

## Sitecore CMS 6.4 – 6.5

# データ定義リファレンス

CMS 管理者、アーキテクト、開発者のためのコンセプトの概要

## 目次

第 1 章	イントロダクション .....	5
1.1	機能 .....	6
1.2	共通の用語 .....	7
1.2.1	テンプレート .....	7
1.2.2	データ インフラストラクチャ .....	7
1.2.3	データ表現 .....	7
1.2.4	データレポジトリ .....	7
1.3	一般情報 .....	8
1.3.1	命名規則 .....	8
	表示名およびフィールド タイトル .....	8
1.3.2	テンプレートのストア場所 .....	8
第 2 章	データテンプレート .....	9
2.1	データテンプレート .....	10
2.1.1	データテンプレート フィールド .....	11
2.1.2	データテンプレート セクション .....	11
	ベーステンプレート .....	12
	多重継承 .....	13
	循環継承 .....	14
2.2	標準値 .....	15
2.2.1	\$name トークン .....	16
2.2.2	ブランクと NULL のフィールド値 .....	16
第 3 章	標準テンプレート .....	17
3.1	標準テンプレートの概要 .....	18
3.2	標準テンプレートの構造 .....	19
3.2.1	[高度な設定] セクション .....	19
3.2.2	[アピアランス (外観と動作)] セクション .....	19
3.2.3	[ヘルプ] セクション .....	20
3.2.4	[挿入オプション] セクション .....	20
3.2.5	[レイアウト] セクション .....	20
3.2.6	[有効期間] セクション .....	20
3.2.7	[パブリッシュ] セクション .....	21
3.2.8	[セキュリティ] セクション .....	21
3.2.9	[統計] セクション .....	21
3.2.10	[タスク] セクション .....	21
3.2.11	[バリデーター] セクション .....	22
3.2.12	[ワークフロー] セクション .....	22

第 4 章	テンプレートフィールド	23
4.1	テンプレートフィールド定義アイテムについて	24
4.1.1	テンプレートフィールドテンプレート	24
	[フィールドネーム]	24
	[タイプ]	24
	[タイトル]	25
	[ソース]	25
	[Blob (バイナリラージオブジェクト)]	26
	[共有]	26
	[バージョン管理なし]	26
	[デフォルト値]	27
	[検証]	27
	[検証テキスト]	27
	[セキュリティフィールド]	27
	[空白に戻す]	27
	[テキスト検索対象外]	27
	[ページエディターボタン]	27
4.2	データテンプレートのフィールドタイプ	28
4.2.1	Analytics フィールドタイプ	28
4.2.2	Simple フィールドタイプ	28
	Checkbox フィールドタイプ	29
	Date および Datetime フィールドタイプ	29
	File フィールドタイプ	30
	Image フィールドタイプ	30
	Integer、Number、Password、Single-Line Text フィールドタイプ	31
	Multi-Line Text フィールドタイプ	31
	Rich Text フィールドタイプ	31
	Word Document フィールドタイプ	32
4.2.3	List フィールドタイプ	32
	Checklist フィールドタイプ	33
	Droplist フィールド	33
	Grouped Droplink フィールドタイプ	34
	Grouped Droplist フィールドタイプ	34
	Multilist フィールドタイプ	34
	Name Value List フィールドタイプ	35
	Treelist および TreelistEx フィールドタイプ	35
4.2.4	Link フィールドタイプ	36
	Droplink フィールドタイプ	36
	Droptree フィールドタイプ	36
	General Link フィールドタイプ	36

Version Link フィールド タイプ.....	36
4.2.5 Developer フィールド タイプ.....	37
Icon フィールド タイプ.....	37
IFrame フィールド タイプ.....	37
Tristate フィールド タイプ.....	37
4.2.6 System フィールド タイプ.....	38
Attachment フィールド タイプ.....	38
Custom フィールド タイプ.....	38
File Drop Area (FDA) フィールド タイプ.....	38
Internal Link フィールド タイプ.....	39
Layout フィールド タイプ.....	40
Rules フィールド タイプ.....	40
Security フィールド タイプ.....	40
Template Field Source フィールド タイプ.....	40
Thumbnail フィールド タイプ.....	41
4.2.7 非推奨のフィールド タイプ.....	41
4.3 Sitecore クエリ.....	42
4.3.1 一般的な構文.....	42
4.3.2 軸.....	44
4.3.3 演算子.....	44
4.3.4 内部アーキテクチャ.....	46
第 5 章 フィールドとアイテムの検証.....	47
5.1 検証のルールとオプションについて.....	48
5.2 デフォルトの検証ルール.....	49
5.2.1 アイテム検証ルール.....	49
5.2.2 フィールド検証ルール.....	49
5.2.3 システム フィールド検証ルール.....	50
5.3 検証オプション.....	51
第 6 章 挿入オプション.....	52
6.1 挿入オプション.....	53
6.1.1 効率的な挿入オプション.....	53
6.1.2 割り当てられた挿入オプション.....	53
6.1.3 挿入ルール.....	54
6.1.4 挿入オプション ルール.....	54
6.1.5 uiGetMasters パイプライン.....	54
6.2 ブランチ テンプレート.....	55
6.2.1 ブランチ テンプレートの使用.....	55
ブランチ テンプレートを使用した結果.....	56
6.3 コマンド テンプレート.....	57
6.3.1 コマンド テンプレートの使用.....	57

# 第 1 章

## イントロダクション

この文書では、Sitecore Web サイトに関連するインフォメーション インフラストラクチャのデザイン、実装、およびメンテナンスを行う際にアーキテクト、デベロッパー、および CMS 管理者が知っておくべき概念について説明します。

この文書には次の章があります：

- 第 1 章 — イントロダクション
- 第 2 章 — データテンプレート
- 第 3 章 — 標準テンプレート
- 第 4 章 — テンプレートフィールド
- 第 5 章 — フィールドとアイテムの検証

この章には次のセクションがあります：

- 機能
- 共通の用語
- 一般情報

## 1.1 機能

本書では、Web サイトのインフォメーション アーキテクチャを定義するとき使用するコンポーネントと概念について詳細に説明します。この文書では、次の機能と目的を詳細に説明します：

**テンプレート** - Sitecore ユーザーは、データテンプレート、ブランチ テンプレートおよびコマンドテンプレートの 3 つのテンプレートのいずれかを使用してアイテムを作成します。データ テンプレートは、アイテムを構築するときのフレームワークとなります。データ テンプレートを使用すると、データの入力方法を制御するフィールドを定義し、他のテンプレートから継承して再利用することができます。

**フィールド** - システムに入力されるデータを制御する領域です。フィールドを**セクション**にグループ化すると、システムに入力されるデータの量と型を編成して制御することができます。フィールドは、フィールドタイプ別に編成されます。

**フィールドタイプ** - データを入力する、またはフィールドで選択するための各種の方法です。システムにデータを入力する方法はさまざまで、フィールドで入力または選択できるデータ型を制御するのがフィールドタイプです。

データはシステム全体で検証されます。検証の種類、および各検証がデータ定義構造の妥当性と正当性を管理するために、システムのデータ定義領域の全体を網羅している点についても説明します。

標準値を使用すると、新しく作成したアイテムにデフォルト データを自動的に挿入することができます。標準値は、既存のアイテムの NULL フィールドに自動的に投入される、または新しく作成したアイテムにデータを指定するデフォルト値です。

**ブランチ テンプレート** - アイテムを 1 つずつではなく、複数のアイテムを一度に作成することができます。

**コマンドテンプレート** - 事前定義された構造ではなく、ロジックに従ってアイテムを挿入することができます。

## 1.2 共通の用語

このセクションでは、よく使用されるデータ定義のいくつかの概念について説明します。

### 1.2.1 テンプレート

Sitecore で使用する「テンプレート」という用語は、Sitecore で新しいデータを作成するときに使用するコンポーネントを指します。データ テンプレート、ブランチ テンプレートおよびコマンド テンプレートという用語は、それぞれ特定の種類のテンプレートを指します。テンプレートは、システムに入力されるデータのスキーマを制御します。

### 1.2.2 データ インフラストラクチャ

「データ インフラストラクチャ」は、データ構造を決定する Sitecore ソリューションの要素を表す用語です。データ インフラストラクチャには、標準値を含むすべての種類のテンプレート、セキュリティなど Web サイトのインフォメーション アーキテクチャ、ユーザーが作成できる対象を制御する挿入オプション、およびその他システムの構造上の要素が含まれます。Sitecore では、データ インフラストラクチャと、コンテンツ (すなわちその構造に入力されるデータ) が区別されます。

### 1.2.3 データ表現

Sitecore のデータ表現には、次のような特徴があります：

- 柔軟 - Sitecore では、使用方法の簡単なブラウザーベースのユーザー インターフェースを使用してデータ構造を定義することができます。
- 抽象化 - データ自体には影響せずに、データ構造定義を変更することもできます。
- 階層型 - データは、暗黙的および明示的な関係を使用して構造化されます。
- XML 指向 - データは XML 表現でも提供されます。
- オブジェクト指向 - Sitecore では、階層型のオブジェクト指向の表現でもデータを提供します。

### 1.2.4 データレポジトリ

Sitecore のデータ レポジトリには、さまざまなデータ サービスの機能があります。データをあらゆる言語で表示することができる一方、厳格なバージョンingとセキュリティで管理されます。システムにおけるデータ フローは、ワークフローに組み込まれる多様なカスタム ルールを使用して設定することができます。また、データの "ライブ" の Web サイトへの公開は、データの公開方法を厳密に監視および管理する各種の設定で変更が可能です。

## 1.3 一般情報

このセクションでは、データ定義に関する一般情報を示します。

### 1.3.1 命名規則

テンプレート、フィールド、フィールド セクションに名前を付けるときは、簡潔で関連性のわかる、理解しやすい名前を使用してください。デフォルトでは、作成時に指定した名前が技術者ユーザーにも一般ユーザーにも表示されます。コンテンツ オーサーなどのビジネス ユーザーにもわかりやすい名前を選択してください。

#### 表示名およびフィールド タイトル

定義の際に Sitecore のユーザー インターフェイスに表示されるのは、実際のアイテム名ではなく表示名およびフィールド タイトルです。表示名の詳細については、『クライアント構成クックブック』を参照してください。

#### メモ

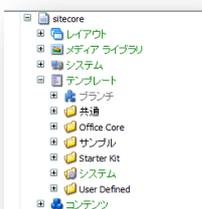
コードはフィールドを名前で参照するため、フィールドには変数やオブジェクトのプロパティに対応した名前を付けてください。ユーザーにとってわかりやすいラベルを使用するには、フィールド タイトルを使用します。

#### ヒント

種類を示す接頭辞または接尾辞を推奨する命名規則は避けるようにしてください。

### 1.3.2 テンプレートのストア場所

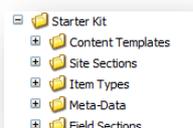
Sitecore における各種のテンプレートは、Master データベースの `/Sitecore/Templates` テンプレートの下にアイテムとしてストアされます。



#### 注意

Sample および User Defined フォルダの下にあるファイルを除いて、`/Sitecore/Templates` アイテムの下にあるフォルダーやデータ テンプレートは、名前を変更したり、削除したり、その他の変更を加えたりしないでください。

Web サイトごと、機能ごと、またはその他の基準でテンプレートを分類するときは、テンプレート フォルダーを作成します。次の図に、Sitecore Starter Kit のテンプレート フォルダ構造を示します。



## 第 2 章

# データ テンプレート

この章では、データ テンプレートと標準値について説明します。データ テンプレートは、あるタイプのアイテムの構造を多数のフィールドとして定義します。データ テンプレートの標準値アイテムが、そのデータ テンプレートで定義されているフィールドのデフォルト値になります。

この章には次のセクションがあります:

