



## Sitecore CMS 6.4

# データの再利用および共有

開発者向けヒントとテクニック

## 目次

Chapter 1	データの再利用および共有 .....	3
1.1	プレゼンテーション コンポーネントを使用したデータの共有および再利用 .....	4
1.1.1	デバイス .....	4
1.1.2	レンダリング プロパティ .....	5
1.1.3	条件付きレンダリング .....	5
1.1.4	CSS .....	6
1.1.5	フィールド値の継承 .....	6
1.1.6	RSS を使用した再利用および共有 .....	7
1.2	アイテムとフィールド値の共有および再利用 .....	8
1.2.1	エイリアス .....	8
1.2.2	選択フィールド .....	9
	アイテム選択データ テンプレート .....	10
1.2.3	公開ターゲット .....	11
1.2.4	プロキシ .....	11
1.2.5	クローン .....	12
1.2.6	ワイルドカード アイテム .....	14
1.2.7	Web サービス、AJAX および他のインターネット インターフェース .....	14
1.3	文字列の共有および再利用 .....	16
1.3.1	Snippet .....	16
1.3.2	ディクショナリ翻訳 .....	16
	ディクショナリ エントリの作成方法 .....	17
1.3.3	トークン置換 .....	17
	トークン置換の実装方法:renderField パイプライン .....	18

# Chapter 1

## データの再利用および共有

Sitecore 開発者は、この文書をもとに Web サイトの複数のページでデータを再利用したり複数のマネージド Web サイトと外部アプリケーションでデータを共有する手段を決定することができます。<sup>1</sup>

Sitecore レイアウト エンジンには、さまざまなページでデータを再利用したり複数のマネージド Web サイトと外部アプリケーション間でデータを共有したりするために各種テクニックを使用することができます。複数のページおよびマネージド Web サイトで、データを再利用および共有するプレゼンテーション コンポーネントを再利用および共有することができます。<sup>2</sup> プレゼンテーション コンポーネントには、任意の数の Sitecore アイテムからデータを組み込むことができます。アイテムを要求するデバイスの種類など、多くの基準に基づいてデータをさまざまな方法でフォーマットすることができます。再利用可能なテキスト Snippet をリッチ テキスト エディターのフィールド値に挿入したり、ディクショナリ エントリや簡易トークンを再利用可能な文字列に動的に置換したりすることができます。複数のマネージド Web サイトと外部アプリケーション間でデータを共有したり、コンテンツ ツリー内の複数の URL や場所にアイテムを表示したりすることができます。<sup>3</sup>

この章には次のセクションがあります：

- プレゼンテーション コンポーネントを使用したデータの共有および再利用
- アイテムとフィールド値の共有および再利用
- 文字列の共有および再利用

---

<sup>1</sup> 複数のマネージド Web サイトの詳細については、<http://sdn5.sitecore.net/Articles/Administration/Configuring%20Multiple%20Sites.aspx> を参照してください。

<sup>2</sup> プレゼンテーション コンポーネントの詳細については、<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。

<sup>3</sup> アイテムの URL の詳細については、<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Dynamic%20Links.aspx> から『ダイナミック リンク ガイド』を参照してください。

## 1.1 プレゼンテーション コンポーネントを使用したデータの共有および再利用

次のテクニックを使用すると、プレゼンテーション コンポーネントを使用してデータを共有および再利用することができます。

### 1.1.1 デバイス

レイアウト エンジンには、Web ブラウザーまたはモバイル デバイスなどのさまざまな種類のデバイスがアイテムを要求する際にさまざまなプレゼンテーション コンポーネントを呼び出すことができます。<sup>4</sup> 各種デバイスを使用することにより、HTTP 要求コンテキスト内の任意の情報に基づいてさまざまな方法でデータをフォーマットすることができます。

#### 重要

各デバイスを起動するための基準を定義する必要があります。<sup>5</sup>

たとえば、コンテンツおよびインフォメーション アーキテクチャが同じである 2 つのマネージド Web サイトが、異なるプレゼンテーション コンポーネントを使用して Sitecore アイテムの単一階層をレンダリングするとします。

この場合、1 つのデバイスを使用してこれらの要件を満たすことができます。2 番目のマネージド Web サイトを関連付けるデバイスを作成します。共有アイテムごとに、2 番目のマネージド Web サイトによって使用されるプレゼンテーション コンポーネントをレイアウト詳細でこのデバイスに関連付けます。2 番目のマネージド Web サイトのマネージド Web サイト定義にこのデバイスを指定します。<sup>6</sup>

