



# Sitecore CMS 6.2

# クライアント構成

# クックブック

CMS アーキテクトとデベロッパーのための機能とヒントとテクニック

## 目次

Chapter 1	イントロダクション.....	6
Chapter 2	共通の手順 .....	7
2.1	共通の手順 .....	8
2.1.1	Sitecore デスクトップでのデータベースの選択方法 .....	8
2.1.2	スタンダード フィールドを表示または非表示にする.....	8
2.1.3	RAW 値を表示または非表示にする .....	9
2.1.4	デベロッパー タブを表示または非表示にする .....	9
2.1.5	アイテムの ID またはパスを Windows クリップボードにコピーする方法 .....	9
2.1.6	クラス シグネチャーを入力する方法 .....	10
Chapter 3	データ テンプレートとアイテム.....	11
3.1	アイテムの外見 .....	12
3.1.1	アイコン .....	12
アイテムへのアイコンの設定方法 .....	12	
データテンプレートに基づくすべてのアイテムにデフォルト アイコンを設定する方法 .....	13	
すべてのアイテムへのデフォルト アイコンの設定方法.....	13	
アイコン選択メニューの構成方法 .....	13	
3.1.2	非表示のアイテム .....	13
非表示アイテムを表示または非表示にする .....	14	
アイテムを表示または非表示にする .....	14	
3.1.3	保護されたアイテム.....	14
アイテムを保護または非保護にする .....	14	
3.1.4	アイテムのスタイル .....	15
コンテンツ ツリーでのアイテム名のスタイルの構成方法 .....	15	
3.1.5	表示名 .....	16
アイテムへの表示名の設定方法 .....	17	
3.1.6	コンテキスト ヘルプ .....	17
アイテムのコンテキスト ヘルプの設定方法 .....	17	
3.2	データ テンプレートのセクション .....	19
3.2.1	データ テンプレートのセクションのアイコン .....	19
3.2.2	データ テンプレートのセクションのアイコン設定方法 .....	20
3.2.3	データ テンプレートのセクションのソート順.....	20
データ テンプレートのセクションのソート方法 .....	20	
データ テンプレート セクション定義アイテムのソート順プロパティの設定方法.....	20	
3.3	データ テンプレート フィールド .....	22

3.3.1	データ テンプレート フィールドのヘッダとコンテキスト ヘルプ .....	22
	データ テンプレート フィールドのタイトルの設定方法 .....	22
	データ テンプレート フィールドにコンテキスト ヘルプを設定する方法 .....	22
3.3.2	データ テンプレート フィールドのスタイルの設定方法 .....	23
3.3.3	データ テンプレートのフィールドのソート順 .....	23
	データ テンプレートのフィールドのソート方法 .....	23
	データ テンプレート フィールド定義アイテムのソート順プロパティの設定方法 .....	23
3.3.4	リッチ テキスト エディター (RTE) の構成 .....	24
	RTE プロファイル定義アイテム リファレンス .....	25
	RTE プロファイルの構成方法 .....	25
	RTE プロファイルへのパスの決定方法 .....	26
	RTE テンプレート フィールド定義アイテムのプロファイルを設定する方法 .....	26
	RTE プロファイルの機能の追加方法 .....	27
	RTE フィールドで使われる CSS を設定する方法 .....	27
	RTE フィールドで表示される CSS スタイルを制限する方法 .....	27
	RTE HTML エlement タイプ ドロップダウン メニューを構成する .....	28
	RTE プロファイルで Snippet を有効化する方法 .....	28
	RTE プロファイルに Snippet を追加する方法 .....	29
	Enter キーで挿入されたマークアップを制御する方法 .....	29
	RTE HTML タブをあるユーザーまたはすべてのユーザーに対して無効にする方法 .....	30
3.4	データ テンプレートのカスタム ユーザー インターフェイス .....	31
3.4.1	IFrame フィールド タイプ .....	31
	IFrame データ テンプレート フィールドの作成方法 .....	31
3.4.2	アイテム エディター .....	32
	アイテム エディターの構成方法 .....	33
	カスタムのアイテム エディターの作成方法 .....	33
3.5	挿入オプション .....	35
Chapter 4	データ 検証 .....	36
4.1	検証の概要 .....	37
4.2	検証の種類 .....	39
4.2.1	データ テンプレートのフィールド検証 .....	39
4.2.2	フィールド タイプの検証 .....	39
4.2.3	アイテムの検証 .....	39
4.2.4	標準の検証 .....	39
4.2.5	グローバル アイテムの検証 .....	39
4.3	検証のエラーレベル .....	40
4.4	検証ルールの構成 .....	41

4.4.1	クイック アクション バーの検証ルールの構成方法 .....	41
4.4.2	バリデート ボタンの検証ルールの構成方法 .....	41
4.4.3	検証 バーの検証ルールの構成方法 .....	41
4.4.4	ワークフローの検証ルールの構成方法 .....	41
4.4.5	ワークフロー コマンドの作成または状態検証アクションの作成方法 .....	42
4.4.6	データ テンプレート フィールド タイプのすべてのインスタンスへの検証ルールの構成方法 .....	42
4.4.7	すべてのアイテムへの検証ルールの構成方法 .....	43
4.4.8	個々のアイテムまたは特定のデータ テンプレートに基づくすべてのアイテムへの検証ルールの構成方法 .....	43
4.4.9	特定のデータ テンプレート フィールドへの検証ルールの構成方法 .....	43
4.4.10	デフォルトのアイテム バリデーター .....	44
4.4.11	デフォルトのフィールド バリデーター .....	44
4.5	バリデーターの登録 .....	46
4.5.1	バリデーターの登録方法 .....	46
4.5.2	正規表現フィールド バリデーターの登録方法 .....	46
4.5.3	整数フィールド バリデーターの登録方法 .....	47
4.5.4	整数範囲フィールド バリデーターの登録方法 .....	47
4.5.5	最大長フィールド バリデーターの登録方法 .....	48
4.5.6	特定のアイテムと特定のテンプレートに基づくアイテムのためのバリデーターの登録方法 .....	48
4.5.7	デフォルトの検証 ルールを無効にする方法 .....	48
4.5.8	バリデーターのデフォルトのエラー レベルを上書きする方法 .....	49
4.6	カスタム バリデーター .....	50
4.6.1	カスタム バリデーターの実装方法 .....	50
4.6.2	カスタム バリデーターの実装方法 .....	50
4.7	検証 アクション .....	52
4.7.1	検証 アクションの作成方法 .....	52
4.7.2	検証 アクションの使用方法 .....	53
Chapter 5	ページ エディター .....	54
5.1	ページ エディターの概要 .....	55
5.1.1	ページ エディターのデザイン ウィンドウ .....	55
5.2	プレースホルダーの設定 .....	56
5.2.1	プレースホルダー設定定義アイテムの作成方法 .....	57
5.2.2	データ テンプレートまたは個々のアイテムへのプレースホルダー設定の構成方法 .....	57
5.3	ページ エディターのモード .....	59
5.3.1	ページ エディターのモードの決定方法 .....	59
5.3.2	ページ エディターのモード .....	60
5.4	エディット フレーム .....	61
5.4.1	エディット フレーム メニュー コマンドの実装方法 .....	62

5.4.2	エディット フレーム メニューの定義方法 .....	62
5.4.3	レイアウトまたはサブレイアウトにエディット フレームを挿入する方法 .....	63
5.4.4	XSL レンダリングにエディット フレームを挿入する方法 .....	64
5.4.5	エディット フレーム プロパティ .....	64
5.5	フィールド エディター .....	65
5.5.1	デフォルトのフィールド エディターの使用方法 .....	66
5.5.2	カスタムのフィールド エディターの実装方法 .....	66
Chapter 6	セキュリティの構成機能 .....	68
6.1	Sitecore のクライアント セキュリティ ロール .....	69
6.2	セキュリティのプリセット .....	72
6.2.1	セキュリティのプリセットの作成方法 .....	72
6.2.2	セキュリティのプリセットの適用方法 .....	73
6.3	データ テンプレートのフィールドのセキュリティ .....	75
	データ テンプレートのフィールド セキュリティの構成方法 .....	75
Chapter 7	Sitecore クライアント RSS フィード .....	76
7.1	Sitecore クライアント RSS フィードの概要 .....	77
7.1.1	ワークフロー フィード .....	77
7.1.2	ワークフロー状態フィード .....	77
7.1.3	アイテム更新フィード .....	77
7.2	Sitecore クライアント RSS フィードの構成 .....	78
7.2.1	ClientFeeds.MaximumItemsInFeed 設定 .....	78
7.2.2	ClientFeeds.ItemExpiration 設定 .....	78

## Chapter 1

# イントロダクション

この Cookbook は CMS アーキテクトとデベロッパーが Sitecore クライアントのユーザー インターフェースのユーザビリティを最適化するための情報、ヒント、テクニックを提供します。<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 — 共通の手順
- Chapter 3 — データ テンプレートとアイテム
- Chapter 4 — データ検証
- Chapter 5 — ページ エディター
- Chapter 6 — セキュリティの構成機能
- Chapter 7 — Sitecore クライアント RSS フィード

---

<sup>1</sup> クライアントのハードウェア要件については <http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%206/Installation.aspx> から『インストールガイド』を参照してください。Internet Explorer の構成については <http://sdn.sitecore.net/Reference/References%20in%20Japanese/IE%20Configuration%20Reference.aspx> から『Internet Explorer 構成リファレンス』を参照してください。

## Chapter 2

### 共通の手順

この章では、この文書と他の文書で記述されている手順で必要となる共通の手順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。

- 共通の手順

## 2.1 共通の手順

このセクションでは、この文書と他の文書で記述されている手順で必要な共通の手順について説明します。

### 2.1.1 Sitecore デスクトップでのデータベースの選択方法

Sitecore デスクトップでデータベースを選択する方法：

1. ブラウザーから <http://localhost/sitecore> などの Sitecore ログイン ページにアクセスします。
2. Sitecore ログイン ページで Sitecore のユーザー名とパスワードを入力します。
3. Sitecore ログイン ページで **[Options]** をクリックし、次に **[Desktop]** をダブルクリックします。ブラウザーで Sitecore デスクトップが表示されます。
4. Sitecore デスクトップの右下の隅で、データベース アイコンをクリックし、次にデータベース名をクリックします。Sitecore デスクトップが再表示され、開いているアプリケーションが閉じられます。Sitecore デスクトップから **[コンテンツ エディター]** または他のアプリケーションを開くと、選択したデータベースにアクセスします。これはログアウトするか、または他のデータベースを選択するまで同様です。
5. Sitecore デスクトップの選択されたデータベースで、左下の隅で、**Sitecore** をクリックし、次に **[コンテンツ エディター]** をクリックします。Sitecore デスクトップの選択されたデータベースで、**[コンテンツ エディター]** が表示され、選択されたデータベースを扱うことができます。

#### 重要

意図しないデータベースの変更を行ってしまう可能性を避けるため、他のデータベースを扱った後では必ず**マスター** データベースを選択します。

#### メモ

特に指定されていない限り、この文書および他の Sitecore の文書の手順の中では**マスター** データベースが選択されているものとします。

### 2.1.2 スタンダード フィールドを表示または非表示にする

スタンダード フィールドを表示または非表示にする方法：

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 **[表示]** タブをクリックします。
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**の **[表示]** タブの、**[表示]** グループで、**[スタンダード フィールド]** を有効または無効にします。

**メモ**

スタンダード フィールドの表示または非表示は Sitecore クライアントのパフォーマンスに影響をあたえます。

### 2.1.3 RAW 値を表示または非表示にする

フィールドの RAW 値を表示することでフィールドのテキスト値を調べることができます。たとえば、フィールドの RAW 値を参照して、Sitecore がある要素をストアする属性を決めることができます。

**メモ**

可能な限り、フィールドの RAW 値に直接アクセスするよりも、フィールドの値を抽象化しているコンストラクトを使用してください。

RAW 値を表示または非表示にする方法：

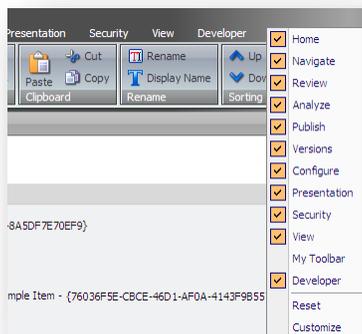
1. テンプレート マネージャーまたはコンテンツ エディターで [表示] タブをクリックします。
2. テンプレート マネージャーまたはコンテンツ エディターの [表示] タブの、[表示] グループで [RAW 値] を有効または無効にします。

### 2.1.4 [デベロッパー] タブを表示または非表示にする

[デベロッパー] タブは Sitecore ソリューションを作成する開発者に便利な機能を提供します。

[デベロッパー] タブを表示または非表示にする方法：

- コンテンツ エディターでタブを右クリックし、次に [デベロッパー] を有効または無効にします。



### 2.1.5 アイテムの ID またはパスを Windows クリップボードにコピーする方法

アイテムのパスを Windows クリップボードにコピーする方法：

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 [アイテム] を選択し、次に [Content] タブをクリックします。
2. [表示] セクションで [ID] または [アイテム Path] の値でマウスをクリックまたはドラッグします。パスを [Windows クリップボード] にコピーするには [CTRL-C] を押すか、選択したテキストを右クリックし、次に[コピー] を選択します。

#### ヒント

[**デベロッパー**] タブの [表示] グループにある [ID] と [Path] コマンドを使って ID またはアイテムへのパスをコピーすることもできます。[**デベロッパー**] タブに関する詳細は「[**デベロッパー**] タブを表示または非表示にする」のセクションを参照してください。

### 2.1.6 クラス シグネチャーを入力する方法

クラスシグネチャーは .NET アセンブリー (.dll ファイル) におけるクラスを識別します。

クラスシグネチャーを入力する方法：

1. 次のプロトタイプを入力します：

```
Namespace.Class, Assembly
```

2. Namespace をクラスを含むネームスペースで置換します。
3. Class をクラスの名前で置換します。
4. Assembly をクラスを含むアセンブリーの名前 (.dll 拡張子なしで) で置換します。

## Chapter 3

# データ テンプレートとアイテム

この章ではデータ テンプレートとアイテムを構成するヒントとテクニックについて説明します。それらはアイテムの制御、セクション、フィールドの外観、カスタム アイテム エディター、挿入オプションについてです。

この章には次のセクションがあります。

- アイテムの見た目
- データ テンプレートのセクション
- データ テンプレート フィールド
- データ テンプレートのカスタム ユーザー インターフェース
- 挿入オプション

## 3.1 アイテムの見たい目

次の機能を使ってコンテンツ ツリーでのアイテムの見たい目を制御することができます。標準テンプレートのフィールドが各アイテムのアイコンを定義します。

### 3.1.1 アイコン

コンテンツ エディターのコンテンツ ツリーを含む Sitecore ユーザー インターフェースは各アイテムのとなりにアイコンを表示します。

開発者はアイコンを次のために使います：

- 特定のアイテムを視覚的に差別化する
- 特定のテンプレートに基づくアイテムを視覚的に差別化する

アイテムがアイコンを指定していない場合、コンテンツ ツリーはアイテムに関連付けられているデータ テンプレートに指定されているアイコンを表示します。そのデータ テンプレートがアイコンを指定していない場合、コンテンツツリーは `web.config` の `/configuration/sitecore/settings/setting` エレメントの `value` 属性で指定された `name DefaultIcon` のアイコンを表示します。

#### メモ

Sitecore はテーマをサポートします。Sitecore は `/sitecore/shell/themes/standard` ディレクトリに相対して相対アイコンパスを解釈します。たとえば、`applications/16x16/star_yellow.png` という値は `/sitecore/shell/themes/standard/applications/16x16/star_yellow.png` と同等です。