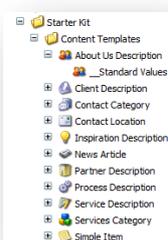
- データ テンプレート
- 標準値

## 2.1 データ テンプレート

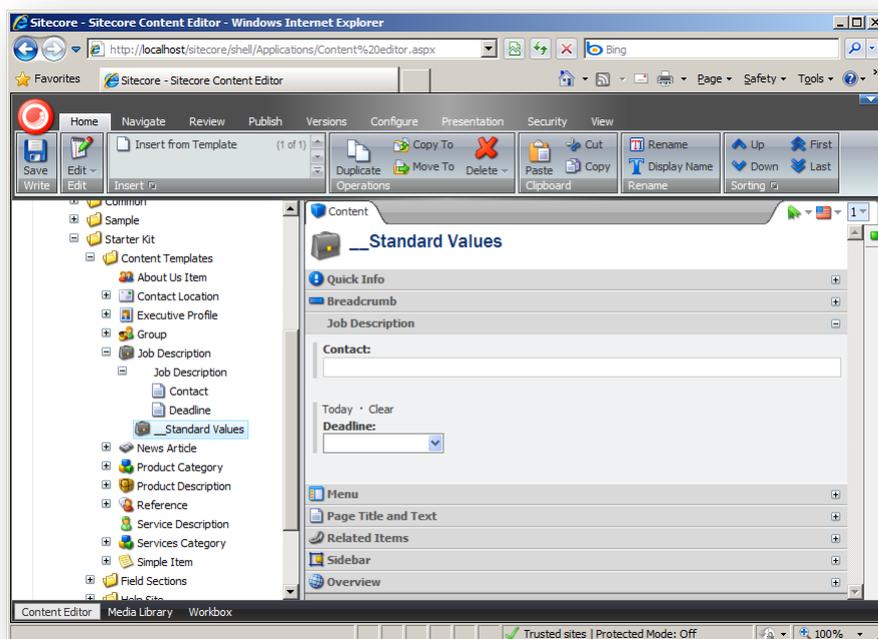
データ テンプレートは、データ型を定義します。Sitecore では、データ テンプレートを各アイテムに関連付けます。データ テンプレートは、そのデータ テンプレートに関連付けられたすべてのアイテムの構造を定義します。

データ テンプレートには多くのデータ テンプレート セクションがあり、そのそれぞれに多くのデータ テンプレート フィールドがあります。データ テンプレート セクションの詳細については、「データ テンプレート セクション」のセクションを参照してください。データ テンプレート フィールドの詳細については、「データ テンプレート フィールド」のセクションを参照してください。データ テンプレート セクションによって、ユーザー インターフェイスを編集する際にデータ テンプレート フィールドが視覚的に整理されます。データ テンプレート フィールドは、そのデータ テンプレートに基づいてアイテムの構造を定義します。

データ テンプレートは、定義アイテムの階層として表されます。ルート アイテムがテンプレートを、その子がフィールド セクションを、その孫がフィールドをそれぞれ定義します。



この例では、Job Description というデータ テンプレートが開かれて Job Description というフィールド セクションがその直下にあり、このフィールド セクションの下に Contact および Deadline の各フィールドがあります。これらは、このデータ テンプレートに関連付けられたアイテムを編集するときに表示されるセクションおよびフィールドに対応しています。次の図は、Job Description データ テンプレートの標準値アイテムを示したもので、これはこのデータ テンプレートに基づいた (関連付けられた) アイテムです。



オブジェクト指向プログラミングでは、データ テンプレートがクラスに該当し、データ テンプレート フィールドがクラスのプロパティに当たります。リレーショナル データベース プログラミングでは、データ テンプレートがテーブルに該当し、データ テンプレート フィールドがそのテーブルの列に当たります。

各データ テンプレートは、ゼロ以上のベース データ テンプレートを継承し、それがベース テンプレートを指定します。上の図では、ベース テンプレートで定義されているフィールドを含むセクションが折り畳まれています。ベース テンプレートの詳細については、「ベース テンプレート」のセクションを参照してください。

ユーザー インターフェイスでは、データ テンプレートのデータ テンプレート セクションおよびフィールドと、すべてのベース テンプレートが組み合わされます。データ テンプレートは、連続継承や複数継承に対応しています。

ほとんどのデータ テンプレートは最終的に Sitecore 標準テンプレートを継承し、標準テンプレートではすべてのアイテムに共通のフィールドが定義されています。標準テンプレートの詳細については、「標準テンプレート」のセクションを参照してください。

データ テンプレートに対する変更は、データ テンプレートに基づいて、またはそのデータ テンプレートから継承するデータ テンプレートに基づいて、すべてのアイテムで直ちに反映されます。

### 2.1.1 データ テンプレート フィールド

データ テンプレートのフィールドは、そのデータ テンプレートから作成される個々のデータ エLEMENTのプロパティを定義します。

データ テンプレート フィールドは、コンテンツ エディターおよびページ エディターにおけるフィールドの動作に影響するユーザー インターフェイス コントロールなどのプロパティを定義します。フィールドの詳細については、「テンプレート フィールド」のセクションを参照してください。

#### メモ

フィールド名を定義する際には、フィールド セクション間であっても重複しないようにしてください。これは、XSLT と .NET のコードはどちらも、セクションを参照せずに単独でフィールド名を使用し、フィールドからコンテンツを抽出するためです。

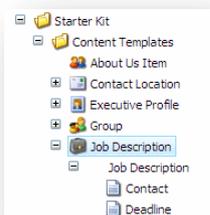
### 2.1.2 データ テンプレート セクション

データ テンプレート セクションは、フィールドを関連グループに整理します。フィールドは常にセクションにストアされます。ただし、セクションが反映されるのはコンテンツ エディターのみです。プログラマーは、フィールドからコンテンツを抽出するときにセクションを指定する必要はありません。

セクションを使用してフィールドを論理上のグループにまとめると、コンテンツ オーサーにとって検索も使用も容易になります。

1 つのセクションと関連する複数のフィールドを持つデータ テンプレートを作成し、複数あるベース テンプレートの 1 つとして使用することを考慮してください。これにより、フィールドの標準セクションを複数持つ新しいデータ テンプレートの構築が容易になります。

セクション定義アイテムは、対応するデータ テンプレート定義アイテムの下にストアされます。次の図では、Job Description というデータ テンプレートがあり、Job Description セクションがその直下にあり、その中に Contact および Deadline の各フィールドがあります。



データ テンプレートに関連付けられているアイテムを編集するときは、各セクションが折り畳み可能なフィールド グループとしてコンテンツ エディターに表示されます。

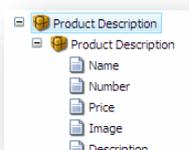
セクション定義の Section Name フィールドは、コンテンツ エディターでセクションのラベルとして使用されます。セクション名はローカライズが可能です。

## ベース テンプレート

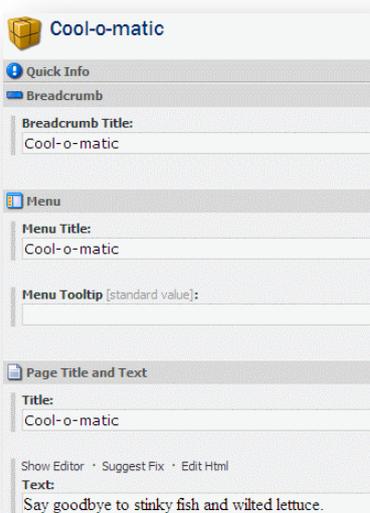
データ テンプレートは、そのベース テンプレートで定義されているセクションとフィールドを継承します。データ テンプレートに関連付けられたベース テンプレートは、テンプレート マネージャーまたはコンテンツ エディターの [継承] タブに表示されます。

一連のフィールドまたはセクションが多くのテンプレートに共通している場合には、1 つのデータ テンプレートにまとめ、必要に応じてそのテンプレートを他のテンプレートに追加することができます。

たとえば、Sitecore Starter Kit の Product Description データ テンプレートについて考えてみます。このデータ テンプレートには、Product Description という 1 つのセクションと、Name、Number、Price、Image、および Description の 5 つのフィールドが定義されています。



ただし、このテンプレートから作成されるアイテムには、追加のセクションとフィールドが含まれています。



このアイテムは、これらのフィールドとセクションを Product Description データ テンプレートのベース テンプレートから継承しています。次の図は、Product Description データ テンプレートに関連付けられたベース テンプレートの階層を表しています。



## 多重継承

データ テンプレートは、1 つのみに限らず任意の数のデータ テンプレートをベースにすることも可能です。継承される複数のテンプレートに、同じフィールドやフィールド セクションが含まれていることもあります。この場合は重複を避けるため、UI によってこれらのフィールドやフィールド セクションがマージされます。

### メモ

データ テンプレートが 1 つのデータ テンプレートを複数回継承した場合、最初のベース テンプレート以外はすべて無視されます。

## 循環継承

データテンプレートが、間接的にでも直接的にでもそれ自身をベースにしている場合は、循環継承と呼ばれます。循環継承は、重大な問題の原因になります。

次のような兆候が現れます:

- 特にデータテンプレートの操作中にシステムが応答しなくなる。
- ASP.NET でアプリケーション エラーが発生する。
- ログ エントリにテンプレートの循環継承が検出されたと記録される。

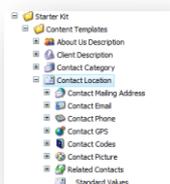
テンプレートの循環継承は、標準テンプレートやそのベース テンプレートに関連付けられているベース テンプレートを開発者が変更すると頻繁に発生します。システムで用意されているデフォルトのテンプレートは変更しないでください。`/Sitecore/Templates/System` ブランチの下にあるテンプレートを変更すると、循環継承の危険があるだけでなく、Sitecore のアップグレード プロセスが煩雑になったり、その他の問題が発生する可能性があります。

他のテンプレートから明示的に継承していないテンプレートも、Sitecore の標準テンプレートから暗黙的に継承します。標準テンプレートは、`/Sitecore/Templates/System/Templates/Sections` で定義されている多くのテンプレートから継承し、それぞれに標準テンプレートのセクションが定義されています。

## 2.2 標準値

標準値は、データ テンプレート フィールドのデフォルト値になります。フィールドの値が NULL の場合、アイテムのそのフィールドには、データ テンプレートでそのアイテムに定義されている標準値が含まれます。

各データ テンプレートの標準値は、\_\_Standard Values という名前のデータ テンプレート定義アイテムの子としてストアされています。各 \_\_Standard Values アイテムは、その親データ テンプレート定義アイテムによって定義されたデータ テンプレートに基づきます。次の図は、ユーザーが Contact Location データ テンプレートを展開したところです。データ テンプレートには、Contact Location データ テンプレート セクション定義アイテムと、Contact Location データ テンプレートの \_\_Standard Values アイテムが含まれています。



標準値アイテムは、データ テンプレートに基づく他のアイテムと同様、データ テンプレートそのものに定義されたすべてのフィールドと、ベース テンプレートから継承されたその他のフィールドが含まれています。標準値アイテムには、データ テンプレートに基づいたアイテムのデフォルト フィールド値が含まれています。標準値は、アイテムのフィールド値が NULL に設定されたときに使用されます。

コンテンツ エディターでアイテムを表示する場合、標準値を含む任意のフィールドにはフィールド タイトルの横にトークン [standard value] が付きます (図を参照)。



標準値は、アイテムが直接基づいているベース テンプレートだけではなく、どのベース テンプレートから継承することもできます。

### ヒント

管理の負荷を軽減し、システム設定を一元管理するために、次の項目は個々のアイテムではなくテンプレートの標準値で定義することをお勧めします:

- レイアウト設定
- 初期ワークフロー
- 挿入オプション

### メモ

フィールドの標準値が、アイテムのベースとなる複数テンプレートの標準値アイテムに設定されている場合、継承リストの最初のテンプレートの値だけが使用されます。他のテンプレートからの値は無視されます。

## 2.2.1 \$name トークン

Sitecore では、標準値で \$name トークンを使用することができます。\$name トークンは、作成中にアイテムの名前に置き換えられます。ただし、この置換が行われるのはアイテムの作成時のみです。アイテム名がフィールドのコンテンツになり、それが標準値の \$name トークンを置き換えます。

### メモ

ユーザーがアイテム名を変更した場合でも、\$name トークンは再評価されません。このため、\$name を使用して値を割り当てられたフィールドには、アイテム名の変更後も元のアイテム名が表示されます。\$name トークンを使用して設定されたフィールド値をリセットした場合、フィールドには \$name 値が表示されるため、ユーザーがフィールド値を適切な内容に設定する必要があります。

