

Oracle® Universal Content Management

File Store Provider インストレーションおよび管理ガイド

10g リリース 3 (10.1.3.3.1)

部品番号 : E05629-01

2007 年 9 月

Oracle Universal Content Management File Store Provider インストールおよび管理ガイド, 10g
リリース 3 (10.1.3.3.1)

部品番号 : E05629-01

原本名 : Oracle Universal Content Management File Store Provider Installation and Administration Guide,
10g Release 3 (10.1.3.3.1)

原本部品番号 : A00041-01

原本協力者 : Bruce Silver, Saroj Aryal

Copyright © 2007 Oracle. 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 USA, Inc., 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目的で使用する際、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。万が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、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation 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、他社の商標の可能性あります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や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の責任を負いかねます。

目次

第1章：はじめに

概要	1-1
このガイドについて	1-2
データ管理	1-2
ファイル管理	1-2
メタデータ管理	1-3
ファイル・ストア	1-3
コンポーネントの機能	1-3
File Store Provider を使用したコンテンツの格納	1-4
Content Server との互換性	1-4
インストール要件	1-4
対象読者	1-5
表記規則	1-5

第2章：インストール

概要	2-1
インストール前の注意事項	2-2
インストール	2-3
コンポーネント・マネージャを使用したコンポーネントの インストール	2-3
Component ウィザードを使用したコンポーネントのインストール ..	2-4
インストールの検証	2-4
リソース表の場所	2-5
File Store Provider のアンインストール	2-5
コンポーネント・マネージャを使用した File Store Provider の アンインストール	2-6
Component ウィザードを使用した File Store Provider の アンインストール	2-7
File Store Provider のロギングの有効化	2-8

第 3 章：構成

概要	3-1
File Store Provider 標準変数の使用	3-2
データベース・オプション	3-2
Content Server のオプション	3-3
追加されるメタデータ・フィールドの構成	3-3
デフォルトのストレージ・ディレクトリの設定	3-4
File Store Provider 標準変数	3-4
File Store Provider の操作	3-5
デフォルト・ファイル・ストアのアップグレード	3-5
パーティションの追加または編集	3-6
File Store Provider の編集	3-6
ストレージ・ルールの追加または編集	3-7
File Store Provider の格納方針の理解	3-8
ストレージ・クラスを決定するための レンディションでのストレージ・ルールの使用	3-8
パス構成と URL 解析の理解	3-10
Content Server のページ	3-12
「Partition Listing」ページ	3-12
「Add/Edit Partition」ページ	3-13
「Edit File Store Provider」ページ	3-16
「Add/Edit Storage Rule」ページ	3-18
パス情報の詳細	3-21
「Provider Listing」ページ	3-22
「Provider Information」ページ	3-23
File Store Provider のリソース表	3-23
PartitionList 表	3-24
StorageRules 表	3-25
PathMetaData 表	3-26
PathConstruction 表	3-27
FileSystemFileStoreAlgorithmFilters 表	3-28
FileStorage 表	3-28
FileCache 表	3-28

第 4 章：実装例

概要	4-1
PathMetaData 表のオプション例	4-2
標準ファイル・パスの構成	4-3
ストレージ・ルールの定義	4-3
パス構成の定義	4-4

Web 表示非対応ストアまたはオプション Web ストアの構成	4-7
ストレージ・ルールの定義	4-7
パス構成の定義	4-7
データベース格納の構成	4-8
ストレージ・ルールの定義	4-8
パス構成の定義	4-9
パス構成とアルゴリズムの変更	4-10
パーティション化の使用	4-10
ディレクトリのファイル数の制限	4-10

付録 A: サード・パーティ・ライセンス

概要	A-1
Apache Software License	A-1
W3C® Software Notice and License	A-2
Zlib License	A-3
一般的な BSD ライセンス	A-4
一般的な MIT ライセンス	A-5
Unicode ライセンス	A-5
その他の帰属	A-7

索引

1

はじめに

概要

バージョン 10g リリース 3 の Content Server では、コンテンツの格納や整理のための従来のファイル・システムにかわって、[データ管理](#)のためにファイル・ストア・システムが実装されます。File Store Provider は、ファイル・ストア機能を Content Server インタフェースで公開します。追加の構成オプションを設定することも可能です。たとえば、ファイル・システムを使用せずに、バイナリ・ラージ・オブジェクト（BLOB）データ型を使用してコンテンツをデータベースに格納するように Content Server を構成できます。これにより次のような利点が得られます。

- ❖ 一貫したバックアップと監視プロセスのために、リポジトリ管理とデータベース管理を統合します。
- ❖ ファイル・システム利用でのディレクトリ構造やディレクトリごとのファイル数に伴う制約を解消できます。
- ❖ システム全体に容易にコンテンツを分散できるため、Content Server の拡張に適しています。
- ❖ 通常はファイル・システムに関連付けられない様々な種類のストレージ・デバイスを使用できます。たとえば、コンテンツ・アドレス・ストレージ・システムや、特定の業務で必要となる書込み専用デバイスです。



警告： File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールまたは無効化を行わないでください。File Store Provider を Content Server の標準インストールでインストールした場合でも、デフォルト・ファイル・ストアをまだアップグレードしていないときは、2-5 ページの「[File Store Provider のアンインストール](#)」の手順に従ってこのコンポーネントをアンインストールすることができます。

この項の内容は、次のとおりです。

- ❖ [このガイドについて](#) (1-2 ページ)
- ❖ [対象読者](#) (1-5 ページ)
- ❖ [表記規則](#) (1-5 ページ)

このガイドについて

この項の内容は、次のとおりです。

- ❖ [データ管理](#) (1-2 ページ)
- ❖ [コンポーネントの機能](#) (1-3 ページ)
- ❖ [Content Server との互換性](#) (1-4 ページ)
- ❖ [インストール要件](#) (1-4 ページ)

データ管理

Content Server は、電子ファイルとそれに関連付けられているメタデータの格納方法を記録することで、コンテンツを管理します。ユーザーがチェックインしたファイル、関連する情報および関連するレンディションを格納して、それらにアクセスできるようにします。ここでは、従来 Content Server で使用されていたデータ管理方法と、File Store Provider での処理方法について説明します。

ファイル管理

データ管理の 2 つの主要機能の 1 つは、Content Server にチェックインされた電子ファイルの格納です。Content Server では、通常、ファイルの格納は従来のファイル・システムを使用して処理され、vault ディレクトリと weblayout ディレクトリを含む階層的なディレクトリ構造に電子ファイルが格納されます。コンテンツ・タイプごとに指定されるリビジョン情報を使用することで、セキュリティ・グループおよびアカウント（使用される場合）、ファイル、関連するレンディションが、vault ディレクトリと weblayout ディレクトリ内の特定のディレクトリに配置されます。たとえば、チェックイン時に指定されたプライマリ・ファイルと代用ファイルは、vault ディレクトリのサブディレクトリに格納されます。特定のファイルの場所は次のように定義されます。

```
<install_dir>/vault/<dDocType>/<account>/<dID>.<dExtension>
```

ここで、*dDocType* はユーザーがチェックイン時に選択するコンテンツ・タイプ、*dID* はこのリビジョンを示す一意のシステム生成 ID、*dExtension* はチェックインされるファイルの拡張子です。この階層モデルでは、システムは *dDocType* メタデータ・フィールドを使用して、vault ディレクトリに作成された階層内にファイルを配置します。同様に、

すべての Web レンディションは <install_dir>/weblayout/groups/ ディレクトリ内の階層に配置されます。Web レンディションは、Web サーバーから利用されるファイルであり、従来のファイル・システム格納方法では、ネイティブ・ファイル、代用ファイル、または Inbound Refinery やその他の変換アプリケーションで生成される Web 表示可能ファイルになります。

ファイル格納場所のこのような簡単な決定方法は、ファイルの場所や操作方法を認識しやすいため、コンポーネントや機能の作成者にとって役立ちます。ただし、ストレージ管理が制約を受ける場合もあります。場所のメタデータを注意深く管理しないと、ディレクトリが一杯になってシステムのパフォーマンスが低下することになります。

メタデータ管理

データ管理のもう 1 つの主要な機能は、電子ファイルに関連付けられているメタデータの格納です。Content Server でのメタデータ管理は、通常はリレーショナル・データベースで行われ、主に 3 つのデータベース表が使用されています。メタデータを使用するとユーザーはコンテンツをカタログ化することができます。また、Content Server 内でのコンテンツ検索を促進するファイル記述子を作成できるようになります。ユーザーに対しては、取得は Content Server で行われ、ファイルが格納される方法や場所は完全に隠されます。ファイルの生成や操作を行う必要があるコンポーネントや機能の作成者に対しては、メタデータによって堅牢なアクセス手段が提供されます。

ファイル・ストア

これまで Content Server で使用されていた従来のファイル・システム・モデルでは、スケーラビリティが制約されます。データ管理のニーズが高まるにつれて、記憶域を増やすためにさらにストレージ・デバイスを追加すると、Web ベース・インタフェースを介した容易なファイル共有が行いにくくなります。ネストされた複雑なファイル構造のためにパフォーマンスが低下することもあります。ネイティブ・ファイル形式が使用される可能性がある場合は、重複した Web 表示可能ファイルの作成を抑制することも困難になります。大規模システム（たとえばコンテンツ・アイテム数が 1 億件以上）を処理するために、Content Server 10g リリース 3 はファイル・ストアの使用に移行しています。ファイル・ストアでは、スケーラビリティ、柔軟性および管理性が向上します。

コンポーネントの機能

このコンポーネントによって FileStoreProvider がインストールされます。FileStoreProvider コンポーネントを使用して、Content Server で管理されるコンテンツの格納とアクセスのためのデータ駆動ルールを定義できます。File Store Provider には次の機能があります。

- ❖ ファイルを簡単に移動する機能
- ❖ 複数のストレージ・デバイス間でファイルをパーティション化する機能

- ❖ Web 表示可能ファイルの作成をオプションにする機能
- ❖ ディレクトリが一杯にならないように管理および制御する機能
- ❖ サード・パーティ・ストレージ・デバイスと統合する機能
- ❖ 異なるストレージ・パラダイムを使用、拡張および強化するための API

File Store Provider を使用したコンテンツの格納

File Store Provider を実装すると、チェックインされたコンテンツとそれに関連するメタデータが調べられ、システム管理者が設定した基準に基づいてストレージ・ルールが割り当てられます。基準には、メタデータ、プロファイル、またはその他の情報が含まれます。ストレージ・ルールによって、ボールド・ファイルと Web ファイルが Content Server で格納される方法や、Web サーバーによってアクセスされる方法が決まります。

Content Server との互換性

File Store Provider は Content Server 10g リリース 3 との互換性があります。以前のバージョンの Content Server はサポートされません。



警告: 本番環境でファイル・ストア・プロバイダを再構成すると、Content Server でファイルの位置を把握できなくなり、見つからないファイルとして報告されることがあります。File Store Provider は、本番環境にインストールする前に開発環境でインストールしてテストする必要があります。

インストール要件

File Store Provider の要件は次のとおりです。

- ❖ Content Server 10g リリース 3 のインスタンス
- ❖ File Store Provider コンポーネント (FileStoreProvider.zip)



警告: File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールまたは無効化を行わないでください。File Store Provider を Content Server の標準インストールでインストールした場合でも、デフォルト・ファイル・ストアをまだアップグレードしていないときは、2-5 ページの「[File Store Provider のアンインストール](#)」の手順に従ってこのコンポーネントをアンインストールすることができます。





対象読者

このガイドは、Content Server のインストールとメンテナンス、およびデータの管理を担当する、Content Server のシステム管理者を対象としています。

表記規則

このガイドでは次の表記規則を使用します。

- ❖ `<Install_Dir>/` という表記は、コンテンツ・サーバー・インスタンスがインストールされているシステム上の場所を参照するために使用されます。
- ❖ スラッシュ (/) は、パス名のディレクトリ・レベルの区切りとして使用されます。ディレクトリ名の末尾には常にスラッシュが付きます。
- ❖ 注意、技術ヒント、重要な通知、および警告には、次の表記規則が使用されます。