#### メモ

アイコンはイメージを指す任意の URL にすることが可能です。

## アイテムへのアイコンの設定方法

アイテムにアイコンを設定する方法：

1. **コンテンツ エディター**または**テンプレート マネージャー**で [アイテム] を選択します。
2. **コンテンツ エディター**または**テンプレート マネージャー**で [設定] タブをクリックします。
3. **コンテンツ エディター**または**テンプレート マネージャー**で [設定] タブの[アピアランス (外観と動作)] グループで [アイコンを設定] コマンドをクリックします。[アイコン選択] メニュー が表示されます。
4. [アイコン選択メニュー] でアイコンを選択するか、または [他のアイコン] をクリックして [アイコン選択] ダイアログを使ってアイコンを選択します。

## メモ

[アイコン選択] ダイアログはひとつのディレクトリにある多数のイメージを表示する場合にはロードに時間がかかる場合があります。

## ヒント

[アイコン選択] ダイアログへのアクセス方法のショートカットとしては、[編集ウィンドウ] で [Content] タブをクリックし、[アイテム タイトル バー] のアイコンをクリックします。

## 重要

個々のアイテムにアイコンを設定するより、データ テンプレートにアイコンを設定します。データ テンプレートのアイコンはテンプレートに基づくアイテム（標準値アイテムを含む）のデフォルト アイコンです。

## データ テンプレートに基づくすべてのアイテムにデフォルト アイコンを設定する方法

データ テンプレートに基づくすべてのアイテムにデフォルト アイコンを設定する方法：

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 [データ テンプレート定義アイテム] を選択します。
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、[データ テンプレート定義アイテム] でアイコンを設定します。アイコンを設定する方法は「アイテムへのアイコンの設定方法」のセクションを参照してください。

## 重要

アイコンをデータ テンプレートの標準値アイテムでなく、各データ テンプレートの定義アイテムで設定します。アイコンをデータ テンプレートの標準値アイテムにのみ設定した場合は、そのアイコンはデータテンプレート自体には適用されません。コンテンツ ツリーのデータ テンプレートに表示されたアイコンはデフォルト アイコンのまま残ります。

## すべてのアイテムへのデフォルトアイコンの設定方法

アイテム、データ テンプレート、データ テンプレートが関連付けられている標準値アイテムがアイコンを定義していないような、すべてのアイテムにデフォルト アイコンを設定するには、web.config ファイルで、DefaultIcon という名前の /configuration/sitecore/settings/setting で value 属性をデフォルトのアイコン パスに設定します。

## [アイコン選択] メニューの構成方法

[アイコン選択] メニューに表示されるアイコンを構成するには、/App\_Config/Icons.config ファイルを編集します。各コレクション エLEMENTが行を定義し、アイコン ファイルへの絶対または相対パスのパイプ区切りのリストを含んでいます。

### 3.1.2 非表示のアイテム

非表示のアイテムは非表示アイテムを表示する権限のないユーザーのコンテンツ ツリー、または非表示アイテムを表示しないことを選択したユーザーのコンテンツ ツリーに表示されません。

## 非表示アイテムを表示または非表示にする

非表示アイテムを表示するには、ユーザーは管理者または Sitecore Client Developing ロールまたは Sitecore Client Maintaining ロールのメンバーである必要があります。

非表示アイテムを表示する方法：

1. **コンテンツ エディター**または**テンプレート マネージャー**で **[ビュー]** タブをクリックします。
2. **コンテンツ エディター**または**テンプレート マネージャー**の **[ビュー]** タブで、**[ビュー]** グループで、**[非表示アイテム]** を有効または無効にします。

## アイテムを表示または非表示にする

アイテムを表示または非表示にするには、ユーザーは管理者または Sitecore Client Configuring ロールのメンバーである必要があります。

アイテム表示または非表示にする方法：

1. **コンテンツ エディター**または**テンプレート マネージャー**で **[アイテム]** を選択します。
2. **コンテンツ エディター**または**テンプレート マネージャー**で **[設定]** タブをクリックします。
3. **コンテンツ エディター**または**テンプレート マネージャー**で**[設定]** タブの **[属性]** グループで **[アイテムを非表示に設定]** をクリックしてアイテムを非表示にするか、または **[アイテムを表示に設定]** をクリックしてアイテムを表示させます。

### 3.1.3 保護されたアイテム

保護されたアイテムは Sitecore ユーザー インターフェースからは編集できません。

## アイテムを保護または保護を解除する

アイテムを保護または保護を解除するには、ユーザーは管理者または **Sitecore Client Configuring** ロールのメンバーである必要があります。

アイテム保護または保護を解除する方法：

1. **コンテンツ エディター**または**テンプレート マネージャー**でアイテムを選択します。
2. **コンテンツ エディター**または**テンプレート マネージャー**で **[設定]** タブをクリックします。
3. **コンテンツ エディター**または**テンプレート マネージャー**で **[設定]** タブの **[属性]** グループで **[アイテムの保護]** をクリックしてアイテムを保護するか、または **[アイテム保護の解除]** をクリックしてアイテムを非保護にします。

### 3.1.4 アイテムのスタイル

アイテムのスタイルはコンテンツ ツリーで各アイテムが表示する名前の見た目を制御します。開発者は特定のアイテムまたは特定のテンプレートに基づくアイテムを強調したり視覚的に差別化するためにアイテムのスタイルを使用することができます。たとえば、ある種のシステム アイテムはデフォルトで緑色の文字で表示され、プロキシ アイテムはグレーアウトされて表示されます。

個々のアイテムにスタイルを設定するには、開発者はアイテムにアイテムのスタイルを適用します。開発者はアイテムのスタイルをデータ テンプレートに適用して、そのテンプレートに基づくすべてのアイテムにスタイルを設定することができます。

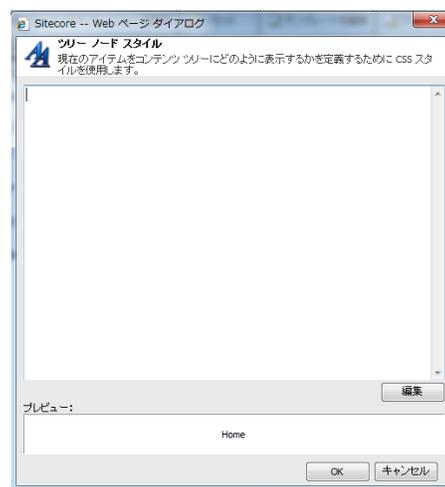
#### コンテンツ ツリーでのアイテム名のスタイルの構成方法

ツリー ノード スタイルを使ってコンテンツ ツリーでのアイテム名のスタイルを構成することができます。

アイテム名のツリー ノード スタイルを構成する方法：

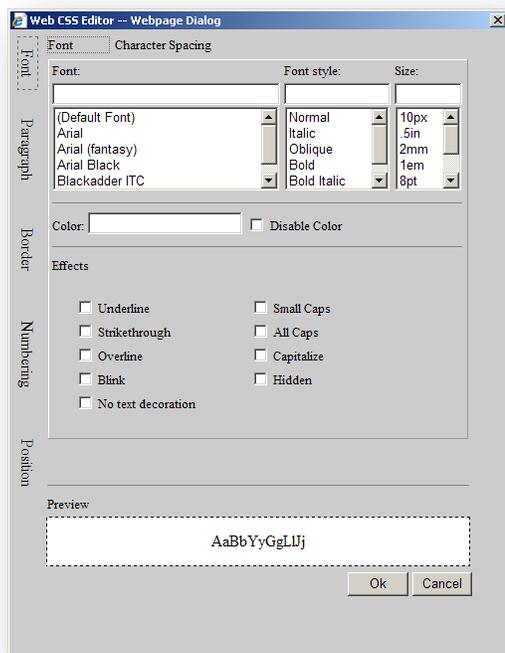
1. テンプレート マネージャーまたはコンテンツ エディターで標準値アイテムまたは個々のアイテムを選択します。
2. テンプレート マネージャーまたはコンテンツ エディターで [設定] タブをクリックします。

テンプレート マネージャーまたはコンテンツ エディターの [設定] タブで、[アピランス (外観と動作)] セクションで [ツリー ノード スタイルを設定] をクリックすると、[ツリー ノード スタイル] ダイアログが表示されます。



3. [ツリー ノード スタイル] ダイアログの [スタイル ダイアログ] で、CSS スタイルを入力するか、または [編集] をクリックし、[Web CSS Editor] ダイアログを使用します。たとえば、COLOR:red; と入力してフォントの色を赤

に設定します。



## メモ

Sitecore は `COLOR:green` をシステム アイテムに、`COLOR:gray` をプロキシ シャドー アイテムに使用します。これらのツリー ノード スタイルを他の目的に使用しないでください。

### 3.1.5 表示名

アイテムの表示名は定義されるとコンテンツ ツリーなどのユーザー インター フェースにアイテム名に代わって表示されます。<sup>2</sup>

開発者は次のために表示名を使用することができます：

- コンテンツ アイテムのデフォルト URL に影響を及ぼすことなく、コンテンツ ツリーでわかりやすいアイテムの名前を表示する
- アイテム名に使用できない文字を使った名前をコンテンツ ツリーに表示する

## 注意

表示名はアイテム名で使用できない文字を表示させる必要がある場合など、限られた場合に使用してください。

<sup>2</sup> 「URL の表示名の使用」については

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Dynamic%20Links.aspx> から『動的リンクガイド』を参照してください。

## アイテムへの表示名の設定方法

アイテムに表示名を設定する方法：

1. **コンテンツ エディター**または**テンプレート マネージャー**でアイテムを選択します。
2. **コンテンツ エディター**または**テンプレート マネージャー**で **[ホーム]** タブをクリックします。
3. **コンテンツ エディター**または**テンプレート マネージャー**で **[ホーム]** タブの **[名前の変更]** グループで **[表示名]** をクリックします。表示名のダイアログが表示されます。
4. 表示されたダイアログ ボックスにアイテムの表示名を入力します。

### 3.1.6 コンテキスト ヘルプ

編集インターフェースは選択されたアイテムとそのデータ テンプレートに関する役立つ情報を表示することができます。

開発者は次のためにコンテキストヘルプを使用することができます：

- 特定のデータ テンプレートに基づくアイテムを扱うユーザーに役立つ情報を提供する
- 特定のアイテムを扱うユーザーに役立つ情報を提供する

アイテム タイトル バーはアイテムのヘルプ プロパティまたはデータ テンプレートで定義された短い説明を表示します。コンテンツ ツリーでマウスをアイテムの上に移動したとき、Sitecore はアイテムまたはそのテンプレートの長い説明をツールチップとして表示します。

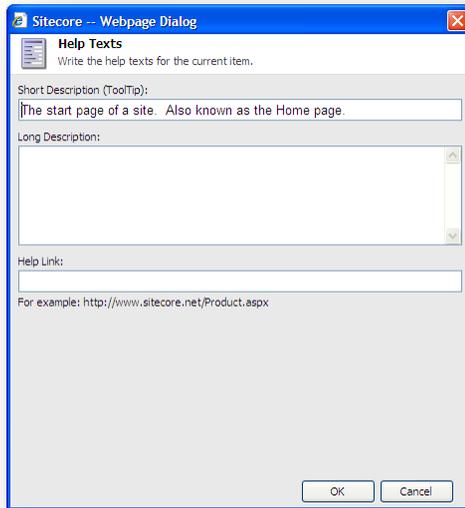
開発者がアイテムのヘルプ プロパティを定義していない場合、Sitecore はアイテムのデータ テンプレートに定義されたヘルプ プロパティを使用します。

## アイテムのコンテキスト ヘルプの設定方法

コンテキスト ヘルプをデータ テンプレートまたは個々のアイテムに設定する方法：

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**で標準値アイテムまたは個々のアイテムを選択します。
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 **[設定]** タブをクリックします。

3. テンプレート マネージャーまたはコンテンツ エディターの [設定] タブの [アピランス (外観と動作)] グループで [ヘルプを設定] をクリックすると、[ヘルプ テキスト ダイアログ] ボックスが表示されます。



4. [概要 (ツールチップ)] フィールドで、アイテムのタイプまたは個々のアイテムの短い説明を入力します。
5. [説明] フィールドで、アイテムのタイプまたは個々のアイテムの長い説明を入力します。

## 3.2 データ テンプレートのセクション

各データ テンプレートにはひとつまたは複数のセクションがあり、各セクションには 1 つまたは複数のフィールドがあります。開発者はデータ テンプレートのセクションを使用して、フィールドを関連グループに整理することができます。フィールドをセクションに整理することで、巨大な単一のデータ エントリー フォームでなく、使わないセクションはユーザーが折り畳むことができるようになります。開発者はデータ テンプレート セクションを使用してその中にフィールドのグループを含めることにより、複数のデータ テンプレートで再利用することができます。

多くのセクションをもつデータ テンプレートや多くのフィールドをもつセクションは、ユーザビリティやパフォーマンスに悪影響を与える場合があります。パフォーマンスを向上させるには多くのリッチテキスト エディター フィールドをもつデータ テンプレートの作成を避けることが非常に重要です。代わりに、開発者は他のフィールド タイプを使うか、別のデータ テンプレートに関連付けられたアイテムの階層を使ったデータを使用することができます。

論理的に整理されたデータ テンプレートのセクションとフィールドはユーザーが更新したいフィールドを見つけるのに役立ちます。プレゼンテーション コンポーネントがページビューでフィールド値をレンダリングするのと同じ順序でセクションとフィールドを並べることを検討してください。または、最も重要か最も頻繁に更新されるセクションとフィールドをデータ テンプレートの最上部に配置するようにします。

適切なデータ テンプレート セクションとフィールド名の使用はユーザーが更新したいフィールドを見つけるのに役立ちます。たとえば、ニュース記事のためのデータ テンプレートが著者や日付などのわずかな数のフィールドのみを定義している場合、それらのフィールドをニュースという名前のセクションに配置し、このセクション名をニュース記事に関係のないフィールドのセクション名に使わないようにします。“データ” などのような汎用的なセクション名は、テンプレート上に数値フィールドが含まれていると解釈されない場合があります。

データ テンプレートとそのベース テンプレートのいずれかが共通の名前のセクションを定義している場合、コンテンツ エディターはそのセクションのすべてのフィールドを単一のビジュアル セクションとしてレンダリングします。セクションとフィールドのソート順プロパティがテンプレートの他のセクションに対する相対的な順序を制御します。

### 3.2.1 データテンプレートのセクションのアイコン

コンテンツ エディターでデータ テンプレート セクションのヘッダーには各データ テンプレート セクションに関連付けられたアイコンが表示されます。開発者がデータ テンプレート セクションのアイコンを指定することで、ユーザーは更新したいフィールドを簡単に見つけることができます。

#### メモ

データ テンプレートのセクション定義アイテムがアイコンを指定していない場合、Sitecore は `web.config` の `/configuration/sitecore/settings/setting` エレメントの `value` 属性で指定された `DefaultIcon` という名前のアイコンを表示します。

### 3.2.2 データ テンプレートのセクションのアイコンの設定方法

データ テンプレートのセクションにアイコンを設定する方法：

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**でデータ テンプレート セクション定義アイテムを選択します。
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**の、データ テンプレート セクション定義アイテムでアイコンを設定します。アイテムにアイコンを設定する方法は「アイテムへのアイコンの設定方法」のセクションを参照してください。

### 3.2.3 データ テンプレートのセクションのソート順

テンプレート ビルダーを使うか、またはデータ テンプレート セクション定義アイテムのソート順プロパティを設定することにより、データ テンプレートのセクションを整理することができます。

