



# Sitecore CMS 6.2

# コンテンツ API クックブック

CMS 開発者のためのコンセプトの概要

## 目次

Chapter 1	イントロダクション .....	5
Chapter 2	データベースの使用 .....	6
2.1	Sitecore データベースへのアクセス方法 .....	7
2.1.1	コンテキスト データベースへのアクセス方法 .....	8
2.1.2	名前でデータベースにアクセスする方法 .....	8
2.1.3	コンテンツ データベースへのアクセス方法 .....	9
2.2	パブリッシュの方法 .....	10
2.2.1	アイテムまたはアイテムのツリーをパブリッシュする方法 .....	10
2.2.2	Master データベースのパブリッシュ方法 .....	10
Chapter 3	アイテムの使用 .....	12
3.1	アイテムへのアクセス方法 .....	13
3.1.1	システム アイテムへのアクセス方法 .....	13
3.1.2	システム データテンプレートへのアクセス方法 .....	13
3.1.3	コンテキスト アイテムへのアクセス方法 .....	14
3.1.4	アイテムへのアクセス権の解決方法 .....	14
3.1.5	アイテムを編集モードにする方法 .....	14
3.2	別のアイテムに関連したアイテムにアクセスする方法 .....	17
3.2.1	アイテムの子にアクセスする方法 .....	17
3.2.2	アイテムのツリーにアクセスする方法 .....	17
3.2.3	アイテムの親にアクセスする方法 .....	18
3.2.4	アイテムの先祖にアクセスする方法 .....	18
3.3	Sitecore クエリを使用してアイテムにアクセスする方法 .....	20
3.4	メディア アイテムへのアクセス方法 .....	22
3.5	アイテムの代替言語にアクセスする方法 .....	23
3.6	アイテムの代替バージョンにアクセスする方法 .....	24
3.7	アイテムの作成方法 .....	26
3.7.1	アイテムのある言語でバージョンを作成する方法 .....	26
3.7.2	メディア ライブラリ アイテムの作成方法 .....	26
ファイル システムを使用したメディア アイテムの作成方法 .....	26	
API を使用したメディア アイテムの作成方法 .....	27	
3.8	アイテムのアイコンへのアクセス方法 .....	28
3.9	アイテムの操作: 名前の変更、移動、コピー、削除 .....	29
3.9.1	アイテムの名前の変更方法 .....	29
3.9.2	アイテムの移動方法 .....	29

3.9.3	アイテムとその子孫をコピーする方法 .....	29
3.9.4	アイテムとその子孫を削除する方法 .....	29
	アイテムの子孫を削除する方法 .....	30
3.10	プロキシアイテムの作成方法 .....	31
3.11	エイリアスの作成方法 .....	32
Chapter 4	フィールドの使用 .....	33
4.1	フィールドへのアクセス方法 .....	34
4.1.1	システム フィールドへのアクセス方法 .....	36
4.1.2	FieldRenderer Web コントロールを使用してフィールドにアクセスする方法 .....	36
4.1.3	Checkbox フィールドへのアクセス方法 .....	37
4.1.4	Date と Datetime フィールドにアクセスする方法 .....	37
4.1.5	File フィールドへのアクセス方法 .....	38
4.1.6	General Link フィールドへのアクセス方法 .....	39
4.1.7	Image フィールドへのアクセス方法 .....	42
4.1.8	Droplink、Droptree、Grouped Droplink フィールドへのアクセス方法 .....	44
4.1.9	Checklist、Multilist、Treelist、TreelistEx フィールドにアクセスする方法 .....	45
4.1.10	FDA (ファイル ドロップ エリア) フィールドへのアクセス方法 .....	46
4.1.11	Word Document フィールドへのアクセス方法 .....	47
4.2	フィールドの標準値にアクセスする方法 .....	49
4.3	フィールドが標準値を持つかどうかを確認する方法 .....	50
4.4	フィールドを標準値にリセットする方法 .....	51
Chapter 5	ダイナミックリンクの使用 .....	52
5.1	コンテンツ アイテムのわかりやすい URL にアクセスする方法 .....	53
5.2	RSS フィードの URL にアクセスする方法 .....	54
5.3	メディア アイテムのわかりやすい URL にアクセスする方法 .....	55
5.4	HTML のダイナミックリンクをわかりやすい URL に変換する方法 .....	56
Chapter 6	シンジケート API .....	57
6.1	Sitecore.Syndication クラス .....	58
6.1.1	Sitecore.Syndication.FeedManager .....	58
6.1.2	Sitecore.Syndication.FeedUtil .....	58
6.1.3	Sitecore.Syndication.PublicFeed .....	58
	例: 複数のアイテムの子を配信可能にする .....	59
6.2	ASP.NET シンジケート クラス .....	60
6.2.1	System.ServiceModel.Syndication.SyndicationItem .....	60
	例: シンジケート エントリのタイトルの長さの制限 .....	60
	例: シンジケート エントリの分類項目 .....	61
Chapter 7	コンテンツ API のトラブルシューティング .....	63
7.1	構成ノードが見つかりません .....	64

7.2	オブジェクトの参照がオブジェクトのインスタンスに設定されていません .....	65
7.3	アイテムが編集モードにありません .....	66
7.4	現在のユーザーにはこのアイテムへの書き込みアクセス権がありません .....	67
7.5	追加するためのアクセスが必要です .....	68

## Chapter 1

# イントロダクション

この文書は CMS 開発者が Sitecore データベースのデータを管理するために使用することができる API（アプリケーション プログラミング インターフェース）のコンセプトの概要を解説します。これにはフィールド値、アイテム間のダイナミックリンク、トラブルシューティングの情報などがあります。<sup>1</sup>

この文書には次の章があります。

- Chapter 1 — イントロダクション
- Chapter 2 — データベースの使用
- Chapter 3 — アイテムの使用
- Chapter 4 — フィールドの使用
- Chapter 5 — ダイナミックリンクの使用
- Chapter 5 — ダイナミックリンクの使用
- Chapter 6 — シンジケート API
- Chapter 7 — コンテンツ API のトラブルシューティング

---

<sup>1</sup> プレゼンテーション コンポーネントで使用される API についての追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20API%20Cookbook.aspx> から『プレゼンテーション コンポーネント API クックブック』を参照してください。

## Chapter 2

# データベースの使用

この章では Sitecore データベースにアクセスするための API、および Master データベースからひとつまたは複数のパブリッシュ ターゲット データベースにパブリッシュするための API についての情報を解説します。

この章には次のセクションがあります。

- Sitecore データベースへのアクセス方法
- パブリッシュの方法

## 2.1 Sitecore データベースへのアクセス方法

Sitecore データベースにアクセスするために `Sitecore.Data.Database` クラスを使用することができます。<sup>2</sup> それぞれの Sitecore のインスタンスは複数の Sitecore のデータベースにアクセスすることが可能です。既定の Sitecore の構成では次の 3 つのデータベースが含まれます：

- Master: CMS ユーザーをサポートするすべてのコンテンツのすべてのバージョンを含みます。
- Web: Web サイトをサポートするマスターからパブリッシュされたバージョンを含みます。
- Core: Sitecore CMS のユーザー インターフェースを制御するデータを含みます。

以降のセクションで解説する技法を使用して Sitecore データベースにアクセスすることができます。

### 重要

Sitecore データベースには SQL コマンドではなく、常に Sitecore API を使用してアクセスしてください。

### 重要

既定の状態ではすべてのコードはコンテキスト ユーザーのセキュリティ コンテキストで実行されます。存在しないデータベース、アイテム、フィールドへのアクセスの試みは null 値が返されるか、または例外が発生します。コンテキスト ユーザーに操作を完了する権利がないことがわかっている場合には、特定のタスクのためのセキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用します。セキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用するための追加情報は、「アイテムへのアクセス権の解決方法」のセクションを参照してください。

### 重要

開発者はアイテムとフィールドにアクセスする際には null 値のチェックを行ってください。この文書のコード例では簡潔性のため、必ずしも null 値のチェックは行っていません。

### メモ

Sitecore API は Sitecore データベースにアクセスします。適切な .NET API を使用して外部のデータストアにアクセスすることも可能です。

### 注意

パブリッシュ ターゲット データベースでアイテムを作成、更新、削除する場合、マスターからパブリッシュ ターゲットへのパブリッシュを行うとそれらの変更を上書きします。パブリッシュ ターゲット データベースでの書き込みは行わないでください。パブリッ

---

<sup>2</sup> Sitecore データベースに関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Content%20Reference.aspx> から『コンテンツ リファレンス』を参照してください。

ユーザーターゲットデータベースへの書き込みのリスクを避けるため、プレゼンテーション コンポーネントから Sitecore データベースへの書き込みを行わないでください。

### メモ

名前、一部のパス、完全なパス、ID、またはその他の抽出条件、たとえば言語やバージョンなどを API メソッドに対して指定することによって、データ テンプレートとフィールドにアクセスすることができます。よりよいパフォーマンスのために、可能な限り ID を使用してください。ただし定数またはその他の機能を使用し、1 つより多くのクラスの文字列のハードコーディングは避けてください。

## 2.1.1 コンテキスト データベースへのアクセス方法

コンテキスト データベースは Web クライアントからアクセスされる論理サイトに関連付けられたプライマリ データベースです。パブリッシュされた Web サイトで稼動しているプレゼンテーション コンポーネントでは、コンテキスト データベースはパブリッシュ ターゲット データベースの 1 つで、たとえば Web データベースです。<sup>3</sup> ページ エディターでは、コンテキスト データベースは Master データベースです。デスクトップ、コンテンツ エディター、その他の CMS ユーザー インターフェイスでは、コンテキスト データベースは Core データベースです。

### 重要

プレゼンテーション コンポーネントは、データベースを名前で参照するのではなく、通常、コンテキスト データベースにアクセスします。

Sitecore.Context.Database プロパティを使用してコンテキスト データベースにアクセスすることができます。たとえば、コンテキスト データベースにアクセスする例を下記に示します：

```
Sitecore.Data.Database context = Sitecore.Context.Database;
```

### 重要

CMS のユーザー インターフェイス コンポーネントはコンテキスト データベースの構成データを使用してコンテンツ データベースのデータを管理します。コンテンツ データベースに関する追加情報は、「コンテンツ データベースへのアクセス方法」のセクションを参照してください。

## 2.1.2 名前でデータベースにアクセスする方法

Sitecore.Configuration.Factory.GetDatabase() メソッドを使用して特定のデータベースにアクセスすることができます。たとえば、Master データベースにアクセスする例を下記に示します：

<sup>3</sup> プレゼンテーション コンポーネントに関する追加情報は

<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Reference.aspx> と <http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Cookbook.aspx> から『プレゼンテーション コンポーネント リファレンス』と『プレゼンテーション コンポーネント クックブック』を参照してください。



```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
```

データをパブリッシュまたはインポートするロジックは通常は Master データベースに名前アクセスします。

#### メモ

第一パラメーターが `web.config` のいずれの `/configuration/sitecore/databases/database` 要素の `id` とも合致しない場合、`Sitecore.Configuration.Factory.GetDatabase()` メソッドは例外を発生します。この比較は大文字と小文字を区別します。

### 2.1.3 コンテンツ データベースへのアクセス方法

コンテンツ エディターなどの CMS ユーザー インターフェースのコンポーネントはコンテンツ データベースを取り扱います。既定のコンテンツ データベースは Master データベースです。ユーザーは Sitecore デスクトップなどのユーザー インターフェースを使用して、コンテンツ データベースを他のデータベースに変更することができます。

`Sitecore.Context.ContentDatabase` プロパティを使用してコンテンツ データベースにアクセスすることができます。たとえば、コンテキスト データベースにアクセスする例を下記に示します:

```
Sitecore.Data.Database content = Sitecore.Context.ContentDatabase;
```

