



**19<sup>c</sup>** ORACLE<sup>®</sup>  
Database

# Oracle SecureFiles

Oracle ホワイト・ペーパー | 2019年2月



ORACLE<sup>®</sup>



## 目次

免責事項 .....	1
概要.....	2
ファイル・データにとって最適なストレージ.....	3
SecureFiles の概要.....	3
SecureFiles のストレージ・パラメータ .....	5
優れたパフォーマンス .....	5
SecureFiles の使用.....	7
拡張機能 .....	8
移行.....	11
インタフェース .....	11
結論.....	12

## 免責事項

下記事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

## 概要

重要なビジネス情報の性質は急速に変化しており、構造化テキストや数値データに限られなくなりました。イメージ、音声、ビデオ、PDF、Word ドキュメントなどの非構造化コンテンツが増加し続けています。Web ベースのアプリケーションにより、企業は以前にも増して、写真、ビデオ、ニュース・クリップなどの非構造化データを取得し、管理することを余儀なくされています。さらに、米国のサーベンス・オクスリー（SOX）法や医療保険の相互運用性と説明責任に関する法律（HIPAA）などの規制の変更が、この傾向に拍車をかけています。

主要なビジネス・アプリケーションでは、リレーショナル・コンテンツ（構造化コンテンツ）と非構造化コンテンツの両方を統合し、ユーザーにシームレスで豊かなユーザー・エクスペリエンスを提供する必要があります。これまで、非構造化コンテンツはファイル・システム内に保管されてきました。これは、ファイル・システムの簡索性やパフォーマンスによるものです。しかしながら、ファイル・システムを選択すると、企業はファイルや非構造化データに対する Oracle 製品のセキュリティ、スケーラビリティ、トランザクション機能、読取り一貫性、高可用性を犠牲にせざるを得ません。そして、このトレードオフには通常は法外なコストがかかります。

構造化データと非構造化データに対して異なるストレージ・メカニズムを使用すると、セキュリティ・モデルや監査モデルの一貫性がなくなり、関連するすべてのエンタープライズ・コンテンツ内での検索が困難になります。また、バックアップやリカバリにおける IT 管理手順はまったく異なるものになります。その結果、最終的に ROI の低下を招く可能性があります。SecureFiles は、Oracle Database の利点を維持しながら、ファイル・データや非構造化データに対して、従来のファイル・システムに匹敵する高いパフォーマンスを発揮できるように設計されている Oracle Database の機能です。

SecureFiles では、構造化データと非構造化データの両者にとって最高のアーキテクチャに非構造化コンテンツを保管できるため、犠牲を伴う必要はなくなります。SecureFiles は、ANSI 規格のラージ・オブジェクト（LOB）のスーパーセットとして設計されており、旧 LOB から BasicFiles の移行を容易にします。アプリケーションは、ユーザーに気付かれることなく、業界標準の暗号化を利用してセキュリティを強化したり、圧縮や重複排除といった高度なストレージ機能を利用したりして、さらなる領域の節約やパフォーマンスの向上を図ることができます。

組織では、SecureFiles により、単一のセキュリティ/監査モデルや統合されたバックアップおよびリカバリ・プロセスを使用して、すべてのリレーショナル・データおよび関連するファイル・データを Oracle で管理でき、全情報にわたってシームレスな検索を実行できるようになります。そのため、Oracle Database は、構造化データ（リレーショナル・データ）および関連する非構造化データの両方を格納するための理想的なソリューションになります。

## ファイル・データにとって最適なストレージ

Oracle Database で SecureFiles が提供されることにより、構造化コンテンツ用のストレージと非構造化コンテンツ用のストレージを区別する必要がなくなります。SecureFiles により、あらゆるエンタープライズ・コンテンツ用のストレージにデータベースを使用できるようになりました。

SecureFiles は、非構造化コンテンツを管理するための Oracle Database の中核的インフラストラクチャです。SecureFiles アーキテクチャは、LOB データ・ブロックとメタデータ・ブロック向けの新しいディスク形式、ネットワーク・プロトコル、領域管理、REDO および UNDO 形式、バッファ・キャッシュ、インテリジェントな I/O サブシステムなどを特徴としています。SecureFiles により、パフォーマンスが大幅に向上するほか、非構造化データ向けに最適化されたストレージが Oracle Database 内で提供されます。