マネージド Web サイト定義は単一セットのアイテムを共有します。Sitecore では、適切なデバイスのレイアウト詳細を評価することにより、適用するプレゼンテーション コンポーネントを決定します。

---

<sup>4</sup> デバイスの詳細については、

<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。

<sup>5</sup> .NET ロジックを使用してデバイスを起動する例については、

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

<sup>6</sup> コンテキスト デバイスのレイアウト詳細がないアイテムを除外するプレゼンテーション コンポーネントのコード例については、

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

### 1.1.2 レンダリング プロパティ

レンダリングにより、デザイン時および実行時に定義できるレンダリング プロパティに基づいてさまざまな出力を生成することができます。<sup>7</sup> レンダリング プロパティを使用することにより、複数のレンダリングで文字列およびアイテムを再利用したり、複数のマネージド Web サイト間で文字列およびアイテムを共有したりすることができます。

レンダリングにはデータ ソース アイテムを渡すことができます。ほとんどのレンダリングでは、レンダリングのデータ ソース アイテムからデータを読み取ります。レンダリングには別のレンダリング プロパティを使用して追加アイテムを渡すことができます。任意の数のマネージド Web サイトで使用される任意の数のレンダリングに同じアイテムを渡すことができます。

たとえば、複数のマネージド Web サイトで共通のフッターの外観を共有するが、このフッターにはサイトごとに異なるデータを表示とします。

この場合、レンダリング プロパティを使用してこれらの要件を満たすことができます。フッター内のデータを表すデータ テンプレートを作成し、マネージド Web サイトごとにこのデータ テンプレートに基づくアイテムを作成します。マネージド Web サイトごとにフッター レンダリングに適したフッター アイテムをデータ ソースとして渡します。

複数のマネージド Web サイトでレンダリングが再利用され、各サイト内の複数の Web ページでフッター データを再利用することができます。別のデバイスに対するフッター レンダリングなど、他のレンダリング内のフッター データ アイテムを再利用することができます。

### 1.1.3 条件付きレンダリング

条件付きレンダリングにより、実行時にレンダリングを制御するロジックを適用します。<sup>8</sup> 条件付きレンダリングを使用することにより、レイアウト エンジンが起動するレンダリングを制御したり、複数の Web ページおよびマネージド Web サイト間でレンダリング、アイテム、文字列を動的に再利用および共有するためのレンダリング プロパティを適用したりすることができます。レンダリング プロパティの詳細については、「レンダリング プロパティ」のセクションを参照してください。

#### メモ

複数のマネージド Web サイトでデータが共有されるがマークアップは共有されない場合、レイアウト詳細では、プレゼンテーション コンポーネント階層を動的に投入する条件付きレンダリングを使用して、1 つまたは複数のプレースホルダーがある最小限のマークアップが含まれるレイアウトを指定することができます。

たとえば、複数のマネージド Web サイトが関係する 1 つのソリューション内の各ページに、次に予定されているイベントの 24 時間前に開始される同じメッセージが表示されるとします。

---

<sup>7</sup> レンダリング プロパティの詳細については、

<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。

<sup>8</sup> 条件付きレンダリングの詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Rules%20Engine%20Cookbook.aspx> から『ルール エンジン クックブック』を参照してください。

この場合、条件付きレンダリングを使用してこれらの要件を満たすことができます。イベントに関する情報を表すレンダリングを再利用するようすべてのマネージド Web サイトを設定します。24 時間以内に発生する予定のイベントを表すアイテムを決定するロジックが含まれるグローバルな条件付きレンダリング条件を追加します。条件付きレンダリング アクションを使用して、レンダリングのデータソースをイベント アイテムに設定します。このようなイベントが発生しない場合、24 時間以内にイベントが発生しないことを示すデフォルト アイテムにレンダリングのデータ ソースを設定するか、この条件下では出力を生成しないようプレゼンテーション コンポーネントを設定します。

### 1.1.4 CSS

CSS (Cascading Style Sheet) ファイルを使用すると、視覚的に異なるスタイルを使用して同じマークアップを表示することができます。CSS に対する参照をマークアップ ストリームに動的に挿入することにより、異なる状況下でデータを別の方法で表示することができます。

たとえば、複数のマネージド Web サイトで同じデータおよびプレゼンテーション コンポーネントを共有するが、プレゼンテーション用として別の CSS ファイルを使用するとします。

この場合、CSS を使用してこれらの要件を満たすことができます。同じホーム アイテムを共有するよう各マネージド Web サイトを設定します。すべてのページの HTML <head> エレメントにレンダリングを実装し、適切な CSS ファイルへのリンクが含まれるようにします。