#### メモ

データ テンプレートとそのベース テンプレートが別のデータ テンプレート セクションを定義している時にセクションの順序を制御するには、データ テンプレート セクション定義アイテムのソート順プロパティを設定します。

### データ テンプレートのセクションのソート方法

データ テンプレートのセクションをソートする方法：

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**でデータ テンプレート定義アイテムを選択します。
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**の、**[編集]** ウィンドウで **[ビルダー]** タブをクリックします。**[テンプレート ビルダー]** が表示されます。
3. **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 **[ビルダー オプション]** タブをクリックします。
4. **テンプレート ビルダー**で、セクション名をクリックしてセクションを選択します。
5. **テンプレート ビルダー**の **[フィールドの並べ替え]** グループで **[上へ移動]**、**[下へ移動]**、**[最初に移動]**、**[最後に移動]** をクリックしデータ テンプレートの他のセクションとの相対的な位置をソートします。

### データ テンプレート セクション定義アイテムのソート順プロパティの設定方法

データ テンプレート セクション定義アイテムのソート順プロパティを設定する方法：

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**でデータ テンプレート セクション定義アイテムを選択します。
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 **[スタンダード フィールド]** を表示します。

スタンダード フィールドを表示する方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。

3. **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**の [アピランス(外観と動作)] セクションの[並び順] フィールドで、数値を入力します。
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**でスタンダード フィールドを非表示にします。  
スタンダード フィールドを非表示にする方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。

## 3.3 データ テンプレート フィールド

このセクションで説明されている機能を使って、データ テンプレート フィールドのユーザビリティを最適化することができます。

### 3.3.1 データ テンプレート フィールドのヘッダーとコンテキスト ヘルプ

コンテンツ エディターは各フィールドの上にフィールド ヘッダーを表示します。開発者はフィールド値を取り扱うユーザーにフィールド ヘッダーを使って文脈に応じた指示と情報を提供します。

各フィールド ヘッダーには下記のエレメントが含まれます：

- フィールド タイトル。これはフィールド名のデフォルトとなります。
- フィールドが標準値を含んでいる場合、[standard value] トークン。
- フィールドがすべての言語のすべてのバージョンで共有される値を含んでいる場合、[shared] トークン。
- フィールドがバージョン管理されていないが、言語によって異なる値を含んでいる可能性がある場合、[unversioned] トークン。
- フィールド タイプに適切なフィールド編集コントロール。

ユーザーがマウスをフィールド タイトルの上に移動させたとき、フィールドの長い説明が表示されます。

ヘルプ リンクを定義するフィールドには、フィールド ヘッダーのタイトルと短い説明が指定された URL にリンクします。

### データ テンプレート フィールドのタイトルの設定方法

データ テンプレート フィールドにタイトルを設定し、コンテンツ エディターのフィールド ラベルをフィールド定義アイテムの名前と異なるものにする方法：

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**でデータ テンプレート フィールド定義アイテムを選択します。
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**の **[Data]** セクションの **[Title]** フィールドで、フィールドのタイトルを入力します。

### データ テンプレート フィールドにコンテキスト ヘルプを設定する方法

コンテキスト ヘルプをデータ テンプレート フィールドに設定する方法：

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**でデータ テンプレート フィールド定義アイテムを選択します。
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 **[設定]** タブをクリックします。
3. **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**の **[アピアランス (外観と動作)]** グループで **[ヘルプを設定]** をクリックすると、**[ヘルプ テキスト]** ダイアログ ボックスが表示されます。

4. **[ヘルプ テキスト]** ダイアログ ボックスの **[概要 (ツールティップ)]** フィールドで、フィールドの短い説明を入力します。
5. **[説明]** フィールドで、フィールドの長い説明を入力します。
6. **[ヘルプ リンク]** フィールドで、フィールドに関する役立つ情報を含むリソースの URL を入力します。

### 3.3.2 データ テンプレート フィールドのスタイルの設定方法

データ テンプレート フィールドにスタイルを設定する方法 :

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**でデータ テンプレート フィールド定義アイテムを選択します。
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 **[設定]** タブをクリックします。
3. **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**の **[アピアランス (外観と動作)]** グループで **[ツリー ノード スタイルを設定]** をクリックすると、**[ツリー ノード スタイル]** ダイアログ ボックスが表示されます。
4. **[ツリー ノード スタイル]** ダイアログ ボックスで、CSS スタイルを入力するか、または **[編集]** ボタンをクリックし CSS スタイル ウィザードを使用します。たとえば、`HEIGHT:600px;` と入力してリッチ テキスト エディターの高さを設定したり、`FONT-WEIGHT:bold;` と入力して単一行テキスト フィールドのテキスト スタイルを設定します。

### 3.3.3 データ テンプレートのフィールドのソート順

データ テンプレート フィールド定義アイテムのソート順プロパティを明示的に設定し、データ テンプレートとそのベーステンプレートが同じデータテンプレートセクションを定義しているときのフィールドの順序を制御します。

#### データテンプレートのフィールドのソート方法

データテンプレートフィールドをソートする方法 :

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**でデータテンプレート定義アイテムを選択します。
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、**[編集]** ウィンドウで **[ビルダー]** をクリックします。**[テンプレート ビルダー]** が表示されます。
3. **テンプレート ビルダー**で、フィールド名をクリックしてフィールドを選択します。
4. **テンプレートビルダー**の **[フィールドの並べ替え]** グループで **[上へ移動]**、**[下へ移動]**、**[最初に移動]**、**[最後に移動]** をクリックし、セクションの他のフィールドとの相対的なフィールドの位置をソートします。

#### データ テンプレート フィールド定義アイテムのソート順プロパティの設定方法

データ テンプレート フィールドのソート順プロパティを設定する方法 :

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**でデータ テンプレート フィールド定義アイテムを編集します。
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**でスタンダード フィールドを表示します。  
スタンダード フィールドを表示する方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。
3. **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**の[**アピランス(外観と動作)**] セクションの[**並び順**] フィールドで、数値を入力します。
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**でスタンダード フィールドを非表示にします。  
スタンダード フィールドを非表示にする方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。

### 3.3.4 リッチ テキスト エディター (RTE) の構成

リッチ テキスト エディター (RTE) のプロファイルはリッチ テキスト エディターのフィールドで利用可能な機能を制御します。

開発者は次のために RTE プロファイルを使用します：

- RTE フィールドから機能を削除する。
- RTE フィールドのデフォルト機能を有効にする。
- RTE フィールドの異なる機能を異なるユーザーに対して表示する。
- RTE フィールドのドロップダウン メニューと他のオプションを構成する。
- RTE フィールドにカスタム機能を追加する。
- 異なるデータ テンプレートと単一のテンプレートの異なる RTE フィールドで異なる機能を利用可能にする。

開発者は各 RTE フィールド定義アイテムのソース プロパティを使って RTE プロファイルを参照することができます。

Sitecore は RTE プロファイルを Core データベースの `/Sitecore/System/Settings/Html Editor Profiles` の下にストアします。開発者が RTE フィールドにソース プロパティを定義しない場合、Sitecore は `/Sitecore/System/Settings/Html Editor Profiles/Rich Text Default RTE` プロファイルを適用します。

各 RTE プロファイルに存在する定義アイテムだけがそのプロファイルを使っているフィールドに表示されます。RTE プロファイルの定義アイテムにアサインされているアクセス権が、ユーザーの RTE 機能へのアクセスを制御します。

#### ヒント

一貫性、アクセシビリティ、リユースのために、デザインをプレゼンテーションから分離します。コンテンツとスタイリング（CSS スタイルなど）やプレゼンテーション（HTML テーブルやイメージ）を結合してしまうと、RTE フィールドはこの目的を達

成できません。RTE フィールド タイプの使用を最小化することに加え、各 RTE フィールドで RTE プロファイルを使って利用できる機能を最小化します。

## RTE プロファイル定義アイテム リファレンス

次のテーブルは RTE プロファイル定義アイテムで使用されるアイテムを説明しています。

Path	機能
/Buttons/HTML View	<b>HTML</b> タブが RTE の最後に表示されるかを制御する
/Class Translation	RTE に切り貼りされたコンテンツのスタイリング情報を対応する CSS スタイルにマッピングする
/Font Names	フォント名のドロップダウン メニューを作成する
/Font Sizes	フォントサイズのドロップダウン メニューを作成する
/Links	カスタム リンクのドロップダウン メニューを作成する
/Paragraphs	エレメントのドロップダウン メニューを作成する
/Ribbon	ページエディターのリボン フォーマットを制御する
/Snippets	Snippets 機能を作成する
/Tidy	このパスは使われなくなりました
/Toolbar 1	第一ツール バーのオプションを制御する
/Toolbar 2	第二ツール バーのオプションを制御する
/Toolbar 3	第三ツール バーのオプションを制御する
/WebEdit Buttons	ページエディターのインライン編集時にフィールドの下に表示されるフォーマット ボタンを制御する
/Zoom	ズーム ドロップダウン メニューを作成する

## RTE プロファイルの構成方法

RTE プロファイルを構成する方法 :

1. Sitecore デスクトップで Core データベースを、**コンテンツ エディター**で Core データベースを使用し、  
/Sitecore/System/Settings/Html Editor Profiles アイテムを選択します。Sitecore デス

クックブックでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

2. Sitecore デスクトップで Core データベースを使い、**コンテンツ エディター**で各カスタム RTE プロファイル定義アイテムの下で、すべてのユーザーには不要なオプションを削除します。item:read アクセス権を使って他のオプションが表示されるかどうかを制御します。
3. Sitecore デスクトップで Master データベースを選択します。データベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

### 重要

**Core** データベースの /Sitecore/System/Settings/Html Editor Profiles アイテムの下にあるデフォルトの RTE プロファイルを編集しないでください。それに代えて、既存の RTE プロファイルを複製し、そのプロファイルを使用する RTE データ テンプレート フィールド定義アイテムを構成します。

### ヒント

デフォルト RTE プロファイルを編集することなく構成するには、または各 RTE フィールド定義のソース プロパティにそのプロファイルを指定することなく構成するには、デフォルト RTE プロファイルを複製しバックアップを作成してからオリジナルを編集します。

## RTE プロファイルへのパスの決定方法

RTE プロファイルへのパスを決定する方法：

1. Sitecore デスクトップで Core データベースを使用して、**コンテンツ エディター**で RTE プロファイル定義アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. Sitecore デスクトップで Core データベースを使用して、**コンテンツ エディター**で RTE プロファイル定義アイテムへのパスを Windows クリップボードにコピーします。アイテムのパスを Windows クリップボードにコピーする方法は「アイテムの ID またはパスを Windows クリップボードにコピーする方法」のセクションを参照してください。
3. Sitecore デスクトップで Master データベースを選択します。データベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」セクションを参照してください。

## RTE テンプレート フィールド定義アイテムのプロファイルを設定する方法

RTE テンプレート フィールド定義アイテムのプロファイルを設定する方法：

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 RTE フィールド定義アイテムを選択します。
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**の [Data] セクションの [Source] フィールドで、RTE プロファイルへのパスを入力します。

## RTE プロファイルの機能の追加方法

RTE プロファイルの機能を追加する方法：

- /Sitecore/System/Settings/Html Editor Profiles/Rich Text Full RTE プロファイル定義から機能をコピーします。

## RTE フィールドで使われる CSS を設定する方法

すべての RTE フィールドの [CSS Class を適用] ドロップダウンの作成に使用される CSS (Cascading Style Sheet) ファイルを設定する方法：<sup>3</sup>

1. リッチ テキスト エディターを閉じます。
2. web.config の /configuration/sitecore/settings/setting エLEMENT の name WebStylesheet で、value 属性に CSS ファイルの IIS Web サイトのドキュメントルートへの相対パスを設定します。
3. リッチ テキスト エディターを開いて変更を確認します。

## RTE フィールドで表示される CSS スタイルを制限する方法

CSS import ディレクティブを使って RTE の [CSS Class を適用] のドロップダウン メニューにスタイルが表示されないようにすることができます。

この [CSS Class を適用] ドロップダウン メニューは web.config の /configuration/sitecore/settings/setting エLEMENT の name WebStylesheet の value 属性で指定された CSS ファイルで定義されたスタイルを含みます。

多くの Web サイトは RTE のコンテンツに関係のない CSS スタイルを使っています。CSS import ディレクティブを使って RTE の [CSS Class を適用] のドロップダウンにこれらのスタイルが表示されないようにすることができます。

たとえば、ユーザーが RTE のコンテンツに適用できるスタイルを contentstyles.css ファイルに配置し、WebStylesheet 設定にそのファイルのドキュメント ルートへの相対パスを設定します。RTE に表示されないようにするスタイルを同じディレクトリの sitestyles.css ファイルに配置し、レイアウトでそのファイルを参照します。

sitestyles.css で contentstyles.css を次のような CSS import ディレクティブを使ってインポートします：

```
@import url(contentstyles.css);
```

contentstyles.css で定義されたスタイルだけが RTE に表示されます。プレゼンテーション コンポーネントは sitestyles.css と contentstyles.css で定義されたスタイルを使うことができます。

<sup>3</sup> RTE に異なる CSS をロードするための情報は

<http://sdn.sitecore.net/Scrapbook/Dynamically%20loading%20web%20stylesheets%20in%20RTE.aspx> を参照してください。

**注意**

RTE は CSS `import` ディレクティブをサポートしません。`web.config` の `WebStylesheet` 設定で指定された CSS ファイルによってインポートされた CSS ファイルで定義されたスタイルは RTE に表示されません。

**注意**

さまざまなレベルでのキャッシュにより、CSS ファイルの変更はブラウザ上で即時には表示されないかも知れません。CSS ファイルの変更が表示されない場合は、次の操作を順に変更が表示されるまで行ってください。リッチテキストエディターをリロードする。ブラウザ キャッシュをクリアする。IIS を再起動する。`web.config` の `/configuration/sitecore/settings/setting` エレメントの `name WebStylesheet` の `value` 属性を変更する。

## RTE HTML エlement タイプ ドロップダウン メニューを構成する

RTE にリストされる HTML エlement タイプを構成する方法：

1. Sitecore デスクトップで Core データベースを使用し、**コンテンツ エディター**で RTE プロファイル定義アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. Sitecore デスクトップで Core データベースを使用し、**コンテンツ エディター**で RTE プロファイル定義アイテムの下で `/Paragraphs` を選択します。
3. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で、HTML エlement 定義アイテムを `/System/Html Editor Profiles/Html Editor List Item` データテンプレートを使用して挿入します。
4. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で HTML エlement 定義アイテムを選択します。
5. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で、`[Data]` セクションの `[Header]` フィールドで Element ドロップダウンで表示する値を入力します。
6. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で、新規アイテムの `[Data]` セクションの `[Value]` フィールドで、ユーザーがこの Element タイプを選択した時に挿入する HTML エlement を入力します。
7. Sitecore デスクトップで Master データベースを選択します。データベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

## RTE プロファイルで Snippet を有効化する方法

RTE プロファイルで Snippet を有効化する方法：

1. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で `/Sitecore/System/Settings/Html Editor Profiles/Rich Text Full/Toolbar 1` アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で `Insert Snippet` アイテムを RTE プロファイルの対応する場所にコピーします。
3. Sitecore デスクトップで Master データベースを選択します。データベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