注：Oracle Database 10g およびそれ以前のリリースの LOB は、'BasicFiles'を使用して引き続きサポートされますが、このホワイト・ペーパーでは、SecureFiles の概要と、この機能の使用例をいくつか紹介しています。

## SecureFilesの概要

SecureFiles には、パフォーマンス、スケーラビリティ、効率的なストレージ、容易な管理性を大幅に向上するためのアーキテクチャ上の機能拡張が多数実施されています。革新的な機能の一部を以下に紹介します。

### Write-Gather Cache (WGC)

SecureFiles WGC は、書き込み操作中、フラッシュする前、または下層ストレージ・レイヤーにコミットする前に、通常は最大 4 MB のデータをバッファリングします。実行中のデータをバッファリングするこの機能によって、連続する大きい領域割当てと大きいディスク I/O が実現します。ディスクを探す負担が減るため、書き込みパフォーマンスが大幅に向上します。

WGC はトランザクションごとに維持されます。1 つの WGC のみが、トランザクションの全 SecureFiles 用で使用されます。4 MB の上限に到達する前に、WGC がディスクにフラッシュされる必要があるかどうか自動的に判断されます。ユーザーの 'コミット' によっても、WGC はディスクにフラッシュされます。

注：WGC のサイズは 4 MB です。WGC はユーザーがチューニングできるパラメータではありません。

### 領域管理

領域の再利用は、バックグラウンドで行われます。コミットされた空き領域とコミットされていない空き領域を別々に管理することで、領域管理は大幅に向上します。そのため SecureFiles では、以前の LOB とは異なり、領域の再利用中に最高水位標の競合に悩まされることはありません。

SecureFiles セグメントには、自動セグメント領域管理 (ASSM) で管理される表領域が必要です。SecureFiles の領域管理は、Oracle Database 内にファイル・データを保管するために最適化されています。インテリジェントな領域マネージャにより、連続する大量のディスク・ブロックが迅速に割り当てられ、空き領域を自動的に再利用することで領域が効率的に削除されます。最近の需要の経験則に基づき、領域の事前割当ても行われます。この事前割当てにより、割当てが行き詰まることや、ひいてはフォアグラウンドのパフォーマンスに影響が及ぶことはありません。領域管理は、自己適応が可能のように設計されており、統計を使用して効率的にメモリや領域が割り当てられます。

## 断片化の削減

BasicFiles や以前の LOB とは異なり、SecureFiles では、動的チャンク（連続する 1 つまたは複数の Oracle ブロック）サイズ設定を使用して、ディスク上の連続した割当てを最大化し、断片化を抑えます。ユーザー指定のチャンク値は、SecureFiles のサイズやセグメントでの使用可能領域といった他の要素とともに、最適なチャンク・サイズを決定するための推奨値として使用されます。SecureFiles チャンクは、完全に内部的な概念です。

固定チャンク・サイズ設定では、設定が大きすぎると内部の断片化が発生し、設定が小さすぎると書き込みパフォーマンスが低下します。動的チャンク・サイズ設定を使用すると、1 回の操作で、大規模なディスク領域の割当てと割当て解除を行うことができます。このように、SecureFiles は、あらゆるサイズの非構造化データやファイル・データで、高いパフォーマンスとストレージ効率を実現するソリューションとなるように一から設計されています。

## インテリジェントなプリフェッチ

読取りパフォーマンスは、データがネットワークで送信されているときに、ディスクからの先読みまたはデータのプリフェッチを行うことで強化されます。SecureFiles は、実際にリクエストが実行される前に、アクセス・パターンを追跡し、データをインテリジェントにプリフェッチします。ネットワーク・ラウンドトリップとデータをプリフェッチするためのディスク I/O が同時に発生するため、読取り待機時間が短縮され、スループットが大幅に向上します。