レイアウト エンジン は、すべてのマネージド Web サイトに対して同じ HTML を提供しますが、異なる CSS を適用します。

### 1.1.5 フィールド値の継承

プレゼンテーション コンポーネントは、コンテキスト アイテムの先祖のフィールド値を再利用することができます。

たとえば、1 つの Web サイト上のすべてのページで同じロゴを使用することができます。CMS ユーザーがページのロゴを指定しない場合、ページのロゴは、ユーザーがロゴを指定するページに最も近い先祖のロゴになります。

この場合、フィールド値を継承してこれらの要件を満たすことができます。種類が Image で名前が Logo であるフィールドが含まれるようページのすべてのデータ テンプレートを設定します。マネージド Web サイトのホーム アイテムに Logo フィールドを投入します。コンテキスト アイテム内のフィールドに値が含まれる場合はこのコンテキスト アイテム内のフィールド値を使用するか、このフィールドの値が含まれるコンテキスト アイテムに最も近い先祖内のフィールド値を使用するロゴ レンダリングを作成します。

プレゼンテーション コンポーネントは、コンテキスト アイテム内の Logo フィールドに値が含まれる場合にこのフィールドで参照されるイメージを使用して HTML <img> エレメントを出力します。それ以外の場合、プレゼンテーション コンポーネントは、このフィールドの値が含まれるコンテキスト アイテムに最も近い先祖の Logo フィールドで参照されるイメージを使用して HTML <img> エレメントを出力します。

#### メモ

多くの場合、デベロッパーは、イメージとテキスト以外にもフィールド継承を使用して、CMS ユーザーがスタイル シートを選択したりプレゼンテーションの他の側面を制御したりできるようにします。

### 1.1.6 RSS を使用した再利用および共有

RSS (Really Simple Syndication) を使用してデータを再利用および共有することができます。<sup>9</sup>

#### メモ

この文書では RSS については説明しません。

#### メモ

データを再利用および共有するために外部 RSS ソースにアクセスするプレゼンテーション コンポーネントを実装することができます。

---

<sup>9</sup> RSS の詳細については、[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 機能の詳細については、<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。

## 1.2 アイテムとフィールド値の共有および再利用

次のテクニックを使用して、Sitecore アイテム内のフィールド値を共有および再利用することができます。

### メモ

レンダリング プロパティを使用してアイテムを共有および再利用することもできます。レンダリング プロパティの詳細については、「レンダリング プロパティ」のセクションを参照してください。

### 1.2.1 エイリアス

エイリアスは、アイテムに別の URL を提供します。エイリアスを使用することにより、複数のマネージド Web サイト間でアイテムを共有することができます。<sup>10</sup>

### メモ

エイリアスはすべてのマネージド Web サイトに適用されます。レイアウト エンジン、エイリアスと一致する HTTP 要求を処理する場合、すべてのマネージド Web サイトについてコンテキスト アイテムを共有アイテムに設定します。エイリアス アイテムは任意の論理サイト定義の範囲を超えて存在でき、すべてのマネージド Web サイトでこのアイテムが共有されます。<sup>11</sup>

たとえば、ホスト名が異なる複数のマネージド Web サイトで FAQ (Frequently Asked Questions) の単一リストが共有されるとします。

この場合、エイリアスを使用してこれらの要件を満たすことができます。共有 FAQ ページを表す新しい単一アイテムを設定します。質問とその回答をストアするためのフィールドを持つ FAQ エントリを表すよう FAQ アイテムの子をそれぞれ設定します。FAQ アイテムでエイリアス `faq` を選択します。

すべてのマネージド Web サイトについて、レイアウト エンジン、URL パス `/faq.aspx` を解釈する際にコンテキスト アイテムを共有 FAQ アイテムに設定します。

### メモ

特定のサイトの FAQ に質問が表示されないようにするには、質問と回答用のデータ テンプレートに選択フィールドを追加します。ユーザーがこのフィールドでサイトを選択しない場合、質問と回答はすべてのサイトに適用されます。ユーザーが 1 つまたは複数のサイトを選択する場合、質問と回答はこれらのサイトのみ適用されます。また、FAQ プレゼンテーション レンダリングには、各マネージド Web サイト内、またはシステム上の任意の場所にストアされている FAQ エントリを含めることができます。

---

<sup>10</sup> エイリアスの詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Content%20Reference.aspx> から『コンテンツ参照マニュアル』を参照してください。

<sup>11</sup> コンテキスト アイテムの詳細については、

<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。



**メモ**

マネージド Web サイトごとに異なる共有データの表示の詳細については、「プレゼンテーション コンポーネントを使用したデータの共有および再利用」のセクションを参照してください。

