

# Sitecore CMS 6.4 以降

# プレゼンテーション コンポーネント クックブック

CMS 管理者、アーキテクト、開発者のための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 開発インフラストラクチャ	7
2.1 要件の分析	8
2.1.1 ネーミング コンベンション	8
2.2 ASP.NET	0
2.2.1 ASP.NET のタグ プリフィックス10	0
2.2.2 ASP.NET コントロール識別子 (ID)11	1
2.2.3 コードビハインド、コードビサイド、コード ファイル?11	1
2.3 デベロッパー センター13	3
2.3.1 デベロッパー センターへのアクセス方法14	4
2.3.2 デベロッパー センターで最近使ったアイテムにアクセスする方法14	4
2.3.3 デベロッパー センターからコンテンツ エディターにアクセスする方法14	4
2.3.4 デベロッパー センターの定型コード ファイル15	5
デベロッパー センターの定型コード ファイルの編集方法15	5
2.4 Microsoft Visual Studio	7
2.4.1 Visual Studio のソリューション エクスプローラーの表示方法18	8
2.4.2 Visual Studio のソリューション エクスプローラーですべてのファイルを表示または非表示に	
する方法 18	
2.4.3 Visual Studio の Web アプリケーション プロジェクトの作成方法18	8
2.4.4 既存のファイルを Web アプリケーション プロジェクトに追加する方法22	2
2.4.5 Sitecore のコントロールを Visual Studio のツールボックスに追加する方法22	2
2.4.6 Visual Studio を使って .NET コードをデバッグする方法24	4
2.4.7 Web サービス メソッドのコレクションの作成方法25	5
2.4.8 Visual Studio のパフォーマンスを最適化する方法	6
Chapter 3 レイアウト詳細29	9
3.1 レイアウト詳細の使い方30	0
3.1.1 デバイス エディター	0
デバイス エディターへのアクセス方法30	0
レイアウトの選択方法31	1
コントロールの追加方法3:	1
コントロールの順序の変更方法31	1
コントロールの削除方法32	2
コントロールの置換方法	2



3.2 レイアウト詳細をスタンダード バリューにリセットする方法3	3
3.3 レイアウト詳細のコピー方法3	\$4
3.4 使用されるプレゼンテーション コンポーネントを判定方法	35
3.5 デバイスを使う3	6
3.5.1 デバイスの作成方法3	6
3.5.2 デバイスのアクティベーションの基準の定義方法3	6
3.6 レイアウト プリセット	8
3.6.1 レイアウト プリセットの作成方法3	8
3.6.2 レイアウト プリセットをデータ テンプレートまたはアイテムと関連付ける方法	39
3.6.3 ページ エディターを使用してレイアウト プリセットを適用する	;9
Chapter 4 コントロール 4	1
4.1 コントロールの出力の表示方法4	2
4.2 レンダリング定義アイテム4	-3
4.3 コントロール プロパティのアクセス方法 4	4
4.4 Visual Studio を使ったコントロール プロパティの設定方法	-5
4.5 共通のコントロール プロパティ4	6
4.5.1 コントロール キャッシュ オプションの構成方法4	6
4.5.2 コントロールのデータ ソースの構成方法4	17
4.5.3 レンダリング パラメーターのコントロールへの渡し方4	-7
4.6 プレースホルダー	9
4.7 サブレイアウト	50
4.8 FieldRenderer Web コントロール	51
4.9 XSL レンダリング5	52
4.9.1 新規の XSL レンダリングの作成方法5	52
4.9.2 XSL レンダリングの出力の表示方法5	53
4.9.3 XSL レンダリングの定型コード ファイル5	53
メインの XSL テンプレート ブロック5	53
4.9.4 カスタムの XSL テンプレート ライブラリ5	54
XSL テンプレート ライブラリの作成方法5	54
XSL レンダリングの XSL テンプレート ライブラリを参照する方法5	55
4.9.5 カスタムの XSL 拡張メソッド5	56
XSL 拡張メソッド ライブラリ クラスの作成方法5	57
カスタムの XSL 拡張メソッド ライブラリの登録方法5	57
.NET XSL 拡張ライブラリの使い方5	58
メソッドを名前空間 sc に追加する方法5	;9
XSL 拡張クラスのプロパティへのアクセス方法5	;9