## ネットワーク・レイヤー

SecureFiles のクライアント/サーバー・ネットワーク・レイヤーにより、クライアントとサーバー間の高速データ送信が可能になります。一時的なステージングなしに、ネットワーク・ソケットからバルク・データを直接読み取ったり、書き込んだりできるため、読取り/書き込みのパフォーマンスが大幅に向上します。

## LOB索引競合の排除

以前の LOB では、セグメントごとのグローバルな B+ツリー索引を使用して i ノードを保持します。i ノードとは、論理オフセットを物理ディスク・ブロックにマッピングするデータ構造です。グローバルな B+ツリー構造は、LOB へのアクセスとナビゲーションの両方で使用され、LOB セグメントの空き領域管理でも使用されます。そのため、同時実行性の高い環境では、競合や大幅なパフォーマンスの低下を招きます。

以前の LOB とは異なり、SecureFiles では、このようなメタデータの管理に LOB 索引を使用しません。代わりに、LOB セグメントのデータ・ブロックとともに割り当てられる“プライベートな”メタデータ・ブロックを使用します。この設計により、とりわけ挿入、削除、再利用が頻繁に行われる実際の環境において、BasicFile LOB で発生していた LOB 索引の競合がなくなり、パフォーマンスが大幅に向上します。

## 容易な管理性

SecureFiles は、自己チューニングと、インテリジェントな領域およびメモリ管理アルゴリズムを備えているため、ユーザーはより少ないパラメータをチューニングするだけで済みます。特に、FREEPOOLS、FREELISTS、FREELIST GROUPS、PCTVERSION は指定する必要がなく、SecureFiles では無視されます。

これらのパラメータの一部は、ワークロードの予測不可能な急増に備えてチューニングするのが困難であるだけでなく、オンラインで変更することもできません。SecureFiles は、内部統計を維持して領域管理アルゴリズムを自己チューニングし、多様なワークロード下で高いパフォーマンスとスケーラビリティを実現します。

## SecureFilesのストレージ・パラメータ

1. ENABLE STORAGE IN ROW (デフォルト) : (他の値は DISABLE STORAGE IN ROW)
2. NOLOGGING
3. CACHE
4. NOCACHE (デフォルト)
5. CACHE READS
6. LOGGING (デフォルト)
7. FILESYSTEM\_LIKE\_LOGGING

CACHE や NOCACHE などのパラメータは、インライン LOB とは無関係です。インライン LOB はヒープ表のセグメント・ブロックに保存され、ヒープ・ブロックの他の非 LOB データと同様に、通常のメカニズムを使用してアクセスされるためです。これらのパラメータは、表外 LOB (LOB セグメントに保存されている LOB) への影響を与えます。

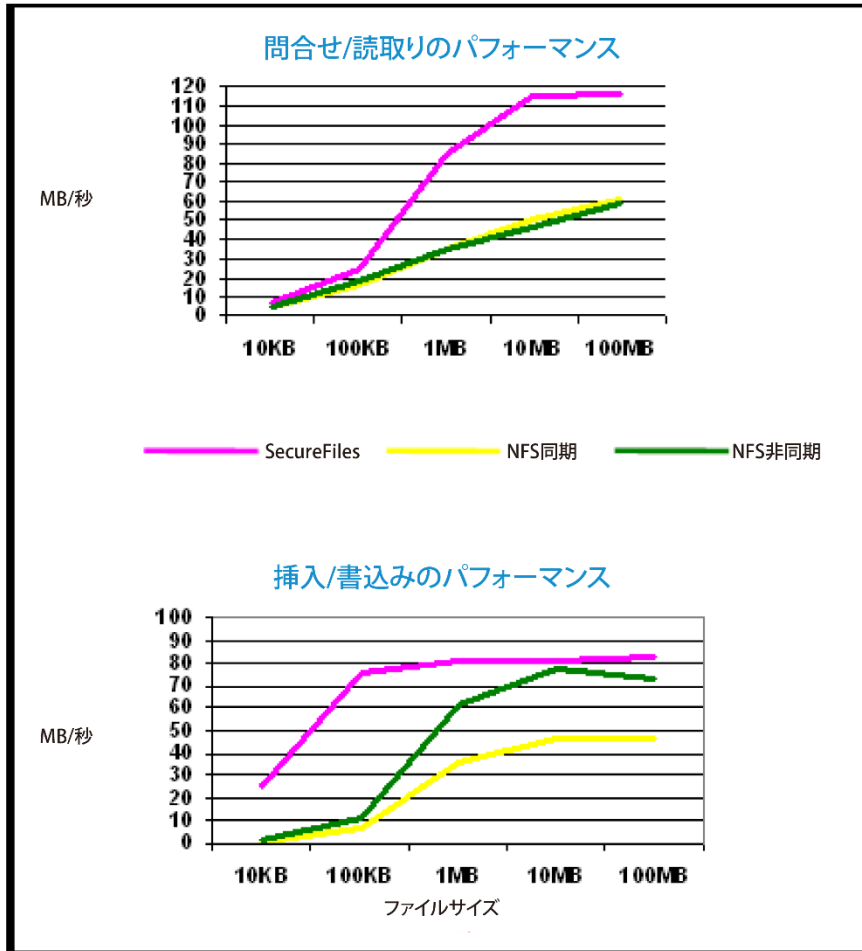
同様に、LOGGING、NOLOGGING もインライン LOB とは無関係です。インライン LOB の変更は、常に対応する REDO に伴って発生します。これらのパラメータは、表外 LOB (LOB セグメントに保存されている LOB) への影響を与えます。