記号	説明
	これは注意です。情報に対し、特に注意を喚起するために使用されます。
	これは技術ヒントです。タスクを容易にするために使用できる情報を示すために使用されます。
	これは重要な通知です。必要な手順または必要な情報を示すために使用されます。
	これは警告です。データの損失または重大なシステム問題の原因となる可能性がある情報を示すために使用されます。

はじめに

2

インストール

概要

File Store Provider は、Content Server のインストール時にオプションとしてインストールできます。Content Server のインストール時にインストールしなかった場合は、コンポーネント FileStoreProvider.zip を個別にインストールできます。FileStoreProvider.zip はインストール・メディアに含まれています。



警告: File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールまたは無効化を行わないでください。File Store Provider を Content Server の標準インストールでインストールした場合でも、デフォルト・ファイル・ストアをまだアップグレードしていないときは、2-5 ページの「[File Store Provider のアンインストール](#)」の手順に従ってこのコンポーネントをアンインストールすることができます。

File Store Provider のインストールおよび構成手順の概要は次のとおりです。

1. File Store Provider コンポーネントを[インストール](#)します。
2. Content Server を再起動します。
3. デフォルト・ファイル・ストアを[アップグレード](#)します。
4. デフォルトの[ストレージ・ルール](#)を編集するか、新しいストレージ・ルールを作成します。
5. 必要であれば1つ以上の新しいパーティションを[作成](#)します。

この項の内容は、次のとおりです。

- ❖ [インストール前の注意事項](#) (2-2 ページ)
- ❖ [インストール](#) (2-3 ページ)
- ❖ [インストールの検証](#) (2-4 ページ)
- ❖ [File Store Provider のアンインストール](#) (2-5 ページ)
- ❖ [File Store Provider のロギングの有効化](#) (2-8 ページ)

インストール前の注意事項

Content Server をインストールしても、File Store Provider 機能にアップグレードしていない場合は、デフォルト・ファイル・ストアによって従来の階層的ファイル・システム格納方法がレプリケートされます。このため、コンテンツは、Content Server のインストール時にファイル・システムに作成された vault ディレクトリと weblayout ディレクトリに格納されるように指定されます。File Store Provider のインストールとアップグレードの前に次の注意事項を確認してください。

- ❖ File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールしないでください。このコンポーネントをアンインストールすると、エラーが発生したり、チェックインや検索機能が無効になったりすることがあります。また、Content Server が起動しなくなる場合もあります。
- ❖ File Store Provider コンポーネントは、管理サーバーのコンポーネント・マネージャを使用して無効にすることができます。これは、Content Server で問題を引き起こす可能性があるストレージ・ルールまたはプロバイダの設定を、構成プロセスで修正する場合に役立ちます。
- ❖ File Store Provider は、本番環境に実装する前に必ず開発環境でテストする必要があります。
- ❖ インストール前に、File Store Provider の影響を受けるファイルを必ずバックアップします。たとえば、provider.hda ファイル (`<install_dir>/<instance_dir>/data/providers/defaultfilestore` ディレクトリ)、intradoc.cfg ファイル (`<install_dir>/<instance_dir>/bin/` ディレクトリ)、fsconfig.hda ファイル (`<install_dir>/<instance_dir>/data/filestore/config/` ディレクトリ) などです。
- ❖ デフォルト・ファイル・ストア・プロバイダをアップグレードする前にコンテンツを Content Server にチェックインすると、そのコンテンツの場所を Content Server で認識できなくなることがあります。
- ❖ パーティションを使用するかどうかを検討します。追加のパーティションは必要ありませんが、File Store Provider をインストールして、デフォルトのファイル・ストアをアップグレードした後で、デフォルトのストレージ・ルールのボールド・パスは \$PartitionRoot\$ に設定されます。\$PartitionRoot\$ の値はパーティションを作成しないと生成されません。パーティションを作成する前にコンテンツをチェックインしようとする、ボールド・パス・ルートの変更や、有効な形式のストレージ・ルールの新規作成が失敗します。詳細は、[ストレージ・ルール](#)および[パス構成](#)の項を含む 3-8 ページの「[File Store Provider の格納方針の理解](#)」を参照してください。
- ❖ ストレージ・ルールの定義方法を検討します。ストレージ・ルールは、チェックイン時にコンテンツに割り当てられ、レンディションの格納およびアクセスの方法を決定します。

インストール

次のいずれかの方法を使用して、File Store Provider コンポーネントのインストールと有効化を行うことができます。

❖ [コンポーネント・マネージャを使用したコンポーネントのインストール](#) (2-3 ページ)

❖ [Component ウィザードを使用したコンポーネントのインストール](#) (2-4 ページ)



注意: File Store Provider が機能するためには NativeOsUtils コンポーネントを有効にする必要があります。NativeOsUtils コンポーネントは、Content Server 10g リリース 3 でデフォルトでインストールされるシステム・コンポーネントです。無効にしないでください。



警告: File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールまたは無効化を行わないでください。File Store Provider を Content Server の標準インストールでインストールした場合でも、デフォルト・ファイル・ストアをまだアップグレードしていないときは、2-5 ページの「[File Store Provider のアンインストール](#)」の手順に従ってこのコンポーネントをアンインストールすることができます。

コンポーネント・マネージャを使用したコンポーネントのインストール

コンポーネント・マネージャを使用して File Store Provider コンポーネントをインストールするには、次のようにします。

1. Content Server に管理者としてログインします。
2. 「Administration」メニューから「Admin Server」を選択します。
3. 適用可能な Content Server インスタンスのボタンをクリックします。
4. 「Component Manager」リンクをクリックします。「Component Manager」ページが表示されます。
5. 「Browse」をクリックし、FileStoreProvider.zip ファイルにナビゲートして選択し、「Open」をクリックします。パスが「Install New Component」フィールドに表示されます。
6. 「Install」をクリックします。インストールされるコンポーネント・アイテムのリストが表示されます。
7. 「Continue」をクリックします。コンポーネント・マネージャによって、File Store Provider コンポーネントをすぐに有効にするか、コンポーネント・マネージャに戻るかが確認されます。コンポーネントを有効にするオプションを選択します。
8. Content Server を再起動します。

Component ウィザードを使用したコンポーネントのインストール

Component ウィザードを使用して File Store Provider コンポーネントをインストールするには、次のようにします。

1. Component ウィザードを起動します。「スタート」→「プログラム」→「Oracle File Store Provider」→「<instance>」→「Utilities」→「Component Wizard」を選択する（Windows）か、/bin ディレクトリの *ComponentWizard* スクリプトを実行します（UNIX）。Component ウィザードのメイン画面と「Component List」画面が表示されます。
2. 「Component List」画面で「Install」をクリックします。「Install」画面が表示されます。
3. 「Select」をクリックします。「Zip File Path」画面が表示されます。
4. **FileStoreProvider.zip** ファイルにナビゲートして選択します。
5. 「Open」をクリックします。インストールされる zip ファイルの内容が、「Install」画面のリストに追加されます。
6. 「OK」をクリックします。Component ウィザードによって、File Store Provider コンポーネントを有効化するかどうかを確認されます。
7. 「Yes」をクリックします。File Store Provider コンポーネントが、「Component List」画面で有効になります。
8. Component ウィザードを終了します。
9. Content Server を再起動します。

インストールの検証

File Store Provider のインストールが正常に終了すると、4 つのページ、3 つのメタデータ・フィールド、5 つのリソース表、2 つのデータベース表が Content Server に追加されます。

次のページが Content Server に追加されます。

- ❖ 「[Partition Listing](#)」 ページ (3-12 ページ)
- ❖ 「[Add/Edit Partition](#)」 ページ (3-13 ページ)
- ❖ 「[Edit File Store Provider](#)」 ページ (3-16 ページ)
- ❖ 「[Add/Edit Storage Rule](#)」 ページ (3-18 ページ)

次のメタデータ・フィールドが Content Server に追加されます。

❖ [xPartitionId](#) (3-3 ページ)

❖ [xWebFlag](#) (3-3 ページ)

❖ [xStorageRule](#) (3-4 ページ)

次のリソース表が Content Server に追加されます。

❖ [PartitionList](#) 表 (3-24 ページ)

❖ [StorageRules](#) 表 (3-25 ページ)

❖ [PathMetaData](#) 表 (3-26 ページ)

❖ [PathConstruction](#) 表 (3-27 ページ)

❖ [FileSystemFileStoreAlgorithmFilters](#) 表 (3-28 ページ)

次の表がデータベースに追加されます。

❖ [FileStorage](#) 表 (3-28 ページ)

❖ [FileCache](#) 表 (3-28 ページ)

リソース表の場所

[PathMetaData](#) 表、[PathConstruction](#) 表および [StorageRules](#) 表は、defaultfilestore ディレクトリの provider.hda ファイルで定義されています。defaultfilestore ディレクトリは、<install_dir>/<instance_dir>/data/providers/ ディレクトリにあります。

PartitionList リソース表は、<install_dir>/<instance_dir>/data/filestore/config/ ディレクトリにある fsconfig.hda ファイルで定義されています。パーティションはこの表に追加され、Content Server 内で「[Add/Edit Partition](#)」ページ (3-13 ページ) を使用して構成されます。



警告: リソース・ファイルは直接編集しないでください。リソース・ファイルを適切に変更するには、Content Server ユーザー・インタフェース内で行うか、追加のコンポーネント開発を行う必要があります。コンポーネント開発の詳細は、『Working With Content Server Components』を参照してください。

FILE STORE PROVIDER のアンインストール

次のいずれかの方法を使用して、File Store Provider コンポーネントをアンインストールできます。

❖ [コンポーネント・マネージャを使用した File Store Provider のアンインストール](#) (2-6 ページ)

- ❖ [Component ウィザードを使用した File Store Provider のアンインストール](#)
(2-7 ページ)



警告: デフォルト・ファイル・ストアをアップグレードした後では、File Store Provider のアンインストールまたは無効化を行わないでください。File Store Provider をアンインストールできるのは、デフォルト・ファイル・ストアをアップグレードしていない場合のみです。

コンポーネント・マネージャを使用した File Store Provider のアンインストール

コンポーネント・マネージャを使用して File Store Provider コンポーネントをアンインストールするには、次のようにします。

1. Content Server に管理者としてログインします。
2. 「Administration」メニューから「Admin Server」を選択します。「Content Admin Server」ページが表示されます。
3. コンポーネントをアンインストールする Content Server インスタンスの名前をクリックします。「Content Admin Server <instance_name>」ページが表示されます。
4. 「Component Manager」をクリックします。「Component Manager」ページが表示されます。
5. 「Enabled Components」リストで「FileStoreProvider」を選択します。
6. 「Disable」をクリックします。
7. 「Start/Stop Content Server」をクリックします。「Content Admin Server <instance_name>」ページが表示されます。
8. Content Server を再起動します。
9. 「Component Manager」をクリックします。「Component Manager」ページが表示されます。File Store Provider コンポーネントが「Disabled Components」リストに含まれています。
10. 「Uninstall Component」ドロップダウン・メニューで File Store Provider コンポーネントを選択します。
11. 「Uninstall」をクリックします。コンポーネント・マネージャによって、コンポーネントをアンインストールするかどうかを確認されます。
12. 「OK」をクリックします。コンポーネント・マネージャによって、File Store Provider コンポーネントが正常にアンインストールされたというメッセージが表示されます。
13. リンクを選択してコンポーネント・マネージャに戻ります。「Component Manager」ページが表示されます。

14. 「**Start/Stop Content Server**」をクリックします。「Content Admin Server <instance_name>」ページが表示されます。

15. Content Server を再起動して変更内容を適用します。



注意: コンポーネントをアンインストールすると、そのコンポーネントが Content Server に提供していた機能がなくなりますが、コンポーネント・ファイルはファイル・システムから削除されません。