## 2.2 パブリッシュの方法

次の方法でパブリッシュを行うことができます: 1つのアイテムをパブリッシュする。1つのアイテムとその子孫のパブリッシュ可能なすべてのバージョンをパブリッシュする。Master データベースをパブリッシュする。

### 2.2.1 アイテムまたはアイテムのツリーをパブリッシュする方法

個々のアイテムをパブリッシュする場合と、アイテムとパブリッシュ可能な子孫のすべてをパブリッシュする場合には同じ API を使用することができます。/Sitecore/Content/Home アイテムをすべての言語ですべてのパブリッシュ ターゲットにパブリッシュする例を下記に示します:

```
DateTime publishDate = DateTime.Now;
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");

Sitecore.Data.Items.Item targets = master.GetItem("/sitecore/system/publishing targets");

foreach (Sitecore.Data.Items.Item target in targets.Children)
{
    string targetDBName = target["target database"];
    Sitecore.Data.Database targetDB =
        Sitecore.Configuration.Factory.GetDatabase(targetDBName);

    foreach (Sitecore.Globalization.Language language in master.Languages)
    {
        Sitecore.Publishing.PublishOptions publishOptions =
            new Sitecore.Publishing.PublishOptions(master,
                targetDB, Sitecore.Publishing.PublishMode.Incremental, language, publishDate);
        publishOptions.Deep = false;
        Sitecore.Publishing.Publisher publisher =
            new Sitecore.Publishing.Publisher(publishOptions);
        publisher.Publish();
    }
}
```

1つのアイテムとパブリッシュ可能な子孫のすべてをパブリッシュするには、Sitecore.Publishing.Publisher オブジェクトを作成する前に publishOptions.Deep プロパティを true に設定します。

### 2.2.2 Master データベースのパブリッシュ方法

Master データベースにあるパブリッシュ可能なすべてのアイテムのすべての言語のパブリッシュ可能なすべてのバージョンをパブリッシュすることができます。たとえば、すべての言語の Master データベースをすべてのパブリッシュ ターゲットにインクリメンタルにパブリッシュする例を下記に示します:

```
DateTime publishDate = DateTime.Now;
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item targets = master.GetItem("/sitecore/system/publishing targets");

foreach (Sitecore.Data.Items.Item target in targets.Children)
{
    string targetDBName = target["target database"];
    Sitecore.Data.Database targetDB =
        Sitecore.Configuration.Factory.GetDatabase(targetDBName);
```

```
foreach (Sitecore.Globalization.Language language in master.Languages)
{
    Sitecore.Publishing.PublishOptions publishOptions =
        new Sitecore.Publishing.PublishOptions(master,
            targetDB, Sitecore.Publishing.PublishMode.Incremental, language, publishDate);
    Sitecore.Publishing.Publisher publisher =
        new Sitecore.Publishing.Publisher(publishOptions);
    publisher.Publish();
}
}
```

## Chapter 3

# アイテムの使用

この章ではアイテムへのアクセス、作成、更新、移動、削除、その他の操作のための API に関する情報を解説します。

この章には次のセクションがあります。

- アイテムへのアクセス方法
- 別のアイテムに関連したアイテムにアクセスする方法
- Sitecore クエリを使用してアイテムにアクセスする方法
- メディア アイテムへのアクセス方法
- アイテムの代替言語にアクセスする方法
- アイテムの代替バージョンにアクセスする方法
- アイテムの作成方法
- アイテムのアイコンへのアクセス方法
- アイテムの操作: 名前の変更、移動、コピー、削除
- プロキシ アイテムの作成方法
- エイリアスの作成方法

## 3.1 アイテムへのアクセス方法

`Sitecore.Data.Items.Item` クラスを使用して任意のアイテムにアクセスすることができます。

### メモ

Sitecore は特定の種類のアイテムを表す専用のクラスを提供しています。たとえば、`Sitecore.Data.Items.TemplateItem` はデータ テンプレートを表し、`Sitecore.Data.Items.MediaItem` はメディア アイテムを表します。

`Sitecore.Data.Database.GetItem()` メソッドを使用して `Sitecore.Data.Item.Item` を取得することができます。`Sitecore.Data.Database.GetItem()` メソッドの第 1 パラメーターとして、アイテムの ID またはアイテムへのパスを指定することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
```

### 重要

アイテムが存在しない場合、またはコンテキスト ユーザーにアイテムへの読み取りアクセス権がない場合、`Sitecore.Data.Database.GetItem()` メソッドは `null` 値を返します。

### メモ

Sitecore はアイテムのパスを評価する際には大文字と小文字を区別します。

### 3.1.1 システム アイテムへのアクセス方法

パスをハードコーディングすることなく、`Sitecore.ItemIDs` クラスのメンバーを使用してシステム アイテムにアクセスすることができます。<sup>4</sup> たとえば、Master データベースの `/Sitecore/Media Library` アイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item mediaRoot = master.GetItem(Sitecore.ItemIDs.MediaLibraryRoot);
```

### 3.1.2 システム データ テンプレートへのアクセス方法

`Sitecore.TemplateIDs` クラスを使用してシステム データ テンプレートにアクセスすることができます。<sup>5</sup> たとえば、Master データベースの 標準テンプレートにアクセスする例を下記に示します:<sup>6</sup>

---

<sup>4</sup> `Sitecore.ItemIDs` クラスのメンバーに関する追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206.aspx> から Sitecore API の文書を参照してください。

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem standard =
    master.Templates[Sitecore.TemplateIDs.StandardTemplate];
```

### 3.1.3 コンテキスト アイテムへのアクセス方法

プレゼンテーションおよびその他のコンポーネントで、多くの処理はコンテキスト アイテムから開始されます。<sup>7</sup>

Sitecore.Context.Item プロパティを使用してコンテキスト アイテムにアクセスすることができます。たとえば、コンテキスト アイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Items.Item contextItem = Sitecore.Context.Item.
```

### 3.1.4 アイテムへのアクセス権の解決方法

Sitecore API は特定のセキュリティ アクセス権をもつ特定のユーザーのコンテキストで動作します。コンテキスト ユーザーに必要なアクセス権がない場合、Sitecore API は null 値を返すか例外を発生する場合があります。アクセス権の制限を回避するには、セキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用します。<sup>8</sup> セキュリティ無効化を使用する例については、「アイテムを編集モードにする方法」のセクションを参照してください。

#### メモ

なるべくセキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用することなく、コンテキスト ユーザーに適切なアクセス権を付与するようにしてください。

### 3.1.5 アイテムを編集モードにする方法

アイテムを更新する Sitecore API は、アイテムが編集モードにない場合には例外を発生する場合があります。

Sitecore.Data.Items.Item.Editing プロパティのメソッドまたは Sitecore.Data.Items.EditContext クラスを使用して、アイテムを編集モードにすることができます。

---

<sup>5</sup> Sitecore.TemplateIDs クラスのメンバーに関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206.aspx> から Sitecore API の文書を参照してください。

<sup>6</sup> 標準テンプレートについての追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Reference.aspx> から『データ定義リファレンス』を参照してください。

<sup>7</sup> コンテキスト アイテムに関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Reference.aspx> から『プレゼンテーション コンポーネント リファレンス』を参照してください。

<sup>8</sup> セキュリティ ユーザー スイッチャーおよびセキュリティ無効化に関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Security%20API%20Cookbook.aspx> から『セキュリティ API クックブック』を参照してください。

たとえば、下記のコードは `Sitecore.Data.Items.Item.Editing` クラスのメソッドを使用し、セキュリティ無効化の内部で Master データベースの `/Sitecore/Content/Home` アイテムを編集モードにします。

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

//TODO: eliminate SecurityDisabler if possible

using (new Sitecore.SecurityModel.SecurityDisabler())
{
    home.Editing.BeginEdit();

    try
    {
        //TODO: update home

        home.Editing.EndEdit();
    }
    catch (Exception ex)
    {
        home.Editing.CancelEdit();
    }
}
```

### メモ

`Sitecore.Data.Items.Item.Editing.CancelEdit()` メソッドを呼び出す場合、または `Sitecore.Data.Items.Item.Editing.EndEdit()` メソッドを呼び出さない場合、Sitecore は変更をコミットしません。

### 重要

開発者はこの例に示した `try/catch` のブロックを使用してください。この文書のコード例では簡潔性のため、必ずしも `try/catch` のブロックを含めていません。

### メモ

この文書のコード例ではセキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用していません。コンテキスト ユーザーはロジックを実行するために必要なアクセス権を持っていることを前提としています。

または、C# の `using` ステートメントと `Sitecore.Data.Items.EditContext` クラスを使用して、アイテムを編集モードにすることもできます。 `using` ステートメントが閉じると

`Sitecore.Data.Items.EditContext.Dispose()` メソッドが起動され、これによってそのコード セグメント内の変更がコミットされます。

たとえば、下記のコードは `Sitecore.Data.Items.EditContext` を使用して、Master データベースの `/Sitecore/Content/Home` アイテムを編集モードにします:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

//TODO: eliminate SecurityDisabler if possible

using (new Sitecore.SecurityModel.SecurityDisabler())
{
    using (new Sitecore.Data.Items.EditContext(home))
```

```
{  
    //TODO: process home  
}
```

## メモ

Sitecore.Data.Items.EditContext クラスを使用する場合、変更を明示的にロールバックすることはできません。using ステートメントの内部のコードが例外を発生する場合、using ステートメントが閉じると例外以前に行われた変更は自動的にコミットされます。



## 3.2 別のアイテムに関連したアイテムにアクセスする方法

このセクションでは別のアイテムに関連したアイテムにアクセスするために使用できる API に関する情報を解説します。

### 3.2.1 アイテムの子にアクセスする方法

Sitecore.Data.Items.Children プロパティを使用して、アイテムの子にアクセスすることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの子にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

foreach(Sitecore.Data.Items.Item child in home.Children)
{
    //TODO: process child
}
```

Sitecore.Data.Items.Item.Children プロパティにアクセスする都度、システムは新規の Sitecore.Collections.Childlist オブジェクトを作成します。C# の foreach ステートメントの実装はプロパティに一度だけアクセスします。アイテムの子に C# の foreach ステートメントでなく C# の for ステートメントを使用してアクセスする場合、Sitecore.Data.Items.Item.Children プロパティの元の値を含む変数を使用し、コレクションを繰り返し再定義することを避けます。たとえば、C# の for ステートメントを使用して Master データベースの /Sitecore/Content/Home アイテムの子にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Collections.ChildList children = home.Children;

for(int childIndex=0; childIndex<children.Count ; childIndex++)
{
    //TODO: process children[childIndex]
}
```

### 3.2.2 アイテムのツリーにアクセスする方法

再帰メソッドと Sitecore.Data.Items.Item.Children プロパティを使用して、ツリーにあるすべてのアイテムにアクセスすることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムとそのアイテムの各子孫を処理する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
ProcessRecursively(home);
...
private void ProcessRecursively(Sitecore.Data.Items.Item item)
{
    //TODO: process item

    foreach(Sitecore.Data.Items.Item child in item.Children)
    {
        ProcessRecursively(child);
    }
}
```

## 注意

再帰メソッドが引数を自身に渡した場合、そのメソッドは無限ループを実装してしまいます。

ツリーのルート アイテムを処理することを避けるため、アイテムを再帰メソッドに渡すのではなく、ループ内の処理ロジックを子に反復する再帰メソッドに移動し、そのロジックの中で子アイテムを処理します。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの子孫のみを処理する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
ProcessRecursively(home);
...
private void ProcessRecursively(Sitecore.Data.Items.Item item)
{
    foreach(Sitecore.Data.Items.Item child in item.Children)
    {
        //TODO: process child

        ProcessRecursively(child);
    }
}
```