## 2.2.2 ブランクと NULL のフィールド値

コンテンツ オーサーがテキスト フィールドなどのフィールドをクリアすると、フィールドのコンテンツがブランクになることがよくありますが、これは NULL と同じではありません。デフォルトでは、このようなフィールドには、フィールドに定義されている標準値ではなくブランク値が表示されます。

ただし、フィールド定義アイテムには [空白に戻す] というチェックボックスがあります。これを選択するとブランク値が NULL に置き換えられます。このため、フィールドに対して [空白に戻す] を有効にし、ユーザーが個々のアイテムのフィールドのコンテンツをブランクにすると、そのアイテムのフィールド値は NULL にリセットされ、コンテンツ エディターにはフィールドの標準値が表示されます。

## 第 3 章

# 標準テンプレート

この章では、Sitecore の標準テンプレートについて詳細に説明します。

この章には次のセクションがあります：

- 標準テンプレートの概要
- 標準テンプレートの構造

### 3.1 標準テンプレートの概要

標準テンプレートは Sitecore のデフォルトのベース テンプレートで、大部分のデータ テンプレートによって共有されます。共有は、継承によって明示的に行うか、(NULL の GUID をベース テンプレートとして指定して) 継承から除外するように明示しない限り、暗黙的に行われます。テンプレートでベース テンプレートを指定しない場合、デフォルトでは Sitecore 標準テンプレートから直接継承します。

標準テンプレートでは、すべてのタイプのアイテムに関するセキュリティやワークフローなどのセクションが定義されています。アイテムをいつ公開するか、どのワークフローにあるか、どのユーザーがアクセスできるかなど、アイテムの管理方法が定義されます。

ほとんどのテンプレートは、直接でも間接でも "最終的" に標準テンプレートに基づく必要があります。

Quick Info セクションは、管理者ユーザーおよびデベロッパー ユーザーに最初に表示されるテンプレートのセクションであり、標準テンプレートのフィールド セクションです。このフィールド セクションには、アイテム位置の詳細、コンテンツ構造におけるテンプレート位置、ストレージ関連の各種 GUID が含まれています。

標準テンプレートのほとんどのフィールドは、全言語の全バージョンで共有されるシステム値を表します。

## 3.2 標準テンプレートの構造

ここからのセクションでは、標準テンプレートの各フィールドについて詳細に説明します。

### 重要

標準テンプレートのフィールド名の先頭には、すべて二重下線が付いています。これらのフィールドには、Sitecore.FieldIDs クラスのプロパティからアクセスしてください。

### 3.2.1 [高度な設定] セクション

[高度な設定] セクションは、データ テンプレートを標準値アイテムに関連付けます。

- [スタンダード バリュール] - このフィールドには、標準値アイテムへのリンクが含まれます。
- [トラッキング] - このフィールドには、アナリティクス プロファイル情報が含まれます。
- [ソース] - このフィールドには、クローンされたアイテムが含まれます。

### 3.2.2 [アピアランス (外観と動作)] セクション

[アピアランス (外観と動作)] セクションは、コンテンツ エディターでのアイテムの表示方法を制御します。

- [コンテキスト メニュー] - このフィールドには、ユーザーがコンテンツ エディターのコンテンツ ツリーでアイテムを右クリックしたとき表示されるメニューへのリンクがあります。
- [表示名] - このフィールドは、コンテンツ エディターのアイテム名より優先されます。
- [エディター] - このフィールドは、アイテムのカスタム エディターを定義します。
- [エディター] - このフィールドは、コンテンツ エディターに表示されるカスタム エディターのタブを制御します。
- [非表示] - このフィールドは、アイテムがコンテンツ エディターに表示されるかどうかを制御します。非表示アイテムの表示状態は、コンテンツ エディターでメイン リボンの [表示] タブにある [非表示アイテム] チェックボックスを選択して管理することができます。
- [アイコン] - このフィールドは、コンテンツ エディターのコンテンツ ツリーと、アイテムのヘッダーで表示されるアイコンを制御します。
- [読み取り専用] - このフィールドは、アイテムを読み取り専用で表示するかどうかを制御します。
- [リボン] - このフィールドは、コンテンツ エディターのリボンのカスタマイズ方法を制御します。
- [スキン] - このフィールドは、フォーマット情報をアイテムと関連付けます。
- [並べ替え順] - このフィールドは、フィールドの表示順序を指定する際に使用します。
- [スタイル] - このフィールドは、フォーマット情報をアイテムと関連付けます。
- [サブアイテムの並べ替え] - このフィールドは、アイテムの子を並べ替えるときのルールを制御します。

- **[サムネイル]** - このフィールドは、サムネイル画像をアイテムと関連付けます。
- **[オリジネーター]** - このフィールドは、アイテムの作成に使用されたブランチ テンプレートを示します。

### 3.2.3 [ヘルプ] セクション

[ヘルプ] セクションには、アイテムについて役立つ情報がストアされています。

- **[ヘルプリンク]** - このフィールドには、このアイテムの詳細なヘルプへのリンクが含まれます。
- **[説明]** - このフィールドには、コンテンツ ツリーでアイテムにマウスを重ねたとき表示されるポップアップ ヘルプが含まれます。
- **[概要 (ツールティップ)]** - このフィールドには、コンテンツ エディターのアイテムのヘッダーに表示される説明が含まれます。

### 3.2.4 [挿入オプション] セクション

[挿入オプション] セクションは、一連の [挿入ルール] と [挿入オプション] を保持するために使用されます。

- **[挿入ルール]** - このフィールドには挿入フィールドが含まれます。
- **[挿入オプション]** - このフィールドには挿入フィールドが含まれます。

### 3.2.5 [レイアウト] セクション

[レイアウト] セクションは主として、データ テンプレートにレンダリングを関連付けるために使用されます。

- **[レンダリング]** - このフィールドには、このアイテムの表示に使用されるレンダリングが含まれます。
- **[レンダラー]** - このフィールドには、このアイテムの表示に使用されるレンダラーが含まれます。
- **[レイアウト プリセット]** - このフィールドには、アイテムに使用できるレイアウト プリセットが含まれています。

#### メモ

[レンダリング] フィールドは管理 Web サイトによって使用されます。[レンダラー] フィールドは Sitecore ユーザー インターフェイスの一部のコンポーネントによって使用されます。

### 3.2.6 [有効期間] セクション

[有効期間] セクションは、このアイテムの各バージョンに対する公開制限を保持するために使用されます。

- **[有効期間の開始]** - このフィールドには、このバージョンの公開が有効になる開始日付が含まれます。
- **[有効期間の終了]** - このフィールドには、このバージョンの公開が有効な最終日付が含まれます。
- **[バージョンを隠す]** - このフィールドは、このバージョンを公開から隠すかどうかを制御します。

### 3.2.7 [パブリッシュ] セクション

[パブリッシュ] セクションは、公開の制限情報を保持するために使用されます。

- [パブリッシュ] – このフィールドには、このアイテムの公開が有効になる開始日付が含まれます。
- [パブリッシュ] – このフィールドには、このアイテムの公開が有効になる開始日付が含まれます。
- [パブリッシュ グループ] – このフィールドには、アイテムで有効な公開のターゲットが含まれます。
- [パブリッシュしない] – このフィールドは、アイテムを公開するかどうかを制御します。[パブリッシュしない] は、[パブリッシュ] および [パブリッシュの解除] より優先されます。

### 3.2.8 [セキュリティ] セクション

[セキュリティ] セクションは、セキュリティ設定を保持するために使用されます。

- [所有者] – このフィールドには、アイテムの現在の所有者であるユーザーの名前が含まれます。仮想の Creator-Owner ロールで使用します。
- [セキュリティ] – このフィールドには、アイテムのセキュリティ アクセス権が含まれます。

### 3.2.9 [統計] セクション

[統計] セクションは、作成、リビジョン、更新に関する基本的な統計を保持するために使用されます。

- [作成日] – このフィールドには、アイテムが作成された日時が含まれます。
- [作成者] – このフィールドには、アイテムを作成したユーザーの名前が含まれます。
- [リビジョン] – このフィールドには、GUID テキスト文字列としてストアされるリビジョン番号が含まれます。
- [更新日] – このフィールドには、アイテムが最後に更新された日時が含まれます。
- [更新者] – このフィールドには、前回の更新を実行したユーザーの名前が含まれます。

### 3.2.10 [タスク] セクション

[タスク] セクションは、関連付けられたタスクに関するリマインダー情報を保持し、アーカイブ日をストアするために使用されます。

- [アーカイブ日] – このフィールドには、アイテムがアーカイブされる日付が含まれます。
- [リマインダー日] – このフィールドには、リマインダー Email がリマインダー受信者に送信される日付が含まれます。
- [リマインダー受信者] – このフィールドには、リマインダーの受信者の Email アドレスが含まれます。Email アドレスが複数の場合には、セミコロン (;) で区切って指定することができます。
- [リマインダー テキスト] – このフィールドには、リマインダー Email の本文が含まれます。

### 3.2.11 [バリデーター] セクション

[バリデーター] セクションは、検証ルールを保持するために使用されます。

- **[クイック アクション バーの検証ルール]** – このフィールドには、クイック アクション バーに表示されるアイテム検証ルールが含まれます。
- **[検証ボタンの検証ルール]** – このフィールドには、コンテンツ ツリーの [レビュー] タブの [校正] グループにある [検証] ボタンで使用されるアイテム検証ルールが含まれます。
- **[検証バーの検証ルール]** – このフィールドには、コンテンツ エディターの右側の [検証] バーに表示されるアイテム検証ルールが含まれます。
- **[ワークフローの検証ルール]** – このフィールドには、ワークフロー検証で使用されるアイテム検証ルールが含まれます。
- **[除外する検証ルール]** – このフィールドは、グローバルなアイテム検証ルールを無効にします。

### 3.2.12 [ワークフロー] セクション

[ワークフロー] セクションは、ワークフロー状態の情報を保持するために使用されます。

- **[ワークフロー]** – このフィールドには、アイテムのワークフロー状態が含まれます。
- **[ワークフローの状態]** – このフィールドには、アイテムの現在のワークフロー状態が含まれます。
- **[ロック]** – このフィールドには、アイテムがロックされているかどうかの状態、ロックしているユーザー、およびロックされた日時に関する情報が含まれます。
- **[デフォルト ワークフロー]** – このフィールドには、このテンプレートから作成されるアイテムのデフォルトのワークフローに関する情報が含まれます。

## 第 4 章

# テンプレート フィールド

この章では、データ テンプレート フィールド定義アイテムと、データ テンプレート フィールドに関連付けられる各フィールド タイプの詳細について説明します。また、特定のデータ テンプレート フィールド定義アイテムのソース プロパティで使用する必要のあるクエリ構文についても説明します。

この章には次のセクションがあります:

- テンプレート フィールド定義アイテムについて
- データ テンプレートのフィールド タイプ
- Sitecore クエリ

## 4.1 テンプレート フィールド定義アイテムについて

データ テンプレート フィールドは、コンテンツ エディターおよびページ エディターにおけるフィールドの動作に影響するユーザー インターフェイス コントロールなどのプロパティを定義します。主としてフィールドの構造を制御する目的で使用され、フィールドにデータを入力するときの自動エラー チェック機能を提供します。

テンプレート フィールドは、オブジェクト指向プログラミングにおけるプロパティ、あるいはリレーショナル データベースにおける列に相当します。

テンプレートの各フィールドはセクション内に存在する必要があります。フィールドはセクションによって、再利用可能な論理グループに整理されます。テンプレート セクションの詳細については、「2.1.2 データ テンプレート セクション」のセクションを参照してください。

### 4.1.1 テンプレート フィールド テンプレート

フィールド定義アイテムは、テンプレート フィールド テンプレートに基づいています。各テンプレート フィールド定義アイテムでは、多くのフィールド プロパティが定義されます。

