限を付与できるかどうかを指定します。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。

---

---

**注意：** 一般管理権限には、インスタンスを作成または削除する権限は含まれません。これらの権限はスーパー・ユーザーの権限です。

---

---

**関連項目：** 3-6 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。

## 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](#)
- [問合せ構文拡張のカスタマイズ](#)
- [Oracle Ultra Search の Query タグ・ライブラリ](#)
- [Oracle Ultra Search クローラ・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](#)

**関連項目：** 『Oracle Ultra Search API Reference』

## 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。
- クローラ・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。クローラはこれを使用して、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、J2EE に準拠した Web アプリケーションです。

## 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広範囲の 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、スケーラビリティを得るために 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。

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

データだけのメソッドは、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。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。

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、定義済のすべてのデータ・グループや属性などを取得します。
- 問合せ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です。
- 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：** [4-3 ページ「問合せパフォーマンスのチューニング」](#)

## 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。Oracle Ultra Search 9.0.1 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。

**関連項目：** [付録 C 「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拡張によって、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、ドキュメントの 'title' (タイトル) という文字列属性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文字列属性に対して使用される場合も同じです。

デフォルトの問合せ構文拡張は、`oracle.ultrasearch.query.Contains` クラスで実装されています。問合せ拡張をカスタマイズするには、`oracle.ultrasearch.query.CtxContains` クラスを使用します。

ここでは、デフォルトの問合せ拡張ルール、および使用している環境向けに問合せ構文拡張を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）。

デフォルトの問合せ構文拡張は、`oracle.ultrasearch.query.Contains` クラスで実装されています。サンプル検索アプリケーションは、コンテンツの検索および文字列属性検索で、この構文拡張を利用します。

## 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 [+] は、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 [-] は、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 [\*] は、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。 <b>注意:</b> [Oracle] の検索は、[Oracle*] とは異なる。
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。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。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・ユーザーの間合せ文字列が [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・ユーザーの間合せは Oracle Text の間合せに拡張されます。拡張された間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、デフォルトの間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>)

ここでは、前述のルールで使用されている項目およびその意味について説明します。

<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 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あり、演算子または二重引用符がない場合のみ使用されます。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は、<plus expression> および <accum expression> が排除されます。

## 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。ctxsys.contains() 演算子を使用した Oracle Text 問合せに関する詳細な知識が必要です。Oracle Text は、問合せ言語の一部として、シソーラス、テーマ、ステミングおよび同音語など、豊富な言語機能のセットを備えています。

### 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』

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、oracle.ultrasearch.query.Contains クラスのかわりに、検索アプリケーション内の oracle.ultrasearch.CtxContains クラスを使用します。CtxContains を使用すると、任意の Oracle Text の問合せを Oracle Ultra Search の問合せの一部として使用できます。次の手順を実行します。

1. ユーザー入力に基づいて Oracle Text の問合せを構成します。たとえば、ユーザー入力がある "cat" の場合、ステミング機能を使用して、Oracle Text の問合せ "\$cat" を構成できます。これにより、"cat" または "cats" が含まれているドキュメントが検索されます。Oracle Text の問合せの構成には、文字列オブジェクトであればどのようなツールでも使用できます。ユーザーの問合せ構文がどの程度複雑であるかにより、一部の既存のレクサーを Java で活用できます。
2. Oracle Text の問合せを使用して、CtxContains を構成します。次に例を示します。

```
String textQuery = "$cat";
oracle.ultrasearch.Query query = new oracle.ultrasearch.CtxContains (textQuery);
```

このコードは、"cat" または "cats" が含まれているドキュメントに対する問合せを構成します。また、次のようにして、問合せの対象をドキュメントのタイトル（コンテンツを含まない）のみに制限することも可能です。

```
String textQuery = "cat";
StringAttribute titleAttribute = instanceMetaData.getStringAttribute("TITLE");
oracle.ultrasearch.Query query = new oracle.ultrasearch.CtxContains (textQuery,
titleAttribute);
```