## RTE プロファイルに Snippet を追加する方法

RTE プロファイルに Snippet を追加する方法：

1. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で `/Sitecore/System/Settings/Html Editor Profiles` ブランチの中で関連する RTE プロファイル定義アイテムの下の **Snippets** アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. コンテンツ エディターで関連する RTE プロファイル定義アイテムの中で Snippet アイテムを選択したままで、Snippet 定義アイテムを `/System/Html Editor Profiles/Html Editor Snippet` データテンプレートを使って挿入します。
3. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で、Snippet 定義アイテムの **[Data]** セクションの **[Header]** フィールドで Snippet ドロップダウン リストで表示するテキストを入力します。
4. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で Snippet 定義アイテムの **[Data]** セクションの **[Value]** フィールドでユーザーが Snippet を選択したときに挿入するマークアップを入力します。
5. Sitecore デスクトップで Master データベースを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

## エンター キーで挿入されたマークアップを制御する方法

`web.config` の `/configuration/sitecore/settings/setting` エLEMENT の `name` `HtmlEditor.LineBreak` の `value` 属性が、ユーザーが [ENTER] キーを押したときにリッチ テキスト エディターが挿入するマークアップを制御します。デフォルトではこの `setting` の `value` は `[p]` で、リッチ テキスト エディターがテキストを段落 エLEMENT で囲みます (“`<p>...</p>`”)。行の最後でテキストを段落 エLEMENT で囲むのではなく、改行エLEMENT (“`<br>`”) を挿入するには、この設定の値を `[br]` に変更します。

**メモ**

/configuration/sitecore/settings/setting エLEMENTの name HtmlEditor.LineBreak の value が [p] である場合、改行のELEMENT (<br>) を [CTRL-ENTER] を入力して挿入できます。この設定の値が [br] である場合、パラグラフELEMENT (<p>...</p>) を [CTRL-M] を入力して挿入できます。

**RTE HTML タブをあるユーザーまたはすべてのユーザーに対して無効にする方法**

すべての CMS ユーザーの HTML タブを RTE から削除したり、アクセス権を使ってどのユーザーが HTML タブをアクセスできるかを制御できます。

RTE から HTML タブを削除または制限する方法：

1. Sitecore デスクトップで **Core** データベースを使用します。**コンテンツ エディター**で RTE エディター プロファイル 定義アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で /Buttons/HTML View アイテムへの読み取りアクセス権を削除または制限します。
3. Sitecore デスクトップで Master データベースを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

## 3.4 データ テンプレートのカスタム ユーザー インターフェース

Sitecore が必要なデータ テンプレート ユーザー インターフェースを提供していない場合、データ テンプレートにカスタム ユーザー インターフェースを実装することができます。

### 3.4.1 IFrame フィールドタイプ

開発者は IFrame データ テンプレート フィールド タイプを使ってコンテンツ エディターでアイテムを取り扱う時に任意の Web アプリケーションをデータ テンプレート フィールドのように見せることができます。

開発者は次のために IFrame フィールド タイプを使うことができます：

- シンプルなカスタム フィールド エディターをデータ テンプレートに追加する。
- レポートと他のカスタム アプリケーションをデータ テンプレートに追加する。

IFrame タイプのフィールドがフィールド定義アイテムのソース プロパティで指定された URL をコンテンツ エディターの IFrame にロードする。Sitecore は次のクエリ文字列パラメータを URL に追加します。

パラメータ	機能
id	ユーザーに選択されたアイテムの GUID
la	ユーザーに選択された言語コード
vs	ユーザーに選択されたバージョン数

開発者は追加のクエリ文字列パラメータを IFrame フィールド定義アイテムのソース プロパティに指定された URL に追加することによってさらに追加の変数を渡すことができます。

### IFrame データ テンプレート フィールドの作成方法

IFrame データ テンプレート フィールドを作成する方法：

1. Visual Studio の Web Applicatoion プロジェクトで iframe で表示するユーザー インターフェースを含む Web ページを作成します。IFrame フィールドに渡されるクエリ文字列パラメータについての詳細は「IFrame フィールドタイプ」のセクションを参照してください。
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**でデータ テンプレート定義アイテムを選択します。
3. **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、[編集] ウィンドウで [ビルダー] タブをクリックします。**テンプレートビルダー**が表示されます。

4. **テンプレート マネージャー**で IFrame タイプのフィールドを追加しデータ テンプレートをセーブします。<sup>4</sup>
5. **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**でセクション定義アイテムを展開し、IFrame データ テンプレート フィールド定義アイテムを選択します。
6. **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**の[Data] セクションの[Source] フィールドで、先に作成された IFrame ユーザーインターフェースを含む Web ページの URL を入力します。

### 3.4.2 アイテム エディター

アイテム エディターはコンテンツ エディターでアイテムを取り扱う方法に代わるユーザーインターフェースを定義します。開発者は次のためにカスタム アイテム エディターを実装することができます：

- 特定のアイテムまたは特定のデータ テンプレートに基づくアイテムのために機能する任意のアプリケーションを提供する
- 個々のアイテムまたは特定のデータ テンプレートに基づくアイテムのためのサマリー、レポート、またはその他のインターフェースを提供する
- アイテムの階層構造を更新するための単一のユーザー インターフェースを提供する

アイテム エディターはユーザーがアイテムを選択したときにコンテンツ エディターのタブのように見えます。コンテンツ エディターの各タブによってそれぞれ別のアイテム エディターが起動します。開発者は任意の数のアイテムエディターをアイテムまたはデータ テンプレートに関連付けることができます。各データ テンプレートの標準値アイテムがそのテンプレートに関連付けられたすべてのアイテムのためのデフォルトのアイテム エディターを定義します。

すべてのアイテムのためのデフォルトのエディターはコンテンツ エディターの [コンテンツ] タブとして見えます。このインターフェースはアイテムのデータ テンプレートとそのベース テンプレートによって定義されたセクションにグループ化されたフィールド エディット コントロールを示します。

Sitecore はデフォルトでさまざまなアイテム エディターを特定のタイプのアイテムに関連付けます。フォルダーのフォルダー アイテム エディターが [フォルダー] タブにリストされる既存の子アイテムと新しい子アイテムを作成するオプションを定義します。データ テンプレートのビルダー アイテム エディターが、テンプレートを構成するデータテンプレート セクションとフィールドを取り扱うための [ビルダー] タブを定義します。

アイテム エディターは URL でアドレスできる Web アプリケーションです。Sitecore はアイテム エディターを Core データベースの /Sitecore/Content/Applications/Content Editor/Editors で管理します。

Sitecore は次のクエリ文字列パラメータをアイテムエディターに渡します。

---

<sup>4</sup> フィールドの追加方法については

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

パラメータ	機能
id	ユーザーに選択されたアイテムの GUID
language	ユーザーに選択された言語コード
version	ユーザーに選択されたバージョン数
database	ユーザーに選択されたアイテムを含むデータベース名

**メモ**

エディター定義アイテムの **[Data]** セクションの **[Url]** フィールドに指定された URL にクエリ文字列パラメータを追加することによって追加の変数を渡すことができます。

**メモ**

アイテム エディターは Sitecore の以前のバージョンの **[編集]** プロパティに優先します。

**アイテム エディターの構成方法**

アイテム エディターを構成する方法：

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**で標準値アイテムまたは個々のアイテムを選択します。
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 **[設定]** タブをクリックします。
3. **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**の **[設定]** タブの **[アピランス(外観と動作)]** グループで **[カスタム エディターを設定]** をクリックすると、**[カスタム エディター]** ダイアログ ボックスが表示されます。
4. **[カスタム エディター]** ダイアログ ボックスの **[すべて]** フィールドで枝を展開し、アイテム エディターをダブルクリックしてそれを **[選択済み]** リストに追加します。
5. アイテム エディター タブの順序を変更するには、カスタム エディター ダイアログの **[選択済み]** フィールドでアイテム エディターを選択し、右にある矢印をクリックして選んだものをソートします。
6. アイテム エディターを削除するには **[カスタム エディター]** ダイアログ ボックスの **[選択済み]** リストでアイテム エディターをダブルクリックします。

**カスタムのアイテムエディターの作成方法**

カスタムのアイテムエディターを作成する方法：

1. Visual Studio の Web Application プロジェクトでカスタムのアイテム エディター ユーザー インターフェースを含む Web ページを作成します。<sup>5</sup>
2. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で /Sitecore/Content/Applications/Content Editor/Editors/Items アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
3. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**で /Sitecore Client/Content Editor/Editor データテンプレートを使用しえ、アイテム エディター定義アイテムを挿入します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
4. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**でアイテム エディター定義アイテムで、**[Data]** セクションの **[Header]** フィールドで、**コンテンツ エディター**がアイテム エディターを起動するタブに表示する値を入力します。
5. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**でアイテム エディター定義アイテムで、**[Data]** セクションの **[Icon]** フィールドで、アイテム エディターを起動するタブに表示するアイコンを入力します。
6. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**でアイテム エディター定義アイテムで、**[Data]** セクションの **[URL]** フィールドで、先に作成されたアイテム エディター ユーザー インターフェースを含む Web ページの URL を入力します。
7. Sitecore デスクトップで Core データベースを使用します。**コンテンツ エディター**でアイテム エディター定義アイテムの **[Data]** セクションで、アイテム エディター定義アイテムをセーブします。
8. Sitecore デスクトップで Master データベースを選択します。

---

<sup>5</sup> カスタムエディターに渡されるクエリ文字列パラメータについての詳細は <http://sdn.sitecore.net/Reference/References%20in%20Japanese/Client%20Configuration%20Reference.aspx> を参照してください。

## 3.5 挿入オプション

挿入オプションはユーザーが既存のアイテムの下に挿入できるアイテムのタイプを制御します。<sup>6</sup>

開発者は次のために挿入オプションを適用することができます：

- ユーザーが既存のアイテムの下に適切なタイプのアイテムを作成できるようにする
- ユーザーが既存のアイテムの下に作成できるアイテムのタイプを制限する
- 既存のアイテムの下にユーザーごとに異なるタイプのアイテムを作成できるようにする
- ユーザーがユーザー インターフェースを使っていくつかのアイテムを一度に作成できるようにする
- アイテムをプログラムから作成できるようにする。たとえば、ウィザードを起動するなど

---

<sup>6</sup> 挿入オプションについての詳細は

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Data%20Definition%20Reference.aspx>  
から『データ定義リファレンス マニュアル』、また  
<http://sdn.sitecore.net/Reference/References%20in%20Japanese/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

# データ検証

この章ではデータ検証（データバリデーション）を構成する手順について説明します。開発者はデータエントリーのルールを徹底させるためにデータ検証を使用することができます。この章には次のセクションがあります。

- 検証の概要
- 検証の種類
- 検証のエラーレベル
- 検証ルールの構成
- バリデーターの登録
- カスタムバリデーター
- 検証アクション

## 4.1 検証の概要

データ検証を構成して Sitecore が無効なデータを取り扱う方法を制御することができます。

クイック アクション バーのバリデーター バーから Sitecore を構成し、ユーザーが [レビュー] タブの [校正] グループで [検証] コマンドを選択したとき、またはユーザーが特定のワークフロー コマンドを選択したときに、検証ルールを起動させることができます。

### メモ

通常、4 種類の検証すべてに同じ検証ルールを選択します。

各データ テンプレート フィールド、またはひとつのデータ テンプレート フィールド タイプのすべてのデータ テンプレート フィールド、または個々のアイテム、またはひとつのデータ テンプレートに基づくすべてのアイテム、またはすべてのアイテム、に対して別のバリデーターを構成することができます。

### 重要

各アイテムに対して、Sitecore はアイテムまたはデータ テンプレートに関連付けられた標準値アイテムで定義されたアイテムの検証ルールおよびグローバル検証ルールを起動します。各データ テンプレート フィールドについて、Sitecore はデータ テンプレート フィールド定義アイテムで定義された検証ルール、およびデータ テンプレート フィールド タイプ検証ルール定義アイテムで定義された検証ルールを起動します。システムがひとつのアイテムに対して起動するバリデーターの数は使用されるサーバーのリソースとクライアントのパフォーマンスに影響があります。

### 重要

`/sitecore/system/Settings/Validation Rules` の下のデフォルトのバリデーター定義アイテムを編集しないでください。代わりに、バリデーター定義アイテムを作成して追加のバリデーターを登録してください。

### メモ

フィールド値の編集を終了した後に、Sitecore は自動的に検証ルールを非同期に起動し、検証が完了した後にユーザー インターフェイスを更新します。`web.config` の `/configuration/sitecore/settings/setting` エレメントの `name Validators.UpdateDelay` の `value` 属性が、編集作業の休止とバリデーターの起動の間の時間の長さを制御します。自動で再検証を行う機能を無効にするには、`web.config` の `/configuration/sitecore/settings/setting` エレメントの `name ValidatorsAutomaticUpdate` の `value` 属性を `False` に設定します。

### メモ

Sitecore のデフォルト構成ではクイック アクション バーの検証は無効になっています。クイック アクション バーの検証を有効にするには [クイック アクション バー] を右クリックして [検証ルール] を選択します。

## ヒント

以前のバージョンの Sitecore で利用できるフィールド定義アイテムの [データ] セクションの [検証] と [検証テキスト] フィールドを使用することは避けて、この章に記載するデータ検証機能を使用してください。

## 4.2 検証の種類

いろいろなタイプのデータの妥当性をチェックすることができます。

### 4.2.1 データ テンプレートのフィールド検証

Sitecore を構成して各データ テンプレートの各フィールド コンテンツの検証を行うことができます。

### 4.2.2 フィールド タイプの検証

Sitecore を構成するフィールド タイプのそれぞれのコンテンツの検証を行うことができます。

### 4.2.3 アイテムの検証

個々のアイテムの検証を構成することができます。

### 4.2.4 標準の検証

データ テンプレートの標準値の検証ルールを構成して、データ テンプレートに基づくすべてのアイテムの検証を構成することができます。

### 4.2.5 グローバル アイテムの検証

すべてのアイテムに適用される検証を構成することができます。

## 4.3 検証のエラー レベル

検証のエラー レベルは検証の結果によって Sitecore が行うアクションを制御します。各バリデーターは `Sitecore.Data.Validators.ValidatorResult` に定義されているエラー レベルのひとつを返します。

エラーレベル	UI の色	機能
Unknown	グレー	検証未完了、結果不明
Valid	緑	有効
Suggestion	緑	ユーザー インターフェイスにアドバイスが表示されます
Warning	オレンジ	ユーザー インターフェイスに注意が表示されます
Error	赤	エラーのためユーザーはワークフロー検証に関連付けられたワークフロー コマンドを完了できません
CriticalError	赤	ユーザーがアイテムをセーブしようとした際に重大なエラーのためモーダルな警告表示が発生し、ユーザーはワークフロー検証に関連付けられたワークフローコマンドを完了できません
FatalError	赤	致命的なエラーのためモーダルな警告表示が発生し、ユーザーはアイテムをセーブできません。

## 4.4 検証ルール構成

クイックアクション バーのバリデーター バーから Sitecore を構成し、ユーザーが [レビュー] タブの [校正] グループで [検証] コマンドを選択したとき、またはユーザーが特定のワークフローコマンドを選択したときに、検証ルールを起動させることができます。

### 4.4.1 クイックアクション バーの検証ルール構成方法

クイックアクション バーでの検証ルールの起動を構成する方法：

1. **コンテンツ エディター**で次のセクションで記述されているように適切な検証ルール定義アイテムを選択します。
2. **コンテンツ エディター**で [検証] セクションの [クイックアクション バー] フィールドで、検証ルールを選択します。

### 4.4.2 [検証] ボタンの検証ルール構成方法

ユーザーが [レビュー] タブの [校正] グループの [検証] コマンドをクリックしたときに検証ルールが起動するように構成する方法：

1. **コンテンツ エディター**で次のセクションで記述されているように適切な検証ルール定義アイテムを選択します。
2. **コンテンツ エディター**で [検証] セクションの [検証ボタン] フィールドで、検証ルールを選択します。