#### [フィールド ネーム]

フィールド ネームはフィールドの作成時に割り当てられ、**[タイトル]** フィールドを指定しない場合に、フィールドのラベルとして使用されます。

#### ヒント

フィールドを名前参照するときコードを読みやすくするように、変数名などのフィールド名には特殊文字を使用しないでください。

#### メモ

フィールド ネームは固有のものである必要があります。1 つのデータ テンプレートで複数のフィールドを同じ名前定義すると、データ テンプレートを保存するときに検証エラーが発生します。ただし、フィールドはベース テンプレートから継承されることもあり、その場合にはアイテムで複数のフィールドが同じ名前になる可能性があります。このような場合、コンテンツ エディターには両方のフィールドが表示されますが、プログラマーは API および XSLT レンダリングを使用するときフィールド名でコンテンツを取得するため、予期しない結果を得ることがあります。

このテンプレートの [データ] セクションのフィールド プロパティは、次のとおりです:

#### [タイプ]

フィールド タイプでは、ユーザー インターフェイスのどのコントロールをコンテンツ エディターに表示してそのフィールドへの入力を受け入れるかを指定し、そのフィールドのストレージ フォーマットも制御します。フィールド タイプの詳細については、「データ テンプレートのフィールド タイプ」を参照してください。

## [タイトル]

タイトルは、ブランクでない限りコンテンツ エディターでフィールドの上に表示されます。ブランクの場合はフィールド名が表示されます。

## [ソース]

ソース プロパティは、コンテンツ エディターのフィールドに関連付けられたユーザー インターフェイス コントロールに影響する情報を提供します。ソース フィールドの動作は、フィールドのタイプによって異なります。

次にいくつか例を挙げます:

- Droplink などのリスト フィールド タイプの場合、フィールドで表示されるリストの項目が並ぶコンテンツ ツリーでの位置をソース プロパティに指定します。
- イメージやファイル タイプの場合、ソース プロパティにはメディア ライブラリ ダイアログで表示される開始フォルダーを指定します。

## メモ

パスの先頭にチルダ ("~") 文字を指定すると、ダイアログは選択した目的のフォルダーで開きますが、ユーザーがツリー全体にアクセスできます。

- リッチ テキスト フィールド定義の場合、ソース プロパティで指定するアイテムには、編集インターフェイスで提供される機能を制御する HTML エディター プロファイルを指定します。
- 選択フィールドの場合、フィールドのソース プロパティには、次の例で示すように実際のクエリ文の前に `query:` 接頭辞を使用する Sitecore クエリを指定することができます。Sitecore クエリの構文の詳細については、「Sitecore クエリ」のセクションを参照してください。

```
query:/sitecore/content/Home/Employees/* [ startswith(@EmployeeName, 'A') ]
```

## ソースに設定できるパラメーター

Droptree、Treelist、TreelistEx などのさまざまなフィールドで、次のパラメーターがサポートされています。これらは、SitecoreQuery の代わりに Sitecore プロパティで使用することができます:

- **DataSource** – フィールドのデータ ソース アイテム。前述したようにフィールドのソース プロパティとしてパスを指定するのと同じです。
- **DatabaseName** – データ ソース アイテムを含むデータベースの名前。
- **IncludeTemplatesForSelection** – ユーザーはこのカンマ区切りのテンプレート名のリストに関連付けられているアイテムを選択することができます。
- **ExcludeTemplatesForSelection** – ユーザーはこのカンマ区切りのテンプレート名のリストに関連付けられているアイテムを選択することができません。
- **IncludeTemplatesForDisplay** – ユーザーはこのカンマ区切りのテンプレート名のリストに関連付けられているアイテムをナビゲートすることができます。

- **ExcludeTemplatesForDisplay** – ユーザーはこのカンマ区切りのテンプレート名のリストに関連付けられているアイテムをナビゲートすることができません。
- **IncludeItemsForDisplay** – ユーザーはこのカンマ区切りの ID 名のリストに対してアイテムを表示することができます。
- **ExcludeItemsForDisplay** – ユーザーはこのカンマ区切りの ID 名のリストに対してアイテムを表示することができません。
- **AllowMultipleSelection** – ユーザーは複数のアイテムを選択することができます。

ソース プロパティでは複数のパラメーターを、次の例のようにアンパサンド ("&") 文字で区切ることができます。

```
DataSource=/sitecore/content/home&IncludeTemplatesForSelection=section,sitemap
```

## [Blob (バイナリ ラージ オブジェクト)]

このフィールドは Sitecore の内部利用に限定され、使用できません。

## [共有]

このチェックボックスを選択すると、サポートされている全言語の各バージョンでフィールドの値が同じになります。[共有] プロパティを設定すると、どの言語またはどのバージョンのアイテムにおけるフィールド値の変更も、他のすべての言語とすべてのバージョンで反映されます。

共有値を考慮する必要があるのは、次の場合のみです:

- フィールドの古い値に関連性がない場合。
- ワークフロー制限がフィールド値に適用されない場合。
- 値が非常に大きい (バージョンングでストレージを消費している) 場合。

### メモ

[共有] チェックボックスと [バージョン管理なし] チェックボックスの両方を選択した場合は、フィールドが共有されます。全言語の全バージョンのアイテムが、このフィールドで 1 つの値を共有します。フィールドの詳細については、「[バージョン管理なし]」のセクションを参照してください。

### メモ

ワークフローと公開の制限は、共有されたフィールドの値には適用されません。

## [バージョン管理なし]

このチェックボックスを選択すると、1 つの言語の各バージョンではフィールドの値が同じになりますが、言語ごとには値が異なります。[バージョン管理なし] のフィールドは共有フィールドと似ていますが、言語が異なれば異なるフィールド値が保持されます。

**メモ**

[共有] チェックボックスと [バージョン管理なし] チェックボックスの両方を選択した場合は、フィールドが共有されます。全言語の全バージョンのアイテムが、このフィールドで 1 つの値を共有します。共有フィールドの詳細については、「[共有]」のセクションを参照してください。

**メモ**

ワークフローと公開の制限は、共有されたフィールドの値には適用されません。

**[デフォルト値]**

このフィールドは Sitecore の内部利用に限定され、使用できません。

**[検証]**

このフィールドには正規表現を指定することができます。フィールドを保存するとき、フィールドの内容がこの正規表現と比較して検証されます。フィールドを保存するには、フィールドの値がこの正規表現と一致する必要があります。

**[検証テキスト]**

フィールドを保存しようとしたときにフィールドの内容が [検証] フィールドの正規表現と一致しない場合、このメッセージが表示されます。

**[セキュリティ フィールド]**

このフィールドは、フィールド定義アイテムの [セキュリティの割り当て] ダイアログを開きます。フィールド読み取りとフィールド書き込みのアクセス権が適用されます。

**[空白に戻す]**

[空白に戻す] チェックボックスを選択すると、このフィールドを空白に設定してアイテムを保存したとき、フィールド値が NULL にリセットされます。NULL フィールドでは、フィールドの標準値が反映されます。空白と NULL のフィールドの詳細については、「空白と NULL のフィールド値」のセクションを参照してください。

**[テキスト サーチ対象外]**

このチェックボックスを選択すると、このフィールドは検索インデックスに含まれません。

**[ページ エディター ボタン]**

このフィールドを使用すると、ページ エディターのフィールドに関連付けられるコントロールを選択することができます。

**メモ**

テンプレート フィールドに対する追加の値が、[検証ルール]、[アピアランス (外観と動作)]、[ヘルプ]、[セキュリティ]、[統計]、および [ワークフロー] セクションにもあります。

## 4.2 データ テンプレートのフィールド タイプ

各フィールドにはフィールド タイプが関連付けられています。各フィールドのタイプは次の内容を制御します:

- Sitecore がコンテンツ エディターやページ エディターなどのユーザー インターフェースのフィールドに表示するユーザー インターフェース コンポーネント。
- フィールドにストアされる値のフォーマット。
- `Sitecore.Data.Fields` ネームスペース、`renderField` パイプライン、`FieldRenderer Web` コントロールのクラスなど、デベロッパーがフィールド値にアクセスするとき使用する .NET のクラスとプログラミング手法。

データベース メディアのバイナリ コンポーネントを除外すると、すべてのフィールド値はテキストとしてストアされます。単一行テキストなど一部のフィールド タイプには、単純なテキスト値がストアされます。Multilist など、それ以外のフィールド タイプは、選択した Sitecore アイテムの GUID を、パイプ文字 ("|") で区切ってストアします。複合フィールド タイプには通例、XML エlementがストアされます。Image フィールドには、他のイメージ プロパティを表す属性とともに 1 つの XML エlementが含まれます。レイアウト詳細のストアに使用されるフィールドには、さらに複雑な XML 構造が含まれます。

データ テンプレートにカスタムのユーザー インターフェースを実装するには、IFrame フィールド タイプを使用して、またはカスタム エディターを実装して、カスタム フィールド タイプを実装します。IFrame フィールドの詳細については、「IFrame フィールド タイプ」のセクションを参照してください。

デベロッパーが適切なフィールド タイプを選択しやすいように、Sitecore のフィールド タイプは次の各セクションで説明するカテゴリに分類されています。

### 4.2.1 Analytics フィールド タイプ

Analytics フィールド タイプは、Sitecore で内部的に利用されます。

#### 注意

Analytics フィールド タイプは使用しないでください。

### 4.2.2 Simple フィールド タイプ

Simple フィールド タイプは、個々の値を表します。個々の値には、Image フィールドの属性、あるいは Rich Text フィールドのイメージやリンクなど複数のプロパティを指定することができます。Sitecore には、次に説明する Simple フィールド タイプがあります。

## Checkbox フィールド タイプ

このフィールド タイプにはトグル ボタンが表示されます。ユーザーがチェックボックスを選択した場合は、値 1 (数字の 1) が入ります。ユーザーがチェックボックスを選択しない場合は、ブランク値が入ります。

### ヒント

Checkbox フィールドでリテラル値 1 をチェックするか、`Sitecore.Data.Fields.CheckboxField.Checked` などのプロパティを使用します。Checkbox フィールドの値を `NULL`、`String.Empty`、0 (数字のゼロ)、または 1 以外の値と比較しないでください。

次の図は、標準テンプレートの [アピランス (外観と動作)] セクションにある [非表示] という名前のチェックボックスを示しています。

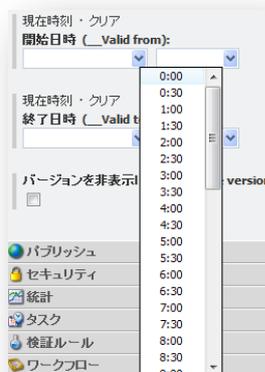


## Date および Datetime フィールド タイプ

Date フィールド タイプは、日付を手動で選択または入力できるユーザー インターフェースを提供します。Datetime フィールド タイプは、時間を手動で選択または入力するユーザー インターフェースを追加します。内容は `yyyyMMddThhmmss` というフォーマットでテキスト文字列としてストアされます。ユーザーが時刻を入力しない場合には、000000 の値 (深夜 0 時) がストアされます。ストアされる値が表すのは、Web サーバーのローカル時刻です。



次の図は、標準テンプレートの [有効期間] セクションにある [有効期間の開始] という Datetime フィールドを示しており、時刻のドロップダウン リストがあります。



## File フィールド タイプ

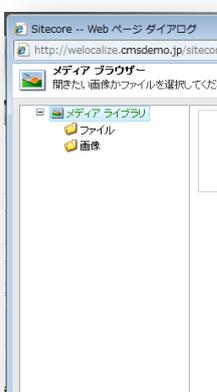
File フィールド タイプを使用すると、ユーザーがメディア ライブラリのアイテムを選択することができます。ユーザーは、[ファイルを開く] をクリックして [メディア ブラウザー] を開き、目的のファイルをメディア ライブラリから選択することができます。

File フィールドのソース プロパティを使用すると、ユーザーが [メディア ブラウザー] を開いたときに選択されるメディア フォルダーを制御したり、この選択インターフェースのルート アイテムを指定したりすることができます。

File フィールドのソース プロパティにアイテムを指定すると、[メディア ブラウザー] はそのアイテムより上にナビゲートできなくなります。ユーザーは指定されたメディア フォルダーより上にナビゲートできません。File フィールドのデフォルトのソースは、メディア ライブラリのルート アイテムです (/Sitecore/Media Library)。