SecureFiles では、NOLOGGING は暗黙的に FILESYSTEM\_LIKE\_LOGGING に変換されるため、メタデータのみジャーナリング・ファイル・システムと意味的に同等の動作となります。

## 優れたパフォーマンス

問合せと挿入の双方における SecureFiles のパフォーマンスを、Linux NFS/Ext3 のパフォーマンスと比較しました。テスト・アプリケーションは、患者のメタデータと DICOM イメージで構成される実在の DICOM アプリケーションを模倣したものです。このテスト・アプリケーションで、SecureFiles のパフォーマンスと NFSv3 ファイル・システムのパフォーマンスを比較しました。どちらの場合も、メタデータは Oracle Database に保存されず、ファイル・システムの場合、DICOM イメージは、NFSv3 経由でクライアント・マシンからアクセスされるファイル・サーバー上に保存されます。SecureFiles の場合、メタデータおよび DICOM イメージは Oracle Database 内に格納されます。

テストでは、SecureFiles へのロギングを FILESYSTEM\_LIKE\_LOGGING モードに設定し、ロギング・レベルと機能性をファイル・システムと比較できるようにしています。そして、NFSv3 のパフォーマンスを非同期モードおよび同期モードの両方で比較しました。



SecureFiles は、読取り操作と書込み操作の両方で、すべてのサイズにおいて NFSv3 アクセスよりも優れていました。小規模なファイル・サイズで示された優位性は、メタデータやファイルが個々のラウンドトリップでアクセスされたり、書き込まれたりする NFSv3 の場合とは異なり、メタデータやデータが 1 回のラウンドトリップでアクセスされるため、少ないラウンドトリップ数で済むことによるものです。

## SecureFilesの使用

### 互換性

SecureFiles の機能は、互換性を Oracle Database 11g Release 1 以上に設定することで使用できます。Oracle Database 12c 以降では、SecureFiles は LOB のデフォルトのストレージ・メカニズムです。Oracle Database 11g Release 1 では、'BASICFILE'がデフォルトの LOB ストレージ・タイプです。SecureFiles を指定する場合は、キーワード'SECUREFILE'を使用します。デフォルトの LOB ストレージは、db\_securefile パラメータを使用して変更できます。

### db\_securefileパラメータ

データベース管理者は、このシステム・パラメータを使用して、SecureFile の使用方針を init.ora ファイルで指定できます。このパラメータは動的で、有効範囲は'ALTER SYSTEM'です。

このパラメータで使用できる値は次のとおりです。

PERMITTED : SecureFiles の作成を許可します (デフォルト) FORCE : すべての LOB を SecureFiles として作成します

ALWAYS : SecureFiles を作成しようとしますが、BasicFiles にフォールバックします IGNORE : SecureFiles を作成しようとする試みを無視します