## メモ

`descendant` 軸または `descendant-or-self` 軸を使用し、Sitecore クエリを使用してツリー全体にアクセスすることも可能です。Sitecore クエリに関する追加情報は、「Sitecore クエリを使用してアイテムにアクセスする方法」のセクションを参照してください。

### 3.2.3 アイテムの親にアクセスする方法

`Sitecore.Data.Items.Item.Parent` プロパティを使用して、アイテムの親にアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの親アイテム (`/Sitecore/Content`) にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item parent = home.Parent;
```

### 3.2.4 アイテムの先祖にアクセスする方法

`Sitecore.Data.Items.Item.Parent` プロパティを再帰メソッドで使用してアイテムの先祖にアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの先祖にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
ProcessRecursively(home.Parent);
...
private void ProcessRecursively(Sitecore.Data.Items.Item item)
{
    //TODO: process item

    if (item.Parent != null )
    {

```

```
        ProcessRecursively(item.Parent);
    }
}
```

### 注意

再帰メソッドが引数を自身に渡した場合、そのメソッドは無限ループを実装してしまいます。

または、`Sitecore.Data.Items.Item.Axes.GetAncestors()` メソッドを使用してアイテムの先祖にアクセスすることもできます。たとえば、コンテキスト アイテムの先祖にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item[] ancestors = home.Axes.GetAncestors();

foreach(Sitecore.Data.Items.Item ancestor in ancestors)
{
    //TODO: process ancestor
}
```

### メモ

これらの技法の重要な相違点は、`Sitecore.Data.Items.Item.Axes.GetAncestors()` メソッドはルートからドキュメント順にアクセスする一方、`Sitecore.Data.Items.Item.Parent` は逆にツリーまたはリーフアイテムから順にアクセスすることです。

### メモ

Sitecore クエリを使用して、アイテムの先祖である `ancestor` または `ancestor-or-self` 軸にアクセスすることも可能です。Sitecore クエリに関する追加情報は、「Sitecore クエリを使用してアイテムにアクセスする方法」のセクションを参照してください。

### 3.3 Sitecore クエリを使用してアイテムにアクセスする方法

`Sitecore.Data.Database.SelectItems()` メソッドを使用して Sitecore クエリに合致するデータベースのアイテムを取得することができます。

#### 重要

Sitecore クエリは、大量のデータを持つリポジトリでアイテムを検索するには、必ずしも最も効率的な方法ではありません。システムが多くのツリーの中でアイテムを頻繁に検索する必要がある場合には、検索インデックスまたは他のソリューションの使用を検討してください。

#### 重要

Sitecore クエリの構文は XPath 構文と同一ではありません。

#### 重要

Sitecore クエリがドキュメント順または逆ドキュメント順にアイテムを返すと想定しないでください。

#### 重要

Sitecore クエリの結果にアクセスする前に必ず null 値のチェックを行ってください。

#### メモ

`web.config` の `/configuration/sitecore/settings/setting` 要素の `name` が `Query.MaxItems` である `value` 属性が Sitecore クエリが返すアイテムの最大数を指定します。

たとえば、Master データベースの `Common/Folder` データ テンプレートに基づくすべてのアイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
string query = String.Format("//*[@@templateid='{0}']",
    Sitecore.TemplateIDs.Folder);
Sitecore.Data.Items.Item[] queried = master.SelectItems(query);

if (queried != null)
{
    foreach (Sitecore.Data.Items.Item item in queried)
    {
        //TODO: process item
    }
}
```

`Sitecore.Data.Items.Item.Axes.SelectItems()` メソッドを使用して、他のアイテムに対して相対的に Sitecore クエリに合致するアイテムにアクセスすることができます。<sup>9</sup> たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `ancestor-or-self` 軸のアイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
```

<sup>9</sup> Sitecore クエリに関する追加情報は、<http://sdn.sitecore.net/Reference/Using%20Sitecore%20Query.aspx> を参照してください。

```
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item[] queried = home.Axes.SelectItems("ancestor-or-self::*");

if (queried!=null)
{
    foreach(Sitecore.Data.Items.Item item in queried)
    {
        //TODO: process item
    }
}
```

### 3.4 メディア アイテムへのアクセス方法

Sitecore.Data.Items.MediaItem クラスを使用してメディア アイテムにアクセスすることができます。メディア アイテムを表す Sitecore.Data.Items.Item を Sitecore.Data.Items.MediaItem のコンストラクターへの第 1 パラメーターとして渡します。たとえば、Master データベースの /Sitecore/Media Library/Files/Sample メディア アイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item item = master.GetItem("/sitecore/media library/files/sample");

if (item==null)
{
    //TODO: handle case that item does not exist
}
else
{
    Sitecore.Data.Items.MediaItem media = new Sitecore.Data.Items.MediaItem(item);

    //TODO: process media
}
```

### 3.5 アイテムの代替言語にアクセスする方法

各アイテムは複数の言語を含むことができます。Sitecore.Data.Database.GetItem() メソッドを使用してアイテムにアクセスする場合に、Sitecore.Globalization.Language クラスを使用して言語を指定することができます。たとえば、/Sitecore/Content/Home アイテムの現在のバージョンに、既定の en 言語でアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Globalization.Language language = Sitecore.Globalization.Language.Parse("en");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home", language);
```

#### メモ

言語を指定しない場合には、Sitecore は既定ではコンテキスト言語を使用します。Sitecore のユーザー インターフェイス コンポーネントはコンテンツ言語を指定します。

Sitecore.Data.Items.Item.Versions.Count プロパティを使用して、あるバージョンがある言語に存在するかどうかを知ることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの各言語の現在のバージョンにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

foreach (Sitecore.Globalization.Language language in home.Languages)
{
    Sitecore.Data.Items.Item langItem = home.Database.GetItem(home.ID, language);

    if (langItem.Versions.Count > 0)
    {
        //TODO: process langItem
    }
    else
    {
        //TODO: handle case that version data exists for language
    }
}
```

## 3.6 アイテムの代替バージョンにアクセスする方法

各アイテムは複数の言語を含むことができます。各言語は複数のバージョンを含むことができます。

`Sitecore.Data.Database.GetItem()` メソッドを使用してアイテムにアクセスする場合に、`Sitecore.Data.Version` クラスを使用してバージョンを指定することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの第 1 バージョンに、既定の `en` 言語でアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Globalization.Language language = Sitecore.Globalization.Language.Parse("en");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home",
    language, Sitecore.Data.Version.Latest);
```

Master データベースの `/Sitecore/Content/Home` アイテムの第 2 バージョンに、既定の `en` 言語でアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Globalization.Language language = Sitecore.Globalization.Language.Parse("en");
Sitecore.Data.Version second = new Sitecore.Data.Version(2);
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home",
    language, second);
```

### メモ

バージョンを指定しない場合には、Sitecore は既定では現在のバージョンを使用します。

`Sitecore.Data.Items.Version.Number` プロパティを使用してバージョンの識別子にアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムのバージョン番号にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
int verNumber = home.Version.Number;
```

`Sitecore.Data.Items.Item.Versions.Count` プロパティを使用して、あるアイテムのあるバージョンがある言語に存在するかどうかを知ることができます。`Sitecore.Data.Items.Item.Versions.GetVersions()` プロパティを使用して、あるアイテムのある言語のすべてのバージョンにアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの各言語のすべてのバージョンにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

foreach (Sitecore.Globalization.Language language in home.Languages)
{
    Sitecore.Data.Items.Item langItem = master.GetItem(home.ID, language);

    if (langItem.Versions.Count > 0)
    {
        foreach (Sitecore.Data.Items.Item verItem in langItem.Versions.GetVersions())
        {
            //TODO: process verItem
        }
    }
    else
    {
        //TODO: handle case that no versions exist in language
    }
}
```



```
}  
}
```

## 3.7 アイテムの作成方法

`Sitecore.Data.Items.Item.Add()` メソッドを使用してアイテムを作成することができます。アイテムを作成する前に、親アイテムと新規アイテムのデータ テンプレートが存在している必要があります。たとえば、Master データベースのデータ テンプレート アイテム `Sample/Sample Item` を使用して `/Sitecore/Content/Home/MyItem` アイテムを作成する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.Item myItem = home.Add("MyItem", sample);
```

### 3.7.1 アイテムのある言語でバージョンを作成する方法

`Sitecore.Data.Items.Versions.Add()` メソッドを使用して、アイテムのある言語でバージョンを作成することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムに、バージョンが存在しない各言語に第 1 のバージョンを追加する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");

foreach (Sitecore.Globalization.Language language in home.Languages)
{
    Sitecore.Data.Items.Item langItem = home.Database.GetItem(home.ID, language);

    if (langItem.Versions.Count < 1)
    {
        langItem = langItem.Versions.AddVersion();
    }
}
```

### 3.7.2 メディア ライブラリ アイテムの作成方法

メディア ライブラリ アイテムを作成するには、Sitecore が監視しているファイル システムにファイルを作成するか、または API を起動します。

#### ファイル システムを使用したメディア アイテムの作成方法

Sitecore のファイル システム (通常は `/upload` ディレクトリ) にファイルをコピーまたは移動することによってメディア ライブラリ アイテムを作成することができます。`web.config` の `/configuration/sitecore/sc.variable` 要素の `name` が `mediaFolder` である `value` 属性によって指定されたディレクトリにディレクトリとファイルを作成すると、Sitecore は対応するメディア フォルダーとメディア アイテムを Master データベースの `/Sitecore/Media Library` の下に作成します。

### 重要

Sitecore は ASP.NET がファイル システム イベントを起こした場合のみメディア アイテムを作成します。メディア ファイルを Sitecore ファイル システムに作成する前に、ASP.NET が処理するリソース (たとえばホーム ページ) を要求して、ASP.NET プロセスがアクティブであることを確認します。

### 注意

過度のファイル システム アクティビティは ASP.NET のワーカー プロセスに過大な負荷を与える場合があります。Sitecore のログ ファイルでメディア インポートのプロセスを監視し、完了するかどうかを確認することができます。メディアのインポートがうまくいかない場合、少量に分割してインポートを試してみてください。

### ヒント

Sitecore がメディア ライブラリ アイテムを作成した後で、ファイルは削除することができます。

## API を使用したメディア アイテムの作成方法

Sitecore.Resources.Media.MediaCreator と

Sitecore.Resources.Media.MediaCreatorOptions クラスを使用して、ファイルからメディア アイテムを作成することができます。たとえば、ファイル C:\temp\sample.jpg から Master データベースの /Sitecore/Media Library/Images/Sample メディア アイテムを作成する例を下記に示します:

```
Sitecore.Resources.Media.MediaCreatorOptions options =
    new Sitecore.Resources.Media.MediaCreatorOptions();
options.Database = Sitecore.Configuration.Factory.GetDatabase("master");
options.Language = Sitecore.Globalization.Language.Parse(
    Sitecore.Configuration.Settings.DefaultLanguage);
options.Versioned = Sitecore.Configuration.Settings.Media.UploadAsVersionableByDefault;
options.Destination = "/sitecore/media library/images/sample";
options.FileBased = Sitecore.Configuration.Settings.Media.UploadAsFiles;
Sitecore.Resources.Media.MediaCreator creator =
    new Sitecore.Resources.Media.MediaCreator();
Sitecore.Data.Items.MediaItem sample =
    creator.CreateFromFile(@"C:\temp\sample.jpg", options);
```

### 3.8 アイテムのアイコンへのアクセス方法

`Sitecore.Data.Items.Item.Appearance.Icon` プロパティを使用してアイテムのアイコンにアクセスすることができます。アイコンがテーマのイメージを含んでいる場合、