File フィールドのソース プロパティがチルダ文字 (~ /Sitecore/Media Library/Files) で始まる場合、[メディア ブラウザー] には指定したメディア フォルダーが選択された状態でメディア ライブラリのコンテンツ ツリー全体が表示されるため、ユーザーはメディア フォルダーより上にナビゲートすることができます。

次の図は、パスにデータが入ったファイル フィールドを示しています。右側には、[メディア ブラウザー] が開いてアイテムが強調表示されています。



## Image フィールド タイプ

Image フィールド タイプを使用すると、ユーザーがメディア ライブラリからイメージを選択してイメージ プロパティを指定することができます。ユーザーは、[ファイルを開く] をクリックして [メディア ブラウザー] を開き、目的のファイルをメディア ライブラリから選択することができます。

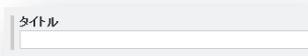
Image フィールドのソース プロパティは、[メディア ブラウザー] で選択したアイテムを制御します。ソース プロパティに関する詳細な情報は「File フィールド タイプ」のセクションを参照してください。

次の図は、Stater Kit の [Home-Products] アイテムの Image フィールドを示しています。



## Integer、Number、Password、Single-Line Text フィールド タイプ

これらのフィールド タイプは、単一行のテキストをストアします。



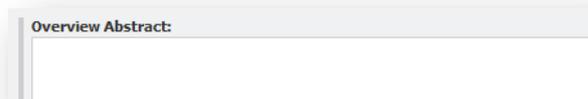
Password フィールド タイプはプレーン テキストをストアしますが、コンテンツ エディターでは入力をマスクします。指定されたテキストは、データベースに格納されるときにハッシュおよびマスクされません。



## Multi-Line Text フィールド タイプ

Multi-line フィールド タイプは、1 つのテキスト値が複数の行にまたがるときに使用します。このフィールドには検証機能がないため、このタイプのフィールドではリッチ テキストはサポートされません。

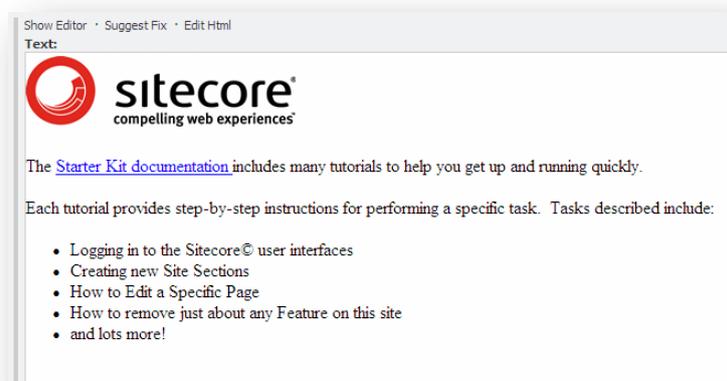
次の図は、スターター キットの [Home-Products] アイテムの [Overview Abstract] という複数行テキスト フィールドを示しています。



## Rich Text フィールド タイプ

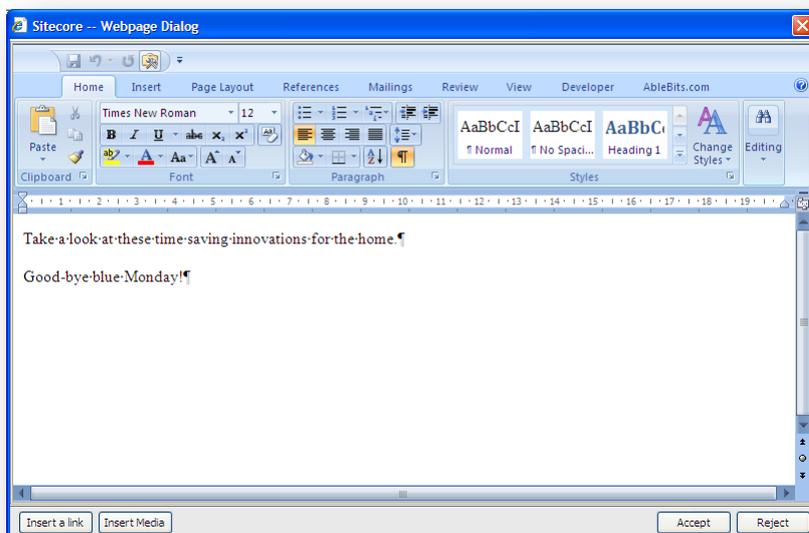
このフィールド タイプは、HTML テキストをストアします。このフィールドでは、ソース HTML をブラウザで表示したときのようにコンテンツが表示されますが、フィールドに実際にストアされるのは文字エンコードされた HTML です。[エディターを表示] ボタンを押すとリッチ テキスト エディターにアクセスでき、[HTML の編集] ボタンを押すとストアされている HTML にアクセスできます。Rich Text フィールドの詳細については、『コンテンツ リファレンス マニュアル』を参照してください。

次の図は、内部リンクとイメージを含むリッチ テキスト フィールドを示しています：



## Word Document フィールド タイプ

Word Document フィールド タイプを使用すると、ユーザーがブラウザーに埋め込まれた Microsoft Word を使用して HTML を編集することができます。



### メモ

Word Document フィールド タイプには、Microsoft Internet Explorer 6 以降が必要です。Internet Explorer の設定方法については、『Internet Explorer 設定ガイド』を参照してください。

web.config で `/configuration/sitecore/settings/setting` エレメントの `value` 属性に `nameWordField.InlineEditing.Width` を設定すると、エディターの幅を制御することができます。  
web.config で `/configuration/sitecore/settings/setting` エレメントの `value` 属性に `WordField.InlineEditing.Height` を設定すると、エディターの高さを制御することができます。  
web.config で `/configuration/sitecore/settings/setting` エレメントの `value` 属性に `WordField.InlineEditing.Padding` を設定すると、エディターの間隔を制御することができます。

### 4.2.3 List フィールド タイプ

各 List フィールド タイプを使用すると、異なる機能を持つ異なるユーザー インターフェイスを使用してユーザーがゼロ以上のアイテムを選択することができます。このフィールドのソース プロパティには、リストでのオプションを指定します。

### メモ

ソース プロパティでアイテムを指定すると、Checklist、Droplist および Multlist のフィールドには指定したアイテムの子が表示され、Droptree、Internal Link、Treelist、Treelist の各フィールド タイプでは指定したアイテムがツリーのルートになります。

## ヒント

Checklist、Droplist、Droptree および Multilist の各フィールド タイプのソース プロパティでは、Sitecore クエリがサポートされています。Sitecore クエリの詳細については、「Sitecore クエリ」のセクションを参照してください。

## ヒント

Checklist、Droptree、Multilist、Treelist および TreelistEx の各フィールド タイプは、ユーザーが選択したアイテムの GUID をパイプ文字 ("|") で区切ってストアします。フィールドに単一アイテムへの参照をストアした場合は、`Sitecore.Data.Fields.ReferenceField` クラスまたは `sc:item()` XSL 拡張メソッドを使用してアクセスすることができます。フィールドに複数の参照をストアした場合は、`Sitecore.Data.Fields.MultilistField` クラスまたは `sc:Split()` XSL 拡張メソッドを使用して、選択したアイテムにアクセスすることができます。

## Checklist フィールド タイプ

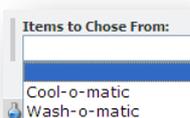
Checklist フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムの子を表す 1 つまたは複数のチェックボックスをユーザーが選択することができます。ユーザーが選択内容を並べ替えることはできません。

次の図は、標準テンプレートの [パブリッシュ] セクションの [パブリッシュ ターゲット] チェックリスト フィールドを示しています:



## Droplist フィールド

Droplist フィールド タイプを使用すると、フィールドのソース プロパティで指定されたリストから、ユーザーがアイテムを 1 つ選択することができます。

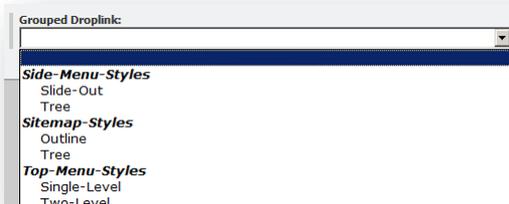


## 重要

Droplist フィールド タイプには、選択したアイテムの名前がストアされます。Droplink フィールド タイプは、その ID をストアします。Droplink フィールド タイプの詳細については、「Droplink フィールド タイプ」のセクションを参照してください。

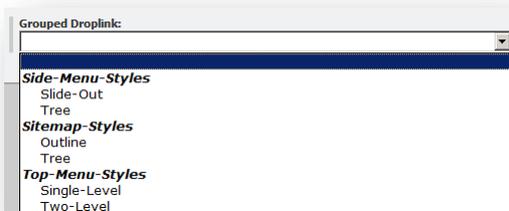
## Grouped Droplink フィールド タイプ

Grouped Droplink フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムの孫をユーザーが 1 つ選択することができます。Grouped Droplink フィールド タイプには、選択したアイテムの GUID がストアされます。



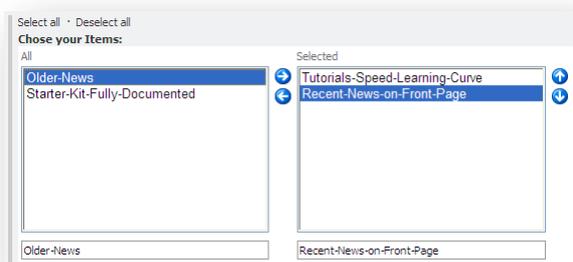
## Grouped Droplist フィールド タイプ

Grouped Droplist フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムの孫をユーザーが 1 つ選択することができます。Grouped Droplist フィールド タイプには、選択したアイテムの名前がストアされます。



## Multilist フィールド タイプ

Multilist フィールド タイプを使用すると、フィールドのソース プロパティで指定されたリストから、ユーザーがゼロ以上のアイテムを選択することができます。ユーザーは選択したアイテムを並べ替えることもできます。

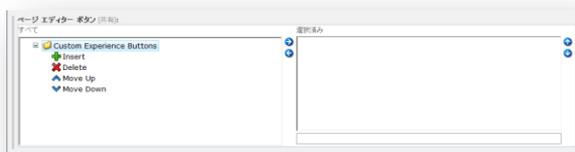


## Name Value List フィールド タイプ

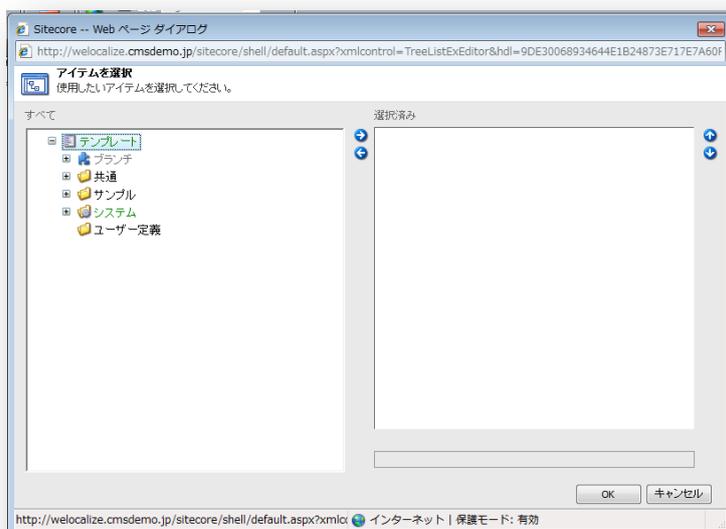
Name Value List フィールド タイプを使用すると、ユーザーがゼロ以上の任意のキーの値を入力することができます。入力された値は、アンパサンド("&") で区切って key=value のペアのリストとしてストアされます。Sitecore は、Name Value List フィールド タイプをレンダリングのパラメーターとして使用します。レンダリング パラメーターの詳細については、『プレゼンテーション コンポーネント リファレンス マニュアル』を参照してください。

## Treelist および TreelistEx フィールド タイプ

Treelist フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムのゼロ以上の子孫をユーザーが選択することができます。ユーザーは選択したアイテムを並べ替えることもできます。