NEVER : 新たな SecureFile の作成を許可しません

### 保存

RETENTION キーワードは、データベースが SecureFiles の読取り一貫性 (CR) を管理するために使用する保存方針を指定するものです。CREATE TABLE または ALTER TABLE 文の lob\_storage\_clause で指定します。SecureFile 列を構成して、SecureFile データの旧バージョンを一定期間保存するために使用されます。このパラメータで使用できる値は次のとおりです。



**AUTO**（デフォルト）：領域を可能な限り効率的に自動で管理します。読取り一貫性機能に十分な UNDO が保持されます。

**MAX**：SecureFiles は、LOB セグメントのサイズが MAXSIZE で指定したしきい値に到達するまで、LOB データの旧バージョンを保持します。MAXSIZE が指定されていない場合は、MAX は AUTO と同じように動作します。

RETENTION：MAX

および

MAXSIZE：UNLIMITED

の組み合わせが使用された場合、RETENTION の値は無視され、

RETENTION：AUTO

および

MAXSIZE：UNLIMITED

の組み合わせが有効になります。

**MIN**：UNDO の保存期間を定められた秒数で指定します。

**NONE**：RETENTION が NONE の場合、領域管理レイヤーでは UNDO は保存されないため、解放された領域は即座に再使用可能になります。

## 拡張機能

以下に挙げる機能は、SecureFile でのみ使用でき、BasicFiles では使用できません。SecureFiles の重複排除、圧縮、暗号化の機能は、個別に設定することも、1 つまたは複数の機能と組み合わせて設定することもできます。3 つの機能すべてが有効化されている場合、重複排除がまず実行され、次に圧縮、暗号化の順に実行されます。

## 重複排除

この機能は、SecureFiles データの複数の冗長コピーを排除するものであり、アプリケーションに対して完全に透過的です。複数の同一の SecureFiles データが自動的に検出され、1 つのコピーのみが保存されるため、ストレージ領域が節約されます。重複排除により、ストレージ管理が簡素化されるだけでなく、とりわけコピー操作でパフォーマンスが大幅に向上します。

DEDUPLICATION は LOB セグメントごとのベースで実行されます。複数の LOB セグメントにまたがる重複は検出されません。lob\_storage\_clause を使用すると、パーティション・レベルで重複排除を指定できます。ただし、パーティション化された SecureFiles の列のパーティションまたはサブパーティションにまたがって重複が検出されることはありません。

DEDUPLICATE キーワードは、SecureFiles の高度な LOB 重複排除チェックを有効化する場合に使用されます。セキュア・ハッシュ索引を使用して、重複する SecureFile データが検出され、同一のコンテンツすべてに対して単一のコピーが保存されます。重複排除はアプリケーションに対して透過的であり、ストレージを削減し、ストレージ管理を簡素化します。KEEP\_DUPLICATES は、重複排除を無効化し、SecureFile データの重複コピーを保持する場合に使用されます。重複排除はデフォルトでは無効化されています。

**注**：高度な LOB 重複排除（Oracle Database 11g での旧名称は SecureFiles LOB 重複排除）は、Oracle Advanced Compression の機能です。

例：

- SECUREFILE LOB 列を含む表を作成し、LOB 重複排除を 1 つのパーティションでのみ有効化します。パーティション p1 に属する LOB のみが重複排除されます。

```
CREATE TABLE t1 ( REGION VARCHAR2(20), mydata BLOB)
  LOB(mydata) STORE AS SECUREFILE )
PARTITION BY LIST (REGION) (
  PARTITION p1 VALUES ('x', 'y')
    LOB(mydata) STORE AS SECUREFILE (
      DEDUPLICATE) ,
  PARTITION p2 VALUES (DEFAULT)
);
```

- SECUREFILE LOB 列の重複排除を無効化します。

```
ALTER TABLE t1 MODIFY
  LOB(mydata) (
    KEEP_DUPLICATES);
```