3. オプションで、And/Or 問合せ演算子で問合せを連結することにより、CtxContains を他の Oracle Ultra Search の問合せと組み合わせることもできます。
4. 構成した問合せオブジェクトで getResult() メソッドを起動して問合せを実行します。

**関連項目：** oracle.ultrasearch.query.CtxContains API の詳細は、『Oracle Ultra Search API Reference』を参照してください。

## 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、および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があります。

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。
filePagePath	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.ultrasearch.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、この属性で利用できる演算子、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="ローケール"	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.ultrasearch.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 >
```

## <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
<code>resultId="名前"</code>	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。
<code>instance="名前"</code>	<code>instance</code> 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。
<code>query="&lt;%= expression %&gt;"</code>	検索対象の問合せオブジェクトを指定します。
<code>queryLocale="ロケール"</code>	問合せオブジェクトのロケールを指定します。
<code>documentLanguage="ロケール"</code>	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。
<code>from="数値"</code>	最初のヒットの索引を指定します。
<code>to="数値"</code>	最後のヒットの索引を指定します。
<code>boostTerm="文字列"</code>	検索オプションで使用され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。
<code>withCount="true   false"</code>	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、 <code>withCount=false</code> 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.ultrasearch.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>
```

## Oracle Ultra Search クローラ・エージェント API

クローラ・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。クローラによるデータ・ソースへのアクセスを可能にするモジュールをクローラ・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 クローラに情報を戻します。Oracle Ultra Search クローラでは、後でクロールするためにその情報をエンキューします。クローラ・エージェントは、Oracle Ultra Search のクローラ・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クローラ・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、クローラ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・パラメータの読取り
- 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

## クローラ・エージェントの概要

クローラ・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クローラの認証
- 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
- 各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クローラに指示
- (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、クローラ処理の前にそのリストをクローラ・キューに保存します。

---

---

**注意:** なんらかの理由でクローラが中断すると、元のクローラの最新タイム・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 のエンキューが終了し、クローラ中の場合、クローラ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 のクローラを終了します。

---

---

2種類のクローラ・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クローラに依存します。標準エージェントとクローラの相互作用は次のとおりです。

- クローラは、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。
- クローラは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。
- クローラは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、クローラ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。クローラは、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、クローラによって 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、クローラ側では削除された URL は検出されません。この場合は、スケジュールのクローラ再クロール・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。

エージェント API `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クローラに通知します。エージェント API `startCrawling(boolean forceRecrawl, Date lastCrawlTime)` によって、クローラは、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。クローラ・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。

## クローラ・エージェントの機能

ここでは、クローラ・エージェントの側面について説明します。

### 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 ではなく、Oracle Ultra Search のバックエンド (サーバー・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。

## ユーザー実装クローラ・エージェント

クローラ・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) クローラによるデータ・ソースへのアクセスを認証できること。
- 各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 で保持する必要が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的 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。

## クローラとクローラ・エージェントの相互作用

クローラは、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。クローラは次の処理を実行できます。

- 定義済みデータ・ソースのクローラ・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)

## クローラ・エージェント API およびクラス

クローラ・エージェント API は、クローラ・エージェントの実装に使用するメソッドの集合です。クローラ・エージェントの実装例 `SampleAgent.java` は、`$ORACLE_HOME/ultrasearch/extension/` にあります。

**UriData:** クローラ・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、`UriData` 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:** クローラ・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:** クローラ・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・パラメータの読み取りおよび書き込みを行います。

**AgentException:** クローラ・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、クローラはユーザー定義のデータ・ソースと通信できます。クローラ・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/extension` 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・コードを表示できます。

`SampleAgent_readme.htm` ファイルと `SampleAgent.java` ファイルがあります。これらは、エージェント API を使用するサンプル・クローラ・エージェント実装用です。