### 4.4.3 [検証] バーの検証ルール構成方法

[検証] バーで検証ルールの起動を構成する方法：

1. **コンテンツ エディター**で次のセクションで記述されているように適切な検証ルール定義アイテムを選択します。
2. **コンテンツ エディター**で [検証] セクションの [検証バー] フィールドで、検証ルールを選択します。

#### メモ

**web.config** の `/configuration/sitecore/settings/setting` エLEMENTの `name` `ContentEditor.ShowValidatorBar` の **value** 属性が `False` に設定されている場合は検証バーはコンテンツ エディターには表示されません。

### 4.4.4 ワークフローの検証ルール構成方法

ユーザーがワークフロー コマンドを選択したときに検証ルールが起動するように構成する方法：

1. **コンテンツ エディター**で次のセクションで記述されているように適切な検証ルール定義アイテムを選択します。
2. **コンテンツ エディター**で [検証] セクションの [ワークフロー] フィールドで、検証ルールを選択します。

#### 4.4.5 ワークフロー コマンドの作成または状態検証アクションの作成方法

ユーザーがワークフロー コマンドを選択したときに、Sitecore が検証ワークフローアクションを起動するように構成する方法：

1. **コンテンツ エディター**でワークフロー状態またはコマンドを選択します。
2. **コンテンツ エディター**で /System/Workflow/Validation Action データ テンプレートを使用して検証ワークフロー アクション定義アイテムを挿入します。
3. **コンテンツ エディター**で検証ワークフロー アクションの **[Data]** セクションの **[Type]** フィールドで次の項目を入力します。

```
Sitecore.Workflows.Simple.ValidatorsAction, Sitecore.Kernel.
```

4. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Max Result Allowed]** フィールドでワークフロー検証アクションが生成することのできる `Sitecore.Data.Validators.ValidatorResult` の最大値を入力します。検証がより高い検証エラー レベルを生成した場合、ワークフロー検証アクションのためにユーザーはワークフロー コマンドを完了させることはできません。このフィールドのデフォルト値は `[Warning]` です。
5. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Unknown Result]** フィールドで、バリデーターが検証の結果として `[Unknown]` を返した場合にユーザーに表示するメッセージを入力します。
6. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Warning Result]** フィールドで、バリデーターが検証の結果として `[Warning]` を返した場合にユーザーに表示するメッセージを入力します。
7. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Error Result]** フィールドで、バリデーターが検証の結果として `[Error]` を返した場合にユーザーに表示するメッセージを入力します。
8. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Critical Error Result]** フィールドで、バリデーターが検証の結果として `[CriticalError]` を返した場合にユーザーに表示するメッセージを入力します。
9. **コンテンツ エディター**で、検証ワークフローアクションの **[Data]** セクションの **[Fatal Error Result]** フィールドで、バリデーターが検証の結果として `[FatalError]` を返した場合にユーザーに表示するメッセージを入力します。

#### 4.4.6 データ テンプレート フィールド タイプのすべてのインスタンスへの検証ルール構成方法

特定のデータ テンプレート フィールド タイプのすべてのインスタンスに起動する検証ルールを構成する方法：

1. **コンテンツ エディター**で /Sitecore/System/Settings/Validation Rules/Field Types アイテムを選択します。

2. **コンテンツ エディター**で既存のデータ テンプレート フィールド タイプの検証ルール定義アイテムを選択するか、  
/System/Validation/Field Type Validation Rules データ テンプレートを使用してデータ テンプレート フィールド タイプの検証ルール定義アイテムを挿入します。フィールド タイプの検証定義アイテムの名前には、フィールド タイプ定義アイテムのデータ セクションのタイプ フィールドに表示されるフィールド タイプの名前を使用します。
3. **コンテンツ エディター**でデータ テンプレート フィールド タイプの検証ルール定義アイテムの **[検証ルール]** セクションで、検証ルールを構成します。

#### 4.4.7 すべてのアイテムへの検証ルール構成方法

すべてのアイテムへの検証ルールを構成する方法：

- **コンテンツ エディター**で /Sitecore/System/Settings/Validation Rules/Global Rules アイテムを選択し、検証ルールを構成します。

#### 4.4.8 個々のアイテムまたは特定のデータ テンプレートに基づくすべてのアイテムへの検証ルールの構成方法

個々のアイテムまたは特定のデータテンプレートに基づくすべてのアイテムへの検証ルールを構成する方法：

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**でスタンダード フィールドを表示します。スタンダード フィールドを表示する方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。
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**でデータ テンプレートの標準値アイテムを選択するか、または個々のアイテムを選択します。
3. **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**で検証ルールを構成します。
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**でスタンダード フィールドを非表示にします。スタンダード フィールドを非表示にする方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。

#### 4.4.9 特定のデータ テンプレート フィールドへの検証ルール構成方法

特定のデータ テンプレート フィールドへの検証ルールを構成する方法：

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**でデータ テンプレート フィールド定義アイテムを選択します。
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、**[検証ルール]** セクションで、フィールド検証ルールを構成します。

#### 4.4.10 デフォルトのアイテムバリデーター

次のデフォルトのバリデーターを使用してアイテムの検証ルールを構成することができます。

バリデーター	機能
Broken Links	無効なリファレンスを識別する
Duplicate Name	同じ親の下で他のアイテムの名前または表示名と同じ名前または表示名をもつアイテム（大文字と小文字は区別する）を識別する
Full Page XHTML	デフォルトのデバイスを使用してアイテムを要求するアウトプットの有効性をチェックする
Media Size Too Big	データベース ストレージに直列化するには大きすぎるメディアを識別する
Url Characters	アイテム名が URL でエスケープ シーケンスを必要とするかチェックする

#### メモ

XHTML バリデーターは XML を `web.config` の `/configuration/sitecore/settings/setting` エLEMENT の `name XHTMLSchemaFile` の `value` 属性で定義されたスキーマに対して検証します。

#### 4.4.11 デフォルトのフィールドバリデーター

次のデフォルトのバリデーターを使用してフィールドの検証ルールを構成することができます。

バリデーター	機能
Broken Links	無効なリファレンスを識別する
Is Email	無効な電子メール アドレスを識別する
Is Integer	無効な整数を識別する
Is XHTML	ローカルの XHTML スキーマに対してフィールドの有効性をチェックする
Max Length 40	40 文字以上を含むフィールド値を識別する
Rating 1 to 9	無効な整数、負数、9 より大きな値を識別する
Required	空のフィールドを識別する
Spellcheck	スペリングのエラーを含むフィールドを識別する
W3C XHTML Validation	リモートの W3C 検証サービスに対する無効性を識別する

バリデーター	機能
Alt Required	メディア ライブラリ イメージ データ テンプレートの Alt テキストの欠落を識別する
Extension May Not Start with a Dot	フィールド名の拡張子にドット文字があるメディア アイテム（それは無効）を識別する
External Link Target	新しいブラウザ ウィンドウで開きタイトルを提供するリッチ テキスト エディター フィールドの外部リンクを識別する
Image Has Alt Text	代替テキストを含まず、代替テキストを含まないメディア アイテムを参照する、イメージ フィールドを識別する
Image Has Alt Text from Media Library	代替テキストを指定せず、代替テキストを含むメディア アイテムを参照する、イメージ フィールドを識別する
Image Size	サイズ制限を超えたメディア アイテムを参照するイメージ フィールドを識別する
Rich Text Image Size	リッチ テキスト エディターで使われるイメージの幅は web.config の /configuration/sitecore/settings/setting エレメントの name Media.MaxImageWidth の value を超えてはならない

## メモ

web.config の /configuration/sitecore/settings/setting エレメントの name HtmlEditor.ValidatorServiceUrl の value 属性が W3C XHTML 検証で使われる URL を制御します。

## 4.5 バリデーターの登録

検証ルール定義アイテムを構成してデフォルトのバリデーターを使うか、またはカスタムのバリデーターを登録することができます。

### 重要

Sitecore のユーザー インターフェースのパフォーマンスを最大化するため、多くのリソース特に処理時間を消費するバリデーター、たとえば外部サービスを起動するか長時間の処理になる可能性のあるものは、別のスレッドで稼働させるのがよいでしょう。別のスレッドを使用することでシステムはひとつのバリデーターが完了する前に別のバリデーターを起動することができます。Sitecore は別のバリデーターが完了するとユーザーインターフェースを更新します。

### 4.5.1 バリデーターの登録方法

フィールドバリデーターを登録する方法：

1. **コンテンツ エディター**で `/Sitecore/System/Settings/Validation Rules/Field Rules` アイテムまたは `/Sitecore/System/Settings/Validation Rules/Item Rules` アイテムの下の適当なプロジェクトに固有のフォルダーを選択します。
2. **コンテンツ エディター**で `/System/Validation/Validation Rule` データ テンプレートを使ってバリデーター定義アイテムを挿入します。
3. **コンテンツ エディター**でバリデーター定義アイテムの **[データ]** セクションの **[種類]** フィールドでクラスシグネチャーを入力します。
4. **コンテンツ エディター**でバリデーター定義アイテムの **[データ]** セクションの **[パラメーター]** フィールドで、URL からエスケープ キーを除いたパラメーター値（アンパサンド文字（`&`）で区切る）を入力します。
5. バリデーターが別のスレッドで稼働する場合には、**コンテンツ エディター**でバリデーター定義アイテムの **[データ]** セクションで **[スレッドを使用する]** を選択します。

### 4.5.2 正規表現フィールド バリデーターの登録方法

正規表現フィールド バリデーターを登録する方法：

1. フィールド バリデーターを登録します。バリデーターを登録する方法は「バリデーターの登録方法」のセクションを参照してください。
2. **コンテンツ エディター**でバリデーター定義アイテムの **[データ]** セクションの **[種類]** フィールドで次の .NET タイプのシグネチャーを入力します。

```
Sitecore.Data.Validators.FieldValidators.RegexValidator, Sitecore.Kernel
```

3. **コンテンツ エディター**でバリデーター定義アイテムの **[パラメーター]** フィールドで下記を入力します。

```
Pattern=RegularExpression&Text=Message
```

4. **コンテンツ エディター**でバリデーター定義アイテムの [データ] セクションの [パラメーター] フィールドで `RegularExpression` を正規表現で置換します。
5. **コンテンツ エディター**でバリデーター定義アイテムの [データ] セクションの [パラメーター] フィールドで [メッセージ] をフィールドがパターンに合致しない場合に表示するメッセージで置換します。Sitecore はメッセージの {0} をフィールド名で置き換えるために `String.Format()` を呼び出します。

### 4.5.3 整数フィールド バリデーターの登録方法

整数フィールドバリデーターを登録する方法：

1. フィールドバリデーターを登録します。バリデーターを登録する方法は「バリデーターの登録方法」のセクションを参照してください。
2. **コンテンツ エディター**でバリデーター定義アイテムの [データ] セクションの [種類] フィールドで次の .NET タイプのシグネチャーを入力します。

```
Sitecore.Data.Validators.FieldValidators.IntegerFieldValidator, Sitecore.Kernel
```

3. **コンテンツ エディター**でバリデーター定義アイテムの [データ] セクションの [パラメーター] フィールドで下記を入力します：

```
AllowNegative=AllowNegative=Boolean&AllowZero=Boolean
```

4. **コンテンツ エディター**でバリデーター定義アイテムの [データ] セクションの [パラメーター] フィールドで `Boolean` を適宜、[True] または [False] で置換します。

### 4.5.4 整数範囲フィールド バリデーターの登録方法

整数範囲フィールドバリデーターを登録する方法：

1. フィールドバリデーターを登録します。バリデーターを登録する方法は「バリデーターの登録方法」のセクションを参照してください。
2. **コンテンツ エディター**でバリデーター定義アイテムを選択します。
3. **コンテンツ エディター**で [データ] セクションの [種類] フィールドで次の .NET タイプのシグネチャーを入力します。

```
Sitecore.Data.Validators.FieldValidators.IntegerRangeValidator, Sitecore.Kernel
```

4. **コンテンツ エディター**で [データ] セクションの [パラメーター] フィールドで次を入力します。

```
Min=Minimum&Max=Maximum
```

5. **コンテンツ エディター**で [データ] セクションの [パラメーター] フィールドで [Minimum] をフィールドに許容される最小値で置換します。

6. **コンテンツ エディター**で **[データ]** セクションの **[パラメーター]** フィールドで **[Maximum]** をフィールドに許容される最大値で置換します。

#### 4.5.5 最大長フィールドバリデーターの登録方法

最大長フィールドバリデーターを登録する方法：

1. フィールドバリデーターを登録します。バリデーターを登録する方法は「バリデーターの登録方法」のセクションを参照してください。
2. **コンテンツ エディター**でバリデーター定義アイテムを選択します。
3. **コンテンツ エディター**で **[データ]** セクションの **[種類]** フィールドで次を入力します。

```
Sitecore.Data.Validators.FieldValidators.MaxLengthFieldValidator, Sitecore.Kernel
```

4. **コンテンツ エディター**で **[データ]** セクションの **[パラメーター]** フィールドで次を入力します。

```
MaxLength=MaximumLength
```

5. **コンテンツ エディター**で **[パラメーター]** フィールドで **[MaximumLength]** をフィールドに許容される最大長の値で置換します。

#### 4.5.6 特定のアイテムと特定のテンプレートに基づくアイテムのためのバリデーターの登録方法

特定のアイテムと特定のテンプレートに基づくアイテムのためのバリデーターを登録する方法：

1. **コンテンツ エディター**で `/Sitecore/System/Settings/Validation Rules/Item Rules` アイテムを選択します。
2. **コンテンツ エディター**で `/System/Validation/Validation Rule` データテンプレートを使って検証ルール定義アイテムを挿入します。
3. **コンテンツ エディター**で検証ルール定義アイテムを選択します。
4. **コンテンツ エディター**で **[データ]** セクションの **[種類]** フィールドでクラスシグネチャーを入力します。
5. **コンテンツ エディター**で、検証ルール定義アイテムの **[データ]** セクションの **[パラメーター]** フィールドで、URL からエスケープキーを除いたパラメータ値（アンパサンド文字（"&"）で区切る）を入力します。

#### 4.5.7 デフォルトの検証ルールを無効にする方法

すべてのアイテムへのデフォルトの検証ルールを無効にする方法：

1. **コンテンツ エディター**で `/Sitecore/System/Settings/Validation Rules/Global Rules` アイテムを選択します。

2. **コンテンツ エディター**で **[検証ルール]** セクションの検証ルールを構成します。
3. **コンテンツ エディター**で `/Sitecore/System/Settings/Validation Rules/Field Types` アイテムを選択します。
4. **コンテンツ エディター**でフィールド タイプ検証ルール定義アイテムで検証を構成します。
5. **コンテンツ エディター**で `/Sitecore/Templates/System/Media` アイテムを選択します。
6. **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**でスタンダード フィールドを表示します。スタンダード フィールドを表示する方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。
7. メディアに使用されるデータ テンプレートのそれぞれに**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**でデータ テンプレート定義アイテムを選択します。[編集] ウィンドウで **[コンテンツ]** タブをクリックし、**[検証ルール]** セクションで検証を構成します。
8. **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**でスタンダード フィールドを非表示にします。スタンダード フィールドを非表示にする方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。

#### 重要

アイテム名のどの文字も `web.config` の `/configuration/sitecore/settings/setting` エLEMENT の `name InvalidItemNameChars` の `value` 属性で指定された正規表現に合致しません。アイテム名は `web.config` の `/configuration/sitecore/settings/setting` エLEMENT の `name ItemNameValidation` の `value` 属性で指定された正規表現に合致しなければなりません。これらの 2 つの設定を変更しないでください。