4.9.6 XSL 拡張メソッドの例	60
GetHome():Sitecore.Data.Items.Item を返す	60
GetRandomSiblings():XML を使って複数の値を返す	61
4.10 Web コントロール	63
4.10.1 Web コントロール クラスの作成方法	63
4.10.2 Web コントロールの登録方法	64
4.10.3 Web コントロールへのプロパティの追加方法	65
4.11 メソッド レンダリング	66
4.11.1 メソッド レンダリング クラスとメソッドの作成方法	66
4.11.2 メソッド レンダリングの登録方法	67
4.12 URL レンダリング	68
4.12.1 URL レンダリングの登録方法	68
4.13 レンダリング設定データ テンプレートを実装する方法	70
Chapter 5 レイアウトとサブレイアウト	71
5.1 レイアウトの作成	72
5.1.1 デベロッパー センターを使ったレイアウトの作成方法	72
5.1.2 Web フォームをレイアウトとして登録する方法	72
5.2 サブレイアウトの作成	74
5.2.1 デベロッパー センターを使ったサブレイアウトの作成方法	74
5.2.2 Web ユーザー コントロールをサブレイアウトとして登録する方法	74
5.3 コントロールをレイアウトまたはサブレイアウトに追加する	76
5.3.1 デベロッパー センターを使ってコントロールをレイアウトまたはサブレイアウトに追加するス	与法
76	
5.3.2 Visual Studio を使ってコントロールをレイアウトまたはサブレイアウトに追加する方法	<del>.</del> 76
5.4 コードビサイドをレイアウトまたはサブレイアウトに追加する	78
5.4.1 既存のファイルを削除してコードビサイドをレイアウトまたはサブレイアウトに追加する方	法 78
5.4.2 ファイルを作成してコードビサイドをレイアウトまたはサブレイアウトに追加する方法	79
5.5 レイアウトまたはサブレイアウトの部分クラス ファイルを追加しコード ファイルをコードビハイ	ンドで
置換する方法	81
Chapter 6 RSS 機能	82
6.1 Sitecore の RSS の概要	83
6.1.1 フィード定義アイテム	83
RSS フィードの作成方法	83
6.1.2 配信アイテム	84
アイテムを配信可能にする方法	85
6.2 RSSの構成	87



6.2.1	Feeds.MaximumItemsInFeed 設定	87
6.2.2	Feeds.ItemExpiration 設定	87



# イントロダクション

このクックブックはプレゼンテーション コンポーネントを取り扱う Sitecore のシステム管理者、 アーキテクト、開発者のためのヒントと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

- Chapter 1 イントロダクション
- Chapter 2 開発インフラストラクチャ
- Chapter 3 レイアウト詳細
- Chapter 4 コントロール
- Chapter 5 レイアウトとサブレイアウト
- Chapter 6 RSS 機能

http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20 Component%20Reference.aspx から『プレゼンテーション コンポーネント リファレンス』を参照して ください。プレゼンテーション コンポーネントの技術的な詳細とサンプル コード は、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u> から『プレゼンテーション コンポーネント XSL リファレンス』を、また <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20API%20Cookbook.aspx</u> 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。プレゼンテーション コンポーネントのトラブル シューティングに関する情報は <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20Troubleshooting%20Guide.aspx</u> から『プレゼンテーション コンポーネント トラブル シューティ ング』を参照してください。

<sup>1</sup> プレゼンテーション コンポーネントに関する詳細については、



# **Chapter 2**

# 開発インフラストラクチャ

この章では Sitecore のレイアウト エンジン プレゼンテーション コンポーネントを使用した開 発環境を構成するためのテクニックを解説します。

この章には次のセクションがあります。

- 要件の分析
- ASP.NET
- デベロッパー センター
- Microsoft Visual Studio



### 2.1 要件の分析

開発の労力を最小限にするために、実装の前に要件の十二分な分析を行うことが重要です。と りわけ、再利用可能なデータ、アプリケーション、プレゼンテーション コンポーネントを認識することが 重要です。情報体系 (Information Architecture)、プレゼンテーション、機能要件を文書化 し、そしてこれらの要件をコンポーネントに対応付けます。コンポーネントにはデータ テンプレート、ワ ークフロー、セキュリティ、プレゼンテーション、挿入オプション、その他のプロパティ、などがあります。各 プレースホルダーのキーと各プレゼンテーション コンポーネントのキャッシュ オプションを文書化します。 できるだけ少ない基準でアウトプット キャッシュをサポートするプレゼンテーション コンポーネントを設 計します。<sup>2</sup>