## 1.2.2 選択フィールド

データ テンプレート内の選択フィールドを使用すると、CMS ユーザーがアイテムの編集時に 1 つまたは複数の Sitecore アイテムを選択できるようになります。任意の数の選択フィールドをデータ テンプレートに追加し、選択したアイテムからデータを読み取るようプレゼンテーション コンポーネントを設定することができます。

データ テンプレートで次のフィールド タイプを使用すると、CMS ユーザーが単一アイテムを選択できるようになります：

- Droplink
- Droptree
- Grouped Droplink

データ テンプレートで次のフィールド タイプを使用すると、CMS ユーザーがゼロ以上のアイテムを選択できるようになります：

- Checklist
- Multilist
- Treelist
- TreelistEx

**メモ**

一定の数のアイテムを選択することをユーザーに求めるデータ検証を選択フィールドに設定することができます。<sup>12</sup>

たとえば、複数のマネージド Web サイトで共通コンテンツ ツリーを共有するとします。サイトごとに一意のプレゼンテーション コンポーネント セットを使用します。1 つのサイトがプロトタイプを提供し、他のすべてのサイトはこのプロトタイプからインフォメーション アーキテクチャとデータの両方を導出してこのインフォメーション アーキテクチャにデータを投入します。導出された各サイトには、プロトタイプ サイトには表示されないアイテムを含めることができます。プロトタイプ サイトには、導出された各サイトには表示されないアイテムを含めることができます。導出されたサイトではプロトタイプ サイトのフィールド値が使用されますが、CMS ユーザーは導出された各サイトでプロトタイプ内のフィールド値を上書きすることができます。

---

<sup>12</sup> データ検証の詳細については、

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

この場合、選択フィールドを使用してこれらの要件を満たすことができます。データ テンプレートごとに Droptree フィールドを作成し、ユーザーがプロトタイプ サイトからアイテムを選択できるようにします。プロトタイプ サイトでは、このフィールドは常にブランクです。プレゼンテーション コンポーネントでは、アイテムにはフィールドの値が含まれず、Droptree フィールドによって指定されたアイテムにこのフィールドの値が含まれる場合、このアイテムのこのフィールドの値が使用されます。

## アイテム選択データ テンプレート

アイテム選択データ テンプレートは、1 つのアイテムから別のアイテムへのシンボリック リンクとして機能します。選択データ テンプレートを使用すると、コンテンツ ツリー内の複数の場所でアイテムを再利用したり、複数のマネージド Web サイト間でアイテムを共有したりすることができます。

アイテム選択データ テンプレートには、単一アイテムを選択することをユーザーに求める選択フィールドが含まれます。アイテム選択データ テンプレートに基づくアイテムを処理する場合、プレゼンテーション コンポーネントは選択されたアイテムからデータを読み取ります。選択フィールドの詳細については、「選択フィールド」のセクションを参照してください。

### メモ

CMS ユーザーはリッチ テキスト エディター、レンダリング プロパティまたは他の機能を使用して、アイテム選択データ テンプレートに基づくアイテムにリンクさせることができます。コード内ではこれらの条件について説明してください。アイテム選択データ テンプレートに基づくアイテムに対する HTTP 要求を処理するために `HttpRequestBegin` パイプライン プロセッサを設定することができます。このプロセッサは、選択したアイテムに対してコンテキスト アイテムを更新したり、エラーをログに記録してこのアイテムにリダイレクトすることができます。また、アイテム選択データ テンプレートに基づくアイテムに対してレイアウト詳細を定義することができます。ここでは、これらのプレゼンテーション コンポーネントは、選択アイテムから一部のデータを読み取り、このフィールドで選択されたアイテムから他のデータを読み取ることができます。

### メモ

各アイテムには、アイテムの選択基準となったアイテムの選択アイテムの数とは関係なく、単一の URL があります。1 つのアイテムに対する複数の URL の有効化の詳細については、「エイリアス」のセクションを参照してください。

### メモ

適切な CMS アクセス権を持つユーザーに対してのみ表示されるレンダリング プロパティとは異なり、アイテムを更新できる任意のユーザーがこのアイテム内のすべての選択フィールドを更新することができます。フィールド セキュリティを使用すると、選択フィールドを更新できるユーザーを絞り込むことができます。<sup>13</sup>

たとえば、レンダリングにより、コンテンツ ツリーを反復してナビゲーション リンクを生成するとします。ナビゲーション内の複数の場所には特定のアイテムを表示する必要があります。