### 4.5.8 バリデーターのデフォルトのエラー レベルを上書きする方法

バリデーターのデフォルトのエラーレベルを上書きする方法：

1. **コンテンツ エディター**でバリデーター定義アイテムを選択します。
2. **コンテンツ エディター**で **[データ]** セクションの **[パラメーター]** フィールドで次を入力します。

```
Result=ErrorLevel
```

3. **コンテンツ エディター**で **[データ]** セクションの **[パラメーター]** フィールドで **[ErrorLevel]** を `Sitecore.Data.Validators.ValidatorResult` のエラー レベルの名前で置換します。

## 4.6 カスタムバリデーター

本章ではカスタムバリデーターを実装する手順を説明します。

### 4.6.1 カスタムバリデーターの実装方法

カスタムバリデーターには2つのコンポーネントが必要です：.NETクラスとバリデーター定義アイテムです。

### 4.6.2 カスタムバリデーターの実装方法

カスタムバリデーターを実装するには：

1. Microsoft Visual Studio の Web アプリケーション プロジェクトで次のプロトタイプに基づくクラスを作成します：

```
using System.Runtime.Serialization;
using Sitecore.Data.Validators;

namespace Namespace.Data.Validators.ItemValidators//TODO:namespace (FieldValidators)
{
    [Serializable]
    public class ClassName :Sitecore.Data.Validators.StandardValidator
    {
        public ClassName(){}//TODO:class name

        public ClassName(
            SerializationInfo info,StreamingContext context):base(info,context)
        {
        }

        public override string Name
        {
            get
            {
                return(GetType().ToString());//TODO:validator name
            }
        }

        protected override ValidatorResult GetMaxValidatorResult()
        {
            return(GetFailedResult(ValidatorResult.Error));//TODO:error level
        }

        protected override ValidatorResult Evaluate()
        {
            if(false)//TODO:validate ControlValidationValue
            {
                return(ValidatorResult.Valid);
            }
            else
            {
                Text = "error message";//TODO:error message
                return(GetFailedResult(ValidatorResult.Error));//TODO:error level
            }
        }
    }
}
```

2. このクラスで `Namespace.Data.ItemValidators` を適宜なネームスペースで置換します。
3. このクラスで `ClassName` のすべてのインスタンスをクラス名で置換します。
4. このクラスで `GetType().ToString()` をバリデーターのフレンドリー ネームで置換します。
5. このクラスですべてのインスタンスを適宜な検証エラーレベルで置換します。
6. このクラスで `[false]` を `ControlValidationValue` のフィールド値をバリデートするロジック、または `GetItem()` で返されるアイテムで置換します。
7. このクラスで `[error message]` をエラーメッセージで置換します。
8. バリデーターを登録します。バリデーターを登録する方法は「バリデーターの登録方法」のセクションを参照してください。

**重要**

`GetMaxValidatorResult()` メソッドの結果が `[FatalError]` または `[CriticalError]` である場合、Sitecore はユーザーインターフェースのセーブまたはワークフローコマンドなどの操作を検証が完了するまでブロックします。検証の間、ユーザー インターフェースがブロックされるのを避けるためには、コストの高いバリデーターの `GetMaxValidtorResult()` メソッドが `[FatalError]` または `[CriticalError]` を返さないようにします。検証バーおよびクイック アクション バーのバリデーターはユーザー インターフェースをブロックしません。

## 4.7 検証アクション

検証アクションは検証エラーを修正するためにユーザー インターフェースからクリックで操作できる処理を示します。

### 4.7.1 検証アクションの作成方法

単一行テキスト データ テンプレート フィールドに検証アクションを作成するには：

1. Visual Studio の Web アプリケーション プロジェクトで次のプロトタイプに基づくクラスを作成します：

```
using Sitecore.Shell.Framework.Commands.ContentEditor.Validators;

namespace Namespace.Shell.Framework.Commands.ContentEditor.Validators//TODO:namespace
{
    public class ClassName:ValidatorCommand//TODO:class name
    {
        public override void Execute(CommandContext context)
        {
            var validator=GetValidator(context);
            if(validator!=null)
            {
                var control = GetControlToValidate(validator);
                if (control!= null)
                {
                    control = control as Sitecore.Web.UI.HtmlControls.Control;
                    if(control!=null)
                    {
                        control.Value=Value;//TODO:logic
                        Validate();
                    }
                }
            }
        }
    }
}
```

2. このクラスで `Namespace.Shell.Framework.Commands.ContentEditor.Validators` をクラスのネーム スペースで置換します。
3. このクラスで `[ClassName]` をクラス名で置換します。
4. このクラスで `[Value]` をフィールドのバリデートされた値で置換します。
5. **コンテンツ エディター**で `/Sitecore/System/Settings/Validation rules/Field Rules` アイテムを選択します。
6. **コンテンツ エディター**で `/System/Menus/Menu Item` データテンプレートを使って検証アクション定義アイテムを挿入します。
7. **コンテンツ エディター**で、検証アクション定義アイテムの **[データ]** セクションの **[表示名]** フィールドで、検証アクションを起動するためにユーザーがユーザー インターフェースから選択するラベルを入力します。
8. `/App_Config/commands.config` ファイルで次のプロトタイプに基づく新しいコマンドを挿入します：

```
<command name="validator:ClassName" type="Namespace.Class,Assembly"/>
```

9. /App\_Config/commands.config ファイルで [ClassName] をクラス名で置換し、  
[Namespace.Class, Assembly] をネームスペースとクラスシグネチャーで置換します。

#### 重要

セーブされていない値は UI 中にもみ存在し、データベースには存在しないため、検証アクションでは Sitecore.Data.Items.Item およびその他の API は使用できません。フィールドを検証するために使われる API はフィールド タイプに依存し、JavaScript を起動する場合があります。（たとえばリッチテキストエディターフィールドの値など）

### 4.7.2 検証アクションの使用法

検証アクションを使用する方法：

1. **コンテンツ エディター**で検証アクションをもつ検証ルールに反するアイテムを編集します。
2. **[検証]** バーで **[検証インジケーター]** を右クリックし、検証アクションを選択します。

## Chapter 5

# ページ エディター

この章はインライン編集やデザイン ウィンドウなどの CMS 機能を提供するページ エディターの構成の手順を説明します。

この章には次のセクションがあります。

- ページ エディターの概要
- プレースホルダーの設定
- ページ エディターのモード
- エディット フレーム
- フィールド エディター

## 5.1 ページ エディターの概要

ユーザーが Web サイトをナビゲートすると、ページ エディターはユーザーが選択したアイテムを編集する機能を含むユーザー インターフェースの要素を重ね合わせます。

### 5.1.1 ページ エディターのデザイン ウィンドウ

ページ エディターではデザイン ウィンドウを使ってコンテンツ アイテムのレイアウト詳細を構成することができます。デザイン ウィンドウは「プレースホルダーの設定」を使って明示的に許可されるもの以外には、ユーザーがプレゼンテーション コンポーネントのバインドができないようにします。

#### メモ

デザイン ウィンドウはコンポーネントのネスト順を強制し、ユーザーがプレゼンテーション コンポーネントを存在しないプレースホルダーにバインドしないようにし、完全修飾のプレースホルダー キーを挿入します。

## 5.2 プレースホルダーの設定

プレースホルダーの設定はユーザーがページ エディターのデザイン ウィンドウでどのプレゼンテーション コンポーネントをプレースホルダーにバインドできるかを制御します。ページエディターのデザイン ウィンドウに関する詳細は「ページ エディターのデザイン ウィンドウ」のセクションを参照してください。

開発者は次のためにプレースホルダーの設定を使います：

- ユーザーがプレースホルダーにバインドする適切なプレゼンテーション コンポーネントを選ぶために役立てる
- ある種のユーザーが特定のプレースホルダーにコンポーネントをバインドできないようにする

レイアウト詳細はプレースホルダー キーまたは完全修飾プレースホルダー キーによってプレースホルダーを参照します。<sup>7</sup> たとえば、レイアウトで [main] キーをもつプレースホルダーにバインドされたサブレイアウトで [content] キーをもつプレースホルダーへの完全修飾プレースホルダー キーは /main/content です。

### メモ

ページ エディターのデザイン ウィンドウは完全修飾プレースホルダー キーを使用しますが、レイアウト詳細は完全修飾プレースホルダー キーを必要としません。

プレースホルダー キーと合致する名前のプレースホルダー設定定義アイテムは、レイアウト詳細がプレースホルダー設定を指定していない限り、それらのプレースホルダーに自動的に適用されます。content という名前のプレースホルダー設定定義アイテムは自動的に [content] というキーをもつすべてのプレースホルダーに適用されます。これにはネストされたプレースホルダーを含みますが、レイアウト詳細がプレースホルダー設定を指定している場合を除きます。プレースホルダーのデフォルト プレースホルダー設定定義アイテムの名前にはプレースホルダー キーを使用します。たとえば、[content] というキーをもつプレースホルダーには content という名前のプレースホルダー設定定義アイテムを編集します。または慣例により、スラッシュ文字 (“/”) をダッシュ文字 (“-”) で置換して完全修飾プレースホルダー キーを使います。たとえば、レイアウトの [main] キーをもつプレースホルダーにバインドするためのサブレイアウトの [content] キーをもつプレースホルダーを制御するには、content という名前のプレースホルダー設定定義アイテムを挿入するか、または main-content という名前のプレースホルダー設定定義アイテムを挿入し、このプレースホルダー設定定義アイテムをレイアウト詳細の完全修飾プレースホルダー キー /main/content と関連付けます。

開発者はプレースホルダー設定定義アイテムを使用してプレースホルダー設定を構成します。開発者はアイコンを各プレースホルダー設定定義アイテムにアサインすることにより各プレースホルダーに表示されるアイコンを制御することができます。開発者はプレースホルダー設定定義アイテムへの書き込みアクセスを制限することにより、どのユーザーがプレースホルダー コンポーネントをプレースホルダーにバインドするかを制限することができます。開発者はプレースホルダー設定定義アイテムの [概要] フィールドに HTML を入力して、ユーザーがマウスをページ エディターのデザイン ウィンドウのプレースホルダーの上で動かしたときに表示されるツールチップを制御することができます。

<sup>7</sup> レイアウト詳細とプレースホルダーに関する詳細は

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/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 マニュアル』を参照してください。

### 重要

Sitecore は対応するネストしたプレースホルダーへの完全修飾プレースホルダー キーを使用して、プレースホルダー設定を自動的に適用しません。たとえば、完全修飾キー `/main/content` を使ったプレースホルダーには、もしそれが存在した場合、`content` という名前のプレースホルダー設定定義アイテムは適用されますが、`main-content` という名前のプレースホルダー設定定義アイテムは適用されません。異なる用途のプレースホルダーに異なるプレースホルダー設定定義アイテムを適用するには、プレースホルダー キーにプレースホルダー設定を構成するか、またはレイアウト詳細を使って完全修飾プレースホルダー キーにプレースホルダー設定を構成します。

## 5.2.1 プレースホルダー設定定義アイテムの作成方法

プレースホルダー設定定義アイテムを作成する方法：

1. **コンテンツ エディター**で `/Sitecore/Layout/Placeholder Settings` アイテムを選択します。
2. **コンテンツ エディター**で `/System/Layout/Placeholder` データテンプレートを使ってプレースホルダー設定定義アイテムを挿入します。プレースホルダー設定定義アイテムにそれが制御するプレースホルダーのキーにちなんだ名前をつけます。
3. **コンテンツ エディター**でプレースホルダー設定定義アイテムを選択します。
4. **コンテンツ エディター**で [データ] セクションの [レンダリング許可] フィールドでユーザーがプレースホルダーにバインドできるプレゼンテーション コンポーネントを選択します。
5. **コンテンツ エディター**で [データ] セクションの [説明] フィールドでプレースホルダーのサマリーを入力します。レイアウトまたはサブレイアウト内にプレースホルダーの場所を示す図を含めることを検討してください。
6. **コンテンツ エディター**でプレースホルダー設定定義アイテムにアイコンを設定し、ページエディターのデザイン ウィンドウのプレースホルダー名の隣に表示されるイメージを制御します。アイテムにアイコンを設定する方法は「アイテムへのアイコンの設定方法」のセクションを参照してください。
7. どのユーザーがサブレイアウトとレンダリングをプレースホルダーにバインドできるかを制御するには、プレースホルダー設定定義アイテムへの書き込みアクセス権を構成します。

## 5.2.2 データ テンプレートまたは個々のアイテムへのプレースホルダー設定の構成方法

データ テンプレートまたは個々のアイテムへのプレースホルダー設定を構成する方法：

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**で標準値アイテムまたは個々のアイテムを選択します。
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 [プレゼンテーション] タブをクリックします。
3. **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**の [プレゼンテーション] タブで、[レイアウト] グループで [詳細] コマンドをクリックします。[レイアウト詳細] ダイアログが表示されます。

4. **[レイアウト詳細]** ダイアログで対象のデバイスの下で **[編集]** をクリックします。**[デバイス エディター]** が表示されます。
5. **[デバイス エディター]** で **[プレースホルダー設定]** タブをクリックし、**[追加]** をクリックします。**[プレースホルダー設定]** ダイアログが表示されます。
6. **[プレースホルダー設定]** ダイアログの **[キー]** でプレースホルダー キーまたは完全修飾プレースホルダー キーを入力します。
7. **[プレースホルダー設定]** ダイアログの **[設定アイテム]** で **[参照]** をクリックし、プレースホルダー設定定義アイテムを選択します。

## 5.3 ページ エディターのモード

各プレゼンテーション コンポーネントは Sitecore にコンポーネントを起動させたユーザー アクションに応じて論理ページ モードで実行されます。

プレゼンテーション コンポーネントが実行されるモードには次のものがあります：

- パブリッシュされた Web サイト
- ブラウザー ベースのデバッガー（プロファイリングとトレースがアクティブであるかどうかにかかわらず）
- ページ エディター（インライン編集がアクティブであるかどうかにかかわらず）
- プレビュー
- ページ エディターのデザイン ウィンドウ
- Web エディット クラシック モード

プレゼンテーション コンポーネントは異なるモードで異なるアウトプットを生成する場合があります。たとえば、プレゼンテーション コンポーネントは HTML<title> エlement にフィールドの値を投入する場合があります。ブラウザーはインライン編集機能を HTML<title> エlement に提供することはできません。もしページがそのフィールドをどこか他で使用しない場合は、ユーザーはフィールドをページ エディターで編集することはできません。この場合、ページの他の場所で使用されているプレゼンテーション コンポーネントはユーザーがインライン編集を使用している場合、ページのどこか他の場所で使用されている値を出力することができます。また別の例として、フィールド値が空の場合 <div> エlement を出力しないレンダリングを考慮してください。フィールドのインライン編集を提供するため、レンダリングはフィールドに値がある場合またはユーザーがページ エディターでインライン編集集中である場合、<div> とフィールド エディット コントロールを出力することができます。

### 5.3.1 ページ エディターのモードの決定方法

`sc:pageMode()` XSL 拡張関数または `Sitecore.Context.PageMode.NET` オブジェクトのプロパティを使ってページ エディターのモードを決定することができます。`sc:pageMode()` XSL 拡張関数と `Sitecore.Context.PageMode.NET` オブジェクトに関する詳細は「ページ エディターのモード」のセクションを参照してください。

たとえば、次の XSL コードは XSL レンダリングがページ エディターのなんらかのモードで稼働しているかどうか（サイトがパブリッシュされたのではなく）を決定します。

```
<xsl:if test="sc:pageMode()/pageEditor">
  <!-- the user is in the Page Editor.-->
</xsl:if>
```