`Sitecore.Resources.Images.GetThemedImageSource()` メソッドを使用して相対パスを完全なパスに変換することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムのアイコンの完全なパスを知る例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
string icon = home.Appearance.Icon;

if (icon.StartsWith("~/"))
{
    icon = Sitecore.StringUtil.EnsurePrefix('/', icon);
}
else if (!(icon.StartsWith("/") && icon.Contains(":")))
{
    icon = Sitecore.Resources.Images.GetThemedImageSource(icon);
}
```

`Sitecore.Data.Items.Item.Appearance.Icon` プロパティを使用してアイテムのアイコンを設定することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムのアイコンを、テーマのイメージ `network/16x16/home.png` に設定する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
home.Editing.BeginEdit();
home.Appearance.Icon = "network/16x16/home.png";
home.Editing.EndEdit();
```

## 3.9 アイテムの操作: 名前の変更、移動、コピー、削除

このセクションではアイテムの名前の変更、移動、コピー、削除の操作を行う API の情報を解説します。

### 3.9.1 アイテムの名前の変更方法

`Sitecore.Data.Items.Item.Name` プロパティを使用してアイテムの名前の変更を行うことができます。たとえば、Master データベースの `/Sitecore/Content/Home/Sample` アイテムの名前を `/Sitecore/Content/Home/Changed` に変更する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
sample.Editing.BeginEdit();
sample.Name = "Changed";
sample.Editing.EndEdit();
```

### 3.9.2 アイテムの移動方法

`Sitecore.Data.Items.Item.MoveTo()` メソッドを使用してアイテムまたはツリーを移動することができます。たとえば、Master データベースの `/Sitecore/Content/Home/Sample` アイテムを `/Sitecore/Content` に移動する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
Sitecore.Data.Items.Item content = master.GetItem(Sitecore.ItemIDs.ContentRoot);
sample.MoveTo(content);
```

### 3.9.3 アイテムとその子孫をコピーする方法

`Sitecore.Data.Items.Item.CopyTo()` メソッドを使用してアイテムとその子孫をコピーすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムとその子孫をコピーして `/Sitecore/Content/Sibling` を作成する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item content = master.GetItem(Sitecore.ItemIDs.ContentRoot);
home.CopyTo(content, "Sibling");
```

### 3.9.4 アイテムとその子孫を削除する方法

`Sitecore.Data.Items.Item.Delete()` メソッドを使用してアイテムとその子孫を削除することができます。たとえば、Master データベースの `/Sitecore/Content/Home/Sample` アイテムを削除する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
sample.Delete();
```

**メモ**

`Sitecore.Data.Items.Item.Delete()` メソッドはアイテムとすべての子孫を削除します。

**アイテムの子孫を削除する方法**

`Sitecore.Data.Items.Item.DeleteChildren()` メソッドを使用してあるアイテムの子孫を削除することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの子孫を削除する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
sample.DeleteChildren();
```

## 3.10 プロキシ アイテムの作成方法

System/Alias データ テンプレートを使用してプロキシ定義アイテムを作成することができます。<sup>10</sup> たとえば、プロキシ定義アイテム /Sitecore/System/Proxies/MyProxy を作成して /Sitecore/Content/Home/Sample とその子孫を Master データベースの /Sitecore/Layout/Sample の代理とする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");

if (master.ProxiesEnabled)
{
    Sitecore.Data.Items.Item proxies = master.GetItem("/sitecore/system/proxies");
    Sitecore.Data.Items.TemplateItem proxy = master.Templates[Sitecore.TemplateIDs.Proxy];
    Sitecore.Data.Items.Item proxyDef = proxies.Add("MyProxy", proxy);
    Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
    proxyDef.Editing.BeginEdit();
    Sitecore.Data.Items.Item parent = master.GetItem(Sitecore.ItemIDs.LayoutRoot);
    proxyDef.Fields[Sitecore.FieldIDs.ProxyTargetItem].Value = parent.ID.ToString();
    proxyDef.Fields[Sitecore.FieldIDs.ProxySourceItem].Value = sample.ID.ToString();
    proxyDef.Fields[Sitecore.FieldIDs.ProxyInsertionType].Value = "Entire sub-tree";
    proxyDef.Editing.EndEdit();
}
else
{
    //TODO: handle case that proxies are disabled in the database
}
```

個別のアイテムのプロキシを設定するには、プロキシ定義アイテムの

Sitecore.FieldIDs.ProxyInsertionType フィールドの値を Root item only に設定します。プロキシアイテムを別のデータベースから設定するには、プロキシ定義アイテムの

Sitecore.FieldIDs.ProxySourceDatabase フィールドの値をソースデータベースの名前に設定します。

### メモ

データベースでプロキシを有効化するには、web.config の適切な id の

/configuration/sitecore/databases/database 要素の <proxiesEnabled> 要素の値を true に設定します。

---

<sup>10</sup> プロキシに関する追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206/Content%20Reference.aspx> から『コンテンツ リファレンス』を、また <http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Cookbook.aspx> から『データ定義クックブック』を参照してください。

### 3.11 エイリアスの作成方法

System/Alias データ テンプレートを使用してエイリアス定義アイテムを作成することができます。<sup>11</sup> たとえば、エイリアス定義アイテム /Sitecore/System/Aliases/MyAlias を作成し、URL /MyAlias.aspx が /Sitecore/Content/Home/Sample を起動するようにする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem alias = master.Templates[Sitecore.TemplateIDs.Alias];
Sitecore.Data.Items.Item aliases = master.GetItem("/sitecore/system/aliases");
Sitecore.Data.Items.Item myAlias = aliases.Add("MyAlias", alias);
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
myAlias.Editing.BeginEdit();
Sitecore.Data.Fields.LinkField linkField = myAlias.Fields["linked item"];
linkField.LinkType = "internal";
Sitecore.Links.UrlOptions urlOptions = Sitecore.Links.LinkManager.GetDefaultUrlOptions();
urlOptions.AlwaysIncludeServerUrl = false;
linkField.Url = Sitecore.Links.LinkManager.GetItemUrl(sample, urlOptions);
linkField.TargetID = sample.ID;
myAlias.Appearance.Icon = sample.Appearance.Icon;
myAlias.Editing.EndEdit();
```

---

<sup>11</sup> エイリアスに関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Content%20Reference.aspx> から『コンテンツ リファレンス』を、また <http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Cookbook.aspx> から『データ定義クックブック』を参照してください。



## Chapter 4

# フィールドの使用

この章では各種のデータ テンプレート フィールド値にアクセスして更新する API に関する情報を解説します。

この章には次のセクションがあります。

- フィールドへのアクセス方法
- フィールドの標準値にアクセスする方法
- フィールドが標準値を持つかどうかを確認する方法
- フィールドを標準値にリセットする方法

## 4.1 フィールドへのアクセス方法

このセクションに記載されている API を使用してフィールド値を読み書きすることができます。メディア アイテムのバイナリデータを保存する Attachment システム フィールド型を除き、Sitecore はすべてのフィールド値をテキストとして保存します。任意のフィールドをシンプル テキストとしてアクセスすることができます。または `Sitecore.Data.Fields` 名前空間の専用のクラスを使用することができます。

いくつかのフィールド型はシンプル テキスト値から構成されています。チェックボックス フィールドは選択された場合、1 文字 ("1") を保存します。リッチ テキスト エディタ (RTE) フィールドは XML エンコードされた HTML または XHTML を含んでいます。多くのフィールド型は単一のアイテムの ID、またはパイプ文字 ("|") で区切られた複数の ID を含みます。その他のフィールド型は XML または独自のフォーマットのデータを含みます。

### ヒント

あるフィールド型の文字列のフォーマットを知るには、コンテンツ エディターを開き、フィールドを含むアイテムを選択し、生のフィールド値を表示します。<sup>12</sup>

`Sitecore.Data.Items.Item` クラスで公開されるコレクションを使用して、任意のフィールドの文字列の値にアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドの値にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
string title = home["title"];
```

### 重要

指定されたフィールドが存在しない場合、`Sitecore.Data.Items.Item` に公開されたコレクションは、`null` 値でなく、空の文字列を返します。

`Sitecore.Data.Items.Item` クラスで公開されるコレクションを使用して、任意のフィールドの値を更新することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドの値を更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
home.Editing.BeginEdit();
home["title"] = "//TODO: replace with appropriate value";
home.Editing.EndEdit();
```

`Sitecore.Data.Items.Item` クラスに公開されるコレクションを使用して、フィールドの値を空の文字列に設定することでクリアすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドをクリアする例を下記に示します:

---

<sup>12</sup> 生のフィールド値を表示する方法については、<http://sdn.sitecore.net/Reference/Sitecore%206.aspx> から『クライアント構成クックブック』を参照してください。

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
home.Editing.BeginEdit();
home["title"] = String.Empty;
home.Editing.EndEdit();
```

### 重要

フィールドの値をクリアすることはそのフィールドを標準値にリセットすることにはなりません。フィールドを標準値にリセットする方法は、「フィールドを標準値にリセットする方法」のセクションを参照してください。

### メモ

フィールド値を null 値に設定することはできません。

または、`Sitecore.Data.Items.Item.Fields` プロパティを使用して、`Sitecore.Data.Fields.Field` クラスのインスタンスとして、任意のフィールドにアクセスすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];

if(titleField!=null)
{
    string title = titleField.Value;
}
```

### 重要

指定されたフィールドがアイテムに存在しない場合、`Sitecore.Data.Items.Item.Fields` コレクションは null 値を返します。

`Sitecore.Data.Fields.Field.Value` プロパティを使用してフィールド値を更新することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドの値を更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];

if(titleField!=null)
{
    home.Editing.BeginEdit();
    titleField.Value = "//TODO: replace with appropriate value";
    home.Editing.EndEdit();
}
```

### 重要

Sitecore は必要な場合に限りデータを取得します。Sitecore があるアイテムのすべてのフィールド値を取得することを確認するためには、`Sitecore.Data.Items.Item.ReadAll()` メソッドを起動します。たとえば、コンテキストアイテムのフィールドを反復する例を下記に示します。

```
Sitecore.Data.Items.Item item = Sitecore.Context.Item;
item.Fields.ReadAll();
```

```
foreach (Sitecore.Data.Fields.Field field in item.Fields)
{
    // TODO: process field
}
```

#### 4.1.1 システム フィールドへのアクセス方法

Sitecore.FieldIDs クラスを使用してシステム フィールドにアクセスすることができます。<sup>13</sup> たとえば、Master データベースの /Sitecore/Content/Home/Sample アイテムのアーカイブ日付フィールドにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
Sitecore.Data.Fields.DateField archiveField =
    sample.Fields[Sitecore.FieldIDs.ArchiveDate];
DateTime archiveWhen = archiveField.DateTime;
```

#### 4.1.2 FieldRenderer Web コントロールを使用してフィールドにアクセスする方法

プレゼンテーション コンポーネントでは、フィールド値を出力するためには可能な限り FieldRenderer Web コントロールを使用します。<sup>14</sup> たとえば、Web コントロールからコンテキスト アイテムで Title フィールドを出力する例を下記に示します:

```
namespace Namespace.Web.UI.WebControls
{
    public class WebControlName :Sitecore.Web.UI.WebControl
    {
        protected override void DoRender(System.Web.UI.HtmlTextWriter output)
        {
            string html = Sitecore.Web.UI.WebControls.FieldRenderer.Render(
                Sitecore.Context.Item, "title");
            output.Write(html);
        }
    }
}
```

または、FieldRenderer Web コントロールをレイアウトまたはサブレイアウトに追加することもできます。たとえば:

```
<sc:FieldRenderer runat="server" ID="fieldControl" />
```

レイアウトまたはサブレイアウトのコードビハインドで、コントロールのプロパティを設定します。たとえば、コンテキスト アイテムで Title フィールドをレンダリングする例を下記に示します:

```
namespace Namespace.Web.UI
{
    public partial class SublayoutName :System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            fieldControl.DataSource = Sitecore.Context.Item.Paths.FullPath;
            fieldControl.FieldName = "title";
        }
    }
}
```

<sup>13</sup> Sitecore.FieldIDs クラスのメンバーに関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206.aspx> から Sitecore API の文書を参照してください。

<sup>14</sup> FieldRenderer Web コントロールに関する追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206.aspx> から『プレゼンテーション コンポーネント リファレンス』を参照してください。

```

    }
  }
}

```

### 重要

FieldRenderer Web コントロールはすべてのフィールド型をサポートしません。FieldRenderer Web コントロールは Date (日付)、Datetime (日時)、Image (イメージ)、Integer (整数)、Multi-Line Text (複数行テキスト)、Number (数値)、Rich Text (リッチテキスト)、Single-Line Text (単一行テキスト) のフィールドで使用することができます。

## 4.1.3 Checkbox フィールドへのアクセス方法

Sitecore.Data.Fields.CheckboxField クラスを使用して Checkbox 型のデータ テンプレート フィールドにアクセスすることができます。ユーザーがチェックボックスを選択したかどうかを知るには

Sitecore.Data.Fields.CheckboxField.Checked プロパティにアクセスします。たとえば、Master データベースの /Sitecore/Content/Home アイテムの CheckboxField という名前のチェックボックスが選択されたかどうかを知る例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.CheckboxField checkboxField = home.Fields["checkboxfield"];

if (checkboxField != null && checkboxField.Checked)
{
  //TODO: handle case that checkbox exists and is selected
}

```

Checkbox フィールドを選択するためには、Sitecore.Data.Fields.CheckboxField.Checked プロパティを設定します。たとえば、Master データベースの /Sitecore/Content/Home アイテムの CheckboxField という名前のチェックボックスが選択されたことを確認する例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.CheckboxField checkboxField = home.Fields["checkboxfield"];

if (checkboxField != null && ! checkboxField.Checked )
{
  home.Editing.BeginEdit();
  checkboxField.Checked = true;
  home.Editing.EndEdit();
}

```

## 4.1.4 Date と Datetime フィールドにアクセスする方法

Sitecore.Data.Fields.DateField クラスを使用して Date と Datetime 型のデータ テンプレート フィールドにアクセスすることができます。Sitecore.Data.Fields.DateField.Value プロパティは Sitecore で使用されている ISO フォーマットの文字列 (yyyyMMddTHH:mm:ss) として日付と時間を含んでいます。<sup>15</sup>

Sitecore.DateUtil.IsodateToDateTime() メソッドを使用して、ISO フォーマットの値を

<sup>15</sup> .NET の日付の書式形式についての追加情報は、<http://msdn.microsoft.com/en-us/library/73ctwf33.aspx> を参照してください。

`System.DateTime` 構造体に変換することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `DateTextField` という名前の `Date` または `Datetime` フィールドにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.DateField dateTimeField = home.Fields["datetimefield"];
string dateTimeString = dateTimeField.Value;
DateTime dateTimeStruct = Sitecore.DateUtil.IsoDateToDateTime(dateTimeString);
```

または、同じ値を表す `System.DateTime` 構造体を含む

`Sitecore.Data.Fields.DateField.DateTime` プロパティにアクセスすることもできます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `DateTextField` という名前の `Date` または `Datetime` フィールドに `System.DateTime` 構造体としてアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.DateField dateTimeField = home.Fields["datefield"];
DateTime dateTimeStruct = dateTimeField.DateTime;
```

`Sitecore.Data.Fields.DateField.Value` プロパティを ISO フォーマットの文字列に更新することによって、`Date` または `Datetime` フィールドの値を更新することができます。`Sitecore.DateUtil.ToIsoDate()` メソッドを使用して `System.DateTime` 構造体を ISO フォーマットに変換することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `DateTextField` という名前の `Datetime` フィールドを現在のシステム日付に更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.DateField dateTimeField = home.Fields["datetimefield"];

if (dateTimeField != null)
{
    home.Editing.BeginEdit();
    dateTimeField.Value = Sitecore.DateUtil.ToIsoDate(DateTime.Now);
    home.Editing.EndEdit();
}
```

#### メモ

ユーザーが `Date` または `Datetime` 型のフィールドに値を指定しなかった場合、

`Sitecore.Data.Fields.DateField.Value` プロパティは空の文字列を含み、

`Sitecore.Data.Fields.DateField.DateTime` プロパティは `System.DateTime.MinValue` を含みます。

#### メモ

既定の状態では、`Date` 型のフィールドは午前 0 時の時刻値を含みます。

### 4.1.5 File フィールドへのアクセス方法

`Sitecore.Data.Fields.FileField` クラスを使用して `File` 型のデータ テンプレート フィールドにアクセスすることができます。`Sitecore.Data.Fields.FileField.MediaItem` プロパティを使用してフィールドで選択されたメ

メディア アイテムにアクセスすることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの FileField という名前の File フィールドで参照されているメディア アイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.FileField fileField = home.Fields["filefield"];
Sitecore.Data.Items.Item file = fileField.MediaItem;

if (fileField==null)
{
    //TODO: handle case that field does not exist
}
else if (file==null)
{
    //TODO: handle case that user has not selected a file
}
else
{
    Sitecore.Data.Items.MediaItem media = new Sitecore.Data.Items.MediaItem(file);
    //TODO: process media
}
```

Sitecore.Data.Fields.FileField.MediaID と Sitecore.Data.Fields.FileField.Src 属性を更新することによって、File フィールドの値を更新することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの FileField という名前の File フィールドが /Sitecore/Media Library/Files/Sample メディア アイテムを指定していることを確認する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.FileField fileField = home.Fields["filefield"];
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/media library/files/sample");

if (fileField == null )
{
    //TODO: handle case that field does not exist
}
else if (sample == null)
{
    //TODO: handle case that sample does not exist
}
else if (fileField.MediaID != sample.ID)
{
    home.Editing.BeginEdit();
    fileField.MediaID = sample.ID;
    fileField.Src = Sitecore.Resources.Media.MediaManager.GetMediaUrl(sample);
    home.Editing.EndEdit();
}
```

#### 4.1.6 General Link フィールドへのアクセス方法

Sitecore.Data.Fields.LinkField クラスを使用して General Link 型のデータ テンプレート フィールドにアクセスすることができます。フィールドのリンクの種類によって、Sitecore.Data.Fields.LinkField クラスの次のようなプロパティを使用することができます。

プロパティ	値
Anchor	HTML <a> 要素の name 属性 (先頭のハッシュ文字 ("#") を除く)
Class	HTML <a> 要素の class 属性
IsInternal	内部リンクには True (真)、メディアその他の種類のリンクには False (偽)
IsMediaLink	メディアリンクには True (真)
LinkType	リンクの種類を識別するトークン (internal、media、external、mailto、anchor、javascript)
MediaPath	メディア アイテムへの完全なパス
QueryString	URL に追加するクエリ文字列パラメーター
Target	HTML <a> 要素の target 属性
TargetID	内部リンクまたはメディアリンクで指定されたアイテムの ID
TargetItem	内部リンクまたはメディアリンクで指定された Sitecore.Data.Items.Item
Text	HTML <a> 要素のテキスト コンテンツ
Title	HTML <a> 要素の title 属性
Url	メディア アイテムを除き、Url プロパティがメディア アイテムの /Sitecore/Media Library に対する相対パスを含むリンクの URL

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドの URL を確認する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
string url = String.Empty;

switch(linkField.LinkType)
{
    case "internal":
    case "external":
    case "mailto":
    case "anchor":
    case "javascript":
        url = linkField.Url;
        break;
    case "media":
        Sitecore.Data.Items.MediaItem media =
```



```

        new Sitecore.Data.Items.MediaItem(linkField.TargetItem);
        url = Sitecore.StringUtil.EnsurePrefix('/',
            Sitecore.Resources.Media.MediaManager.GetMediaUrl(media));
        break;
    case "":
        break;
    default:
        string message = String.Format("{0} :Unknown link type {1} in {2}",
            this.GetType(), linkField.LinkType, home.Paths.FullPath);
        Sitecore.Diagnostics.Log.Error(message, this);
        break;
    }
}

```

Sitecore.Data.Fields.LinkField.Clear() メソッドを使用して General Link 型のフィールドからデータを削除することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドからリンクを削除する例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
home.Editing.BeginEdit();
linkField.Clear();
home.Editing.EndEdit();

```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドを /Sitecore/Content/Home/Sample アイテムに更新する例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");
home.Editing.BeginEdit();
linkField.Clear();
linkField.LinkType = "internal";
Sitecore.Links.UrlOptions urlOptions = Sitecore.Links.LinkManager.GetDefaultUrlOptions();
urlOptions.AlwaysIncludeServerUrl = false;
linkField.Url = Sitecore.Links.LinkManager.GetItemUrl(sample, urlOptions);
linkField.TargetID = sample.ID;
home.Editing.EndEdit();

```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドを /Sitecore/Media Library/Files/Sample メディア アイテムに更新する例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/media library/files/sample");
home.Editing.BeginEdit();
linkField.Clear();
linkField.LinkType = "media";
linkField.Url = sample.Paths.MediaPath;
linkField.TargetID = sample.ID;
home.Editing.EndEdit();

```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドを外部 URL <http://sitecore.net> に更新する例を下記に示します:

```

Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
home.Editing.BeginEdit();

```

```
linkField.Clear();
linkField.LinkType = "external";
linkField.Url = "http://sitecore.net";
home.Editing.EndEdit();
```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドをアンカー namedAnchor に更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
home.Editing.BeginEdit();
linkField.Clear();
linkField.LinkType = "anchor";
linkField.Url = "namedAnchor";
home.Editing.EndEdit();
```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドを電子メール アドレス email@domain.tld に更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
home.Editing.BeginEdit();
linkField.Clear();
linkField.LinkType = "mailto";
linkField.Url = "mailto:email@domain.tld";
home.Editing.EndEdit();
```

Master データベースの /Sitecore/Content/Home アイテムの GeneralLinkField という名前の General Link フィールドを JavaScript 関数に更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.LinkField linkField = home.Fields["generallinkfield"];
home.Editing.BeginEdit();
linkField.Clear();
linkField.Text = "//TODO: replace with appropriate value";
linkField.LinkType = "javascript";
linkField.Url = @"javascript:alert('javascript')";
home.Editing.EndEdit();
```

## 4.1.7 Image フィールドへのアクセス方法

Sitecore.Data.Fields.ImageField クラスを使用して Image 型のデータテンプレートフィールドにアクセスすることができます。Sitecore.Data.Fields.ImageField.MediaType プロパティを使用してフィールドで選択されたメディア アイテムに Sitecore.Data.Items.Item としてアクセスすることができます。フィールドがイメージを指定していない場合、Sitecore.Data.Fields.ImageField.MediaType プロパティは null 値です。

Sitecore.Resources.ImageBuilder クラスを使用して HTML <img> 要素を作成することができます。

Sitecore.Resources.Media.MediaTypeManager.GetMediaUrl() メソッドを使用してメディア アイテムの URL を確認することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの ImageField という名前の Image フィールドの値に基づいて HTML <img> 要素を作成する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.ImageField imageField = home.Fields["imagefield"];
```

```
if (imageField!=null && imageField.MediaItem!=null)
{
    Sitecore.Data.Items.MediaItem image =
        new Sitecore.Data.Items.MediaItem(imageField.MediaItem);
    string src = Sitecore.StringUtil.EnsurePrefix('/',
        Sitecore.Resources.Media.MediaManager.GetMediaUrl(image));
    string imgTag = String.Format("<img src=\"\"{0}\"\" alt=\"\"{1}\"\" />", src, image.Alt);
}
```