Component ウィザードを使用した File Store Provider のアンインストール

Component ウィザードを使用して File Store Provider コンポーネントをアンインストールするには、次のようにします。

1. Component ウィザードを起動します。「**スタート**」→「**プログラム**」→「**Oracle File Store Provider**」→「<instance>」→「**Utilities**」→「**Component Wizard**」を選択する (Windows) か、/bin ディレクトリの *ComponentWizard* スクリプトを実行します (UNIX)。Component ウィザードのメイン画面と「Component List」画面が表示されます。
2. 「Component List」画面で File Store Provider コンポーネントを選択し、「**Disable**」をクリックします。
3. Content Server を再起動します。
4. 「Component List」画面で File Store Provider コンポーネントを選択し、「**Uninstall**」をクリックします。Component ウィザードによって、File Store Provider コンポーネントをアンインストールするかどうかを確認されます。
5. 「**Yes**」をクリックします。「Uninstall」画面が表示され、アンインストールされる zip ファイルの内容がリスト表示されます。
6. 「**OK**」をクリックします。File Store Provider コンポーネントが、「Component List」画面から削除されます。
7. Component ウィザードを終了します。
8. Content Server を再起動して変更内容を適用します。



注意: コンポーネントをアンインストールすると、そのコンポーネントが Content Server に提供していた機能がなくなりますが、コンポーネント・ファイルはファイル・システムから削除されません。

FILE STORE PROVIDER のロギングの有効化

filestore トレースは、Content Server の「System Audit Information」ページで使用できます。インストール後に問題が発生した場合は、*filestore* をトレース・セクションに追加してサーバー出力を表示することで、トレースを有効にしてみてください。トレースの詳細は、Content Server のトラブルシューティング・ガイドを参照してください。

3

構成

概要

バージョン 10g リリース 3 の Content Server では、コンテンツの格納や整理のための従来のファイル・システムにかわって、[データ管理](#)のためにファイル・ストア・システムが実装されます。File Store Provider コンポーネントをインストールした後、システム管理者がデフォルト・ファイル・ストアを[アップグレード](#)すると、コンポーネントによって公開される機能を利用できるようになります。デフォルト・ファイル・ストアをアップグレードしたら、ボールドおよび Web URL パス式を変更できます。



注意: File Store Provider をインストールして、デフォルト・ファイル・ストアを[アップグレード](#)した後、デフォルトのストレージ・ルールのボールド・パスは `$PartitionRoot$` に設定されます。`$PartitionRoot$` の値はパーティションを作成しないと生成されません。パーティションは File Store Provider を実行するためには必要ありませんが、パーティションを作成する前にコンテンツをチェックインしようとする、ボールド・パス・ルートの変更や、有効な形式のストレージ・ルールの新規作成が失敗します。詳細は、[ストレージ・ルール](#)および[パス構成](#)の項を含む 3-8 ページの「[File Store Provider の格納方針の理解](#)」を参照してください。



警告: リソース・ファイルは直接編集しないでください。リソース・ファイルを適切に変更するには、Content Server ユーザー・インターフェース内で行うか、追加のコンポーネント開発を行う必要があります。コンポーネント開発の詳細は、『[Working With Content Server Components](#)』を参照してください。

他の 3 つのリソース表は、ファイル・パスの定義と処理に使用されます。[PathMetaData 表](#)と [PathConstruction 表](#)ではデフォルトがほとんどの状況に対応します。[StorageRules 表](#) (3-25 ページ) には、ストレージ・ルールを定義するときに指定した値が格納されます。これら 3 つの表はプロバイダ固有であり、defaultfilestore ディレクトリの provider.hda ファイルで定義されています。defaultfilestore ディレクトリは、

<install_dir>/<instance_dir>/data/providers/ ディレクトリにあります。4 つ目の表である [FileSystemFileStoreAlgorithmFilters](#) 表を変更するには、コンポーネントと Java コードが必要です。

この項の内容は、次のとおりです。

- ❖ [File Store Provider 標準変数の使用](#) (3-2 ページ)
- ❖ [File Store Provider の操作](#) (3-5 ページ)
- ❖ [File Store Provider の格納方針の理解](#) (3-8 ページ)
- ❖ [Content Server のページ](#) (3-12 ページ)
- ❖ [File Store Provider のリソース表](#) (3-23 ページ)

FILE STORE PROVIDER 標準変数の使用

File Store Provider をインストールすると、Content Server データベース、Content Server メタデータ・フィールドおよびその他の構成ファイルに変更が加えられ、構成オプションを設定できるようになります。

この項の内容は、次のとおりです。

- ❖ [データベース・オプション](#) (3-2 ページ)
- ❖ [Content Server のオプション](#) (3-3 ページ)

データベース・オプション

場合によっては、データベースに格納されているコンテンツをファイル・システムに強制的にコピーする必要があります。たとえば、Inbound Refinery が変換のためにファイルにアクセスする必要があるときです。ファイル・システムに強制的にコピーしたファイルは、一時的なキャッシュとみなされます。次の構成値を使用して、一時的にキャッシュしたファイルをいつクリーンアップするかを制御します。エントリが [FileCache](#) 表 (3-28 ページ) にあるファイルのみがシステムによってクリーンアップされることに注意してください。

変数	説明
FsCacheThreshold	最大キャッシュ・サイズを MB 単位で指定します（デフォルトは 100）。しきい値に達すると、Content Server は、FsMinimumFileCacheAge パラメータで指定された最小保存期間を過ぎたファイルの削除を開始します。
FsMaximumFileCacheAge	ファイルが削除されるまでの経過日数です。デフォルトは 365 です。
FsMinimumFileCacheAge	キャッシュされたファイルの削除が可能になるまでの最小保存期間。このパラメータは FsCacheThreshold パラメータとともに使用し、キャッシュされたファイルを削除する時期を決定します。

Content Server のオプション

File Store Provider をインストールすると、いくつかの Content Server メタデータ・フィールドが追加され、構成ファイルで追加のオプションが使用できるようになります。

この項の内容は、次のとおりです。

[追加されるメタデータ・フィールドの構成](#) (3-3 ページ)

[デフォルトのストレージ・ディレクトリの設定](#) (3-4 ページ)

[File Store Provider 標準変数](#) (3-4 ページ)

追加されるメタデータ・フィールドの構成

File Store Provider をインストールすると、次の 3 つのメタデータ・フィールドが Content Server に追加されます。

- ❖ **xPartitionId:** このメタデータ・フィールドを PartitionList 表とともに使用して、コンテンツ・アイテム・ファイルのルート・ロケーションを判別します。このフィールドはユーザー・インタフェースでは非表示にすることをお勧めします。パーティション選択アルゴリズムによって値が提供されるためです。
- ❖ **xWebFlag:** このメタデータ・フィールドを使用して、コンテンツ・アイテムに Web 表示可能ファイルが含まれるかどうかを判別します。結果として、ポールド・ファイルのみのコンテンツ・アイテムがシステムにある場合に、このメタデータ・フィールドを削除すると、システムが Web 表示可能ファイルの存在を予期するため悪影響を受けることがあります。このメタデータ・フィールドは、構成値 WebFlagColumnn によって指定できます。

- ❖ **xStorageRule:** このメタデータ・フィールドを使用して、ファイルの格納方法を決定するルールを記録します。このメタデータ・フィールドは、構成値 `StorageRuleField` によって指定できます。



注意: これらのメタデータ・フィールドは、File Store Provider によって起動時に追加されます。削除しても、Content Server が再起動すると再び追加されます。メタデータ・フィールドを永続的に削除する必要がある場合は、`intradoc.cfg` ファイルで構成変数 `FsAddExtraMetaFields=false` を設定して、フィールドの自動作成を無効にします。`intradoc.cfg` ファイルは `<install_dir>/<instance_dir>/bin/` ディレクトリにあります。

デフォルトのストレージ・ディレクトリの設定

`StorageDir` パラメータは、`PartitionRoot` 列の値が指定されていない場合にすべてのパーティションで使用されるルート・ディレクトリと同じに設定できます。この場合、ストレージ・ディレクトリとパーティション名が `PartitionRoot` パラメータを作成するために使用されます。`StorageDir` パラメータは、`<install_dir>/<instance_dir>/bin` ディレクトリにある `intradoc.cfg` ファイルで設定されます。

File Store Provider 標準変数

`<install_dir>/<instance_dir>/data/providers/defaultfilestore` ディレクトリにある `provider.hda` では、次のパラメータとクラスがファイル・システム・ストアの標準として設定されています。

```
ProviderType=FileStore
ProviderClass=intradoc.filestore.BaseFileStore
IsPrimaryFileStore=true

# Configuration information specific to a file system store provider.
ProviderConfig=intradoc.filestore.filesystem.FileSystemProviderConfig
EventImplementor=intradoc.filestore.filesystem.FileSystemEventImplementor
DescriptorImplementor=intradoc.filestore.filesystem.FileSystemDescriptorImplementor
AccessImplementor=intradoc.filestore.filesystem.FileSystemAccessImplementor
```


FILE STORE PROVIDER の操作

File Store Provider がインストールされてデフォルト・ファイル・ストアがアップグレードされているとき、チェックインされたコンテンツとそれに関連するメタデータが調べられ、システム管理者が設定した基準に基づいてストレージ・ルールが割り当てられます。基準には、メタデータ、プロファイル、またはその他の情報が含まれます。ストレージ・ルールによって、ボールド・ファイルと Web ファイルが Content Server で格納およびアクセスされる方法や、Web サーバーによってアクセスされる方法が決まります。ファイルは、データベースに格納するか、1 つ以上のファイル・システムまたはストレージ・メディアに配置できます。格納場所を管理するためにパーティションを作成することができますが、必須ではありません。

この項の内容は、次のとおりです。

- ❖ [デフォルト・ファイル・ストアのアップグレード](#) (3-5 ページ)
- ❖ [パーティションの追加または編集](#) (3-6 ページ)
- ❖ [File Store Provider の編集](#) (3-6 ページ)
- ❖ [ストレージ・ルールの追加または編集](#) (3-7 ページ)
- ❖ [File Store Provider の格納方針の理解](#) (3-8 ページ)

デフォルト・ファイル・ストアのアップグレード

File Store Provider コンポーネントをインストールした後、システム管理者がデフォルト・ファイル・ストアをアップグレードすると、コンポーネントによって公開される機能を利用できるようになります。デフォルト・ファイル・ストアをアップグレードしたら、ボールドおよび Web URL ファイル・パス式を変更できます。



警告: File Store Provider をインストールしてデフォルト・ファイル・ストアをアップグレードした後では、アンインストールまたは無効化を行わないでください。File Store Provider を Content Server の標準インストールでインストールした場合でも、デフォルト・ファイル・ストアをまだアップグレードしていないときは、2-5 ページの「[File Store Provider のアンインストール](#)」の手順に従ってこのコンポーネントをアンインストールすることができます。

デフォルト・ファイル・ストアをアップグレードするには、次のようにします。

1. Content Server にシステム管理者としてログインします。
2. 「Administration」トレーを開いて「Providers」をクリックします。「[Provider Listing](#)」ページが表示されます。
3. DefaultFileStore プロバイダの横の「Action」列の「Info」をクリックします。「[Provider Information](#)」ページが表示されます。

4. 「Upgrade」をクリックします。「Edit File Store Provider」ページが表示されます。
5. 「Update」をクリックして変更内容を送信します。「Provider Listing」ページが表示されます。



注意: 「Update」をクリックして変更内容を送信する前に、「Edit File Store」ページから別の場所に移動しないでください。更新しないと、アップグレードされたファイル・ストア・プロバイダが有効になりません。

6. Content Server を再起動します。

パーティションの追加または編集

パーティションを作成して、Content Server によって管理されるファイルに追加のルート・パスを定義することができます。ただし、これには別の場所または異なる種類のメディアに記憶域が必要です。パーティションは「Partition Listing」ページ (3-12 ページ) を使用して作成します。新しいパーティションが作成されると、`<install_dir>/<instance_dir>/data/filestore/config/` ディレクトリにある `fsconfig.hda` ファイルの PartitionList リソース表が、File Store Provider によって変更されます。