次の C# コードは .NET コンポーネントがページ エディターのなんらかのモードのコンテキストで稼働しているかどうか（サイトがパブリッシュされたのではなく）を決定します。

```
if (Sitecore.Context.PageMode.IsPageEditor)
```

```
{
  // the user is in the page editor.
}
```

## メモ

ユーザーはページ エディターの複数のモードに同時にアクセスすることができます。

### 5.3.2 ページ エディターのモード

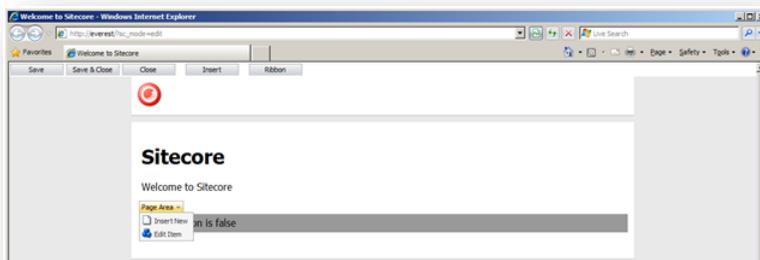
次のテーブルにページ モードを決定する XSL と .NET プログラミングコンストラクトを示します。

XSL	.NET	モードの説明
<code>sc:pageMode ()</code>	<code>Sitecore.Context.PageMode</code>	
<code>/pageEditor</code>	<code>IsPageEditor</code>	ページエディターにアクセスする
<code>/pageEditor/classic</code>	<code>IsPageEditorClassic</code>	Sitecore 5 のクラシックモードとコンテンツマーカーを使用する
<code>/pageEditor/design</code>	<code>IsPageEditorDesigning</code>	ページ エディターのデザイン ウィンドウを使用する
<code>/pageEditor/edit</code>	<code>IsPageEditorEditing</code>	ページ エディターのインライン編集
<code>/pageEditor/navigate</code>	<code>IsPageEditorNavigating</code>	ページ エディターのナビゲート中だがインライン編集ではない
<code>/preview</code>	<code>IsPreview</code>	プレビュー
<code>/normal</code>	<code>IsNormal</code>	パブリッシュされた Web サイト
<code>/profile</code>	<code>IsProfiling</code>	デバッグ、プロファイリング
<code>/debug</code>	<code>IsDebugging</code>	デバッグ

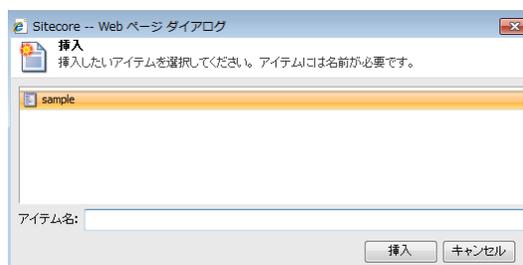
## 5.4 エディット フレーム

エディット フレームはユーザーがページ エディターでインライン編集中にマークアップ構造周辺でのコンテキストメニュー付きの挿入された境界線を提供します。エディット フレーム コマンドは Sitecore のユーザー インターフェイス コマンドを起動します。

<sup>8</sup> エディットフレームは <div> エレメント、およびその上にエディット フレームを生成します。メニューの中の各エディット フレーム コマンドが Sitecore コマンドを起動します。



エディット フレーム コントロールの [ボタン] プロパティは、エディット フレーム メニューに表れるエディットフレーム メニュー アイテムを表す子を含む Core データベースのアイテムを指定します。エディット フレーム メニューを指定しない場合、Sitecore は /Sitecore/Content/Applications/WebEdit/Edit Frame Buttons/Default で定義されるデフォルトのエディット フレーム メニューを使用します。デフォルトのエディット フレーム メニューは、ユーザーがエディット フレームのデータソースに定義された挿入オプションを使ってアイテムを挿入できるインサート エディット フレーム コマンドを含みます。



カスタムのエディット フレーム コマンドを含むエディット フレームの実装の例は「カスタムのフィールド エディターの実装方法」のセクションを参照してください。

---

<sup>8</sup> エディットフレームを使ってフィールド エディターを開く例については <http://trac.sitecore.net/DefaultFieldEditor/>を参照してください。

## メモ

`web.config` の `/configuration/sitecore/settings/setting` エレメントの `nameWebEdit.DefaultButtonPath` の `value` 属性を設定することによりデフォルト エディット フレーム メニューを構成することができます。

## メモ

Sitecore はネストしたエディット フレームはサポートしません。エディット フレームは他のどんなコントロールを含むこともできません。これにはインライン編集コントロールを生成する `FieldRenderer Web` コントロールを含みますが、他のエディット フレームを含むことはできません。<sup>9</sup>

### 5.4.1 エディット フレーム メニュー コマンドの実装方法

エディットフレームメニューコマンドを実装する方法：

1. Visual Studio の Web アプリケーション プロジェクトで `Sitecore.Shell.Applications.WebEdit.Commands.WebEditCommand` から継承するエディット フレーム メニュー コマンド クラスを作成します。
2. このクラスで `Execute()` メソッドを実装します。
3. エディット フレーム メニューにエディット フレーム メニュー コマンド定義アイテムを追加します。詳細は「エディット フレーム メニューの定義方法」のセクションを参照してください。

### 5.4.2 エディット フレーム メニューの定義方法

エディット フレーム メニューを定義する方法：

1. Sitecore デスクトップで Core データベースを使い、**コンテンツ エディター**で `/Sitecore/Content/Applications/WebEdit/Edit Frame Buttons` アイテムを選択します。Sitecore デスクトップでデータベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。
2. **コンテンツ エディター**で `/System/WebEdit/Edit Frame Button Folder` データ テンプレートを使ってエディット フレーム メニュー定義アイテムを挿入します。

## ヒント

または `/Sitecore/Content/Applications/WebEdit/Edit Frame Buttons/Default` エディット フレームメニュー定義を複製することもできます。

<sup>9</sup> `FieldRenderer Web` コントロールに関する詳細は <http://sdn.sitecore.net/Reference/References%20in%20Japanese/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 マニュアル』を参照してください。

3. **コンテンツ エディター**でエディット フレーム メニュー定義アイテムの下に /System/WebEdit/Edit Frame Small Button データ テンプレートを使って、一つまたは複数のエディット フレーム メニュー コマンド定義アイテムを挿入します。

#### ヒント

/Sitecore/Content/Applications/WebEdit/Edit Frame Buttons/Default/Insert アイテムを複製してエディット フレーム メニュー定義アイテムを作成することもできます。

#### ヒント

既存のエディット フレーム メニュー コマンド定義アイテムを複製しそのプロパティを更新することもできます。

#### ヒント

デフォルトのフィールド エディターのエディット フレーム メニュー コマンドを挿入するには「デフォルトのフィールド エディターの使用方法」のセクションを参照してください。

4. **コンテンツ エディター**で、各エディット フレーム メニュー コマンド定義アイテムで [データ] セクションの[ヘッダー] フィールドで、ページ エディターのメニュー コマンドに表示されるメニュー ラベルを入力します。
5. **コンテンツ エディター**で、各エディット フレーム メニュー コマンド定義アイテムで [データ] セクションで、アイコン フィールドをメニュー アイテムのアイコンに設定します。

#### ヒント

このアイコンをエディット フレーム メニュー コマンド定義アイテム自体に使うこともできます。<sup>10</sup>

6. **コンテンツ エディター**で、各エディット フレーム メニュー コマンド定義アイテムで [データ] セクションで、[クリック] フィールドをメニューコマンドが起動する UI コマンドに設定します。
7. **コンテンツ エディター**で、各エディット フレーム メニュー コマンド定義アイテムで [データ] セクションで、[ツールチップ] フィールドをメニュー アイテムのためのホバー テキストに設定し、エディット フレーム メニュー コマンド定義アイテムをセーブします。
8. Sitecore デスクトップで Master データベースを選択します。データベースを選択する方法は「Sitecore デスクトップでのデータベースの選択方法」のセクションを参照してください。

### 5.4.3 レイアウトまたはサブレイアウトにエディット フレームを挿入する方法

レイアウトまたはサブレイアウトにエディット フレームを挿入するには、次のようなコードを追加します：

```
<sc:editframe runat="server">
  ...
```

<sup>10</sup> アイテムにアイコンを設定する方法は

<http://sdn.sitecore.net/reference/References%20in%20Japanese/client%20configuration%20cookbook.aspx>を参照してください。

```
</sc:editframe>
```

「エディット フレーム プロパティ」セクションに説明されている方法で、エディット フレームのプロパティを設定することができます。

#### 5.4.4 XSL レンダリングにエディット フレームを挿入する方法

XSL レンダリングにエディット フレームを挿入するには、次のようなコードを追加します：

```
<sc:editFrame>
...
</sc:editFrame>
```

「エディット フレーム プロパティ」セクションに説明されている方法でエディット フレームのプロパティを設定することができます。

#### 5.4.5 エディット フレーム プロパティ

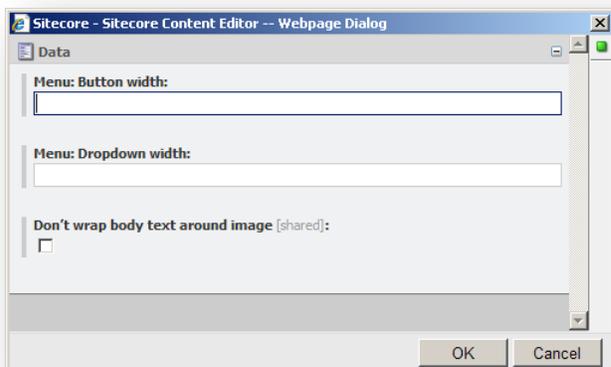
次のようなエディット フレームのプロパティを設定することができます。

プロパティ (.NET)	属性 (XSL)	説明
Buttons	Buttons	エディット フレーム メニュー アイテムが利用出来るコマンドを定義する
CssClass	class	エディット フレームで生成された <div> エレメントの class 属性の値
DataSource	select	エディット フレーム メニュー コマンドに渡されるアイテム
Debug	debug	Sitecore がエディット フレームによって生成された <div> エレメントを整形するかどうか制御する
FramePadding	frame-padding	コンテンツのまわりに表示されるエディットフレームを示されるピクセル数だけ拡張する
HintOffset	hint-offset	エディット フレーム ホバーリングメニューとエディット フレーム領域の間のピクセル数

Style	style	エディット フレームで生成された <div> エレメントの style 属性の値
Title	title	エディット フレーム メニューのタイ トル
Tooltip	Tooltip	エディット フレーム メニューのツ ールチップ

## 5.5 フィールド エディター

フィールド エディターはページ エディターの中でユーザーにページ エディターがインライン編集コントロールを提供しないフィールドを編集できるインターフェースを提供します。フィールド エディターのユーザー インターフェースはコンテンツ エディターのユーザー インターフェースと比較するとかなり簡素化されています。



エディット フレーム コマンドを構成してフィールド エディターを起動することができます。Sitecore は指定したフィールドを編集できるフィールド エディターを起動するために使用できるエディット フレーム コマンドを提供しています。フィールド エディターをページ エディターの中から起動してインライン編集コントロールを提供しないフィールドを編集することができます。また一つ以上のアイテムからのフィールドを同時に編集することができます。<sup>11</sup> エディット フレームに関する詳細は「エディット フレーム」のセクションを参照してください。

### 注意

フィールド エディターはアイテムの検証を強制しません。

<sup>11</sup> 標準のテンプレートでは定義されずインライン編集をサポートしないアイテムのすべてのフィールドを含むフィールドエディターを開くためにエディットフレームメニューコマンドを提供する例については、<http://trac.sitecore.net/DefaultFieldEditor/>を参照してください。

### 5.5.1 デフォルトのフィールド エディターの使用方法

デフォルトのフィールド エディターは指定したフィールドを含みます。

デフォルトのフィールド エディターを使用する方法：

1. フィールド エディター メニューを定義します。フィールド エディター メニューを定義する方法は「[エディット フレーム メニューの定義方法](#)」のセクションを参照してください。
2. **コンテンツ エディター**でフィールド エディター メニュー定義アイテムの下に System/WebEdit/Field Editor Button データ テンプレートを使って、デフォルトのフィールド エディター コマンド定義アイテムを挿入します。

#### メモ

このフィールド エディター コマンド定義アイテムは `webedit:fieldeditor` コマンドを起動します。

3. **コンテンツ エディター**でデフォルト フィールド エディター コマンド定義アイテムで **[データ]** セクションの **[ヘッダー]** フィールドで編集アイテムを入力します。
4. **コンテンツ エディター**で、デフォルト フィールド エディター コマンド定義アイテムで **[データ]** セクションで、**[アイコン]** フィールドに `people/16x16/cubes_blue.png` を設定します。アイテムにアイコンを設定する方法は「[アイテムへのアイコンの設定方法](#)」のセクションを参照してください。

#### ヒント

このアイコンをエディット フレーム メニュー コマンド定義アイテム自体に使うこともできます。

5. **コンテンツ エディター**で、デフォルト フィールド エディター コマンド定義アイテムで **[データ]** セクションの**[フィールド]** フィールドで、フィールド エディターに含むフィールドの名前または ID をパイプ文字 ('|') 区切りで入力します。

#### メモ

フィールド名は大文字と小文字を区別しません。

#### メモ

指定されたフィールドのいずれかが存在しない場合、Sitecore はエラーを出しません。

6. Sitecore デスクトップでマスター データベースを選択します。データベースを選択する方法は「[Sitecore デスクトップでのデータベースの選択方法](#)」のセクションを参照してください。

### 5.5.2 カスタムのフィールド エディターの実装方法

フィールドを動的に含むフィールドエディターを実装する方法：

1. Visual Studio の Web アプリケーションプロジェクトで  
`Sitecore.Shell.Applications.WebEdit.Commands.FieldEditorCommand` クラスから継承するクラスを作成します。
2. このクラスで、フィールド エディターに含むフィールドのリストを作成する `GetOptions()` メソッドを実装します。
3. `GetOptions()` メソッドで  
`Sitecore.Shell.Applications.WebEdit.PageEditFieldEditorOptions` クラスのコンストラクターにフィールドのリストを渡します。
4. `GetOptions()` メソッドで  
`Sitecore.Shell.Applications.WebEdit.PageEditFieldEditorOptions` オブジェクトの追加プロパティをセットします。
5. `GetOptions()` メソッドで  
`Sitecore.Shell.Applications.WebEdit.PageEditFieldEditorOptions` オブジェクトを返します。
6. 適当なエディット フレーム メニュー定義にフィールド エディター コマンドを追加します。エディット フレーム メニュー定義アイテムにコマンドを追加する方法は「エディット フレーム メニューの定義方法」のセクションを参照してください。
7. エディット フレーム メニューを使用するには、レイアウトまたはサブレイアウトまたは XSL レンダリングでエディット フレームを構成します。レイアウト、サブレイアウト、XSL レンダリングを構成してエディット フレーム メニューを使用する方法は「レイアウトまたはサブレイアウトにエディット フレームを挿入する方法」または「XSL レンダリングにエディット フレームを挿入する方法」のセクションを参照してください。

#### メモ

フィールド エディターはすべてのフィールドを表示された 1 つのセクションにマージします。

`Sitecore.Shell.Applications.WebEdit.PageEditFieldEditorOptions` オブジェクトのタイトルとアイコン プロパティを設定してこのセクションに使われるアイコンラベルを制御することができます。または `PreserveSections` プロパティを `[true]` に設定し、元のセクション構造を保ちます。