この場合、アイテム選択データ テンプレートを使用してこれらの要件を満たすことができます。Droptree のフィールドを持つアイテム選択データ テンプレートを作成します。コンテンツ ツリーにリンクを表示する必要があるアイテム選択データ テンプレートに基づくアイテムを作成します。アイテム選択データ テンプレートに基づくアイテムごとに、この場所に表示するアイテムを選択します。アイテム選択データ テンプレートに基づくアイテムを処理する際に選択したアイテムにリンクするようナビゲーション レンダリングを更新します。

<sup>13</sup> フィールド セキュリティの詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Client%20Configuration%20Cookbook.aspx> から『クライアント設定クックブック』を参照してください。

### 1.2.3 公開ターゲット

公開ターゲットは、Sitecore が公開先として使用可能なデータベースを表します。<sup>14</sup> 公開ターゲットを使用すると、コンテンツ デリバリー環境をスケール変更したり、公開データを Sitecore インスタンスごとに分離したりすることができます。

公開制限内で公開ターゲットを選択しない限り、Sitecore では、すべての公開ターゲットに対してすべてのデータが公開され、すべてのコンテンツ デリバリー環境内でデータを再利用および共有できるようになります。

#### 重要

Sitecore では、共通の先祖を持つ子孫が、この祖先の公開先ではない公開ターゲットに公開されることはありません。

### 1.2.4 プロキシ

プロキシを使用すると、アイテムまたはアイテムのブランチがコンテンツ ツリー内の複数の場所に表示されます。<sup>15</sup> プロキシを使用して、アイテムまたはアイテムのブランチをコンテンツ ツリー内の複数の場所で再利用したり、アイテムのブランチを複数のマネージド Web サイトで共有したりすることができます。

#### 注意

Sitecore 6.4 ではプロキシのサポートは終了予定で、今後の Sitecore のバージョンではプロキシはサポートされません。

#### 注意

プロキシを有効にすると、Sitecore インストール環境のパフォーマンス全体に悪影響を与える可能性があります。データベース内でプロキシを使用する必要がない場合は、このデータベースに対してプロキシを有効にしないでください。

プロキシの使用に関するパフォーマンスへの悪影響は、次の 2 つの要因によって大幅に増加します：

- インストール環境内のプロキシ定義の数
- プロキシ定義によって参照されるコンテンツ ツリーのサイズ

パフォーマンスの低下を回避するために、次の各事項に注意することをお勧めします：

- CMS ビジネス ユーザーによるプロキシの作成を許可しないようにしてください。
- 大量のアイテムのブランチに対してプロキシを設定しないようにしてください。

---

<sup>14</sup> 公開ターゲットの詳細については、

<http://sdn.sitecore.net/Developer/Configuring%20Production%20Environments.aspx> から『本番環境の設定ガイド』を参照してください。

<sup>15</sup> プロキシの詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Content%20Reference.aspx> から『コンテンツ参照マニュアル』を参照してください。

- 本番環境で使用可能なプロキシ定義の数を無限に許可しないようにしてください。
- ソリューションを本番環境でリリースする前に、テスト環境で予想されるプロキシの数を慎重にテストしてください。

**重要**

公開ターゲット データベースではプロキシを無効にし、Master データベースで仮想アイテムを公開するよう設定することをお勧めします。

**メモ**

プロキシを使用する方法によっては、プロキシを使用すると単一アイテムに対して複数の URL が使用される結果として、検索エンジンのランキングに影響を与える可能性があります。「デバイス」のセクションおよび「アイテム選択データ テンプレート」のセクションで説明されている例の要件を満たす場合のように、プロキシなしでデータを共有および再利用することができます。

**メモ**

アイテムごとに、各デバイスによって単一レイアウトがトリガーされます。単一アイテムに別のレイアウトを適用するために別のデバイスを使用することができます。レンダリング プロパティ、条件付きレンダリングおよび他のテクニックを使用して、状況に応じて再利用および共有されるデータをフォーマットすることができます。コンテキストごとに異なるデータのフォーマットの詳細については、「プレゼンテーション コンポーネントを使用したデータの共有および再利用」のセクションを参照してください。

## 1.2.5 クローン

クローンはプロキシに類似しており、アイテムのコピーを共有します。アイテムのコピーはクローニングに非常に似た操作ですが、クローン アイテム作成後の動作がコピーと異なります。クローニングはコピーよりも柔軟で、プロキシのパフォーマンス オーバーヘッドがありません。

クローンは「インテリジェントな」標準値リンクを持つアイテムで、オリジナルと呼ばれる別のアイテムをポイントします。オリジナルとは、クローンのベースとなる元のアイテムのことです。クローンには、オリジナルと同じデータ テンプレートが備わっています。Sitecore は、標準テンプレートの [Advanced] セクションにある [\_\_Source] フィールドを使用して、クローンのソースを指定します。[\_\_Source] フィールドの URI が有効なアイテムが有効なクローンです。クローニングのロジックは、クローン アイテム用のフィールド値を取得する際に適用されます。