パーティションを Content Server に追加するには、次のようにします。

1. 「Administration」を開いて「File Store Administration」をクリックします。
2. パーティションが定義されていない場合は、「Add Partition」をクリックします。パーティションが定義されている場合は「Add/Edit Partition」ページが表示されます。
3. パーティション名を入力します。一意の名前を指定する必要があります。
4. パーティション・ルート、複製方法およびその他の関連するパラメータを変更します。詳細は、「Add/Edit Partition」ページ (3-13 ページ) を参照してください。
5. 「Is Active」が有効になっていることを確認します。
6. 「Update」をクリックします。「Partition Listing」ページが表示されます。

File Store Provider の編集

デフォルト・ファイル・ストア・プロバイダはいつでも編集できます。ファイル・ストアを編集するには、次のようにします。

1. Content Server にシステム管理者としてログインします。
2. 「Administration」トレーを開いて「Providers」をクリックします。「Provider Listing」ページが表示されます。
3. DefaultFileStore プロバイダの横の「Action」列の「Info」をクリックします。「Provider Information」ページが表示されます。

4. 「Edit」をクリックします。「Edit File Store Provider」ページが表示されます。
5. 必要な変更を行い、「Update」をクリックして変更内容を送信します。「Provider Listing」ページが表示されます。



注意：「Update」をクリックして変更内容を送信する前に、「Edit File Store」ページから別の場所に移動しないでください。

6. Content Server を再起動します。

ストレージ・ルールの追加または編集

ファイル・ストアには複数のストレージ・ルールを追加できます。



重要：ストレージ・ルールを削除することはできません。ストレージ・ルールを作成する前によく検討してください。



警告：コンテンツが Content Server にチェックインされた後でストレージ・ルールを変更すると、Content Server でコンテンツの場所を認識できなくなる場合があります。

ストレージ・ルールを追加または編集するには、次のようにします。

1. Content Server にシステム管理者としてログインします。
2. 「Administration」トレイを開いて「Providers」をクリックします。「Provider Listing」ページが表示されます。
3. DefaultFileStore プロバイダの横の「Action」列の「Info」をクリックします。「Provider Information」ページが表示されます。
4. 「Edit」をクリックします。「Edit File Store Provider」ページが表示されます。
5. 「Add new rule」を選択します。または、編集するルールの名前を「Storage Rules」選択リストから選択して、「Edit rule」をクリックします。「Add/Edit Storage Rule」ページが表示されます。
6. 必要に応じてストレージ・ルールを変更し、「OK」をクリックします。「Edit File Store Provider」ページが表示されます。
7. 「Update」をクリックします。「Provider Listing」ページが表示されます。



重要：ストレージ・ルールで定義された Web URL ファイル・パスで使用する Web ルートが、Content Server に対して定義されたデフォルトの weblayout ディレクトリでない場合、ストレージ・ルールで使用する Web ルートについて、Web サーバーにエイリアスまたは仮想ディレクトリを追加する必要があります。このようにしないと、Content Server がファイルにアクセスする場所を認識できません。Web サーバーに仮想ディレクトリを追加する方法の詳細は、Web サーバーのドキュメントを参照してください。

FILE STORE PROVIDER の格納方針の理解

Content Server にチェックインするとき、コンテンツ・アイテムは、メタデータと、ユーザーが選択したプライマリ・ファイルで構成されます。代替ファイルが含まれる場合もあります。代替ファイルは、ユーザーが選択してチェックインすることもでき、Web 表示可能ファイルとみなされます。Content Server のファイル・システムでは、プライマリ・ファイルは `<install_dir>/<instance_dir>/` のルートの vault ディレクトリに格納され、ネイティブ・ファイルと呼ばれます。代替ファイルも、チェックイン時には vault に格納されますが、weblayout ディレクトリにコピーされるか、変換アプリケーション (Inbound Refinery など) に渡されます。代替ファイルをチェックインしない場合は、ネイティブ・ファイルが vault ディレクトリから weblayout ディレクトリにコピーされて、2 箇所が存在します。Inbound Refinery がインストールされているときに代替ファイルをチェックインしない場合は、ネイティブ・ファイルのレンディションが作成されて weblayout ディレクトリに格納されます。

Content Server のファイル・システムでは、指定されたディレクトリにコンテンツを格納することで、コンテンツのパスが定義されます。コンテンツに Web ブラウザからアクセスするには、静的な Web URL ファイル・パス (コンテンツが特定の場所にあることがわかっている場合) を使用するか、GET_FILE などの動的な Content Server サービス・リクエスト (コンテンツの場所がわからない場合) を使用します。File Store Provider がインストールされているとき、コンテンツがファイル・システムに格納される場合とされない場合があります。このため、コンテンツのパスを定義するために新しい方法を利用する必要があります。

File Store Provider の設定方法に応じて、静的な Web URL が得られる場合と得られない場合があります。特定の場所がわからない場合は、Content Server の動的サービス・リクエストを使用することで、コンテンツにアクセスできます。File Store Provider では、静的な Web URL は Web URL ファイルとして定義され、動的アクセスは単に Web URL と呼ばれます。File Store Provider ユーザー・インタフェースでは、静的な Web URL ファイル・パスしか構成できません。ただし、静的な Web URL を Content Server のサーバー・リクエストとして処理し、実質的な動的アクセスとすることができます。

この項の内容は、次のとおりです。

- ❖ [ストレージ・クラスを決定するための レンディションでのストレージ・ルールの使用](#) (3-8 ページ)
- ❖ [パス構成と URL 解析の理解](#) (3-10 ページ)

ストレージ・クラスを決定するための レンディションでのストレージ・ルールの使用

コンテンツをチェックインする際には、Content Server によって管理されるコンテンツのすべてのバージョンはレンディションとみなされます。このようなレンディションに

は、ネイティブ・ファイルや Web 表示可能ファイル、および Inbound Refinery または サード・パーティの変換アプリケーションでレンダリングされるその他のファイルが含まれます。

レンディションはストレージ・クラスにグループ分けされます。ストレージ・クラスによって、レンディションのアクセス場所とアクセス方法が決まります。ストレージ・クラスは、[ストレージ・ルール](#)にグループ分けされます。ストレージ・ルールでは、ストレージ・クラスを介して [ボルト](#)、[Web](#) および [Web URL](#) パス式が定義されます。また、ストレージ・ルールによって、レンディションを格納しない ([web 表示非対応](#)ファイル・ストアなど) か、またはレンディションをファイル・システム以外の別のデバイスに格納する ([データベース](#)など) かも決まります。

次の例では、ストレージ・ルールによって、様々なコンテンツ・アイテムの格納場所と格納方法がどのように決まるかを示します。

例：

ストレージ・ルールは、[「Add/Edit Storage Rule」ページ](#) (3-18 ページ) で「File system only」として定義されています。「Is Webless File Store」は無効です。この場合、システムはプライマリ・ファイルのコピーを作成して、[weblayout](#) ディレクトリに配置します。

通常、このような従来のファイル・システム格納の例では、データベース格納よりもアクセス速度が速いという利点が得られます。ファイル・システム階層が複雑な場合、またはファイル・システムが一杯になった場合、あるいはコンテンツ・アイテムの量が増加するにつれて、利点は小さくなります。

例：

ストレージ・ルールは、[「Add/Edit Storage Rule」ページ](#) (3-18 ページ) で「File system only」として定義されています。「Is Webless File Store」は有効です。この場合、プライマリ・ファイルのコピーは作成されません。レンディションはネイティブ・ファイルのみです。Web 表示可能ファイルへのリクエストは、[vault](#) に格納されているネイティブ・ファイルにルーティングされます。

このような従来のファイル・システム格納の例には、前の例と同じく、コンテンツへのアクセス速度が速いという利点があります。また、[vault](#) ディレクトリから [weblayout](#) ディレクトリにコンテンツのバージョンをコピーしないため、記憶領域も節約できます。かわりに、Web 表示可能ファイルへのアクセスを [vault](#) ディレクトリのコンテンツにリダイレクトします。チェックインされているネイティブ・ファイルのほとんどが Web 表示可能形式の場合、またはブラウザで表示する必要のないコンテンツを管理するために [Content Server](#) を使用している場合には、この方法が役立ちます。

例：

ストレージ・ルールは、「[Add/Edit Storage Rule](#)」ページ (3-18 ページ) で「JDBC Storage」として定義されています。「Renditions」選択リストでは何も選択されていません。この場合、ボールド・ファイルと Web ファイルはどちらもデータベースに格納されます。

このデータベース格納の例には、リポジトリ管理がデータベース管理と統合される利点があり、一貫したバックアップと監視プロセスが実現します。ファイル・システム利用での、ディレクトリ構造やディレクトリごとのファイル数に伴う制約を解消できます。



重要：索引作成や変換の際など、必要に応じて、データベースに格納されるコンテンツ・アイテムをファイル・システムに強制的にコピーすることができます。ファイル・システム上のファイルは、`<install_dir>/<instance_dir>/config` ディレクトリにある `config.cfg` ファイルに指定されたパラメータに従い、一時的なキャッシュとして処理されて削除されます。使用されるパラメータの詳細は、[FileCache 表](#) (3-28 ページ) を参照してください。

例：

ストレージ・ルールは、「[Add/Edit Storage Rule](#)」ページ (3-18 ページ) で「JDBC Storage」として定義されています。「Renditions」選択リストでは「Web Files」が選択されています。この場合、ボールド・ファイルはデータベースに格納され、Web ファイルはファイル・システムに永続的に格納されます。

このようにネイティブ・ファイルをデータベースに格納し、Web 表示可能ファイルをファイル・システムに格納する混在方法では、ネイティブ・ファイルに対しては前の例のデータベースの利点（バックアップと監視が統合され、ファイル・システムの制約が解消される）が得られ、Web 表示可能レンディションに対しては高速 Web アクセスが得られます。最初の例と同じく、ファイル・システム構造が複雑な場合やファイルの数量が非常に多い場合にはこの利点は小さくなります。

パス構成と URL 解析の理解

Content Server に格納されるコンテンツのパスは、[PathConstruction 表](#)の PathExpression 列で定義されます。パスを構成する各要素はスラッシュ (/) で区切られています。各要素は、静的な文字列または動的パーツの組合せで構成されます。動的パーツは \$ で囲まれています。パーツは、アルゴリズム、IdocScript 変数、環境変数またはメタデータ参照を介して計算され、次のように解釈されます。

- ❖ PathMetaData 表で定義されたフィールドの場合があります。PathMetaData 表に定義されている場合は、アルゴリズムにマップできます。たとえば、`$dDocType$` です。
- ❖ 接頭辞 `#env.` が付いている場合は環境変数です。たとえば、`$#env.VaultDir$` です。
- ❖ IdocScript 変数の場合もあります。たとえば `$HttpWebRoot$` です。

たとえば、標準のボールドの場所は次のように定義されます。

```
$PartitionRoot$/vault/$dDocType$/dDocAccount$/dID$$ExtensionSeparator$$dExtension$
```

このパス式を解析すると 5 つの要素に分かれ、PathMetaData 表に指定されたルールに従って次のように解釈されます。

- ❖ **\$PartitionRoot\$**: partitionSelection アルゴリズムにマップされ、PartitionList 表を参照するために xPartitionId を使用してパーティション・ルートを判別します。
- ❖ **/vault/**: 文字列です。計算や置換は行われません。
- ❖ **\$dDocType\$**: PathMetaData 表によって定義されます。ファイル・パラメータ内の参照です。
- ❖ **\$dDocAccount\$**: documentAccount アルゴリズムにマップされます。このアルゴリズムは、dDocAccount を受け取り、適切なすべてのデリミタを含む Content Server の標準アカウント表現に解析します。
- ❖ **\$dID\$\$ExtensionSeparator\$\$dExtension\$**: この要素には 3 つのパーツが含まれます。
 - **\$dID\$**: dDocType と同様に、ファイル・パラメータに定義されている必須フィールドです。
 - **\$ExtensionSeparator\$**: アルゴリズムによって判別されます。デフォルトではピリオド (.) を返します。
 - **\$dExtension\$**: dDocType と同様です。