TreelistEx フィールドは Treelist フィールドと同じ機能を持ちますが、ユーザーがリストの編集を選択して新しいブラウザ ウィンドウが開くまで選択したアイテムが表示される点のみ異なります。コンテンツ エディターには、Treelist フィールドより TreelistEx フィールドの方が迅速にロードされます。



### ヒント

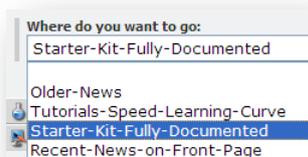
ユーザーが頻繁に編集するフィールドには Treelist フィールド タイプを使用し、編集頻度が低いフィールドには TreelistEx フィールド タイプを使用してください。

## 4.2.4 Link フィールド タイプ

Link フィールド タイプを使用すると、アイテム、外部 URL、アンカー、Email アドレス、および JavaScript 関数へのリンクをユーザーが入力することができます。

### Droplink フィールド タイプ

Droplink フィールド タイプを使用すると、フィールドのソース プロパティで指定されたリストから、ドロップダウン リストを使用してユーザーがアイテムを 1 つ選択することができます。



### Droptree フィールド タイプ

Droptree フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムの子孫をユーザーが 1 つ選択することができます。



### General Link フィールド タイプ

General Link フィールド タイプを使用すると、アイテム、URL、アンカー、Email アドレス、または JavaScript 関数にリンクさせることができます。

次の図は、URL への外部リンクが入った General Link フィールドを示しています:



### Version Link フィールド タイプ

このフィールドは Sitecore の内部利用に限定されます。

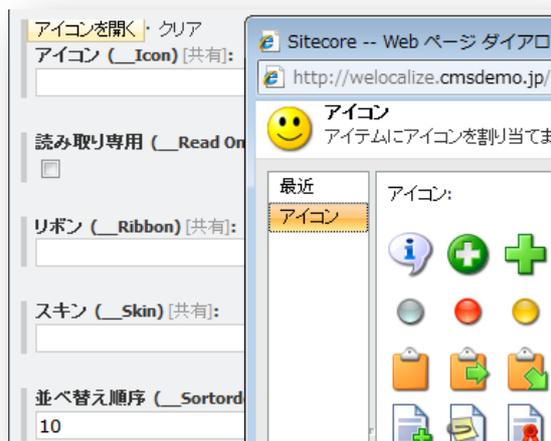
## 4.2.5 Developer フィールド タイプ

Developer フィールド タイプは主として、コンテンツ エディターでのアイテムの外観を制御します。標準的な Web サイトのデベロッパーは、エンド ユーザー データに対して Developer フィールド タイプを使用しません。

### Icon フィールド タイプ

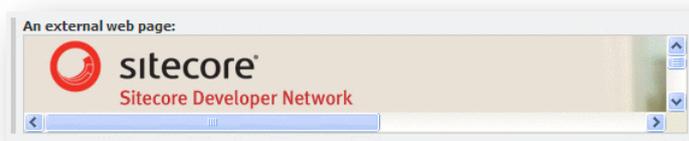
Icon フィールド タイプは、テーマ別のアイコンを各アイテムに関連付けます。

次の図は、[Open file] ダイアログが開いた状態の [アイコン] フィールドを示しています。



### IFrame フィールド タイプ

IFrame フィールド タイプには、フィールド内の Web ページが表示されます。このフィールドのソース プロパティに表示する URL を指定します。次の図は、外部 Web ページのサンプルを示しています。



### Tristate フィールド タイプ

Tristate フィールド タイプを使用すると、ユーザーが [デフォルト]、[はい]、[いいえ] の 3 つからいずれかのオプションを選択することができます。ユーザーが [デフォルト] を選択した場合、フィールドの値は空の文字列になります。[はい] を選択した場合、このフィールドの値は数値 1 になります。[いいえ] を選択した場合、このフィールドの値は数値 0 になります。



## 4.2.6 System フィールド タイプ

System フィールド タイプには、特定の要件に対応する高度な機能があります。標準的な Web サイトの開発者は、エンドユーザー データに対して System フィールド タイプを使用しません。

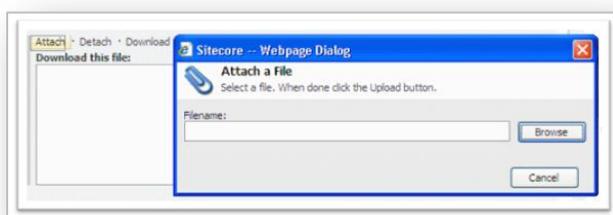
### Attachment フィールド タイプ

データベース メディア アイテムは、Attachment フィールド タイプを使用してバイナリ データをストアします。

#### 注意

データ テンプレートでは Attachment フィールド タイプを使用しないでください。

次の図は、[ファイルを添付] ダイアログが開いた状態の Attachment タイプのフィールドを示しています。



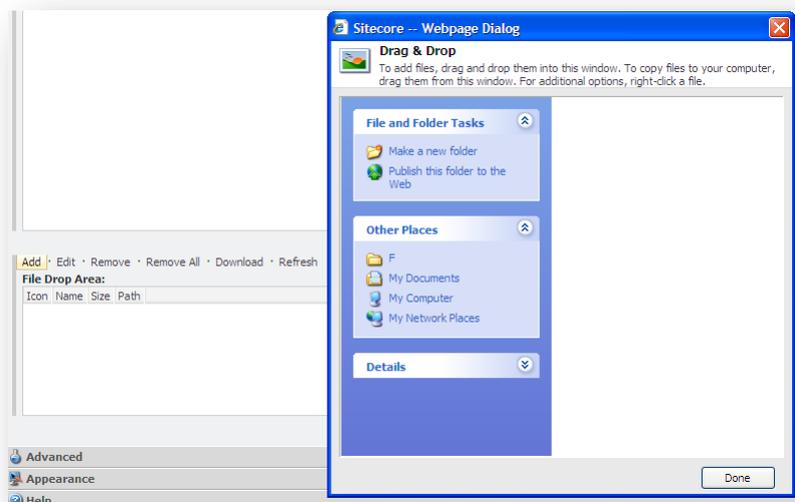
### Custom フィールド タイプ

関連性のないエントリをフィールド タイプのリストに含まないように、Sitecore には使用頻度の低いカスタム フィールドのための Custom フィールド タイプが用意されています。たとえば、レイアウト詳細で使用される CachingField、ExperienceTunerField および TestLaboratoryField Custom の各フィールド タイプには Custom フィールド タイプが使用されます。

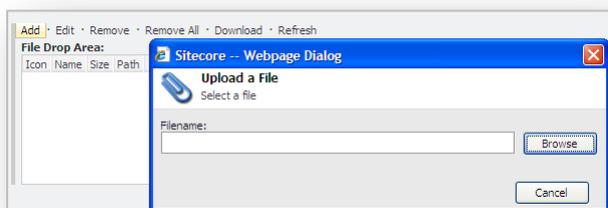
### File Drop Area (FDA) フィールド タイプ

File Drop Area (FDA) フィールド タイプを使用すると、ユーザーが WebDAV プロトコルを使用してファイルを管理することができます。詳細については、『WebDAV の構成』リファレンスを参照してください。[File Drop Area] フィールドの値は、対応するメディア アイテムを含むメディア ライブラリ フォルダを参照します。ソース プロパティは、[File Drop Area] フィールドに対して何の効果も持ちません。

[File Drop Area] フィールドを有効にした場合、[追加] をクリックすると [ドラッグ & ドロップ] ダイアログ ボックスが表示されます:



[File Drop Area] フィールドを無効にした場合、[追加] をクリックすると [アップロード] ダイアログ ボックスが表示されます:



アイテムを公開するとき、Sitecore はメディアの変更や削除も含めて、そのアイテムの [File Drop Area] フィールドで参照されているメディア アイテムを自動的に公開します。

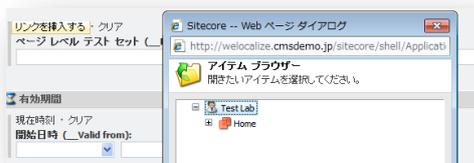
## Internal Link フィールド タイプ

Internal Link フィールド タイプを使用すると、フィールドのソース プロパティで指定されたアイテムの子孫をユーザーが 1 つ選択することができます。

### 注意

データ テンプレートでは Internal Link フィールド タイプを使用しないでください。このフィールド タイプは Sitecore の System フィールド タイプであり、Sitecore の内部利用およびカスタマイズに限定されます。パスおよびアイテム名にポイントするため、エンド ユーザー データには適していません。このリンクは、エンド ユーザーによってアイテムの名前が変更されると、簡単に切断されてしまいます。代わりに、Droplink、Droptree、または General Link を使用してください。

次の図は、[内部リンク] フィールドの上にある [挿入] から呼び出したアイテム選択ダイアログを示しています。



## Layout フィールド タイプ

Layout フィールド タイプを使用すると、ユーザーが各アイテムのレイアウト詳細を操作することができます。

### 注意

データ テンプレートでは Layout フィールド タイプを使用しないでください。

次の図は、[レイアウト] フィールドを示しています。



## Rules フィールド タイプ

Rules フィールド タイプを使用すると、条件付きのレンダリングを設定することができます。

### 注意

データ テンプレートでは Rules フィールド タイプを使用しないでください。

## Security フィールド タイプ

Security フィールド タイプを使用すると、各アイテムのセキュリティ設定を操作することができます。

### 注意

データ テンプレートでは Security フィールド タイプを使用しないでください。

## Template Field Source フィールド タイプ

Template Field Source フィールド タイプを使用すると、ユーザーがテンプレートフィールド定義アイテムのソースプロパティを選択することができます。

### 注意

データ テンプレートでは Template Field Source フィールド タイプを使用しないでください。

## Thumbnail フィールド タイプ

Thumbnail フィールド タイプを使用すると、ユーザーがメディア ライブラリからサムネイル画像を選択することができます。

### 注意

データ テンプレートでは Thumbnail フィールド タイプを使用しないでください。

## 4.2.7 非推奨のフィールド タイプ

次の表に、非推奨のフィールド タイプと、その代替タイプを示します：

Sitecore 5.3 の非推奨フィールド タイプ	Sitecore 6 の代替フィールド タイプ
html	Rich Text
link	General Link
lookup	Droplink
memo	Multi-Line Text
reference	Droplink
server file	None
text	Single-Line Text
tree	Droptree
tree list	Treelist
valuelookup	Droplist

### 注意

データ テンプレートでは非推奨のフィールド タイプを使用しないでください。上の表にリストした代替フィールド タイプを使用してください。

## 4.3 Sitecore クエリ

Sitecore クエリでは、XPath 構文の簡易版を使用して Sitecore のコンテンツ ツリーからアイテムをフィルターおよび取得するクエリ文字列構文を使用することができます。デベロッパーは、.NET コードおよびフィールド定義アイテムのソースフィールドで Sitecore クエリ文字列を使用します (ソースフィールドの先頭に "query:" 接頭辞が必要です)。

### 4.3.1 一般的な構文

Sitecore クエリ構文では、コンテキスト アイテムという概念を利用し、次の記号を使用して関連アイテムを参照します:

記号	意味
/	コンテンツ ツリーまたは親子関係のルート
text	アイテム名に一致
#	ダッシュ (-) を含むエスケープ テキスト。例: #meta-data#
*	任意のアイテム名に一致するワイルドカード
..	コンテキスト アイテムの親
[ ]	フィールドと XML エレメントの属性に関連する検索基準
@	アイテムのベース テンプレートで定義されているフィールド
@@	XML エレメントの属性。すべての Sitecore アイテムは "item" エレメントとして扱われ、次の属性が含まれます。 <ul style="list-style-type: none"> <li>name アイテムの名前</li> <li>key すべて小文字にしたアイテムの名前</li> <li>id アイテムの GUID</li> <li>tid アイテムのベース テンプレートの GUID</li> <li>mid アイテムの作成に使用するブランチ テンプレート (存在する場合)</li> <li>sortorder アイテムの並び順</li> <li>template アイテムのベース テンプレートの名前</li> <li>parentid アイテムの親の GUID</li> </ul>