クローン アイテムは、以下のフィールドをオリジナルから継承しません。

- Lock
- Workflow
- Workflow State
- Updated
- Created

- Updated By
- Created By
- Source
- Revision

### アイテムのクローニング

アイテムのクローンを作成する場合、Sitecore ではアイテムの子孫ごとにクローンを作成します。Sitecore では、クローンにはどのフィールド値も格納しませんが、オリジナルの値はすべて引き継ぎます。ただし、フィールドが変更されると、クローンはその変更を保存して、値のローカル コピーを作成します。

アイテムをクローニングするには、コンテンツ ツリー内の該当するアイテムを選択します。コンテンツ エディターのリボンにある [設定] タブの [クローン グループ] で、クローンおよびアンクローン制御を使用します。

### アンクローン

アンクローン操作を行うとクローンがアイテムに変わります。この操作では、すべてのフィールド値がオリジナルからクローンにコピーされ、[\_\_Source] リンクが削除されて、実質的に 2 つの独立したアイテムが作成されます。

### 通知

新しいアイテム、フィールド、またはバージョンなどの変更がオリジナルにある場合は、その UI に通知がプッシュされるため、エンドユーザーはその変更を受け入れたり拒否したりすることができます。各通知は、その通知が適用されるアイテムを含んでいる Sitecore データベースの通知テーブルにレコードとして格納されます。

### クローンの削除

クローンのあるアイテムを削除すると、そのクローンが正常なコンテンツ アイテムになります。これは、実質的にアンクローン操作と同じです。

#### メモ

クローンは Sitecore の正常なアイテムです。プロキシと異なり、特殊なオブジェクトや背後のデータ構造はありません。クローンでは、他の Sitecore アイテムと同様に、ワークフロー、公開制限、リマインダ、レイアウト定義を使用できます。

#### メモ

データ テンプレートまたはサイト セクションをクローンすることはできません。

## 1.2.6 ワイルドカード アイテム

ワイルドカード アイテムは、別の方法ではアイテムの子にマッピングされるすべての URL パスをこのアイテムの共通の子にマッピングします。ワイルドカード アイテムを使用すると、Sitecore アイテムを含む任意のデータを再利用および共有することができます。

アイテムの名前がアスタリスク ("\*") である場合、このアイテムはワイルドカードです。URL パスのディレクトリ部分がアイテムにマッピングされるが、URL パスのファイル名部分と一致する名前を持つ子がこのアイテムにはなく、その場所にワイルドカード アイテムが存在する場合、Sitecore はコンテキスト アイテムをワイルドカード アイテムに設定します。

たとえば、複数のマネージド Web サイトで外部 PIM (Product Information Management) システム内のデータを共有するとします。任意のマネージド Web サイト上の任意の製品の URL は `/product/<ProductID>.aspx` です。

この場合、ワイルドカード アイテムを使用してこれらの要件を満たすことができます。各 Web サイトの下に `/product` アイテムを作成します。各 `/product` アイテムの下に、名前がアスタリスク ("\*") であるアイテムを作成します。URL 内の製品識別子を使用して PIM システムからデータを読み取るレンダリングが含まれるようワイルドカード アイテムのレイアウト詳細を設定します。

レイアウト エンジンは、ワイルドカード アイテムに対してコンテキスト アイテムを設定し、他の任意のアイテムの場合と同じように、このアイテムのレイアウト詳細に指定されているプレゼンテーション コンポーネントを起動します。

### メモ

各 Web サイトの下に `/product` アイテムを作成するのを回避するためにエイリアスを使用することができます。エイリアスの詳細については、「エイリアス」のセクションを参照してください。

### メモ

Sitecore ワイルドカード アイテムはネストをサポートしています。たとえば、URL `/products/category/product.aspx` と一致する他のアイテムが存在しない場合は、`/Sitecore/Content/Home/Products/*/*` アイテムが処理されます。

## 1.2.7 Web サービス、AJAX および他のインターネット インターフェース

Web サービスなどのインターフェースにより、ネットワーク接続を介してデータ、API および他のサービスを公開します。Web サービス、AJAX および他のインターネット テクノロジを使用することにより、Sitecore 内のデータを再利用したり他のアプリケーションと共有したりすることができます。これには、クライアント内の AJAX も含まれます。