## サンプル・クローラ・エージェントの設定

ここでは、サンプル・クローラ・エージェントの設定方法について説明します。

### 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 ではなく、Oracle Ultra Search のバックエンド (サーバー・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 `ultrasearch.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 ファイルを作成するには、次のように入力します。

```
/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 クローラが電子メール・ソースを検索する場合、クローラ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、README ファイルおよびスタイルシートを示します。

ファイル	説明
SampleAgent_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
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する背景イメージ

## 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、2-12 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 リライター API

URL リライターは、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。クローラは、アクティブ化された URL リライターを使用して、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 クローラ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)

---

---

**注意：** すべての URL は、パス・ルールをチェックする前に、ドメイン・ルールを通過する必要があります。パス・ルールを使用すると、クロール領域をさらに制限できます。パス・ルールはホストに固有ですが、ホストごとに2つ以上のパス・ルールを指定できます。たとえば、同じホスト上にパス files://host/doc および除外パス files://host/doc/unwanted を含めることができます。

---

---

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せん。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

クローラが抽出した 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、考えられる基準は非常に大規模になる可能性があります。

## 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、クローラ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、クローラは表示 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、クローラは表示 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、クローラ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-qa.us.com:9393/index.html  
http://www.acme.com/index.html
```

URL リンク `http://www.acme-qa.us.com:9393/index.html` が抽出されると、キューに挿入される前に、クローラによって、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。

アクセス URL:

```
http://www.acme-qa.us.com:9393/index.html
```

表示 URL:

```
http://www.acme.com/index.html
```

抽出された URL リンクはリライトされ、クローラは、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、クローラ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
```

クローラ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、クローラ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 リライターの作成および使用

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/extension/` 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. 7-20 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・ソースのクローラ・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。クローラ・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 リライターの使用を確認します。



---

---

**注意：** URL リライタは、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 API Reference』を参照してください。
- サンプル URL リライタ `SampleRewriter.java` は、`$ORACLE_HOME/ultrasearch/extension/` にあります。
- 7-20 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・クローラ・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・クローラ・エージェントを使用して、独自のクローラ・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・クローラ・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:port/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、OracleAS 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 OracleAS 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、電子メールの読取り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 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スケーラビリティを強化するために、他のコンピュータ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。

## サンプル検索アプリケーション

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、Oracle Application Server 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スケーラビリティを向上させるために、Oracle データベース・サーバーとは別のコンピュータで実行されます。Oracle データベース・サーバーは、Oracle Ultra Search の索引を保持しません。

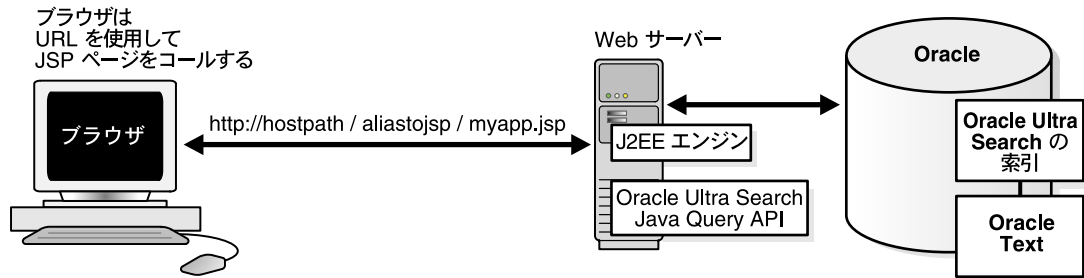
JSP アプリケーションは、実行時に Java サブレットにコンパイルされます。コンパイルされたサブレットは、1 つ以上の Java Virtual Machine (Java VM) で実行されます。JSP アプリケーションは、Oracle JDBC ドライバを使用して Oracle データベース・サーバーと通信します。

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。

- `$ORACLE_HOME/ultrasearch/lib/ultrasearch_query.jar`
- `$ORACLE_HOME/lib/mail.jar`
- `$ORACLE_HOME/lib/activation.jar`

図 8-1 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 8-1 JSP のコール





---

---

# 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。

**関連項目：** 1-10 ページ [「ドキュメント検索オプション」](#)

## 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
```