## 圧縮

Oracle では、SecureFiles データが圧縮可能かどうかを検出され、業界標準の圧縮アルゴリズムを使用して圧縮されます。圧縮で節約が得られない場合や、データがすでに圧縮されている場合は、SecureFiles ではそのような LOB の圧縮がオフになります。圧縮により、ストレージが大幅に削減されるだけでなく、I/O、バッファ・キャッシュの必要量、REDO 生成、暗号化のオーバーヘッドが低減されるため、パフォーマンスが向上します。また、ファイル全体を圧縮解除しなくとも、圧縮された SecureFile のランダムな読み取りおよび書き込みが可能です。読み取りおよび書き込みが行われる論理オフセットに対応する圧縮ファイルのサブ部分（圧縮単位）のみが圧縮解除されるため、CPU と I/O が節約されます。

高度な LOB 圧縮には、LOW、MEDIUM、HIGH の 3 つのレベルがあります。高度な LOB 圧縮では、デフォルトで MEDIUM レベルが使用されます。MEDIUM レベルでは通常、3~5% という少ない CPU オーバーヘッドで適度な圧縮が実行されます。高度な LOB 圧縮の LOW は、高パフォーマンス向けに最適化されています。高度な LOB 圧縮の LOW では、MEDIUM によって達成される圧縮の約 80% が、3 分の 1 の CPU 使用量で維持されます。高度な LOB 圧縮の HIGH では、ストレージの節約が最大になりますが、CPU オーバーヘッドも最大になります。

COMPRESS キーワードは、SecureFile の高度な LOB 圧縮を有効化する場合に使用されます。SecureFile の圧縮は、表や索引の圧縮とは全く異なります。表や索引の圧縮の設定は、SecureFile の圧縮には影響を与えず、その逆も同様です。パーティション化された表では、lob\_storage\_clause を使用してパーティション・レベルで圧縮を指定します。

**注：**高度な LOB 圧縮 (Oracle Database 11g での旧名称は SecureFiles LOB 圧縮) は、Oracle Advanced Compression の機能です。

例：

- SECUREFILE LOB 列を含む表を作成し、LOB 圧縮を 1 つのパーティションでのみ有効化します。パーティション p1 に属する LOB のみが圧縮されます。

```
CREATE TABLE t1 ( REGION VARCHAR2(20), mydata BLOB)
  LOB(mydata) STORE AS SECUREFILE )
PARTITION BY LIST (REGION) (
```

```

PARTITION p1 VALUES ('x', 'y')
      LOB(mydata) STORE AS SECUREFILE (
          COMPRESS ),
PARTITION p2 VALUES (DEFAULT)
);

```

- SECUREFILE LOB の圧縮を無効化します。

```

ALTER TABLE t1 MODIFY
      LOB(mydata) (
          NOCOMPRESS
      );

```

## 暗号化

SecureFiles では、透過的データ暗号化 (TDE) 構文の使用がサポートされます。データベースでは、表内のすべての SecureFile 列の自動的な鍵管理がサポートされ、データおよびバックアップが透過的に暗号化または復号化されます。アプリケーションに変更を加える必要はなく、TDE セマンティックを使用して SecureFiles を活用できます。SecureFiles では、次の暗号化アルゴリズムがサポートされます。

**3DES168** : 168 ビット・サイズの鍵を使用するトリプル・データ暗号化規格

**AES128** : 128 ビット・サイズの鍵を使用する高度暗号化規格

**AES192** : 192 ビット・サイズの鍵を使用する高度暗号化規格 (デフォルト)

**AES256** : 256 ビット・サイズの鍵を使用する高度暗号化規格

ENCRYPT および DECRYPT キーワードは、SecureFile の暗号化を有効化または無効化し、使用する暗号化アルゴリズムをオプションで選択する場合に使用されます。暗号化はブロック・レベルで実行され、指定した列のすべての SecureFiles が暗号化されます。暗号化は相当な CPU を消費するため、SecureFile に対する列レベルおよび表領域レベルの暗号化を同時に有効化してはなりません。この場合、どちらか一方は不要です。

例：

- SECUREFILE LOB 列を含む表を作成し、LOB 暗号化を 1 つのパーティションでのみ有効化します。パーティション p1 に属する LOB のみが暗号化されます。

```

CREATE TABLE t1 ( REGION VARCHAR2(20), mydata BLOB)
      LOB(mydata) STORE AS SECUREFILE (
          CACHE
          ENCRYPT USING 'AES128' )
      PARTITION BY LIST (REGION) (
          PARTITION p1 VALUES ('x', 'y')
          PARTITION p2 VALUES (DEFAULT)
      );