これらの記号を組み合わせると、特定のアイテムまたはアイテム グループを参照することができます。次はその例です:

Sitecore クエリ文字列	結果セット
<code>/*</code>	コンテンツ ツリーのルート
<code>/sitecore/content/home</code>	Sitecore ホーム アイテム
<code>/sitecore/content/home/*[startswith(@title, 'P')]</code>	ホーム アイテムの下にあり、"P" で始まる Title フィールドを含む直属のサブアイテム
<code>*[@__hidden='1']</code>	コンテキスト アイテムの下にあるすべての非表示サブアイテム
<pre>query: /*/content/#meta-data#/colors/*[@show='1']</pre>	[表示] チェックボックス フィールドがオンにされた状態で Color アイテムの下にあるすべてのサブアイテムを選択するソース フィールド
<code>./*[@@tid="{A87A00B1-E6DB-45AB-8B54-636FEC3B5523}"]</code>	フォルダー テンプレートに基づいたコンテキスト アイテムの下にあるサブアイテム

### 4.3.2 軸

クエリの軸コンポーネントは、コンテキスト ノードに関してノード選択の方向を決定します。軸は、方向のあるクエリと考えることができます。

次の表に、一般的な軸の一部を示します:

軸	説明
ancestor	コンテキスト アイテムのすべての先祖を返す (XPath と同じ)。
ancestor-or-self	コンテキスト アイテムと、コンテキスト アイテムのすべての先祖を返す (XPath と同じ)。
child	(/*) は、コンテキスト アイテムのすべての子孫を返す (XPath と同じ)。
descendant	(//*) は、コンテキスト アイテムのすべての子孫、子の子孫、または子の子などを返す (XPath と同じ)。
descendant-or-self	コンテキスト アイテムと、コンテキスト アイテムのすべての子孫を返す (XPath と同じ)。
following	コンテキスト ノードに続くすべての兄弟を返す (XPath の following-sibling と同じ)。
parent	(..) は、コンテキスト アイテムのすべての親アイテムを返す (XPath と同じ)。
preceding	コンテキスト アイテムに先行するすべての兄弟を返す (XPath の preceding-sibling と同じ)。
self	(.) は、コンテキスト アイテムを返す (XPath と同じ)。
[int]	指定したインデックスを持つ子アイテムを返す。

### 4.3.3 演算子

Sitecore クエリ式では、次の表に挙げる演算子を使用することができます:

演算子	説明	例	戻り値
	結合	//Products //Shapes	Products または Shapes という名前のすべてのアイテム
+	加算	6 + 4	10.
-	減算	6 - 4	2.
*	乗算	6 * 4	24.

演算子	説明	例	戻り値
div	除算	8 div 4	2.
=	等しい	position()=3	position() が 3 の場合は True、それ以外の場合は False
!=	等しくない	position() !=3	position() が 3 の場合は False、それ以外の場合は True
<	より小さい	position() <4	position() が 4 より小さい場合は True、それ以外の場合は False
<=	以下	position() <=4	position() が 4 以下の場合は True、それ以外の場合は False
>	より大きい	position() >4	position() が 4 より大きい場合は True、それ以外の場合は False
>=	以上	position() >=4	position() が 4 以上の場合は True、それ以外の場合は False
or	論理 OR	position()=3 or position()=4	position() が 3 または 4 の場合は True、それ以外の場合は False
and	論理 AND	position() >3 and position() <7	position() が 3 より大きく、かつ 7 より小さい場合は True、それ以外の場合は False
mod	剰余 (除算の余り)	5 mod 2	1.

### 4.3.4 内部アーキテクチャ

Sitecore では、可能な限り高速なテクノロジーを使用してクエリが処理されます。要求されたクエリがデータ プロバイダーでサポートされている場合には SQL データベース、サポートされていない場合には Sitecore データ マネージャーです。SQL データベースでは最高のパフォーマンスを得られますが、その代わりに、一部のクエリがサポートされなくなります。

SQL データベースでは、アイテムをパスで解決する `"/sitecore/content/home"` などのクエリや、一連のアイテムを名前で検索する `"/home"` がサポートされています。

これは、実際にはどのように動作するのでしょうか。SQL データベースを使用する大規模なサイトがあり、*needle* という名前のコンテンツ アイテムをすべて検索する場合を考えてみます。次の API コードで、この結果セットが取得されます：

```
Item content = Sitecore.Context.Database.GetItem("/sitecore/content");  
Item[] needles = content.Axes.SelectItems("//needle");
```

SQL Server のデータ プロバイダーはこの種のクエリをサポートしているため、このクエリは直接データベースで解決され、データベースに多くのコンテンツ アイテム数が含まれている場合でも、かなり高速で実行されます。

ただし、検索基準を追加すると、Sitecore での検索方法が変わることがあります。前の例で、アイテムに `IsHidden` という名前のチェックボックス フィールドが含まれていると仮定します。非表示ではない `needles` を検索するクエリは、次のようになります：

```
Sitecore.Data.Items.Item content =  
    Sitecore.Context.Database.Items["/sitecore/content"];  
Sitecore.Data.Items.Item[] needles =  
    content.Axes.SelectItems("//needle[@IsHidden != '1']");
```

SQL Server のデータ プロバイダーでは、述語 (角かっこで囲まれた検索文字列の一部、`[@IsHidden != '1']`) がサポートされていません。このため、Sitecore ではクエリ API を使用してデータ マネージャーでこのクエリが解決されます。解決するには、述語をアイテムごとに評価し、一致するアイテムを返せるように、クエリのスコープにある全アイテム (この例では `/sitecore/content` のすべての子孫) がロードされます。残念ながら、このように多くのアイテムをロードするときにはパフォーマンスが犠牲になり、アイテムのセットが大きい場合には大幅に速度が低下します。

以上のことを踏まえると、最初に `needle` という名前のアイテムをすべて検索し、次にメモリーにある結果セットを検索し、非表示ではないアイテムを探すという方法が最適であることがわかります。

## 第 5 章

### フィールドとアイテムの検証

この章では、フィールドとアイテムに対して検証を行い、入力データの正当性を制御するさまざまな方法について詳細に説明します。検証の詳細については、『クライアント構成クックブック』を参照してください。

この章には次のセクションがあります:

- 検証のルールとオプションについて
- デフォルトの検証ルール
- 検証オプション

## 5.1 検証のルールとオプションについて

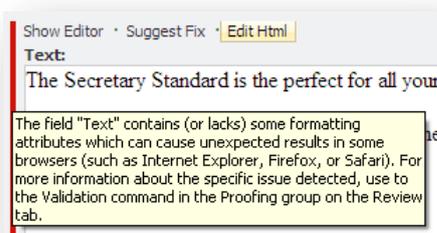
フィールド値は検証のルールとオプションによって検証され、このときのフィールド検証ルールは、web.config ファイルで指定されている設定に基づいてオンザフライ方式で解決されます。検証設定を保持するセクションには、直接的な式の設定、パイプライン、およびイベントハンドラー、そしてコンテンツ エディターで [検証] バーをオフにする機能など単純な設定が含まれています。

これらの検証ルールは、1 つの言語バージョンのアイテムで 1 つのフィールド値を検証します。各アイテムの検証は、任意の数のアイテム検証ルールを使用して行うことができ、その場合はアイテム名などのグローバル値と、複数のフィールド値を伴う条件も検証されます。

アイテムとフィールドを検証するときに検証ルールで使用される基本的なテクノロジーは同じですが、アイテム検証およびフィールド検証のルールでは、主として次のような相違があります：

- フィールド検証ルールは、1 つの言語バージョンで 1 つまたは複数のフィールドの値を検証します。それに対し、アイテム検証ルールはすべてのバージョンに共有されたアイテム データを検証します。アイテム名や、複数のフィールド値を伴う条件などです。
- ユーザー インターフェイスには、/sitecore/system/Settings/Validation Rules/Field Rules からのフィールド検証ルールと、/sitecore/system/Settings/Validation Rules/Item Rules からのアイテム検証ルールが表示されます。
- テンプレートにおいて、フィールド検証ルールはテンプレート フィールド定義によって定義されますが、アイテム検証ルールはテンプレート標準値と個々のアイテムによって定義されます。

フィールド検証の問題はすべて、コンテンツ エディターでフィールド値の左に色付きのバーとして表示されます。表示される色は、エラーがなければグレー、警告の場合には黄色、エラーの場合には赤です。次の図では、フィールド検証の警告 (赤いバー) が表示され、ツールチップの説明としてエラーに関する詳細が表示されています。



## 5.2 デフォルトの検証ルール

次の表は、あらかじめ用意されているアイテム検証およびフィールド検証のルールの例と、そのルールで実行される検証の簡単な説明です。新しいバージョンでは、その他のルールがあらかじめ用意されていることがあります。

### 5.2.1 アイテム検証ルール

アイテム検証ルール	検証する内容
Broken Links	全言語と全バージョンで、1 つまたは複数のフィールドにおける破損リンクをチェックする。
Duplicate Name	アイテム名が兄弟間でユニークであることをチェックする。
Full Page XHTML	サイト訪問者がアイテムを要求したときに生成される XHTML を検証する。
Media Size Too Big	メディアライブラリのアイテムが指定サイズを超えているかどうかをチェックする。
Url Characters	アイテム名に、URL をレンダリングする際にエスケープの必要な文字が含まれているかどうかをチェックする。このような文字は検索エンジンのインデックス作成に悪影響を及ぼすことがあるため。

### 5.2.2 フィールド検証ルール

フィールド検証ルール	検証する内容
Broken Links	フィールドに破損リンクが含まれているかどうかをチェックする。
Is Email	フィールドに Email アドレスが含まれているかどうかをチェックする。
Is Integer	フィールドに整数が含まれているかどうかをチェックする。
Is XHTML	フィールドに XHTML が含まれているかどうかをチェックする。
Max Length 40	フィールドに含まれる値が 40 文字以下かどうかをチェックする。
Rating 1 to 9	フィールドに含まれる値が 1 から 9 の間にあるかどうかをチェックする。
Required	フィールドに値が含まれているかどうかをチェックする。
Spellcheck	RAD エディターのスペル チェック機能を使用してスペリングをチェックする。リッチ テキスト エディターでも使用される。
Type and Assembly	アクセス可能なアセンブリにおけるクラスを値が適切に参照しているかどうかをチェックする。
W3C XHTML Validation	W3C の検証サービスを使用してフィールドの HTML を検証する (CMS がインターネット アクセスできることが必要)。

### 5.2.3 システム フィールド検証ルール

システム フィールド検証 ルール	検証する内容
Alt Required	alt テキストが指定されていることをチェックする。
Extension May Not Start with a Dot	メディア ファイルの拡張子がドットで始まっていることをチェックする。
Extern Link Target	外部リンク (他サイトへのリンク) をチェックする。
Image Has Alt Text	イメージに alt テキストがあるかどうかをチェックする。
Image Has Alt Text from Media Library	メディア アイテムにデフォルトの alt テキストがあるかどうかをチェックする。デフォルトの alt テキスト (メディア ライブラリから) が使用される。
Image Size	イメージ フィールドを通じて参照されるイメージのサイズをチェックする。
Rich Text Image Size	リッチ テキスト フィールドに含まれるイメージのサイズについて、サイトのデザイン上イメージが大きすぎないかどうかをチェックする。

## 5.3 検証オプション

検証オプションは、データ テンプレート標準値の **[検証ルール]** セクションとテンプレート フィールド定義で選択することができます。Sitecore には、一連の検証オプションがあらかじめ用意されています。次の表は、検証オプションと、実行される検証の簡単な説明です。

検証オプション	制御される内容
[クイック アクション] バー	コンテンツ エディターの左側にある <b>[クイック アクション]</b> バーに検証の問題が表示される。
[検証] ボタン	<b>[レビュー]</b> タブの <b>[校正]</b> グループから [検証] を選択したときと、ワークフロー検証アクションを含むワークフロー状態への変換をユーザーが呼び出したときに、検証の問題が表示される。
[検証] バー	コンテンツ エディターの左側にある <b>[検証]</b> バーに検証の問題が表示される。
[ワークフローの検証ルール]	ワークフロー検証アクションに関連付けられたワークフロー コマンドをユーザーが選択したとき、ユーザー インターフェース検証の問題が表示される。すべての検証エラーを解決しない限り、ユーザーはワークフロー アクションを完了できない。
[除外する検証ルール]	個々のコンテンツ アイテムに対するグローバルな検証ルールを無効にすると、[クイック アクション] バーに検証の問題が表示されなくなります。