使用するコンピュータには、Java 1.2 以上に準拠した **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：** A-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 Web site</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：** [A-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>
</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:complexType>
        </xsd:element>
      </xsd:sequence>
      <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: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:complexType>
  </xsd:element>
  <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: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>
```

---

---

## クローラの Java CLASSPATH の変更

Oracle Ultra Search クローラは、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、デフォルトのクローラ 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、クローラ CLASSPATH として特定のインスタンスで使用されます。

## クローラの Java CLASSPATH を変更するケース

通常は、クローラの 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。

## クローラ CLASSPATH とリモート・クローラ CLASSPATH の違い

クローラ CLASSPATH とは、Oracle Ultra Search のバックエンド（サーバー・コンポーネント）と同じホストで実行するクローラの CLASSPATH です。ただし、Oracle Ultra Search には、スケーラビリティを得るために他のホストで実行できるリモート・クローラがあります。リモート・クローラの起動には、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・クローラの 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：** 4-6 ページ「[リモート・クローラの使用方法](#)」

## Oracle Ultra Search Server ホストでのクローラの Java CLASSPATH 変更

1. Oracle Ultra Search のバックエンド（サーバー・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/wk0addcpath.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・ユーザーまたはスーパー・ユーザー権限を付与されたデータベース・ユーザーとして wk0addcpath.sql スクリプトを実行します。（このスクリプトは、CRAWLER\_CONFIG\_DEFAULT 表のみを更新します。クローラを再構成して、WK\$CRAWLER\_CONFIG 表を正しく更新する必要もあります。）
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 のバックエンド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（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・クローラ・ホストでのクローラの Java CLASSPATH 変更

1. Oracle Ultra Search Middle-Tier がインストールされているリモート・クローラ・ホストにログオンします。UNIX コンピュータの場合は、次のファイルの位置を特定して開きます。`$ORACLE_HOME/ultrasearch/tools/remotecrawler/scripts/unix/define_env`  
Windows コンピュータの場合には、次のファイルの位置を特定して開きます。`$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、4-6 ページの「[リモート・クローラの使用法](#)」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、WK\_QUERYEXP PL/SQL パッケージに含まれています。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。

この付録では、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 [+] は、その後続くトークンが、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 [-] は、その後続くトークンが、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 [\*] 演算子は、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。

## ルールのサマリー

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、<code>[Oracle]</code> 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> <p><b>注意:</b> <code>[Oracle]</code> の検索は、<code>[Oracle*]</code> とは異なる。</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、<code>[Oracle Applications]</code> 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 <code>[+]</code> 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、<code>[Oracle+Applications]</code> 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。<b>注意:</b> 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 <code>[+]</code> を付けることができる。</p>
強制的な排除 [-]	<p>単語の前に <code>[-]</code> 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、<code>[Oracle-Applications]</code> 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。<b>注意:</b> 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 <code>[-]</code> 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、<code>["Oracle Applications"]</code> 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 <code>[*]</code> 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、<code>[Ora*]</code> 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、<code>[A*e]</code> 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。

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>)
```

ここでは、前述のルールで使用されている項目およびその意味について説明します。

<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あり、演算子または二重引用符がない場合のみ使用されます。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は、<plus expression> および <accum expression> が排除されます。

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、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、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。