### 重要

Web サービスが実行されるセキュリティ コンテンツを常に明らかにしてください。

### メモ

Sitecore では、URL `/sitecore/shell/webservice/service.aspx` でデフォルトの Web サービスを公開します。任意の Sitecore データまたは API を公開するカスタム Web サービスを実装することもできます。

#### メモ

インターネットを介してデータを公開するための Web サービスは必ずしも必要ではありません。クライアント アプリケーションは、特定のデバイスを起動するためのクエリ文字列パラメーターを使用して、Sitecore サーバー上のアイテムの URL を要求することができます。このデバイスのレイアウトにより、外部アプリケーションに必要なデータをフォーマットすることができます。デバイスの詳細については、「デバイス」のセクションを参照してください。

## 1.3 文字列の共有および再利用

次のテクニックを使用して、Sitecore アイテムのフィールド値にはストアしない文字列を共有および再利用することができます。

### メモ

レンダリング プロパティを使用して文字列を再利用することもできます。レンダリング プロパティの詳細については、「レンダリング プロパティ」のセクションを参照してください。

### 1.3.1 Snippet

Snippet を使用して、CMS ユーザーが事前定義された文字列をリッチ テキスト エディターのフィールド値に挿入できるようにします。<sup>16</sup> Snippet を使用すると、CMS ユーザーが使用可能なフレーズ、トークンおよび他の文字列をリッチ テキスト エディターのフィールド値に挿入できるようになります。

### 重要

Snippet を更新しても、この Snippet が含まれるフィールド値は更新されません。

たとえば、ある組織が頻繁にスローガンをリッチ テキスト エディターのフィールド値に挿入するとします。

これらの要件を満たすには、ユーザーがスローガンを毎回キーボードで入力するのではなくマウスを数回クリックするだけでスローガンを挿入できるようにするための Snippet を設定します。

この Snippet は、ユーザーがリッチ テキスト エディターでこの Snippet をド롭ダウン メニューから選択すると、フィールド値に挿入されます。

### 1.3.2 デクショナリ翻訳

Sitecore デクショナリは、用語および翻訳をストアするためのアイテムを使用して特定の文字列を別の言語に翻訳します。デクショナリを使用すると、フレーズおよび文字列を言語ごとに再利用可能な値に翻訳することができます。

`Sitecore.Globalization.Translate.Text()` メソッドを使用して、トークンを翻訳することができます。デクショナリ エントリ アイテムの `Key` フィールドの値が、`Sitecore.Globalization.Translate.Text()` メソッドの最初のパラメーターに渡された文字列と一致する場合、このメソッドは、このデクショナリ エントリ内の `Phrase` フィールドの値を返します。それ以外の場合、`Sitecore.Globalization.Translate.Text()` は、このメソッドに渡された値を返します。

### メモ

`Sitecore.Globalization.Translate.Text()` メソッドは大文字と小文字を区別します。

---

<sup>16</sup> Snippet の詳細については、

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



ディクショナリ翻訳を使用してデータを再利用する例については、「トークン置換」のセクションを参照してください。

## ディクショナリ エントリの作成方法

ディクショナリ エントリを作成する方法:

1. コンテンツ エディターで `/Sitecore/System/Dictionary` アイテムを選択します。プロジェクト期間全体にわたってディクショナリ内のエントリが少ない場合、ディクショナリ エントリを `/Sitecore/System/Dictionary` アイテムの真下に挿入することができます。それ以外の場合は、`System/Dictionaries/Dictionary Folder` データ テンプレートをを使用してディクショナリ フォルダーを挿入します。たとえば、文字 `a` で始まる用語を収容するために "`a`" という名前のディクショナリ フォルダーを作成するとします。この場合、ネストしたディクショナリ フォルダーを作成することにより、数百ものアイテムが含まれるフォルダーが作成されないようにします。
2. コンテンツ エディターで、`/Sitecore/System/Dictionary` アイテムまたはディクショナリ フォルダー アイテムが選択された状態で、`System/Dictionaries/Dictionary Entry` データ テンプレートをを使用してディクショナリ エントリ アイテムを挿入します。CMS ユーザーがその値を保持する際に用語を識別しやすいような名前をディクショナリ エントリ アイテムに付けます。

### メモ

CMS ユーザーが `/Sitecore/System/Dictionary` フォルダーにアクセスするために `/Sitecore/System` ブランチにアクセスできない場合、このブランチを `/Sitecore/Content/Dictionary` などのいずれかの場所に表示するためのプロキシを使用することができます。プロキシの詳細については、「プロキシ」のセクションを参照してください。

3. コンテンツ エディターで、新しいディクショナリ エントリ アイテム内の Data セクション内の Key フィールドに、翻訳するトークンを入力します。
4. コンテンツ エディターで、新しいディクショナリ エントリ アイテム内の Data セクション内の Phrase フィールドに、現在の言語の値を入力します。