標準構成の URL には、dDocName と拡張子だけでなくセキュリティと dDocType の情報も含まれます。URL および Web の場所は次のように構成されます。

```
.../groups/$dSecurityGroup$/dDocAccount$/documents/$dDocType$/  
dDocName$.dWebExtension$
```

groups セパレータは、その後のディレクトリが、コンテンツ・アイテムが属するセキュリティ・グループとアカウントの名前であることを Content Server に知らせます。アカウントはオプションであり、アルゴリズムによって計算されます。セキュリティ情報の後には documents セパレータがあります。この直後に dDocType が指定されます。URL の最後の部分は、dDocName とその形式を示す拡張子です。

URL はこの形式であると予想されているため、Content Server によってメタデータが正常に抽出されます。さらに重要なのは、コンテンツ・アイテムのセキュリティ情報を判別できることと、特定のユーザーのアクセス権限を導出できることです。

解析のガイドラインが拡張され、Web ディレクトリの分散が許可されています。groups セパレータはそのまま使用されますが、documents セパレータは sg で置き換えることができます。解析で sg セパレータが検出された場合、URL の残りの部分が

/sg/\$dDocName\$.dWebExtension\$ であるという想定は行われません。かわりに、解析では分散の終了マークである d が検索されます。d が検出されると、以前と同じく、その後の情報に dDocName と dWebExtension が含まれていると想定されます。現在では、次のフォームの URL もシステムで正しく解析することができます。

```
.../groups/$dSecurityGroup$/dDocAccount$/sg/<dispersion>/<dispersion>.../d/  
dDocName$.dWebExtension$
```


CONTENT SERVER のページ

File Store Provider をインストールして有効にすると、次のページが Content Server に追加されます。

- ❖ 「[Partition Listing](#)」 ページ (3-12 ページ)
- ❖ 「[Add/Edit Partition](#)」 ページ (3-13 ページ)
- ❖ 「[Edit File Store Provider](#)」 ページ (3-16 ページ)
- ❖ 「[Add/Edit Storage Rule](#)」 ページ (3-18 ページ)

「Partition Listing」 ページ

Partition Name	Partition Root	Is Active	Action
myPartition	\$#env.StorageDir:	TRUE	<div> Edit Delete </div>

Add Partition

「Partition Listing」 ページには、すべての現行パーティションのリストが表示され、ルートとステータスが示されます。「Partition Listing」 ページにアクセスするには、コンテンツのトレイ領域で「Administration」トレイを開き、「File Store Provider Administration」をクリックします。リストの各パーティションの要素は、「[Add/Edit Partition](#)」 ページ (3-13 ページ) を使用して変更できます。値は、`<install_dir>/<instance_dir>/data/filestore/config/` ディレクトリにある `fsconfig.hda` ファイルの PartitionList リソース表に格納されます。

要素	説明
Partition Name	「 Add/Edit Partition 」 ページ (3-13 ページ) を使用してパーティションの作成時に定義したパーティションの名前を表示します。パーティション名は、コンテンツを格納するときに Content Server で使用されるパス式に含まれます。

要素	説明
Partition Root	このパーティションのコンテンツが格納されているルート・レベルを表示します。これは、 Content Server がコンテンツの格納場所の選択に使用するアルゴリズムに渡される引数の 1 つです。値は、静的文字列 (C:/vault など)、式 (\$#env.VauldDir\$ など)、または Idoc Script 変数 (\$HttpWebRoot\$ など) になります。
Is Active	パーティションがアクティブ (TRUE) かアクティブでない (FALSE) かを表示します。アクティブなパーティションにはコンテンツを格納できます。
Action	各パーティションに対するアイテムの処理メニューを表示します。このメニューを使用して、パーティションの編集または削除を選択できます。
Add Partition	「Add Partition」をクリックすると、「 Add/Edit Partition ページ (3-13 ページ) が表示されます。このページを使用して、 Content Server で使用される新しいパーティションの追加やアクティブ化を行うことができます。

「Add/Edit Partition」 ページ

Partition Name:

Partition Root:

Capacity Check Interval:

Slack Bytes:

Duplication Methods:

Is Active: ☒

「Add/Edit Partition」 ページを使用して、**Content Server** がコンテンツを格納するパーティションを作成または変更します。このページにアクセスするには、「[Partition Listing](#) ページ (3-12 ページ) で「Add Partition」をクリックします。このページに入力する値は、<install_dir>/<instance_dir>/data/filestore/config/ ディレクトリにある fsconfig.hda ファイルの [PartitionList](#) 表に格納されます。

要素	説明
Partition Name	パーティションの一意の名前を定義します。パーティション名は「 Partition Listing ページ (3-12 ページ)」に表示されます。また、Content Server がコンテンツの格納に使用するパス式に含まれます。このため、パーティション名は作成する各パーティションで一意であることが必要です。Idoc Script や HTML のパス式と同じ文字の制約があります。
Partition Root	このパーティションでコンテンツが格納されているパスのルート・レベルを定義します。これは、Content Server がコンテンツの格納場所の選択に使用するアルゴリズムに渡される引数の 1 つです。
Capacity Check Interval	このパーティションで使用可能なディスク領域を確認する間隔を指定します。秒単位で指定します。プラットフォームによっては、この引数を使用できない場合があります。
Slack Bytes	パーティションが一杯でコンテンツを受け入れられなくなる状態を指定します。パーティションの使用可能領域が、指定する遊びバイト数を下回る場合、そのパーティションは新しいコンテンツを受け入れません。
Duplication Methods	<p>ネイティブ・ファイルが Web 表示可能レンディションに変換されない場合の、ネイティブ・ファイルの処理方法を指定します。たとえば、多くのイメージ・ファイルでは、Web 表示可能なレンディションは必要ありません。ネイティブ・ファイルを Web パスにコピーするかわりに、ネイティブ・ファイルをリンクすることで、記憶領域の管理が容易になります。</p> <ul style="list-style-type: none"> • copy (デフォルト) : ネイティブ・ファイルを Web パスにコピーします。 • link: Web パスを vault のネイティブ・ファイルに解決します。
Is Active	パーティションがアクティブで新しいコンテンツに対して使用可能かどうかを指定します。

要素	説明
Update	指定した情報を送信して、パーティションを作成または更新します。
Reset	パーティションを更新する前の状態に情報をリセットします。

「Edit File Store Provider」 ページ

Edit File Store Provider

Provider Name
DefaultFileStore

Provider Description
Default File Store Provider

Provider Class
intradoc.filestore.BaseFileStore

Connection Class

Configuration Class
intradoc.filestore.filesystem.FileSystemProviderConfig

Access Implementor
intradoc.filestore.filesystem.FileSystemAccessImplementor

Descriptor Implementor
intradoc.filestore.filesystem.FileSystemDescriptorImplementor

Event Implementor
intradoc.filestore.filesystem.FileSystemEventImplementor

Metadata Implementor
intradoc.filestore.filesystem.FileSystemMetadataImplementor

Storage Rules
default
Edit rule...

Update
Reset

「Edit File Store Provider」 ページは、既存のプロバイダを変更するために使用されます。このページにアクセスするには、「[Provider Information](#)」 ページ (3-23 ページ) で「Edit」をクリックします。このページに入力する情報は、`<install_dir>/<instance_dir>/data/providers/defaultfilestore` ディレクトリにある `provider.hda` ファイルに格納されます。デフォルト値でほとんどの格納の組合せに対応できます。プロバイダの詳細は、Content Server のドキュメントに含まれる『Managing System Settings and Processes』を参照してください。

要素	説明
Provider Name	プロバイダの名前を定義します。
Provider Description	「 Provider Listing 」 ページ (3-22 ページ) に表示される、プロバイダを識別する説明です。
Provider Class	プロバイダの機能を制御する Java クラス・ファイルのパス。デフォルトのクラス・ファイルは <code>BaseFileStore</code> です。

要素	説明
Connection Class	これは、File Store Provider に適用できない Java クラス・ファイルのパスです。値を入力しないでください。
Configuration Class	ファイル・ストア・プロバイダの機能を構成するために使用される Java クラス・ファイルのパス。
Access Implementor	コンテンツにアクセスするためにコールされる Java クラス・ファイルのパス。
Descriptor Implementor	コンテンツを説明するときにコールされる Java クラス・ファイルのパス。
Event Implementor	イベント（索引作成または検索など）を実装するときにコールされる Java クラス・ファイルのパス。
Metadata Implementor	コンテンツの情報が必要なときにコールされる Java クラス・ファイルのパス。
Storage Rules	プロバイダに対して使用されるストレージ・ルールを表示します。編集するルールを選択するか、「Add rule」を選択して追加のルールを作成します。
Edit Rule	ストレージ・ルールの追加または変更のために「 Add/Edit Storage Rule 」ページ（3-18 ページ）にアクセスします。

「Add/Edit Storage Rule」 ページ

Storage Rule Name default

☒ **File system only**
Files will be stored on the file system only. The location of the files is specified by the storage path information below. Files are stored in both the vault and weblayout directories unless the 'Is Webless' option is checked. The webless filestore will apply only to files that require conversion and is tied to the metadata field xWebFlag.

☐ **JDBC Storage**
Files will be stored in the database unless the system has been configured to store specified renditions on the file system. Use the options below to force renditions to be stored on the filesystem and not in the database.

Renditions

☐ **Is Webless File Store**

Path Information [Show Path Metadata](#)

The following rules are applied to web locations where there is expectations of parsing the relative URL for security and content ID information. By default, the URL may have its security attributes preceded by the 'groups' specifier and completed by the 'documents' specifier. After documents, the parser assumes that the next directory is the Content Type i.e. dDocType, which is then followed by the dDocName plus extension. However, the parser also recognizes the 'sg' directory as the beginning of the dispersion directory. Once 'sg' is encountered as a segment in the URL path, the parser searches for the 'd' segment. If this segment is found, it proceeds to determine the dDocName plus extension information.