---

## 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]	<code>((({Oracle Applications}) *2, (({Oracle}; {Applications}) *2, ({Oracle}, {Applications}))) within TITLE_31) *2, (({Oracle Applications}) *2, (({Oracle}; {Applications}) *2, ({Oracle}, {Applications}))))</code>
タイトル属性が「MyTitle」に、作成者属性が「MyAuthor」に限定された [Oracle Applications]	<code>((({Oracle Applications}) *2, (({Oracle}; {Applications}) *2, ({Oracle}, {Applications}))) within TITLE_31) *2, (({Oracle Applications}) *2, (({Oracle}; {Applications}) *2, ({Oracle}, {Applications})))) &amp; (((({MyTitle}) WITHIN TITLE_31) &amp; (({MyAuthor}) WITHIN AUTHOR_32)) *10) *10</code>





# 索引

## C

CTXSYS ユーザー, 3-4

## D

data-sources.xml ファイル, 2-22

DB\_CACHE\_SIZE パラメータ, 4-3

DBMS\_JOB パッケージ, 1-3

## E

Email API, 1-3, 8-27

## F

federator.rar, 7-29

Federator サーチレット, 7-29

## H

HTTPS, 5-3, 7-20, 7-27

## I

INSO フィルタ, 2-5

## J

Java CLASSPATH, B-1

JAZN, 5-8

jazn-data.xml, 5-9

JDBC, 2-21, 2-22, 2-23, 2-27, 4-4, 4-6, 4-12, 4-13,  
5-8, 7-15, 8-2, 8-12, 8-24, 8-33, 8-34, A-2

JOB\_QUEUE\_PROCESSES 初期化パラメータ, 3-4

## O

OC4J, 2-12, 2-14, 2-15, 2-16, 2-18, 4-4, 5-8, 8-33

Oracle Enterprise Manager, 2-15, 2-25, 3-2, 7-4

Oracle Internet Directory, 1-11, 2-6, 5-3, 5-9, 7-4

Oracle Text, 1-2, 1-3, 2-2, 2-9, 3-4, 4-10, 6-2,  
6-6, 6-8, 8-4, C-1

Oracle Ultra Search

Administration ツール, 7-2

API, 1-3

instance

デフォルト, 2-11

Middle-Tier, 2-12

OID との統合, 1-11, 5-9

Oracle Application Server との統合, 1-5

Oracle Ultra Search Server コンポーネント, 1-3

Real Application Clusters, 4-10

アップグレード, 3-12

方法, 3-12

インスタンス, 7-7, 7-10

作成, 7-7

スナップショット, 7-7

インスタンス管理者, 1-12, 5-3

概要, 1-2

管理権限, 7-42

グローバルゼーション, 7-43

クローラ, 1-2, 6-2

言語, 7-13, 7-43

検索ポートレット, 1-7

構成, 1-13, 3-2

コンポーネント, 1-2

システム要件, 2-3

スーパー・ユーザー, 1-12, 5-3

スナップショット・インスタンス, 7-8

チューニング, 4-2

デフォルト・インスタンス, 5-3  
バックエンド (サーバー・コンポーネント), 2-4  
メタデータ・ローダー, A-1  
ユーザー, 7-41  
ユーザーの管理, 7-41  
リモート・クローラ, 6-10  
ログオン, 7-4  
Oracle Ultra Search サーチレット, 7-28  
OracleAS Portal, 1-5

## P

---

PROCESSES 初期化パラメータ, 3-4

## Q

---

Query API, 1-3, 1-8, 8-2  
Query タグ・ライブラリ, 8-10

## R

---

REDO ログ・ファイル  
    サイズ変更, 3-2  
robots.txt ファイル, 1-9, 7-22, 8-30

## S

---

Single Sign-On Server, 1-5  
SORT\_AREA\_RETAINED\_SIZE 初期化パラメータ,  
    3-4  
SORT\_AREA\_SIZE 初期化パラメータ, 3-4

## U

---

ultrasearch.rar, 7-28  
UNDO 領域  
    サイズ変更, 3-3  
URL 発行, 7-37  
URL リライタ, 1-8, 7-21, 8-29  
    作成, 8-32  
    使用, 8-32  
URL リライタ API, 1-3  
URL リンクのフィルタ処理, 8-30  
URL リンクのリライト, 8-31  
URL ルーピング, 4-2