### 重要

各ディクショナリ エントリ アイテムで、言語ごとに Data セクション内の Phrase フィールドに値を入力します。

### 重要

ディクショナリ フォルダーおよびディクショナリ エントリを公開することを忘れないでください。

## 1.3.3 トークン置換

トークン置換により、代替値を持つデータ内の文字列を置換します。プレゼンテーション コンポーネント内のトークン置換を使用して、ページ間および場合によっては言語間でデータを再利用したり、複数のマネージド Web サイト間でこのデータを共有したりすることができます。

### 重要

これ以外の方法ではフィールド値に表示されないトークンを選択します。

**重要**

ユーザーがページ エディターでインライン編集を行っている場合は、インライン編集をサポートするフィールドのトークンを置換しないでください。

**メモ**

Sitecore アイテム、設定ファイル、.NET リソース ファイルまたは他の情報ストアのフィールドにストアされている値を持つトークンを置換することができます。

たとえば、ある組織のスローガンが定期的に変更されるとします。このスローガンはリッチ テキスト エディターのフィールドに平文として表示される必要があります。

この場合、トークン置換を使用してこれらの要件を満たすことができます。\$slogan などのトークンをフィールド値に挿入し、次のセクションで説明されているように、renderField パイプラインを使用して実行時にこのトークンをスローガンに置換します。スローガンは単一値であるため、スローガンに対してディクショナリ翻訳を使用することができます。

## トークン置換の実装方法: renderField パイプライン

レイアウト エンジンは、renderField パイプラインを呼び出すことにより、内部表現の HTML への変換や動的リンクの拡張など、フィールド値の処理を行います。前のセクションで定義されている要件を満たす次の例に基づいてプロセッサを renderField に追加することにより、トークン置換を実装することができます。

1. 次の例に基づくクラスを Visual Studio プロジェクトに追加します:

```
namespace Sitecore.Sharedsource.Pipelines.RenderField
{
    using System;

    public class ApplySlogan
    {
        public string Token
        {
            get;
            set;
        }

        public void Process(Sitecore.Pipelines.RenderField.RenderFieldArgs args)
        {
            if (Sitecore.Context.PageMode.IsPageEditorEditing)
            {
                return;
            }

            Sitecore.Diagnostics.Assert.IsNotNullOrEmpty(this.Token, "Token");
            string slogan = Sitecore.Globalization.Translate.Text(this.Token);
            Sitecore.Diagnostics.Assert.IsNotNullOrEmpty(slogan, "slogan");

            if (!String.IsNullOrEmpty(args.Result.FirstPart))
            {
                args.Result.FirstPart = args.Result.FirstPart.Replace(
                    this.Token,
                    slogan);
            }

            if (!String.IsNullOrEmpty(args.Result.LastPart))
            {
                args.Result.LastPart = args.Result.LastPart.Replace(
```

```
        this.Token,  
        slogan);  
    }  
}  
}
```

2. `renderField` パイプラインに対するプロセッサを `Sitecore.Pipelines.RenderField.AddBeforeAndAfterValues` プロセッサの直前に追加します:

```
<renderField>  
...  
  <processor type="Sitecore.Sharedsource.Pipelines.RenderField.ApplySlogan, assembly">  
    <token>$$slogan</token>  
  </processor>  
  <processor  
    type="Sitecore.Pipelines.RenderField.AddBeforeAndAfterValues, Sitecore.Kernel" />  
  ...  
</renderField>
```

この例の場合、「ディクショナリ エントリの作成方法」のセクションで説明されているようにディクショナリ エントリを作成します。ディクショナリ エントリ アイテムで、Data セクション内の Key フィールドにトークン `$$slogan` を入力します。Data セクション内の Phrase フィールドに、現在のスローガンを入力します。ディクショナリ翻訳の詳細については、「ディクショナリ翻訳」のセクションを参照してください。

ユーザーがページ エディターでインライン編集を行っていない限り、このプロセッサにより、`renderField` パイプラインを使用してレンダリングされたフィールド値内のトークン `$$slogan` は、対応するディクショナリ エントリの値に置換されます。

#### メモ

スローガンごとに個々のデータ テンプレートを作成し、このデータ テンプレートの標準値にスローガンをストアすることができます。

#### メモ

CMS ユーザーがトークンを記憶しなくてもいいようにするには、トークンが含まれる Snippet を作成します。たとえば、トークン `$$slogan` を挿入する Snippet を定義することができます。Snippet の詳細については、「Snippet」のセクションを参照してください。