Vault Path

Web-viewable Path

Web URL File Path

「Add/Edit Storage Rule」 ページを使用して、Content Server にチェックインされたコンテンツを各プロバイダが格納する方法と場所を指定します。このページでは、コンテンツがファイル・システムとデータベースのどちらに格納されるか、Web レンディションが作成されるか、およびコンテンツのパスをどのように構成するかを定義します。このページにアクセスするには、「[Edit File Store Provider](#)」 ページ (3-16 ページ) で「Storage Rules」 選択リストの横にある「Edit」 をクリックします。

要素	説明
File system only	Content Server にチェックインされるコンテンツが、データベースではなく指定されたファイル・システムのみで格納されるように指定します。「Is Webless File Store」オプションが有効になっていない場合、これにはネイティブ・ファイルと Web 表示可能ファイルの両方が含まれます。
Is Webless File Store	Web 表示可能レンディション・コンテンツを作成しないことを指定します。
JDBC Storage	Content Server にチェックインされるコンテンツが、ファイル・システムではなくデータベースのみに格納されるように指定します。「Renditions」選択リストでオプションが選択されていない場合、これにはネイティブ・ファイルと Web 表示可能ファイルの両方が含まれます。
Renditions	<p>「JDBC Storage」が有効になっているときに、レンディションがファイル・システムに格納されるように指定します。</p> <ul style="list-style-type: none"> 選択なし（デフォルト）：ネイティブ・ファイルと Web 表示可能ファイルの両方がデータベースに格納されます。 Web Files: Web 表示可能ファイルがファイル・システムに、ネイティブ・ファイルがデータベースに格納されます。 Vault Files: ネイティブ・ファイルがファイル・システムに、Web 表示可能ファイルがデータベースに格納されます。
Show Path Metadata	プロバイダが使用するパスの構成に使用されるメタデータの 詳しい情報 を表示します。
Vault Path	このルールを使用するプロバイダ用に、Content Server にチェックインされるネイティブ・コンテンツが格納される vault の場所へのパスを定義する式です。
Web-viewable Path	Web 表示可能レンディションのファイル・システム上のパスを定義する式です。

要素	説明
Web URL File Path	ブラウザで Web 表示可能レンディションにアクセスするために使用される URL です。



重要: ストレージ・ルールで定義された Web URL ファイル・パスで使用する Web ルートが、Content Server に対して定義されたデフォルトの weblayout ディレクトリでない場合、ストレージ・ルールで使用する Web ルートについて、Web サーバーにエイリアスまたは仮想ディレクトリを追加する必要があります。このようにしないと、Content Server がファイルにアクセスする場所を認識できません。Web サーバーに仮想ディレクトリを追加する方法の詳細は、Web サーバーのドキュメントを参照してください。

パス情報の詳細

Path Information Hide Details		
The table below lists the current configuration of the path metadata that can be used in path construction. Fields can be tied to algorithms and require the existence of content metadata. The field is computed when it is referenced as a component in the path construction expression.		
Field Name	Description	Generation Algorithm
dID	Standard content id field	
dDocName	Standard content name field	
dDocAccount	Standard account field	documentAccount
dDocType	Standard content type field	
dExtension	Native file extension	
dWebExtension	Web file extension	
dSecurityGroup	Security group	
dRevisionID	Standard revision id field	
dReleaseState	Release state	
dStatus	Status	
PartitionRoot	Computed partition root	partitionSelection
ExtensionSeparator	Computed extension separator usually '.'	extensionSeparator
xWebFlag	Flag used to determine existence of a web file	
RenditionId	Rendition specifier	
RevisionLabel	Computed revision label	revisionLabel3
RenditionSpecifier	Additional rendition specifier	renditionSpecifier
RenditionPrefix	Additional rendition prefix computation	renditionPrefix

プロバイダが使用するパスの構成に使用されるメタデータの詳しい情報は、「[Add/Edit Storage Rule](#)」 [ページ](#) (3-18 ページ) の「Show Path Metadata」をクリックすると表示できます。

「Provider Listing」 ページ

Providers					
Provider	Description	Type	Connection State	Last Activity Date	Action
SystemDatabase	System I	database	15 out of 15 conn	3/11/07 5:32 PM	Info Test
SystemServerSocket	System :	incoming	good	3/11/07 5:32 PM	Info Test
refinery_10gR3	Connect	outgoing	good	3/11/07 5:32 PM	Info Test
DefaultFileStore	Default I	FileStore	good		Info Test

Create a New Provider		
Provider Type	Description	Action
outgoing	Configuring an outgoing provider.	Add
database	Configuring a database provider.	Add
incoming	Configuring an incoming provider.	Add
preview	Configuring a preview provider.	Add
ldapuser	Configuring an LDAP user provider.	Add

File Store Provider がインストールされている場合、Content Server の「Provider Listing」 ページを変更すると、ファイル・ストア・プロバイダの作成や変更が可能になります。既存のファイル・ストア・プロバイダの横にある「Info」をクリックすると、「[Provider Information](#)」 ページ（3-23 ページ）が表示されます。新しいファイル・ストア・プロバイダを作成するには、FileStore プロバイダ・タイプの「Action」列の「Add」をクリックします。「[Edit File Store Provider](#)」 ページ（3-16 ページ）が表示されます。

「Provider Information」 ページ

File Store Provider Information for 'DefaultFileStore'

Provider Name:	DefaultFileStore
Provider Description:	Default File Store Provider
Connection State:	good
Last Activity Date:	None

Provider Type:	FileStore
Provider Class:	intradoc.filestore.BaseFileStore
Provider Connection:	

ファイル・ストア・プロバイダの「Provider Information」ページには、選択したプロバイダの情報（接続状態や最新アクティビティ日付、プロバイダのタイプ、クラスおよび接続など）が表示されます。「Provider Information」ページの「Edit」をクリックすると「[Edit File Store Provider](#)」ページ（3-16 ページ）が表示されます。このページではプロバイダの詳細を変更できます。

FILE STORE PROVIDER のリソース表

この項の内容は、次のとおりです。

- ❖ [PartitionList 表](#)（3-24 ページ）
- ❖ [StorageRules 表](#)（3-25 ページ）
- ❖ [PathMetaData 表](#)（3-26 ページ）
- ❖ [PathConstruction 表](#)（3-27 ページ）
- ❖ [FileSystemFileStoreAlgorithmFilters 表](#)（3-28 ページ）
- ❖ [FileStorage 表](#)（3-28 ページ）
- ❖ [FileCache 表](#)（3-28 ページ）

PartitionList 表

PartitionList 表では、partitionSelection アルゴリズムで使用可能なパーティションが定義されます。この表は、<install_dir>/<instance_dir>/data/filestore/config/ ディレクトリにある fsconfig.hda ファイルで定義されています。Content Server ユーザー・インターフェースの「[Add/Edit Partition](#)」ページを使用して変更できます。表の列は次のように使用されます。

列	説明
PartitionName	パーティションの名前を指定します。この名前はパス式で参照されます。
PartitionRoot	partitionSelection アルゴリズムに渡される引数。
IsActive	パーティションが現在アクティブで新しいファイルを受け入れるかどうかを判別します。
CapacityCheckInterval	使用可能なディスク領域を確認する間隔（秒）を指定します。これはプラットフォームによっては使用できません。
SlackBytes	コンテンツの格納に十分な領域がパーティションにあるかどうかを判別します。使用可能な領域が遊びバイトよりも少ない場合、パーティションは非アクティブ化され、投稿に使用されなくなります。
DuplicationMethods	<p>ネイティブ・ファイルが Web 表示可能レンディションに変換されない場合の、ネイティブ・ファイルの処理方法を指定します。</p> <ul style="list-style-type: none"> • copy（デフォルト）: ネイティブ・ファイルを Web パスにコピーします。 • link: Web パスを vault のネイティブ・ファイルに解決します。 <p>コピーとリンクは、Content Server がインストールされているオペレーティング・システムの機能を利用しています。このため、プラットフォームによっては使用できない方法があります。</p>

StorageRules 表

StorageRules 表では、コンテンツ・アイテムの格納に使用するルールが定義されます。ルールによって、使用されるパス式、対象となるストレージ・クラス、コンテンツ・アイテムの格納方法が指定されます。

この表は、<install_dir>/<instance_dir>/data/providers/defaultfilestore/ ディレクトリにある provider.hda ファイルで定義されています。Content Server ユーザー・インタフェースの「[Add/Edit Storage Rule](#)」ページを使用して変更できます。表の列は次のように使用されます。

列	説明
StorageRule	ストレージ・ルールの名前。動的インクルードから計算され、コンテンツ・アイテムの <i>xStorageRule</i> メタデータ・フィールドに格納されます。
StorageType	記憶域の実装を決定します。 <ul style="list-style-type: none"> • FileStorage: ファイルがファイル・システムに格納されます。 • JdbcStorage: ファイルがデータベースに格納されます。
IsWeblessStore	システムで Web 表示非対応ファイルが許可されるかどうかを指定するために使用されます。 <ul style="list-style-type: none"> • true: デフォルトでは、新たに作成されるコンテンツ・アイテムに Web 表示可能ファイルは含まれません。特定の状況では、Web 表示可能ファイルが必要になります。そのような場合は、コール・コードの引数を使用して、Web 表示可能ファイルを作成する必要があることを指定できます。Web 表示可能ファイルの有無に関する情報は、<i>xWebFlag</i> メタデータ・フィールドに格納されます。 • false: デフォルトでは、新たに作成されるコンテンツ・アイテムに Web 表示可能ファイルが含まれます。
RenditionsOnFileSystem	データベースではなくファイル・システムに格納されるファイルがあるかどうかを判別するために、JdbcStorage によって使用されます。

PathMetaData 表

PathMetaData 表では、ファイルの場所を判別するために使用されるメタデータが定義されます。コンテンツ・アイテムのメタデータを直接使用することも、アルゴリズムを使用してメタデータを計算することもできます。PathMetaData 表は、defaultfilestore ディレクトリの provider.hda ファイルで定義されています。defaultfilestore ディレクトリは、<install_dir>/<instance_dir>/data/providers/ ディレクトリにあります。

表の列は次のように使用されます。

列	説明
FieldName	パス式に表示されるフィールドの名前。
GenerationAlgorithm	フィールドの値を解決または計算するために使用されるアルゴリズムを指定します。
RequiredForStorage	<p>メタデータが必要なストレージ・クラスを定義します。</p> <ul style="list-style-type: none"> • #all: ボールト・レンディションと Web 表示可能レンディションの両方でメタデータが必要です。 • web: Web 表示可能レンディションのみでメタデータが必要です。 • vault: ネイティブ・ファイル・レンディションのみでメタデータが必要です。 <p>指定されていないすべてのレンディションについては、このフィールドはオプションです。結果として、この列が空の場合、このメタデータ・フィールドはすべてのレンディションまたはストレージ・クラスでオプションになります。アルゴリズムが指定されている場合、この値は空になります。アルゴリズムは、ArgumentFields 列に指定された値を使用して、どのフィールドが必要かを決定します。</p>
Arguments	GenerationAlgorithm フィールドに指定されたアルゴリズムに渡されるオプション引数。
ArgumentFields	Arguments 列に定義された引数で必要なフィールドのカンマ区切りリスト。これらのフィールドは、GenerationAlgorithm フィールドに指定されたアルゴリズムでも必要です。

PathConstruction 表

PathConstruction 表では、ファイルをパスにマップします。PathConstruction 表は、defaultfilestore ディレクトリの provider.hda ファイルで定義されています。defaultfilestore ディレクトリは、<install_dir>/<instance_dir>/data/providers/ ディレクトリにあります。詳細は、3-10 ページの「パス構成と URL 解析の理解」も参照してください。



警告: PathConstruction 表で提供されるデフォルトは、ほとんどの状況に対応できます。このリソース・ファイルは直接編集しないでください。適切に変更するには、追加のコンポーネント開発を行う必要があります。コンポーネント開発の詳細は、『Working With Content Server Components』を参照してください。

PathConstruction 表の列は次のように定義されます。

列	説明
FileStore	計算される記憶域のパスを指定します。 <ul style="list-style-type: none"> • web: Web 表示可能ファイルのパス。 • vault: ネイティブ・ファイルのパス。 • weburl: Content Server によって生成されます。多くの場合、GET_FILE が使用されます。 • weburl.file: ブラウザで Web 表示可能レンディションにアクセスするための正しく構成された URL。
PathExpression	パスを定義します。
AutoCreateLimit	作成できるディレクトリの深さを指定します。
StorageRule	このパス構成が属するストレージ・ルールを指定します。

FileSystemFileStoreAlgorithmFilters 表

FileSystemFileStoreAlgorithmFilters 表は、アルゴリズム名を FilterImplementor インタフェースの実装にマップするために使用されます。このアルゴリズムは [PathMetaData 表](#) で参照でき、必要なパス・フィールドを計算するために使用されます。このアルゴリズムを実装するクラスは、ファイル・パラメータ・オブジェクトが null の場合、計算に使用する必須のメタデータ・フィールドを返す必要があります。doFilter メソッドには、ExecutionContext を介して、フィールドやコンテンツ・アイテム、およびコールを行ったファイル・ストア・プロバイダに関する情報が渡されます。特に、ファイル・システム・プロバイダについては、ExecutionContext を介してアルゴリズムに次の情報が渡されます。その他のファイル・ストア・プロバイダでは、より多くの情報または異なる情報が渡される可能性があることに注意してください。

```
Properties fieldProperties = (Properties)
    context.getCachedObject("FieldProperties");
Parameters data = (Parameters)
    context.getCachedObject("FileParameters");
Map localData = (Map) context.getCachedObject("LocalProperties");
String algorithm = (String) context.getCachedObject("AlgorithmName");
```