#### メモ

`Sitecore.Shell.Applications.WebEdit.PageEditFieldEditorOptions` オブジェクトの `DialogTitle` プロパティを設定してフィールド エディター ブラウザー ウィンドウのタイトルを制御することができます。

## Chapter 6

# セキュリティの構成機能

この章ではコンテンツオーサーのユーザー エクスペリエンスを最適化するために開発者が使用できるセキュリティ機能（Sitecore セキュリティロールを含む）を説明します。

この章には次のセクションがあります。

- Sitecore のクライアント セキュリティ ロール
- セキュリティのプリセット
- データ テンプレートのフィールドのセキュリティ

## 6.1 Sitecore のクライアント セキュリティ ロール

セキュリティ管理者は CMS ユーザーを 1 つまたは複数の Sitecore クライアントセキュリティロールに関連付け、Sitecore ユーザーインターフェースから使用できる機能を制御します。

Sitecore アプリケーションはすべての機能をすべてのユーザーに見せるわけではありません。使用できる機能を制限することには下記のようなメリットがあります：

- ユーザーインターフェースのエレメントの数を減らすことでパフォーマンスとユーザビリティが改善される
- 選択肢を減らすことでユーザーが不注意によって機能を実行することを防ぐことができる
- 不要な機能を非表示にすることでトレーニング要件を最小化することができる

Sitecore の各ロールに関連付けられた機能のリストは、この文書では対象外として掲載していません。

次の表はデフォルトの各 Sitecore クライアント セキュリティ ロールのメンバーシップに提供されているオプションの要約です。

ロール	機能
Sitecore Account Managing	ユーザー、ロール、ドメイン管理機能にアクセスする
Sitecore Client Authoring	コンテンツ エディター、ページ エディター、その他のコンテンツ オーサーのためのアプリケーションのコンテンツ オーサリング機能にアクセスする
Sitecore Client Configuring	コンテンツ エディターのアイテムの構成機能にアクセスする
Sitecore Client Designing	コンテンツ エディターとページ エディターのデザイン機能にアクセスする
Sitecore Client Developing	デベロッパー センターとその他の開発機能にアクセスする
Sitecore Client Maintaining	データテンプレート管理機能にアクセスする
Sitecore Client Publishing	サイトとアイテムのパブリッシュ機能にアクセスする
Sitecore Client Securing	セキュリティ エディターとアクセス権をアサインすることに関連付けられたその他の機能にアクセスする
Sitecore Client Translating	コンテンツの翻訳機能にアクセスする

ロール	機能
Sitecore Client Users	ユーザーに Sitecore クライアントへのログインを許可する。他のすべての Sitecore クライアントロールは Sitecore Client Users ロールのメンバーである。どの Sitecore クライアントロールのユーザーも自動的に Sitecore Client Users ロールのメンバーとなる。

セキュリティ管理者は下記のロールを使って Sitecore Client Authoring ロールのメンバーの利用できる機能を制限します。管理者は Sitecore Client Authoring ロールをアサインしたあとでこれらのロールのメンバーにすることで、より使用できる機能に制限のあるユーザーにすることができます。

ロール	機能
Sitecore Limited Content Editor	機能を単純化されたコンテンツ エディター インターフェースに限定する
Sitecore Limited Page Editor	機能を単純化されたページ エディター インターフェースに限定する
Sitecore Minimal Page Editor	ページエディターからリボンを削除する

次のロールはデフォルトのアクセス権をいろいろなデータベースのブランチに与え、関連する Sitecore のクライアント ロールのメンバーに含めることで、セキュリティ管理を簡素化します。

ロール	機能
Author	コンテンツツリーの <code>/sitecore/content</code> ブランチにアクセス権を与える。Sitecore Client Authoring のメンバー。
Designer	デザイン変更をする際に必要なコンテンツツリーの部分へのアクセス権を与える。Sitecore Client Designing のメンバー。
Developer	サイトの開発とメンテナンスをする際に必要なコンテンツツリーの部分へのアクセス権を与える。Author、Designer、Sitecore Client Developing、Sitecore Client Maintaining、Sitecore Client Configuring のメンバー。

Sitecore は Online Marketing Suite (OMS) の機能へのアクセスを制御するデフォルトのロールも提供しています。

ロール	機能
Analytics Content Profiling	アナリティクス プロファイルのメンテナンスへのアクセス権を与える。
Analytics Maintaining	マーケティング センターへのアクセス権を与える。
Analytics Reporting	Sitecore アナリティクス レポートへのアクセス権を与える。

デフォルトの Sitecore のロールがあなたのユーザー インターフェースの制御要件に合致していない場合は、Core データベースのアクセス権によって Sitecore が各ユーザーまたは各ロールに提示する機能のより精密な制御を行うことが可能です。

#### 重要

ユーザー インターフェースの不要な機能を取り除き、ユーザーが誤って機能を起動する可能性を最小化するため、ユーザーはできるだけ最小限のロールのメンバーとします。このロールの唯一の例外は先に説明した限定的な最小限のロールです。ユーザーが必要な場合にはこれらのロールのメンバーとなることができます。多くのユーザーは Sitecore Client Authoring ロールのメンバーのみになるのがよいでしょう。ユーザーに追加の機能が必要な場合にセキュリティ管理者が追加のロールを与えることができます。

#### 重要

管理者はロールのメンバーシップにかかわらず、すべての機能へのアクセス権を持ちます。管理者の数は最小限にします。

## 6.2 セキュリティのプリセット

セキュリティのプリセットを使うとユーザーはセキュリティ管理者によってあらかじめ決められたアイテムへのアクセス権のリストを適用することができます。

セキュリティ管理者は次のためにセキュリティのプリセットを使います：

- エンドユーザーのコンテンツへのアクセス権の適用のプロセスを簡素化する
- ユーザーがコンテンツに適用する共通のアクセス権のリストを定義する
- 1 クリックのユーザー インターフェイス コンポーネントを提供し、そうでなければ複数のアクションが必要なアクセス権のリストを適用する

セキュリティのプリセットにはコンテンツ エディターの [プリセット] グループの [セキュリティ] タブから、またセキュリティ エディターの [プリセット] グループからアクセスできます。

### ヒント

セキュリティ管理を最小化するため、可能な限りセキュリティの継承を使用します。

### 6.2.1 セキュリティのプリセットの作成方法

セキュリティ プリセット定義アイテムの名前には、リボンの中からこのセキュリティ プリセットを起動するコマンドにふさわしい値を使います。

ユーザー インターフェイスからアクセス権を定義してセキュリティ プリセットを作成する方法：

1. **コンテンツ エディター**で `/Sitecore/System/Settings/Security/Presets` アイテムを選択します。
2. **コンテンツ エディター**で `/System/Security/Security Preset` データ テンプレートを使ってセキュリティ プリセット定義アイテムを挿入します。
3. **コンテンツ エディター**で [データ] セクションの [セキュリティ プリセット] フィールドでアクセス権を定義します。

既存のソースアイテムからアクセス権をコピーしてセキュリティ プリセットを作成する方法：

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**でソース標準値アイテムまたは個々のアイテムを選択します。
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**でスタンダード フィールドを表示します。スタンダード フィールドを表示する方法は「スタンダード フィールドを表示または非表示にする」のセクションを参照してください。
3. **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 RAW フィールド値を表示します。RAW フィールド値を表示する方法は「RAW 値 RAW 値を表示または非表示にする」のセクションを参照してください。

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**の **[セキュリティ]** セクションの **[セキュリティ]** フィールドで、値を選択し、それを Windows のクリップボードにコピーします。
5. **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 **[スタンダード フィールド]** を非表示にします。スタンダード フィールドを非表示にする方法は「RAW 値 RAW 値を表示または非表示にする」のセクションを参照してください。
6. **コンテンツ エディター**で `/Sitecore/System/Settings/Security/Presets` アイテムを選択します。
7. **コンテンツ エディター**で `/System/Security/Security Preset` データ テンプレートを使ってセキュリティ プリセット定義アイテムを挿入します。
8. **コンテンツ エディター**で **[セキュリティ]** セクションの **[セキュリティ]** フィールドで Windows クリップボードの値を貼り付けます。
9. **コンテンツ エディター**で RAW フィールド値を非表示にします。RAW フィールド値を非表示にする方法は「RAW 値を表示または非表示にする」のセクションを参照してください。

## 6.2.2 セキュリティのプリセットの適用方法

デフォルトではセキュリティ プリセットを適用するにはユーザーは管理者または Sitecore Client Security ロールのメンバーである必要があります。

**コンテンツ エディター**または**セキュリティ エディター**を使ってセキュリティプリセットを適用します。

### ヒント

複数のアイテムにアクセス権を適用する場合にはセキュリティ エディターを使用します。

**コンテンツ エディター**を使用してセキュリティプリセットを適用するには：

1. **コンテンツ エディター**でアイテムを選択します。
2. **コンテンツ エディター**で **[セキュリティ]** タブをクリックします。
3. **コンテンツ エディター**で **[セキュリティ]** タブの **[プリセット]** グループで **[セキュリティ プリセット]** コマンドをクリックします。Sitecore はセキュリティ プリセット定義アイテムの **[セキュリティ プリセット]** フィールドで定義されたアクセス権を選択されたアイテムにコピーします。

**セキュリティ エディター**を使用してセキュリティプリセットを適用する方法：

1. Sitecore デスクトップの左下の隅で、**Sitecore** ボタンをクリックし、次に **[セキュリティ エディター]** をクリックします。**セキュリティ エディター**が表示されます。
2. **セキュリティ エディター**でアイテムを選択します。

3. **コンテンツ エディター**で **[セキュリティ]** タブの **[プリセット]** グループで **[セキュリティ プリセット]** コマンドをクリックします。Sitecore はセキュリティ プリセット定義アイテムの **[セキュリティ プリセット]** フィールドで定義されたアクセス権を選択されたアイテムにコピーします。

## 6.3 データ テンプレートのフィールドのセキュリティ

データ テンプレートのフィールドのセキュリティは、アイテムへのアクセスを制御するセキュリティ権限と、そのアイテムの個々のフィールド値へのアクセスを制御するセキュリティ権限とを区別します。

開発者は次のためにフィールド セキュリティを使います：

- あるユーザーの編集インターフェースにあるフィールドを非表示にする
- あるユーザーの編集インターフェースにあるフィールドを読み取り専用にする
- あるアカウントにはフィールド値の更新を許可し、他のアカウントにはこれらの値の更新ができないようにする

たとえば、SEO (search-engine optimization 検索エンジン最適化) のために、データ テンプレートフィールドの値を使って HTML の <title> エlementを生成する組織では、他のフィールドへの書き込みアクセス権をもつユーザーがそのフィールドへの書き込みアクセスは不要かもしれません。それらのアイテムへの書き込みアクセスをもつある種のユーザーのみが SEO に用いるフィールドを更新できるべきでしょう。フィールド セキュリティを使用してこれらの要件を満たすことができます。

デフォルトではアイテムのアクセス権がアイテムのすべてのフィールドのアクセス権を定義します。あるフィールドがアクセス権を定義すると、ユーザーは明示的なフィールド セキュリティ アクセスとフィールドを含むアイテムへのアクセスを必要とします。

### データ テンプレートのフィールド セキュリティの構成方法

データ テンプレートのフィールド セキュリティを構成する方法：

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**でデータ テンプレート フィールド定義アイテムを選択します。
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 **[セキュリティ]** タブをクリックします。
3. **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**の **[セキュリティ]** タブで、**[セキュリティ]** グループで **[割り当て]** コマンドをクリックします。アクセス権ダイアログが表示されます。
4. アクセス権ダイアログでフィールドの読み取りとフィールドの書き込みの権利を一つまたは複数のアカウントに許可または不許可にします。

## Chapter 7

# Sitecore クライアント RSS フィード

この章では Sitecore クライアントの RSS フィード機能について説明します。<sup>12</sup> RSS フィードは情報を標準の XML フォーマットで配信します。Sitecore クライアントの RSS フィードは CMS ユーザーに RSS の XML フォーマットで情報と機能を公開します。CMS ユーザーは任意の RSS リーダーを使用して Sitecore クライアントの RSS フィードにアクセスすることができます。

この章はまず Sitecore クライアント RSS 機能の概要を説明し、次に Sitecore クライアント RSS 構成オプションを説明します。

この章には次のセクションがあります。

- Sitecore クライアント RSS フィードの概要
- Sitecore クライアント RSS フィードの構成

---

<sup>12</sup> Sitecore RSS 機能についての詳細は

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/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、また <http://sdn.sitecore.net/Reference/References%20in%20Japanese.aspx> から『コンテンツ オーサー クックブック』を参照してください。

## 7.1 Sitecore クライアント RSS フィードの概要

Sitecore クライアントの RSS フィードは CMS 機能を RSS (Really Simple Syndication) フィードとして公開します。<sup>13</sup>

Sitecore は Core データベースの `/Sitecore/Content/Applications/Syndication/Feeds` アイテムの下に Sitecore クライアント RSS フィード定義アイテムを管理します。

### 7.1.1 ワークフロー フィード

ワークフロー フィードはワークフローにあるアイテムについての情報を公開します。

ワークフロー フィードはワークフローにある各アイテムの各状態の変化の RSS エントリーを含みます。

ワークフロー フィードはワークボックスに示される各ワークフローのタイプの RSS アイコンとして見えます。

### 7.1.2 ワークフロー状態フィード

ワークフロー状態フィードはあるワークフロー状態に入るまたは出るアイテムについての情報を公開します。

ワークフロー状態フィードはワークフロー上の各アイテムのワークフロー状態を出入するそれぞれの時点の RSS エントリーを含みます。

ワークフロー状態フィードはワークボックスに示される各ワークフローの各状態の RSS アイコンとして見えます。

### 7.1.3 アイテム更新フィード

アイテム更新フィードはアイテムの更新についての情報を公開します。またオプションとして、その子孫の更新についても公開します。

ワークフローに関連付けられたアイテムについては、アイテム更新フィードは各ワークフロー状態の変更の RSS エントリーを含みます。ワークフローに関連付けられていないアイテムについては、アイテム更新フィードは各言語の新バージョンごとの RSS エントリーを含みます。

アイテム更新フィードは **コンテンツ エディター** の **[レビュー]** タブの **[校正]** グループの **[購読]** コマンドで表示されます。アイテムの変更またはアイテムとその子孫の変更の購読を選択することができます。

---

<sup>13</sup> RSS に関する詳細は [http://en.wikipedia.org/wiki/RSS\\_\(file\\_format\)](http://en.wikipedia.org/wiki/RSS_(file_format)) を参照してください。Sitecore RSS 機能についての詳細は <http://sdn.sitecore.net/Reference/References%20in%20Japanese/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。

## 7.2 Sitecore クライアント RSS フィードの構成

本セクションは Sitecore クライアントの RSS フィードの構成について説明します。

### 7.2.1 ClientFeeds.MaximumItemsInFeed 設定

web.config の /configuration/sitecore/settings/setting エLEMENT の name ClientFeeds.MaximumItemsInFeed の value 属性が Sitecore クライアント RSS フィードに含むアイテムの最大数を指定します。

### 7.2.2 ClientFeeds.ItemExpiration 設定

web.config の /configuration/sitecore/settings/setting エLEMENT の name ClientFeeds.ItemExpiration の value 属性が Sitecore クライアント RSS 配信アイテムの有効期限を制御します。ClientFeeds.ItemExpiration 設定に値としてゼロ (“0”) を設定するとアイテムは期限が切れることはありません。この値として正数を設定すると、管理されている Web サイトの RSS フィード アイテムは指定された日数ののち、RSS フィードから有効期限切れとなります。