### 重要

Sitecore メディア ライブラリをビジネス ユーザーによって投稿されるイメージやその他のメディアのために使用します。ファイル システムとソース コード管理システムを開発者によって管理されるイメージのために使用します。

Sitecore.Data.Fields.ImageField.Clear() メソッドを使用してイメージ フィールドのコンテンツをクリアすることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの ImageField という名前の Image フィールドをクリアする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.ImageField imageField = home.Fields["imagefield"];
home.Editing.BeginEdit();
imageField.Clear();
home.Editing.EndEdit();
```

Sitecore.Data.Fields.ImageField クラスを使用して Image フィールドを更新することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの ImageField という名前の Image フィールドを /Sitecore/Media Library/Images/Sample イメージに更新する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item sampleItem =
    master.GetItem("/sitecore/media library/images/sample");
Sitecore.Data.Items.MediaItem sampleMedia =
    new Sitecore.Data.Items.MediaItem(sampleItem);
Sitecore.Data.Fields.ImageField imageField = home.Fields["imagefield"];

if (imageField.MediaID != sampleMedia.ID)
{
    home.Editing.BeginEdit();
    imageField.Clear();
    imageField.Src = Sitecore.Resources.Media.MediaManager.GetMediaUrl(sampleMedia);
    imageField.MediaID = sampleMedia.ID;
    imageField.MediaPath = sampleMedia.MediaPath;

    if (!String.IsNullOrEmpty(sampleMedia.Alt))
    {
        imageField.Alt = sampleMedia.Alt;
    }
    else
    {
        imageField.Alt = sampleMedia.DisplayName;
    }

    home.Editing.EndEdit();
}
```

## 4.1.8 Droplink、Droptree、Grouped Droplink フィールドへのアクセス方法

Sitecore.Data.Fields.ReferenceField クラスを使用して、ユーザーは1つのアイテムを選択してフィールド型にアクセスすることができます。これには Droplink、Droptree、Grouped Droplink を含みます。

Sitecore.Data.Fields.ReferenceField.TargetItem プロパティはフィールドに指定された Sitecore.Data.Items.Item または null 値を含みます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの ReferenceField という名前の Droptree フィールドで指定されているアイテムにアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.ReferenceField referenceField = item.Fields["referencefield"];

if (referenceField==null)
{
    //TODO: handle case that field does not exist
}
else if (referenceField.TargetItem==null)
{
    //TODO: handle case that user has not selected an item
}
else
{
    Sitecore.Data.Items.Item referencedItem = referenceField.TargetItem;
    //TODO: process referencedItem
}
```

アイテムの ID に Sitecore.Data.Fields.ReferenceField.Value プロパティを設定して、サポートされている型の1つのフィールドを更新することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの ReferenceField という名前の Droptree フィールドが /Sitecore/Content/Home/Sample アイテムを指定していることを確認する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.ReferenceField referenceField = home.Fields["referencefield"];

if (referenceField==null)
{
    //TODO: handle case that field does not exist
}
else
{
    Sitecore.Data.Items.Item sample = master.GetItem("/sitecore/content/home/sample");

    if (sample==null)
    {
        //TODO: handle case that sample does not exist
    }
    else if (sample.ID.ToString() != referenceField.Value)
    {
        home.Editing.BeginEdit();
        referenceField.Value = sample.ID.ToString();
        home.Editing.EndEdit();
    }
}
```

### 4.1.9 Checklist、Multilist、Treelist、TreelistEx フィールドにアクセスする方法

Sitecore.Data.Fields.MultilistField クラスを使用して、ユーザーは複数のアイテムを選択してフィールド型にアクセスすることができます。これには Checklist、Multilist、Treelist、TreelistEx を含みます。

Sitecore.Data.Fields.MultilistField.GetItems() メソッドを使用して、フィールドに指定されたアイテムを表す Sitecore.Data.Item.Item オブジェクトのリストにアクセスすることができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの Multiselect という名前の Multilist フィールドで指定されているアイテムに対して反復する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.MultilistField multiselectField = home.Fields["multiselect"];

if (multiselectField==null)
{
    //TODO: handle case that field does not exist
}
else
{
    Sitecore.Data.Items.Item[] items= multiselectField.GetItems();

    if (items!=null && items.Length>0)
    {
        for (int i=0; i<items.Length ; i++)
        {
            //process items[i]
        }
    }
}
```

#### メモ

Sitecore.Data.Fields.MultilistField.GetItems() メソッドが返すリストの個々のメンバーが null 値となることはありません。あるアイテムをユーザーがそのアイテムへの参照を更新することなく削除してしまった場合、Sitecore.Data.Fields.MultilistField.GetItems() メソッドはそのアイテムを結果から除外します。

#### メモ

Sitecore.Data.Fields.MultilistField クラスを使用して Droplink、Droptree、Grouped Droplink のフィールド型にアクセスすることも可能です。この方法では、1 つまたは複数の Sitecore アイテムの ID を保存するすべてのフィールド型に単一のプログラミング モデルを提供します。このためフィールドの型を変更する場合にコードを更新する必要を少なくすることができます。Droplink、Droptree、Grouped Droplink は複数のアイテムの選択をサポートしないため、これらのフィールド型の更新には Sitecore.Data.Fields.Multilist クラスを使用するべきではありません。

Sitecore.Data.Fields.MulitlistField.Add() メソッドを使用して、サポートされているフィールド型にアイテムを追加することができます。また Sitecore.Data.Fields.MulitlistField.Remove() メソッドを使用してアイテムを削除することができます。たとえば、Master データベースの /Sitecore/Content/Home アイテムの TreelistEx フィールド Multiselect が /Sitecore/Content/Home/Sample1 アイテムを指定している一方で /Sitecore/Content/Home/Sample2 アイテムは指定していないことを確認する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
```

```
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.Item sample1 = master.GetItem("/sitecore/content/home/sample1");
Sitecore.Data.Items.Item sample2 = master.GetItem("/sitecore/content/home/sample2");
Sitecore.Data.Fields.MultilistField multiselectField = home.Fields["multiselect"];

if(multiselectField.Contains(sample2.ID.ToString())
|| !multiselectField.Contains(sample1.ID.ToString()))
{
    home.Editing.BeginEdit();

    if(!multiselectField.Contains(sample1.ID.ToString()))
    {
        multiselectField.Add(sample1.ID.ToString());
    }

    if(multiselectField.Contains(sample2.ID.ToString()))
    {
        multiselectField.Remove(sample2.ID.ToString());
    }

    home.Editing.EndEdit();
}
```

#### 4.1.10 FDA (ファイル ドロップ エリア) フィールドへのアクセス方法

Sitecore.Data.Fields.FileDropAreaField クラスを使用して、FDA (ファイル ドロップ エリア) 型のフィールドの値にアクセスすることができます。Sitecore.Data.Fields.FileDropAreaField.GetMediaItems() メソッドは FDA フィールドに関連付けられたメディア アイテムを返します。

次の例に示すような Web コントロールを実装して、FDA フィールドに関連付けられたメディア アイテムへのリンクの不規則なリストを生成することができます。

```
namespace Sitecore.Sharedsource.Web.UI.WebControls
{
    using System;

    public class FDALinks :Sitecore.Web.UI.WebControl
    {
        public string FieldName
        {
            get;
            set;
        }

        protected override void DoRender(System.Web.UI.HtmlTextWriter output)
        {
            if (this.FieldName == null
                || Sitecore.Context.Item == null
                || output == null)
            {
                return;
            }

            Sitecore.Data.Fields.FileDropAreaField fdaField =
                Sitecore.Context.Item.Fields[this.FieldName];

            if (fdaField == null)
            {
                return;
            }

            Sitecore.Collections.ItemList mediaItems = fdaField.GetMediaItems();
        }
    }
}
```

```
if (mediaItems.Count < 1)
{
    return;
}

output.Write("<ul>");

foreach (Sitecore.Data.Items.Item mediaItem in mediaItems)
{
    string mediaUrl = Sitecore.StringUtil.EnsurePrefix(
        '/',
        Sitecore.Resources.Media.MediaManager.GetMediaUrl(mediaItem));
    string markup = String.Format(
        @"<li><a href="{0}">{1}</a>",
        mediaUrl,
        mediaItem.Name);
    output.Write(markup);
}

output.Write("</ul>");
}
}
```

#### 4.1.11 Word Document フィールドへのアクセス方法

`Sitecore.Data.Fields.WordDocumentField` クラスを使用して Word Document フィールドの値にアクセスすることができます。`Sitecore.Data.Fields.WordDocumentField.Html` プロパティを使用してフィールド値の HTML 表現にアクセスすることができます。`Sitecore.Data.Fields.WordDocumentField.PlainText` プロパティを使用してフィールド値のプレーン テキスト表現にアクセスすることができます。

`Sitecore.Data.Fields.WordDocumentField.Styles` プロパティを使用して HTML に関連付けられた CSS (カスケード スタイル シート) コードにアクセスすることができます。たとえば、次の例に示すような Web コントロールを実装して、Word Document フィールドのスタイルと HTML を出力することができます。

```
namespace Sitecore.Sharedsource.Web.UI.WebControls
{
    using System;
    using System.Text.RegularExpressions;

    public class RenderWordDocumentFieldRenderer : Sitecore.Web.UI.WebControl
    {
        public string FieldName
        {
            get;
            set;
        }

        protected override void DoRender(System.Web.UI.HtmlTextWriter output)
        {
            Sitecore.Data.Items.Item item = this.GetItem();

            if (this.FieldName == null || item == null)
            {
                return;
            }

            Sitecore.Data.Fields.WordDocumentField wordField = item.Fields["WordField"];

            if (wordField == null
```

```
        || wordField.BlobId == Sitecore.Data.ID.Null
        || Regex.IsMatch(wordField.PlainText, @"\s*$"))
    {
        return;
    }

    output.WriteLine(String.Format(
        "<style type=\"text/css\">{0}</style>",
        wordField.Styles));
    output.WriteLine(Sitecore.Web.UI.WebControls.FieldRenderer.Render(
        item,
        wordField.InnerField.Name));
    }
}
```

`Sitecore.WordOCX.WordOCXUrlManager.GetDownloadLink()` メソッドを使用して、Word Document フィールドの値にワードの文書 (.docx ファイル) としてアクセスできる URL を取得することができます。

たとえば、コンテキスト アイテムの `WordDocument` という名前の Word Document フィールドにワードの文書ファイルとしてアクセスする URL を取得する例を下記に示します:

```
Sitecore.Data.Items.Item item = Sitecore.Context.Item;
Sitecore.Data.Fields.WordDocumentField wordField = item.Fields["WordDocument"];

if (wordField != null
    && wordField.BlobId != Sitecore.Data.ID.Null
    && !Regex.IsMatch(wordField.PlainText, @"\s*$"))
{
    Dictionary<string, string> parameters = new Dictionary<string, string>();
    parameters["db"] = item.Database.Name;
    parameters["blobId"] = wordField.BlobId.ToString();
    Sitecore.WordOCX.WordOCXUrlManager wordManager =
        new Sitecore.WordOCX.WordOCXUrlManager(parameters);
    string url = wordManager.GetDownloadLink();
}
```