FileSystemFileStoreAlgorithmFilters 表は File Store Provider に含まれており、変更するためにはコンポーネントと Java コードが必要です。



警告: FileSystemFileStoreAlgorithmFilters 表で提供されるデフォルトは、ほとんどの状況に対応できます。このリソース・ファイルは直接編集しないでください。適切に変更するには、Java を使用するか、追加のコンポーネント開発を行う必要があります。コンポーネント開発の詳細は、『Working With Content Server Components』を参照してください。

FileStorage 表

FileStorage 表は、File Store Provider がインストールされると Content Server に追加されます。この表は、コンテンツがデータベースに格納されるときに、JdbcStorage ストレージ・タイプによって排他的に使用されます。FileStorage 表には、コンテンツ・アイテムのレンディションが含まれ、コンテンツ・アイテムとレンディションの dID を使用して、どのレンディションがどのコンテンツ・アイテムに属するかを一意に識別します。

FileCache 表

FileCache 表は、File Store Provider がインストールされると Content Server に追加されます。この表は、ファイル・システムに配置されたレンディションを記憶するために、JdbcStorage ストレージ・タイプによって排他的に使用されます。データベースに格納されるレンディションが、ファイル・システムに配置されるのは、索引作成や変換など固有の処理で必要となる場合です。多くの場合、これらは一時的なファイルであり、予定された処理の一環として指定期間が過ぎると削除されます。

4

実装例

概要

ここでは、それぞれの例についてプロバイダ定義ファイル（provider.hda）に含まれる表の内容を明示的に示します。provider.hda ファイルは手動で編集しないでください。provider.hda ファイルを適切に変更するには、「[Add/Edit Partition](#)」ページ（3-13 ページ）を使用して Content Server ユーザー・インタフェース内で行うか、追加のコンポーネント開発を行う必要があります。[PathMetaData](#) 表、[PathConstruction](#) 表および [FileSystemFileStoreAlgorithmFilters](#) 表といったその他のリソース表で提供されるデフォルト・オプションは、柔軟性が高く、ほとんどの状況に対応できます。

この項の内容は、次のとおりです。

- ❖ [PathMetaData](#) 表のオプション例（4-2 ページ）
- ❖ [標準ファイル・パスの構成](#)（4-3 ページ）
- ❖ [Web 表示非対応ストアまたはオプション Web ストアの構成](#)（4-7 ページ）
- ❖ [データベース格納の構成](#)（4-8 ページ）
- ❖ [パス構成とアルゴリズムの変更](#)（4-10 ページ）

PathMetaData 表のオプション例

ほとんどの例では、[PathMetaData 表](#)の次の構成定義が使用されます。わかりやすいように、例に関係ない一部の列は省略しています。

```
@ResultSet PathMetaData
6
FieldName
GenerationAlgorithm
RequiredForStorage
    <省略された列>
dID

#all
dDocName

#all
dDocAccount
documentAccount

dDocType

#all
dExtension

#all
dWebExtension

weburl
dSecurityGroup

#all
dRevisionID

#all
dReleaseState

#all
dStatus

web
xPartitionId
partitionSelection
```

```

ExtensionSeparator
extensionSeparator

xWebFlag

RenditionId

#all
RevisionLabel
revisionLabel

RenditionSpecifier
renditionSpecifier

@end

```

標準ファイル・パスの構成

ファイル・システムで Content Server の標準の場所にコンテンツを配置するように、File Store Provider を構成することができます。

ストレージ・ルールの定義

まず、ストレージ・ルールを定義します。この場合、すべてのコンテンツをファイル・システムに格納するため、ストレージ・ルールのタイプは *FileStorage* です。

例：

```

@ResultSet StorageRules
4
StorageRule
StorageType
IsWeblessStore
RenditionsOnFileSystem
default
FileStorage

@end@

```

パス構成の定義

次は、ルールストレージ・クラスそれぞれに**パス構成**を定義します。通常、パスの最後の部分は、すべての使用例で標準のものを使用する必要があります。そうしないと、Content Server が hcs* ファイルに対して正常に機能しません。ただし、Web URL ファイル・パス・ルートの変更が、コンテンツ・サーバー Web ルートとして Web サーバーに正しく認識されれば、機能に影響を与えずにルート・パスを変更できます。

この構成では、vault、Web、Web URL の各ストレージ・クラスを **PathConstruction** 表 (3-27 ページ) で定義する必要があります。**vault** のパス式については、第 3 章の「**パス構成と URL 解析の理解**」で説明しています。ユーザーに表示されるのは Web パス式のみです。これは、ルートのみが Web URL と異なります。つまり、Web パスはファイル・システムの絶対パスであり、Web URL は Web サーバーによって提供される URL です。

例：

```
@ResultSet PathConstruction
4
FileStore
PathExpression
AutoCreateLimit
IsWritable
StorageRule
vault
$#env.VaultDir$$dDocType$/$dDocAccount$/$dID$$ExtensionSeparator$$dExtension$
6
true
default
weblink
$HttpWebRoot$groups/$dSecurityGroup$/$dDocAccount$/documents/$dDocType$/
  $dDocName$$RenditionSpecifier$$RevisionLabel$$ExtensionSeparator$
  $dWebExtension$
3
false
default
web
$#env.WeblayoutDir$groups/$dSecurityGroup$/$dDocAccount$/documents/
$dDocType$/$dDocName$$RenditionSpecifier$$RevisionLabel$$ExtensionSeparator$$dWebExtension$
3
true
default
@end
```

Web パス構成は次のように定義されます。

```
$#env.WeblayoutDir$groups/$dSecurityGroup$/$dDocAccount$/documents/$dDocType$/  
$dDocName$$RenditionSpecifier$$RevisionLabel$$ExtensionSeparator$$dWebExtension$
```

これは、次のように各パーツに解析されます。

パスのセグメント	説明
\$#env.WeblayoutDir\$	共有環境で値 WeblayoutDir を検索します。これは、 weblayout ディレクトリの物理ルート・パスとして Content Server によって定義されます。
\$HttpWebRoot\$	Web URL に対する代替 IdocScript 変数。
/groups/	文字列。
\$dSecurityGroup\$	PathMetaData 表によって使用されます。これは必須フィールドです。コール元または記述子作成者が提供する必要があります。コンテンツ・アイテムのメタデータ情報に含まれます。
\$dDocAccount\$	documentAccount アルゴリズムにマップされます。このアルゴリズムは、 dDocAccount を受け取り、適切なすべてのデリミタを含む Content Server の標準アカウント表現に解析します。
/documents/	文字列。
\$dDocType\$	PathMetaData 表によって使用されます。これは必須フィールドです。コール元または記述子作成者が提供する必要があります。コンテンツ・アイテムのメタデータ情報に含まれます。
\$dDocName\$	PathMetaData 表によって使用されます。これは必須フィールドです。コール元または記述子作成者が提供する必要があります。コンテンツ・アイテムのメタデータ情報に含まれます。
\$RenditionSpecifier\$	システムがサムネイルなど追加レンディションを作成している場合のみ、 renditionSpecifier によって提供されます。それ以外の場合は、空の文字列が返されます。

パスのセグメント	説明
\$RevisionLabel\$	リビジョン・ラベルは <code>revisionLabel</code> アルゴリズムによって提供されます。このアルゴリズムは、コンテンツ・アイテムのステータスに応じて、 <code>~dRevLabel</code> をパスに追加します。
\$ExtensionSeparator\$	<code>extensionSeparator</code> アルゴリズムがここで使用されます。デフォルトではピリオド (.) を返します。
\$dWebExtension\$	<code>dWebExtension</code> は Web ストレージ・クラスおよび Web URL ストレージ・クラスの必須フィールドです。ファイル・パラメータを介して渡されます。

Web 表示非対応ストアまたはオプション Web ストアの構成

前の例のストレージ・ルールで `IsWeblessStore` が `true` に構成されているため、この例では、Web 表示可能ファイルはデフォルトでは作成されません。ただし、`Inbound Refinery` または `WebForms`、あるいは Web 表示可能ファイルを必要とするその他のコンポーネントでドキュメントが処理される場合は、Web ファイルが作成されます。ファイルの場所は、前の例と同じ標準構成です。ただし、ファイルに Web レンディションがない場合があるため、Web URL パスを調整する必要があります。また、Web URL ファイルの使用方法にも注意してください。これは、Web 表示可能ファイルが実際に存在するときに URL を計算するために使用されます。メタデータ・フィールド `xWebFlag` を使用して、ファイルがブラウザに提供される方法を判別します。

ストレージ・ルールの定義

例：

```
@ResultSet StorageRules
4
StorageRule
StorageType
IsWeblessStore
RenditionsOnFileSystem
default
FileStorage
true
@end@
```

パス構成の定義

例：

```
@ResultSet PathConstruction
4
FileStore
PathExpression
AutoCreateLimit
IsWritable
vault
$#env.VaultDir$$dDocType$/$dDocAccount$/$dID$$ExtensionSeparator$$dExtension$
6
true
```

```

default
weburl
$HttpCgiPath$?IdcService=GET_FILE&dID=$dID$
    &dDocName=$dDocName$&allowInterrupt=1&noSaveAs=1&fileName=$dOriginalName$
3
false
default
weburl.file
$HttpWebRoot$groups/$dSecurityGroup$/$dDocAccount$/documents/$dDocType$/
    $dDocName$$RenditionSpecifier$$RevisionLabel$$ExtensionSeparator$
    $dWebExtension$
3
false
default
web
$#env.WeblayoutDir$groups/$dSecurityGroup$/$dDocAccount$/documents/
    $dDocType$/$dDocName$$RenditionSpecifier$$RevisionLabel$
    $ExtensionSeparator$$dWebExtension$
3
true
default
@end

```

データベース格納の構成

データベースにファイルを格納するには、タイプが *JdbcStorage* のストレージ・ルールが必要です。デフォルトでは、このルールに属するすべてのコンテンツ・アイテムのファイルがデータベースに格納されます。ただし、ファイルがデータベースに格納されていても、基礎となるファイル・システムやシステムにおいて、ファイルを一時的にファイル・システムにキャッシュすることが必要になる可能性があります。これは、特に索引作成や変換に際して発生することがあります。



技術ヒント: 所定のストレージ・クラスに属するレンディションを常にファイル・システムに格納するようにルールを構成できます。これが最も役立つのは、ボールド・ファイルをデータベースに格納し、Web ファイルをファイル・システムに格納するシステムです。

ストレージ・ルールの定義

次のデフォルト・ルールではすべてのファイルがデータベースに格納されますが、*filesInWeb* ルールでは、ボールド・ファイルがデータベースに、Web ファイルがファイル・システムに格納されます。

例：

```

@ResultSet StorageRules
4
StorageRule
StorageType
IsWeblessStore
RenditionsOnFileSystem
default
JdbcStorage
filesInWeb
JdbcStorage

web
@end@

```

パス構成の定義

```

@ResultSet PathConstruction
4
FileStore
PathExpression
AutoCreateLimit
IsWritable
StorageRule
vault
$#env.VaultDir$$dDocType$/$dDocAccount/$dID$$ExtensionSeparator$$dExtension$
6
true
default
webserv.file
$HttpWebRoot$groups/$dSecurityGroup$/$dDocAccount$/documents/$dDocType$/
    $dDocName$$RenditionSpecifier$$RevisionLabel$$ExtensionSeparator$
    $dWebExtension$
3
false
default
web
$#env.WeblayoutDir$groups/$dSecurityGroup$/$dDocAccount$/documents/
    $dDocType$/$dDocName$$RevisionLabel$$RenditionSpecifier$
    $ExtensionSeparator$$dWebExtension$
3
true
default
@end