## W

---

Web クロール, 8-29  
WK\_INST デフォルト・インスタンス, 5-3  
WK\_TEST インスタンス管理者, 5-3  
wk0migrate.sql スクリプト, 3-14, 3-15  
wk0pref.sql ファイル, 3-7, 6-2  
wk0upgrade.sql スクリプト, 3-13, 3-15  
WKSYS.WK\_QRY パッケージ, 4-5  
WKSYS データベース・ユーザー, 2-5, 2-28, 3-6,  
    3-8, 3-14, 5-4, 7-4, B-2  
    パスワードの変更, 3-2  
WKUSER ロール, 3-6, 7-42

## X

---

XML DB, 1-7, 2-6

## あ

---

アクセス URL, 6-3, 8-20, 8-31  
アクセス制御リスト, 1-6  
値リスト (LOV), 1-8, 1-9, 1-10, 6-4, 7-18, 7-43,  
    7-44, 8-10, 8-24, A-1

## い

---

移行ログ, 3-15  
インスタンス・スナップショット, 1-11

## か

---

管理グループ, 1-11

## く

---

クローラ, 6-2  
    CLASSPATH, B-1  
    概要, 6-2  
    クローラ・エージェント, 6-3  
    クロール・プロセス, 6-4  
    設定, 6-2, 7-12  
    データ・ソース, 6-2  
    統計, 7-16  
    パラメータ, 7-2, 7-41  
    リモート・クローラ, 7-15

クローラ・エージェント, 1-5  
API, 8-19  
機能, 8-22  
サンプル・エージェント・ファイル, 8-24  
スマート・エージェント, 8-21  
設定, 8-24  
標準エージェント, 8-21  
クローラ・エージェント API, 1-3

## け

---

検索オプション, 1-10, 7-38  
制限事項, 1-10  
検索属性, 1-9, 7-18

## さ

---

サーチレット, 1-6  
索引  
最適化, 7-36  
変更, 3-7, 6-2  
サンプル検索アプリケーション, 1-7, 8-33

## す

---

スケジュール  
索引最適化, 7-36  
データの同期, 7-32  
ストップリスト, 3-7  
デフォルト, 3-7  
変更, 3-8

## せ

---

セキュアな検索, 1-6, 2-6, 2-8

## て

---

データ・グループ, 7-3, 7-37  
データ収集モード, 1-11  
データ・ソース, 7-20  
Web, 7-20  
電子メール, 7-25  
同期, 6-3  
表, 7-23  
同期, 4-13

ファイル, 7-26  
ユーザー定義, 6-3, 7-30  
デフォルト・インスタンス, 5-3

## と

---

問合せ構文拡張, 1-13, 8-4, C-1  
問合せ統計, 7-39  
ドキュメント属性, 1-9, 6-4  
ドキュメントのキャッシュ, 6-5  
ドキュメントのキューイング, 6-5  
ドキュメントの索引付け, 6-8  
ドメイン・ルール, 8-30  
トリガー, 4-15

## に

---

認証, 1-12  
シングル・サインオン, 1-12, 5-3, 7-4

## は

---

パス・ルール, 7-26, 8-30

## ひ

---

表示 URL, 6-3, 7-22, 7-24, 7-27, 8-20, 8-31

## ふ

---

プロキシ・サーバー, 7-17

## め

---

メタデータ, 6-4  
ロード, A-1  
メタデータ・ローダー, 1-10

## り

---

リソース・アダプタ, 1-6  
リモート・クローラ, 4-6, 6-10  
プロファイル, 7-15  
リモート・クローラ・ホスト  
インストール, 2-27

## れ

---

連携した検索, 1-5

## ろ

---

ロボットの除外, 1-9, 7-22