## 4.2 フィールドの標準値にアクセスする方法

`Sitecore.Data.Fields.Field.GetStandardValue()` メソッドを使用してフィールドの標準値にアクセスすることができます。<sup>16</sup> たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの Title フィールドの標準値にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];
string standardTitle = titleField.GetStandardValue();
```

---

<sup>16</sup> フィールドの標準値についての追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%20.aspx> から『データ定義リファレンス』および『データ定義クックブック』を参照してください。

## 4.3 フィールドが標準値を持つかどうかを確認する方法

`Sitecore.Data.Fields.Field.ContainsStandardValue` プロパティを使用して、あるフィールドが標準値を持つかどうかを確認することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドが標準値を持つかどうかを確認する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];

if (titleField.ContainsStandardValue)
{
    //TODO: handle case that field contains standard value
}
```

### メモ

フィールドに標準値がない場合、`Sitecore.Data.Fields.Field.ContainsStandardValue` プロパティは `False` (偽) です。

### メモ

フィールドはその標準値と同じ値を、実際には標準値を持つことなく、含むことができます。たとえば、あるアイテムがその標準値を持つとします。ユーザーがそのフィールドを更新すると、そのフィールドはすでに標準値を持ちません。ユーザーが再度フィールドを更新し、そのフィールドの標準値と同じ値をフィールド値として設定したものの、標準値のリセットはしなかったとします。その場合、フィールドは標準値と同じ値を持ちますが、標準値は持ちません。フィールドの値はその標準値と同じですが、`Sitecore.Data.Fields.Field.ContainsStandardValue` プロパティは `False` (偽) です。

## 4.4 フィールドを標準値にリセットする方法

`Sitecore.Data.Fields.Field.Reset()` メソッドを使用してフィールドをその標準値にリセットすることができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Title` フィールドの値をその標準値にリセットする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];
home.Editing.BeginEdit();
titleField.Reset();
home.Editing.EndEdit();
```

### 重要

`Sitecore.Data.Fields.Field` の値を空の文字列に更新しても、フィールドの値はデータ テンプレートで関連付けられた標準値アイテムで定義されている値には戻されません。フィールドをその標準値にリセットするには `Sitecore.Data.Fields.Field.Reset()` メソッドを使用します。

### 重要

`Sitecore.Data.Fields.Field` の値をフィールドの標準値に更新しても、フィールドの値はデータ テンプレートで関連付けられた標準値アイテムで定義されている値には戻されません。フィールドをその標準値にリセットするには `Sitecore.Data.Fields.Field.Reset()` メソッドを使用します。

### 重要

フィールドを標準値にリセットしたとき、Sitecore は標準値の `$_name` などのトークンを展開しません。トークンを置換するには `MasterVariablesReplacer` を使用します。

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.Field titleField = home.Fields["title"];
home.Editing.BeginEdit();
titleField.Reset();
home.Editing.EndEdit();
home.Editing.BeginEdit();
Sitecore.Data.MasterVariablesReplacer replacer =
    Sitecore.Configuration.Factory.GetMasterVariablesReplacer();
replacer.ReplaceField(home, titleField);
home.Editing.EndEdit();
```

### メモ

フィールド値に `MasterVariablesReplacer` を使用する前にリセット操作をコミットする必要があります。

## Chapter 5

# ダイナミックリンクの使用

この章では Sitecore のダイナミックリンク管理の API を構成し使用するための情報を解説します。<sup>17</sup>

この章には次のセクションがあります。

- コンテンツ アイテムのわかりやすい URL にアクセスする方法
- RSS フィードの URL にアクセスする方法
- メディア アイテムのわかりやすい URL にアクセスする方法
- HTML のダイナミックリンクをわかりやすい URL に変換する方法

---

<sup>17</sup> Sitecore のダイナミックリンクに関する追加情報は、  
<http://sdn.sitecore.net/Reference/Sitecore%206/Dynamic%20Links.aspx> から『ダイナミックリンクの使用のガイド』  
を参照してください。

## 5.1 コンテンツ アイテムのわかりやすい URL にアクセスする方法

`Sitecore.Links.LinkManager.GetItemUrl()` メソッドを使用してコンテンツ アイテムのわかりやすい URL にアクセスすることができます。<sup>18</sup> たとえば、Master データベースの `/Sitecore/Content/Home` アイテムのわかりやすい URL にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
string url = Sitecore.Links.LinkManager.GetItemUrl(home);
```

---

<sup>18</sup> わかりやすい URL に関する追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Dynamic%20Links.aspx> から『ダイナミック リンクの使用のガイド』を参照してください。

## 5.2 RSS フィードの URL にアクセスする方法

RSS フィードの URL はフィード定義アイテムの既定の URL です。<sup>19</sup> RSS フィードの URL にアクセスする API は任意のアイテムの URL にアクセスするものと同じです。アイテムの URL にアクセスするために使用する API についての追加情報は、前のセクション「コンテンツアイテムのわかりやすい URL にアクセスする方法」を参照してください。

`Sitecore.Syndication.FeedManager.GetFeedUrl()` メソッドを使用して RSS の URL を取得することができます。`Sitecore.Syndication.FeedManager.GetFeedUrl()` の第 1 パラメーターはフィード定義アイテムです。第 2 パラメーターは URL に認証情報を含むかどうかを示します。

### メモ

RSS URL 認証機能は Sitecore クライアント RSS フィードのみに使用します。<sup>20</sup>

シンジケート API に関する追加情報は、Chapter 6「シンジケート API」を参照してください。

---

<sup>19</sup> Sitecore の RSS 機能についての追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Cookbook.aspx> から『プレゼンテーション コンポーネント クックブック』を参照してください。

<sup>20</sup> Sitecore クライアント の RSS フィードについての追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Client%20Configuration%20Cookbook.aspx> から『クライアント構成クックブック』を参照してください。

## 5.3 メディア アイテムのわかりやすい URL にアクセスする方法

`Sitecore.Resources.Media.MediaManager.GetMediaUrl()` メソッドを使用してメディア アイテムのわかりやすい URL にアクセスすることができます。たとえば、Master データベースの `/Sitecore/Media Library/Images/Sample` メディア アイテムのわかりやすい URL にアクセスする例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sampleItem = master.GetItem(
    "/sitecore/media library/images/sample");
Sitecore.Data.Items.Item sampleMedia = new Sitecore.Data.Items.MediaItem(sampleItem);
string url = Sitecore.StringUtil.EnsurePrefix('/',
    Sitecore.Resources.Media.MediaManager.GetMediaUrl(sampleMedia));
```

### 注意

Sitecore は、メディアの URL の先頭のスラッシュ文字 ("/") を自動的に含めません。これはメディア アイテムの相対 URL となり、IIS がチルダ文字 ("~") によってドキュメント ルートから解決します。非常に深い情報体系を持つソリューションでは、相対メディア URL が Web クライアントまたは Web サーバーによる制限を超過することがあります。先の例で示した `Sitecore.StringUtil.EnsurePrefix()` メソッドを使用して、メディア URL が先頭のスラッシュ文字を含むことを確認してください。

`Sitecore.Resources.Media.MediaUrlOptions` クラスを使用してメディア オプションを指定することができます。たとえば、Master データベースの `/Sitecore/Media Library/Images/Sample` メディア アイテムのサムネイルの URL を取得する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item sampleItem = master.GetItem(
    "/sitecore/media library/images/sample");
Sitecore.Data.Items.MediaItem sampleMedia =
    new Sitecore.Data.Items.MediaItem(sampleItem);
Sitecore.Resources.Media.MediaUrlOptions mediaOptions =
    new Sitecore.Resources.Media.MediaUrlOptions();
mediaOptions.Thumbnail = true;
string url = Sitecore.StringUtil.EnsurePrefix('/',
    Sitecore.Resources.Media.MediaManager.GetMediaUrl(sampleMedia, mediaOptions));
```

## 5.4 HTML のダイナミック リンクをわかりやすい URL に変換する方法

ダイナミックリンクをわかりやすい URL に変換するためには、FieldRenderer Web コントロール、または renderField パイプライン、または `Sitecore.Links.LinkManager.ExpandDynamicLinks()` メソッドを使用することができます。<sup>21</sup>

`Sitecore.Links.LinkManager.ExpandDynamicLinks()` メソッドを使用して、リッチ テキスト エディタ (RTE) フィールドのダイナミックリンクを、コンテンツとメディアリンクを含め、わかりやすい URL に変換することができます。たとえば、Master データベースの `/Sitecore/Content/Home` アイテムの `Text` フィールドのダイナミックリンクを変換する例を下記に示します:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
string containsDynamicLinks = home.Fields["text"].Value;
string containsFriendlyLinks = Sitecore.Links.LinkManager.ExpandDynamicLinks(
    containsDynamicLinks, Sitecore.Configuration.Settings.Rendering.SiteResolving);
string finalMarkup = System.Text.RegularExpressions.Regex.Replace(containsFriendlyLinks,
    "([^\s/])~/media", "$1~/media");
```

---

<sup>21</sup> FieldRenderer Web コントロールに関する追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Reference.aspx> から『ブレンディング コンポーネント リファレンス』を参照してください。



## Chapter 6

# シンジケート API

この章では Sitecore の RSS (Really Simple Syndication) 機能に関連した API について解説します。<sup>22</sup>

この章ではまず Sitecore のシンジケート API を説明し、次に ASP.NET のシステム シンジケート API を説明します。

この章には次のセクションがあります。

- Sitecore.Syndication クラス
- ASP.NET シンジケート クラス

---

<sup>22</sup> Sitecore の RSS 機能についての追加情報は、<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Cookbook.aspx> から『プレゼンテーション コンポーネント クックブック』を参照してください。

## 6.1 Sitecore.Syndication クラス

このセクションでは Sitecore の RSS 機能を実装している `Sitecore.Syndication` 名前空間のクラスについて解説します。

### 6.1.1 Sitecore.Syndication.FeedManager

`Sitecore.Syndication.FeedManager.GetFeedUrl()` メソッドを使用して RSS フィードの URL にアクセスすることができます。`Sitecore.Syndication.FeedManager.GetFeedUrl()` メソッドに関する追加情報は、「RSS フィードの URL にアクセスする方法」のセクションを参照してください。

### 6.1.2 Sitecore.Syndication.FeedUtil

`Sitecore.Syndication.FeedUtil` クラスはシンジケートに関連した静的なユーティリティ メソッドを含みます。

`Sitecore.Syndication.FeedUtil.IsConfiguredForFeed()` メソッドは RSS フィードに含めることのできるアイテムには `True` (真) を返します。`Sitecore.Syndication.FeedUtil.IsConfiguredForFeed()` メソッドの使用例は「例: 複数のアイテムの子を配信可能にする」のセクションを参照してください。

`Sitecore.Syndication.FeedUtil.IsFeed()` メソッドはパラメーターがフィード定義アイテムである場合に `True` (真) を返します。

### 6.1.3 Sitecore.Syndication.PublicFeed

`Sitecore.Syndication.PublicFeed` クラスはフィード定義アイテムを使用して RSS フィードを作成します。

`Sitecore.Syndication.PublicFeed` クラスは、Sitecore クエリによって識別されたアイテム、またはフィード定義アイテムで指定されたデータ ソース アイテムの子に識別されたアイテムを配信可能にします。<sup>23</sup>

`Sitecore.Syndication.PublicFeed` クラスをカスタムのフィードを開発するための基本クラスとして使用することができます。クラスをフィード定義アイテムの **[Extensibility]** セクションの **[Type]** フィールドに指定します。フィード定義アイテムの **[Extensibility]** セクションの **[Type]** フィールドに値を指定しなかった場合、Sitecore は `Sitecore.Syndication.PublicFeed` クラスを使用してフィードをフォーマットします。