```

パス構成とアルゴリズムの変更

これまでの例では、標準構成のファイル・パスを使用していました。非常に大規模な実装の場合には、ディレクトリが一杯になったりパフォーマンスの低下を招いたりする可能性があります。次の例では、ファイルをいくつかの格納オプションに分散することができます。

パーティション化の使用

File Store Provider では、パーティションを容易に使用して、粗いディレクトリ構造を作成できます。デフォルトでは、`xPartitionId` メタデータ・フィールドが使用され、これがコンテンツ・アイテム・リビジョンのメタデータ情報に含まれるようになります。使用パーティションをパーティション選択アルゴリズムで判別しないように、**Content Server** ユーザー・インタフェースでこのフィールドを無効にすることをお勧めします。パーティション選択アルゴリズムは、すべてのアクティブなパーティションを調べます。新しいコンテンツがシステムに入ると、パーティションが順番に選択されます。各パーティションは [PartitionList](#) 表にエントリがあり、アクティブであることを宣言できます。`PartitionRoot` は `xPartitionId` から計算されます。この値は [PartitionList](#) 表の参照キーです。`xPartitionId` が指定されていない場合は、使用可能でアクティブな次のパーティションがシステムによって検索され、場所の計算にその値が使用されます。その後、`xPartitionId` は、コンテンツ・アイテムのメタデータの一部として格納されます。

パーティション選択を使用するには、`vault` ストレージ・クラスを `PathConstruction` 表に次のように定義します。

```
vault
  $PartitionRoot$/$dDocType$/$dDocAccount$/$dID$$ExtensionSeparator$$dExtension$
  6
  true
```

システム管理者が、投稿されないようにパーティションを閉じる必要がある場合、たとえばストレージ・デバイスのメンテナンスが必要な場合は、[「Add/Edit Partition」ページ](#) (3-13 ページ) を使用していつでもパーティションを非アクティブ化することができます。

ディレクトリのファイル数の制限

ファイルを分散させるもう 1 つの方法は、パスを変更して、コンテンツ・アイテムの `dID` ごとにファイルを分割することです。次の例では、ディレクトリの制限は、10,000 ファイルに予備ファイル分（追加レンディション用）を加えたものです。

パス式の内容が次の場合について考えます。

```
$dID[-12:-10:0]/$dID[-10:-8:0]/$dID[-8:-4:0]/$
```

`dID` が 1234567890 であれば、結果は 00/12/3456 です。

このパス式の `dID[-12:-10:0]` に注意してください。これは次のように解釈されます。

- ❖ 文字列の最後から 12 番目の文字から取得し始め、文字列の最後から 10 番目の文字まで取得します。
- ❖ 結果の文字列の長さが 2（すなわち 12-10）になるように文字 0 を埋め込みます。



サード・パーティ・ライセンス

概要

この付録には、この製品に付属するすべてのサード・パーティ製品のサード・パーティ・ライセンスの説明が含まれます。

- ❖ [Apache Software License](#) (A-1 ページ)
- ❖ [W3C® Software Notice and License](#) (A-2 ページ)
- ❖ [Zlib License](#) (A-3 ページ)
- ❖ 一般的な [BSD ライセンス](#) (A-4 ページ)
- ❖ 一般的な [MIT ライセンス](#) (A-5 ページ)
- ❖ [Unicode ライセンス](#) (A-5 ページ)
- ❖ [その他の帰属](#) (A-7 ページ)

APACHE SOFTWARE LICENSE

```
* Copyright 1999-2004 The Apache Software Foundation.  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*   http://www.apache.org/licenses/LICENSE-2.0  
*
```

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

W3C® SOFTWARE NOTICE AND LICENSE

- * Copyright © 1994-2000 World Wide Web Consortium,
- * (Massachusetts Institute of Technology, Institut National de
- * Recherche en Informatique et en Automatique, Keio University).
- * All Rights Reserved. <http://www.w3.org/Consortium/Legal/>
- *
- * This W3C work (including software, documents, or other related items) is
- * being provided by the copyright holders under the following license.By
- * obtaining, using and/or copying this work, you (the licensee) agree that
- * you have read, understood, and will comply with the following terms and
- * conditions:
- *
- * Permission to use, copy, modify, and distribute this software and its
- * documentation, with or without modification, for any purpose and without
- * fee or royalty is hereby granted, provided that you include the following
- * on ALL copies of the software and documentation or portions thereof,
- * including modifications, that you make:
- *
- * 1. The full text of this NOTICE in a location viewable to users of the
- * redistributed or derivative work.
- *
- * 2. Any pre-existing intellectual property disclaimers, notices, or terms
- * and conditions.If none exist, a short notice of the following form
- * (hypertext is preferred, text is permitted) should be used within the
- * body of any redistributed or derivative code: "Copyright ©
- * [\$date-of-software] World Wide Web Consortium, (Massachusetts

* Institute of Technology, Institut National de Recherche en
* Informatique et en Automatique, Keio University).All Rights
* Reserved. <http://www.w3.org/Consortium/Legal/>"
*
* 3. Notice of any changes or modifications to the W3C files, including the
* date changes were made.(We recommend you provide URIs to the location
* from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission.Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty.In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

一般的な BSD ライセンス

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

一般的な MIT ライセンス

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE ライセンス

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>. Unicode Software includes any source code published in the Unicode Standard or under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

その他の帰属

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc

サード・パーティ・ライセンス

索引

記号

\$#env.WeblayoutDir\$, 4-5
 \$dDocAccount\$, 4-5
 \$dDocName\$, 4-5
 \$dDocType\$, 4-5
 \$dSecurityGroup\$, 4-5
 \$dWebExtension\$, 4-6
 \$ExtensionSeparator\$, 4-6
 \$HttpWebRoot\$, 4-5
 \$RenditionSpecifier\$, 4-5
 \$RevisionLabel\$, 4-6

A

Access Implementor, 3-17
 Action, 3-13
 Add Partition, 3-13
 ArgumentFields, 3-26
 Arguments, 3-26
 AutoCreateLimit, 3-27

C

Capacity Check Interval, 3-14
 CapacityCheckInterval, 3-24
 Configuration Class, 3-17
 Connection Class, 3-17
 Content Server のオプション, 3-3
 Content Server のページ, 3-12
 「Edit File Store Provider」ページ, 3-16
 「Partition Listing」ページ, 3-12
 「Provider Information」ページ, 3-23
 「Provider Listing」ページ, 3-22
 ストレージ・ルールの追加ページ, 3-18
 パーティションの編集ページ, 3-13

D

Descriptor Implementor, 3-17
 Duplication Methods, 3-14
 DuplicationMethods, 3-24

E

「Edit File Store Provider」ページ, 3-16
 Access Implementor, 3-17
 Configuration Class, 3-17
 Connection Class, 3-17
 Descriptor Implementor, 3-17
 Edit Rule, 3-17
 Event Implementor, 3-17
 Metadata Implementor, 3-17
 Provider Class, 3-16
 Provider Description, 3-16
 Provider Name, 3-16
 Storage Rules, 3-17
 Edit Rule, 3-17
 Event Implementor, 3-17

F

FieldName, 3-26
 FileStorage, 3-25
 FileStore, 3-27
 FsCacheThreshold, 3-3
 FsMaximumFileCacheAge, 3-3
 FsMinimumFileCacheAge, 3-3

G

GenerationAlgorithm, 3-26

I

Is Active, 3-13, 3-14
Is Webless File Store, 3-19
IsActive, 3-24
IsWeblessStore, 3-25

J

JDBC Storage, 3-19
JdbcStorage, 3-25

M

Metadata Implementor, 3-17

P

「Partition Listing」 ページ, 3-12
Action, 3-13
Add Partition, 3-13
Is Active, 3-13
Partition Name, 3-12
Partition Root, 3-13
Partition Name, 3-12, 3-14
Partition Root, 3-13, 3-14
PartitionList 表
CapacityCheckInterval, 3-24
DuplicationMethods, 3-24
IsActive, 3-24
PartitionName, 3-24
PartitionRoot, 3-24
SlackBytes, 3-24
PartitionName, 3-24
PartitionRoot, 3-24
PathConstruction 表
AutoCreateLimit, 3-27
FileStore, 3-27
PathExpression, 3-27
StorageRule, 3-27
PathExpression, 3-27
PathMetaData 表, 4-2
ArgumentFields, 3-26
Arguments, 3-26
FieldName, 3-26
GenerationAlgorithm, 3-26
RequiredForStorage, 3-26
Provider Class, 3-16

Provider Description, 3-16
Provider Name, 3-16

R

Renditions, 3-19
RenditionsOnFileSystem, 3-25
RequiredForStorage, 3-26
Reset, 3-15

S

Show Path Metadata, 3-19
Slack Bytes, 3-14
SlackBytes, 3-24
Storage Rules, 3-17
StorageRule, 3-25, 3-27
StorageRule 表
IsWeblessStore, 3-25
RenditionsOnFileSystem, 3-25
StorageRule, 3-25
StorageType, 3-25
StorageType, 3-25

U

Update, 3-15

V

Vault Path, 3-19

W

Web URL File Path, 3-20
Web-viewable Path, 3-19

X

xPartitionId, 3-3
xStorageRule, 3-4
xWebFlag, 3-3

あ

アンインストール, 2-5

い

- インストール, 2-3
 - Component ウィザード使用, 2-4
 - 検証, 2-4
 - コンポーネント・マネージャ, 2-3

き

- 記憶域タイプ
 - FileStorage, 3-25
 - JdbcStorage, 3-25

こ

- 構成
 - Web 表示非対応ストア, 4-7
 - オプション Web ストア, 4-7
 - データベース・ストア, 4-8
 - 標準ファイル・パス, 4-3
- 構成変数, 3-2

し

- システム監査情報, 2-8

す

- ストレージ・ルールの追加ページ, 3-18
- ストレージ・ルールの編集
 - 手順, 3-7
- ストレージ・ルールの編集ページ, 3-18
 - File system only, 3-19
 - Is Webless File Store, 3-19
 - JDBC Storage, 3-19
 - Renditions, 3-19
 - Show Path Metadata, 3-19
 - Vault Path, 3-19
 - Web URL File Path, 3-20
 - Web-viewable Path, 3-19
- ストレージ・ルール・ページ, 3-18

て

- データベース・オプション, 3-2
 - FsCacheThreshold, 3-3
 - FsMaximumFileCacheAge, 3-3

- FsMinimumFileCacheAge, 3-3

データベース表

- FileCache 表, 3-28
- FileStorage 表, 3-28

手順

- ストレージ・ルールの編集, 3-7

と

- トレース, 2-8

は

- パーティション化, 4-10
- パーティションの追加ページ, 3-13
 - Capacity Check Interval, 3-14
 - Duplication Methods, 3-14
 - Is Active, 3-14
 - Partition Name, 3-14
 - Partition Root, 3-14
 - Reset, 3-15
 - Slack Bytes, 3-14
 - Update, 3-15
- パーティションの編集ページ, 3-13
- パス構成, 4-10

ひ

表

- FileCache 表, 3-28
- FileStorage 表, 3-28
- FileSystemFileStoreAlgorithmFilters 表, 3-28
- PartitionList 表, 3-24
- PathConstruction 表, 3-27
- PathMetaData 表, 3-26
- StorageRules 表, 3-25

ふ

- ファイルの制限, 4-10

め

- メタデータ・フィールド
 - xPartitionId, 3-3
 - xStorageRule, 3-4
 - xWebFlag, 3-3

り

- リソース表, 3-23
 - FileCache 表, 3-28
 - FileStorage 表, 3-28
 - FileSystemFileStoreAlgorithmFilters 表, 3-28
 - PartitionList 表, 3-24
 - PathConstruction 表, 3-27
 - PathMetaData 表, 3-26
 - StorageRules 表, 3-25
- 場所, 2-5

れ

- 例
 - PathMetaData 表, 4-2
 - Web 表示非対応ストア, 4-7
 - オプション Web ストア, 4-7
 - データベース・ストア, 4-8
 - 標準ファイル・パス, 4-3

ろ

- ロギング, 2-8