```

- 3DES168 を使用して LOB 暗号化を有効化します。

```

ALTER TABLE t1 MODIFY
      (mydata BLOB ENCRYPT USING '3DES168');

```

## 移行

BasicFile から SecureFile に移行するには、オンライン表再定義を利用することが推奨されます。オンライン再定義を使用して、BasicFile の LOB 列/パーティションを含む表を移行すると、表/パーティションをオフラインにする必要がないというメリットがあります。しかしながら、これには表/パーティションによって使用される領域に等しいか、またはそれをわずかに上回るストレージが追加が必要となります（オンライン表再定義を使用するには、現在使用されている領域の少なくとも 2 倍の領域を確保する必要があります）。移行先の表がパーティション化されている場合、パーティション化をパラレル実行するための通常の方法が適用され、オンライン再定義が並列で実行されます。グローバル索引は再構築する必要があります。

BasicFile LOB 列の移行中は、REDO 領域の生成によって、パフォーマンス上の問題が生じる可能性があります。そのような場合は、移行先の SecureFile 列に NOLOGGING ストレージ・パラメータを設定し、移行が完了したら LOGGING を有効化することが推奨されます。

### 例：

BasicFile CLOB 列'c'を含む表'tab1'をアップグレードします。

**手順 1：**ソース・テーブルのレプリカである仮表を作成し、CLOB 列に SecureFile 属性を指定します。

```
CREATE TABLE tab1_tmp(id number not null, c clob)
  LOB(c) STORE AS SECUREFILE (tablespace tbs_1);
```

**手順 2：**オンライン再定義を実行します。

```
DECLARE
  error_count pls_integer := 0;
BEGIN
  DBMS_REDEFINITION.start_redef_table('scott', 'tab1', 'tab1_tmp', 'id id, c c');
  DBMS_REDEFINITION.copy_table_dependents('scott', 'tab1', 'tab1_tmp', 1, true,
  true, true, false, error_count);
  DBMS_OUTPUT.put_line('errors := ' || to_char(error_count));
  DBMS_REDEFINITION.finish_redef_table('scott', 'tab1', 'tab1_tmp');
end;
/
```

## インタフェース

SecureFiles では、SQL、PL/SQL、OCI、ODBC、JDBC などの標準クライアント・インタフェースがサポートされます。

## 結論

SecureFiles は、構造化、非構造化を問わず、あらゆるエンタープライズ・コンテンツを、シームレス、効率的、かつ高パフォーマンスな単一の Oracle Database で確実に管理できるようにする Oracle Database の機能です。SecureFiles によって、リレーショナル・データや構造化データだけでなく、組織内の'全データ'をデータベース内に格納し、管理することができます。

SecureFiles は、ファイルや他の非構造化データをデータベース外で保管することによるパフォーマンス上の障壁を打ち破るように設計されています。この設計により、普及しているファイル・システムに匹敵するパフォーマンスと簡素性が提供されます。また SecureFiles では、ファイル・システムでは実現できない、複数の高度なデータベース機能を利用できます。

SecureFiles のデータと他のリレーショナル・コンテンツは、一貫性のあるシームレスな方法でまとめて管理できます。Oracle Database のトランザクションに加わることで、原子性、読取り一貫性、リカバリ機能が実現します。

SecureFiles を使用すると、すべての情報は、同じデータベース・ツールを使用して管理できます。さらに、すべてのアプリケーションで、効率的でシームレスな一貫性のある方法で使用できます。

**ORACLE®**

Oracle Corporation, World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口  
電話：+1.650.506.7000  
ファクシミリ：+1.650.506.7200

CONNECT WITH US

 [blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)  
 [facebook.com/oracle](https://facebook.com/oracle)  
 [twitter.com/oracle](https://twitter.com/oracle)  
 [oracle.com](https://oracle.com)

### Hardware and Software, Engineered to Work Together

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0219



Oracle is committed to developing practices and products that help protect the environment