`Sitecore.Syndication.PublicFeed.FeedItem` プロパティはフィードに関連付けられたフィード定義アイテムを公開します。

`Sitecore.Syndication.PublicFeed.GetSourceItems()` メソッドを上書きし、カスタム ロジックを使用して、配信可能にするアイテムを決めるフィードを実装することができます。

<sup>23</sup> Sitecore クエリについての追加情報は、

<http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Reference.aspx> から『データ定義リファレンス』を参照してください。

Sitecore.Syndication.PublicFeed.GetSourceItems() メソッドを上書きする例については、「例: 複数のアイテムの子を配信可能にする」のセクションを参照してください。

## 例: 複数のアイテムの子を配信可能にする

次の例をもとにカスタムの RSS フィードを実装することができます。この例ではフィード定義アイテムの **Sources** という名前のフィールドで選択されたアイテムのすべての子を配信可能にします。

1. Visual Studio の Web アプリケーション プロジェクトで、次の例をもとにクラスをコンパイルします:

```
namespace Sitecore.Shareddata.Syndication
{
    public class SelectionFeed : Sitecore.Syndication.PublicFeed
    {
        public override System.Collections.Generic.IEnumerable<Sitecore.Data.Items.Item>
            GetSourceItems()
        {
            Sitecore.Data.Fields.MultilistField entries = this.FeedItem.Fields["Sources"];

            if (entries == null || entries.Count < 1)
            {
                return new Sitecore.Data.Items.Item[0];
            }

            Sitecore.Collections.ItemList list = new Sitecore.Collections.ItemList();

            foreach (Sitecore.Data.Items.Item parent in entries.GetItems())
            {
                foreach (Sitecore.Data.Items.Item child in parent.Children)
                {
                    if (Sitecore.Syndication.FeedUtil.IsConfiguredForFeed(child))
                    {
                        list.Add(child);
                    }
                }
            }

            if (list.Count < 1)
            {
                return new Sitecore.Data.Items.Item[0];
            }

            list.Sort(new Sitecore.Data.Comparers.UpdatedComparer());
            return list.ToArray();
        }
    }
}
```

1. テンプレート マネージャー または コンテンツ エディター で、System/Feeds/RSS Feed データテンプレートの [Extensibility] セクションで、**Sources** という名前の **Treelist** フィールドを追加し、このフィールドの [Source] プロパティを [/sitecore/content] アイテムに設定します。
2. コンテンツ エディター のフィード定義アイテムで [Extensibility] セクションの [Type] フィールドにクラスのシグネチャを入力します:

```
Sitecore.Shareddata.Syndication.MultiParentFeed, Assembly
```

3. コンテンツ エディター のフィード定義アイテムで [Extensibility] セクションの [Sources] フィールドで配信可能にする子を持つアイテムを選択します。

## 6.2 ASP.NET シンジケート クラス

次のセクションで説明するクラスを使用して RSS フィードのエントリを操作することができます。

### メモ

`System.ServiceModel.Syndication` 名前空間のクラスを使用するためには、Visual Studio の Web アプリケーション プロジェクトで `System.ServiceModel.Web` アセンブリへの参照を追加します。

`System.ServiceModel.Web` 参照の **Copy Local** プロパティを **False** に設定します。

### 6.2.1 System.ServiceModel.Syndication.SyndicationItem

`System.ServiceModel.Syndication.SyndicationItem` クラスはシンジケート フィードの 1 つのアイテムを表します。`System.ServiceModel.Syndication.SyndicationItem` クラスを使用して RSS フィードのエントリのプロパティを操作することができます。

### 例:シンジケート エントリのタイトルの長さの制限

次の例をもとにカスタム フィードを実装することができます。この例では RSS エントリのタイトルの長さを制限します。

1. Visual Studio の Web アプリケーション プロジェクトで、次の例をもとにクラスを追加します:

```
namespace Sitecore.Sharedsource.Syndication
{
    using System;
    using System.ServiceModel.Syndication;

    public class LimitedTitleFeed : Sitecore.Syndication.PublicFeed
    {
        protected override SyndicationItem RenderItem(Sitecore.Data.Items.Item item)
        {
            SyndicationItem entry = base.RenderItem(item);
            int titleLengthLimit;

            if (this.FeedItem != null
                && Int32.TryParse(this.FeedItem["TitleLengthLimit"], out titleLengthLimit))
            {
                string title = entry.Title.Text.Substring(0, titleLengthLimit - 3);
                title = title.TrimEnd(new[] { ' ', ',', ';' }) + "...";
                entry.Title = new TextSyndicationContent(title);
            }

            return entry;
        }
    }
}
```

2. **テンプレート マネージャー** または **コンテンツ エディター** で `System/Feeds/RSS Feed` データ テンプレートから継承するカスタム フィード データ テンプレートを作成し、**TitleLengthLimit** という名前の整数フィールドを追加します。
3. **コンテンツ エディター** で、そのカスタム フィード データ テンプレートをもとにフィードを作成します。

4. **コンテンツ エディター** で、フィード定義アイテムの **[TitleLengthLimit]** フィールドで、シンジケート エントリのタイトルに許容される最大の文字数を入力します。
5. **コンテンツ エディター** のフィード定義アイテムで **[Extensibility]** セクションの **[Type]** フィールドに .NET タイプのシグネチャを、たとえば次の例に示すように入力します:

```
Sitecore.Sharedsource.Syndication.SelectionFeed, Assembly
```

## 例:シンジケート エントリの分類項目

RSS フォーマットを使用して、シンジケート エントリのそれぞれを複数のトピックの分類項目に関連付けることができます。分類項目のリストを、分類項目定義アイテムを含むフォルダーとして管理することができます。データ テンプレートのそれぞれに選択フィールドを追加することができ、これによって CMS ユーザーが配信可能なアイテムに分類項目定義アイテムをいくつでも選択することができます。分類項目定義アイテムの名前を持つ

`System.ServiceModel.Syndication.SyndicationItem.Categories` リストを作成することによって、各シンジケート エントリに分類項目を含めることができます。

次の例をもとにカスタム フィードを実装することができます。この例ではシンジケート エントリに分類項目を含めます。

1. Visual Studio の Web アプリケーション プロジェクトで、次の例をもとにクラスを追加します:

```
namespace Sitecore.Sharedsource.Syndication
{
    using System.ServiceModel.Syndication;

    public class CategorizedFeed : Sitecore.Syndication.PublicFeed
    {
        protected override SyndicationItem RenderItem(Sitecore.Data.Items.Item item)
        {
            Sitecore.Diagnostics.Assert.ArgumentNotNull(item, "item");
            SyndicationItem entry = base.RenderItem(item);
            Sitecore.Data.Fields.MultilistField entries = item.Fields["Categories"];

            if (entries != null)
            {
                foreach (Sitecore.Data.Items.Item category in entries.GetItems())
                {
                    if (category != null)
                    {
                        entry.Categories.Add(new SyndicationCategory(category.Name));
                    }
                }
            }

            return entry;
        }
    }
}
```

2. **コンテンツ エディター** のフィード定義アイテムで **[Extensibility]** セクションの **[Type]** フィールドに .NET タイプのシグネチャを、たとえば次の例に示すように入力します:

```
Sitecore.Sharedsource.Syndication.CategorizedFeed, Assembly
```

3. **コンテンツ エディター** で分類項目定義アイテムを含むフォルダーを作成します。

4. **テンプレート マネージャー** または **コンテンツ エディター** で、シンジケート エントリに使用したデータ テンプレートに **Categories** という名前の選択フィールドを追加し、**[Categories]** フィールドの **[Source]** プロパティを前のステップで作成した分類項目定義アイテムを含むフォルダーに設定します。
5. **コンテンツ エディター** で、シンジケート エントリの **[Categories]** フィールドで、関連する分類項目を選択します。

#### ヒント

**[Categories]** フィールドを含むデータ テンプレートを作成し、そのデータ テンプレートをシンジケートをサポートするテンプレートのためのベース データ テンプレートとして追加します。

## Chapter 7

# コンテンツ API のトラブルシューティング

この章では Sitecore 開発者がコンテンツ API を使用する際に経験する、よくある問題に関するトラブルシューティングについて説明します。

この章には次のセクションがあります。

- 構成ノードが見つかりません
- オブジェクトの参照がオブジェクトのインスタンスに設定されていません
- アイテムが編集モードにありません
- 現在のユーザーにはこのアイテムへの書き込みアクセス権がありません
- 追加するためのアクセスが必要です

## 7.1 構成ノードが見つかりません

`Sitecore.Configuration.Factory.GetDatabase()` メソッドを使用して存在しないデータベースにアクセスしようとした場合や、`web.config` で構成された名前と（大文字と小文字の区別も含めて）同じデータベース名を入力しなかった場合に、次のようなメッセージがブラウザーに表示される場合があります。

*System.InvalidOperationException: Could not find configuration node*

`Sitecore.Configuration.Factory.GetDatabase()` メソッドに第 1 パラメーターとして渡した値が `web.config` の適切な `/configuration/sitecore/databases/database` 要素の `id` 属性と合致していることを確認します。



## 7.2 オブジェクトの参照がオブジェクトのインスタンスに設定されていません

存在しないアイテムやパブリッシュされていないアイテムにアクセスしようとした場合や、コンテキスト ユーザーに `item:read` セキュリティ アクセス権がないアイテムにアクセスしようとした場合に、次のようなメッセージがブラウザーに表示される場合があります。

*System.NullReferenceException: Object reference not set to an instance of an object*

アイテムに指定された ID またはパスが正しいこと、コードが正しいデータベースにアクセスしていること、アイテムがデータベースに存在すること、コンテキスト ユーザーにアイテムへの `item:read` アクセス権があることを確認します。アイテムまたはそのデータテンプレートをパブリッシュする必要がある場合があります。またはセキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用する必要がある場合があります。セキュリティ ユーザー スイッチャーまたはセキュリティ無効化を使用するための追加情報は、「アイテムへのアクセス権の解決方法」のセクションを参照してください。

## 7.3 アイテムが編集モードにありません

アイテムが編集モードでないときにそれを更新しようとした場合に次のようなメッセージがブラウザーに表示される場合があります。

*Sitecore.Exceptions.EditingNotAllowedException: Item is not in editing mode*

アイテムを編集モードにするための情報は「アイテムを編集モードにする方法」のセクションを参照してください。

## 7.4 現在のユーザーにはこのアイテムへの書き込みアクセス権がありません

コンテキストユーザーが `item:write` セキュリティ アクセス権のないアイテムを更新しようとした場合に、次のようなメッセージがブラウザーに表示される場合があります。

*System.UnauthorizedAccessException: The current user does not have write access to this item*

コンテキストユーザーがアイテムへの `item:write` アクセス権を持っていることを確認して下さい。セキュリティユーザー スイッチャーまたはセキュリティ無効化を使用する必要がある場合があります。セキュリティユーザー スイッチャーまたはセキュリティ無効化を使用するための追加情報は、「アイテムへのアクセス権の解決方法」のセクションを参照してください。

## 7.5 追加するためのアクセスが必要です

コンテキストユーザーが `item:create` セキュリティ アクセス権のないアイテムの下にアイテムを追加しようとした場合に、次のようなメッセージがブラウザーに表示される場合があります。

*Sitecore.Exceptions.AccessDeniedException: AddFromTemplate - Add access required*

コンテキストユーザーが親アイテムへの `item:create` アクセス権を持っていることを確認して下さい。セキュリティユーザー スイッチャーまたはセキュリティ無効化を使用する必要がある場合があります。セキュリティユーザー スイッチャーまたはセキュリティ無効化を使用するための追加情報は、「アイテムへのアクセス権の解決方法」のセクションを参照してください。