## 第 6 章

### 挿入オプション

この章では挿入オプションについて説明します。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種類を制御します。挿入オプションには、データ テンプレート、ブランチ テンプレート、およびコマンド テンプレートを含めることができます。

この章には次のセクションがあります:

- 挿入オプション
- ブランチ テンプレート
- コマンド テンプレート

## 6.1 挿入オプション

Sitecore の管理者とデベロッパーは、挿入オプションを設定して、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種類を制御します。挿入オプションには、次を含めることができます：

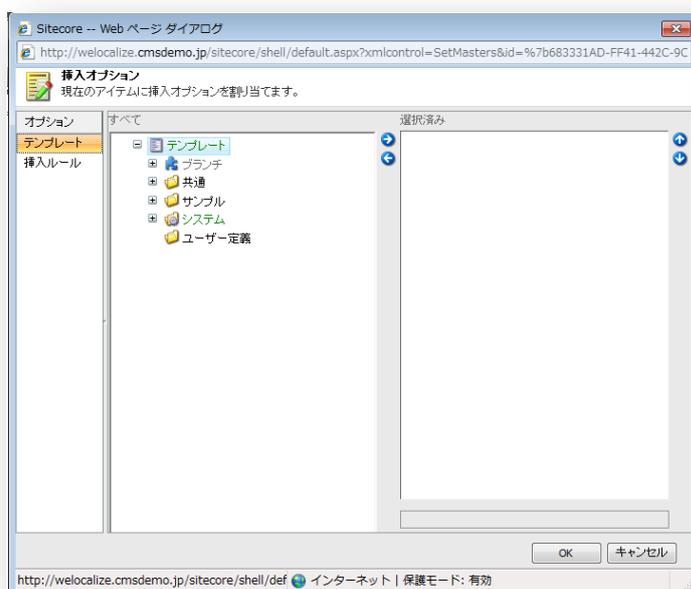
- データ テンプレート
- ブランチ テンプレート
- コマンド テンプレート

### 6.1.1 効率的な挿入オプション

Sitecore は、次のセクションで説明するリソースを使用して、アイテムごと、ユーザーごとに効率的な挿入オプション (ユーザーが新しいアイテムを挿入するために使用可能なデータ テンプレート、ブランチ テンプレートおよびコマンド テンプレート) を決定します。

### 6.1.2 割り当てられた挿入オプション

データ テンプレートの標準値で、挿入オプションを割り当てることができます。割り当てる挿入オプションには、データ テンプレート、ブランチ テンプレート、コマンド テンプレート、および挿入ルールを含めることができます。



#### 重要

可能な限り、挿入オプションは個々のアイテムに割り当てるのではなく、標準値で割り当てるようにしてください。

### 6.1.3 挿入ルール

管理者が挿入オプションで挿入ルールを割り当てると、効率的な挿入オプションを実行時のユーザーに動的に再定義することができます。デベロッパーは、管理者が選択できるように、カスタムの挿入ルールを実装することができます。

### 6.1.4 挿入オプション ルール

管理者は、挿入オプション ルールを実装して効率的な挿入オプションを定義し、ルール パラメーターを選択するユーザー インターフェースを提供することができます。特定のアイテムに割り当てられる挿入ルールとは異なり、挿入オプション ルールはすべてのアイテムに適用されます。ただし、条件によって適用を制限するのが一般的です。挿入ルールの詳細については、「挿入ルール」のセクションを参照してください。効率的な挿入オプションに追加のメカニズムを提供するルール エンジンの詳細については、『ルール エンジン クックブック』を参照してください。

### 6.1.5 uiGetMasters パイプライン

Sitecore では、`web.config` で定義されている `uiGetMasters` パイプラインを呼び出すことによって効率的な挿入オプションのリストを構築します。`uiGetMasters` パイプラインには、ユーザーが新しいアイテムを挿入するときに使用可能なデータ テンプレート、コマンド テンプレートおよびブランチ テンプレートのリストが定義されています。

`uiGetMasters` パイプラインは、次のプロセスでアイテムの効率的な挿入オプションを定義します：

1. アイテムに割り当てられていて、コンテキスト ユーザーが読み取りアクセス権を持つ個々の挿入オプションを、効率的な挿入オプションのリストに追加します。
2. アイテムに関連付けられている挿入ルールを、効率的な挿入オプションのリストに追加します。挿入ルールの詳細については、「挿入ルール」のセクションを参照してください。
3. 挿入オプション ルールを、効率的な挿入オプションのリストに適用します。挿入オプション ルールの詳細については、「挿入オプション ルール」のセクションを参照してください。
4. コンテキスト ユーザーが `insert:show` アクセス権を持っていないアイテムがあれば削除します。

実行時のユーザーにとって効率的な挿入オプションを決定するロジックを含む `uiGetMasters` パイプラインに、デベロッパーはカスタム プロセッサを追加することができます。これらのルールを参照するアイテムのみに適用される挿入ルールとは異なり、`uiGetMasters` パイプラインはすべてのアイテムに適用されます。

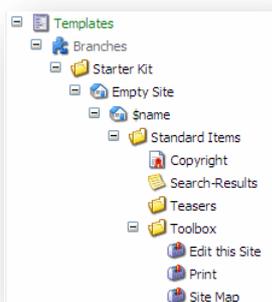
## 6.2 ブランチ テンプレート

ブランチ テンプレートを使用すると、Sitecore 管理者により事前定義されているアイテムの構造を使用して、ユーザーがアイテムを挿入することができます。ブランチ テンプレートは、子孫をすべて含むブランチ テンプレート定義アイテムで構成されています。

ユーザーが挿入オプションからブランチ テンプレートを選択すると、そのブランチ テンプレート定義アイテムの下位に定義されている子孫のコレクションすべてのコピーが作成され、このコレクションがサブアイテムのグループとして現在選択されているアイテムの下位に配置されます。

ブランチ テンプレートの子孫で定義されているフィールド コンテンツはすべてコピーされますが、フィールドとアイテムの名前に割り当てられている \$name トークンは、挿入の操作時にユーザーが指定した名前に置き換えられます。

次の図は、Sitecore Starter Kit に含まれている Empty Site のブランチ テンプレートを示しています。ブランチ テンプレート定義アイテムの直下にあるのは、\$name という名前のサブアイテムのみです。このブランチ テンプレートに基づいてアイテムを作成すると、名前を要求され、\$name トークンは作成されたアイテムの名前に置き換えられます。\$name アイテムの下に各種のフォルダーおよびアイテムが表示されています。\$name ルートの下にこれらが作成されると、新しく作成されたアイテムの下にコンテンツの完全なサブセットが作成されます。



### 6.2.1 ブランチ テンプレートの使用

ブランチ テンプレートを使用すると、ユーザーは事前定義された再利用可能な構造を使用して複数のアイテムを作成しやすくなります。ブランチ テンプレートを使用すると、作成したアイテムに初期のフィールド値をコピーすることもできます (標準値からフィールド値を継承するのではなく)。これには、標準テンプレート フィールドのフィールド値 (アイテムのアイコンやアクセス権の割り当てのように、リボン コントロールを介して割り当てられる) と、カスタム データ テンプレートに定義されているフィールド (製品番号など) の両方が含まれます。

ブランチ テンプレートは、コンテンツ オーサーが複数のアイテム (兄弟または子孫) を一度に作成したいときにも便利です。コンテンツの複数のサブツリーは、ブランチ テンプレートを使用して作成することができます。

## ブランチ テンプレートを使用した結果

ユーザーがブランチ テンプレートを呼び出すと、システムで次のような処理が実行されます:

- すべてのフィールド値を含めて、ブランチ テンプレート定義アイテムの子孫がコピーされ、新しいアイテムが作成されます。
- 新しいアイテムでトークン置換が実行され、ブランチ テンプレートの呼び出し時にユーザーが入力する名前によって、アイテム名とフィールド値の両方で `$name` などのトークンが置き換えられます。
- ブランチ テンプレートが 1 つの子アイテムとゼロ以上の子孫で構成される場合、その子の名前は通常 `$name` であるため、ユーザーは作成するルート アイテムの名前を指定することができます。ブランチ テンプレートのアイテム名については、`$name` のみサポートされます。
- ブランチ テンプレートの標準値を上書きしないフィールドの場合、新しいアイテムで対応するフィールドには標準値が含まれています。ただし、標準値に `$name` などのトークンが含まれ、それも指定されるアイテム名で置き換えられる場合を除きます。

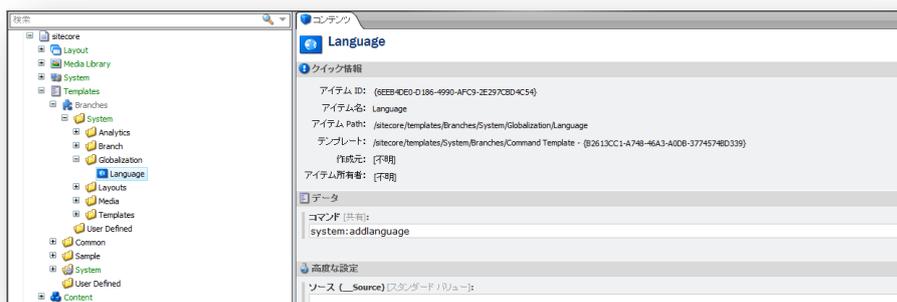
### メモ

ブランチ テンプレートのフィールド値は、データが重複する原因となることがよくあります。このため、ブランチ テンプレートのフィールド値より、テンプレートの標準値の方が優先されるのが普通です。ブランチ テンプレートのフィールド値は、テンプレート標準値のように継承されるのではなくコピーされるという点にも注意してください。したがって、ブランチを使用してアイテムを作成した後でそのブランチを変更した場合、以前に作成済みのアイテムには変更が反映されません。

## 6.3 コマンド テンプレート

コマンド テンプレートは、挿入の操作時に呼び出されるクラスとメソッドを定義します。データ テンプレートとブランチ テンプレートは事前定義された構造によって構成されますが、コマンド テンプレートは Sitecore の UI コマンドを参照してウィザードやその他のロジックを呼び出し、新しいアイテムを作成します。

次の図は、新しい言語を作成するときに使用される [言語] コマンド テンプレートを示しています。右ペインには [コマンド] フィールドがあり、そこにコマンド テンプレートを呼び出したときに呼び出されるコマンドの名前（ここでは `system:addlanguage`）が表示されています。



### 6.3.1 コマンドテンプレートの使用

コマンド テンプレートを使用すると、事前定義された構造ではなく、ロジックに従ってアイテムを挿入することができます。コマンド テンプレートを使用して、ユーザーが挿入オプションを呼び出すとき複数のアイテムを挿入することもできますが、コマンド テンプレートの方がブランチ テンプレートより柔軟です。

コマンド テンプレートは、アイテムと標準値に挿入オプションとして割り当てることができます。コマンド テンプレートの挿入オプションは、データ テンプレートやブランチ テンプレートの挿入オプションと同じように見えます。唯一の相違は、コマンド テンプレートの挿入オプションが Sitecore UI コマンドをトリガーするという点です。

コマンド テンプレートは通常、ユーザーから情報を収集するウィザード アプリケーションを呼び出し、プログラマ的に適切なアイテム セットを作成します。

Sitecore で定義されているコマンド テンプレートの例の 1 つには、テンプレートの作成が含まれています：

- `/Sitecore/System/Languages` の挿入オプションに、`/Sitecore/Templates/Branches/System/Globalization/Language` コマンド テンプレートが含まれています。
- これが、コントロール パネルで使用されるコマンドを呼び出し、新しい言語を作成します。
- 各コマンド テンプレートの [コマンド] フィールドの値は、`/App_Config/Commands.config` ファイルのエントリに対応しており、ユーザーがコマンド テンプレートを呼び出すとき、指定されたクラスのメソッドをシステムで呼び出します。