### 2.1.1 ネーミング コンベンション

プロジェクトの各コンポーネントは標準化されたネーミング コンベンションに沿う必要があります。す べてのプロジェクトは名前をもつ必要があります。たとえば MyWebSite などです。多くのプロジェクト には略号や短い名前があります。たとえば、MWS や MyWebSite のための mws、などです。

開発者はプロジェクトの略号をプロジェクトの ASP.NET のタグのプリフックスとしてよく使用しま す。これについては「ASP.NET のタグ プリフィックス」のセクションで説明します。管理者はプロジェク ト名や略号をファイル システムのパスやその他のオブジェクトの名前によく使用します。たとえば、管 理者は MyWebSite プロジェクトのインストール先を C:¥inetpub¥sitecore¥MyWebSite や C:¥inetpub¥sitecore¥MyWebSite¥MWS などのフォルダーにすることができます。

開発者は MyWebSite プロジェクトのレイアウトやサブレイアウトのファイルを /layout/mws ディ レクトリに、対応する定義アイテムを /sitecore/layout/layouts/mws と /sitecore/layout/sublayouts/mws に保存することができます。

開発者は MyWebSite プロジェクトの XSL レンダリングを /xsl/mws ディレクトリに、対応する 定義アイテムを /sitecore/layout/renderings/mws に保存することができます。

開発者はその他のリソース、たとえば JavaScript や CSS ファイルなど保存する際に、プロジェクト名や略号にちなんだ名前をディレクトリ名に使ったり、ファイル名に使ったりすることができます。

管理者はプロジェクト名や略号をプリフィックスとしてリレーショナル データベースの名前に使うことが できます。たとえば、開発者は MyWebSite プロジェクトに関連したデフォルトのデータベースの名 前を mwsCore や mwsMaster や mwsWeb などとすることができます。

<sup>&</sup>lt;sup>2</sup> Sitecore のアウトプット キャッシュ関する詳細については、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20Reference.aspx</u>から『プレゼンテーション コンポーネント リファレンス』を参照してください。



開発者はプロジェクト名や略号を .NET アセンブリの名前に使うことができます。たとえば、 MyWebSite ソリューションが .NET アセンブリを 1 つ含む場合、開発者は MyWebSite.dll とい う名前のアセンブリを生成する Visual Studio プロジェクトを構成することができます。 MyWebSite ソリューションが複数の .NET アセンブリを含む場合、開発者は mws プリフィックスで 始まるファイル名、たとえば mwh.bl.dll をビジネス ロジックとし、mwh.web.dll を Web コンポー ネントのファイル名とした、アセンブリを生成する Visual Studio プロジェクトを構成することができ ます。

一般に、各 .NET アセンブリはクラスを共通の名前空間に含むようにします。たとえば、 MyWebSite.dll アセンブリ、または mwh.bl.dll アセンブリは、名前空間 MyWebSite 内で各種 の名前空間の中にクラスを含むことができます。一方、mwh.web.dll アセンブリは 名前空間 MyWebSite.Web.UI のみにクラスを含むことができます。

### 重要

ある Sitecore のインスタンスと別のインスタンスで重複したリソースなどの管理を簡素化するため に、プレゼンテーション コンポーネント定義アイテムをファイルへのパスに対応したプロジェクトに特有 のフォルダーに保存します。たとえば、レイアウト ファイル /layouts/mws/Web.aspx には、レイアウ ト定義アイテム /Sitecore/Layout/Layouts/mws/Web を使います。

### 重要

web.config にある Sitecore の定義アイテムと要素は .NET コンポーネントを名前空間、クラ ス、アセンブリ名を使って参照します。ふさわしい名前を選択してそれを使い続けることで、将来再 構成が必要になることを避けることができます。

### ヒント

.NET フレームワークまたは Sitecore API の名前空間に対応した名前空間を使います。たと えば、Microsoft は名前空間 System.Web.UI.WebControls を Web コントロールに使いま す。Sitecore は Sitecore.Web.UI.WebControls を使います。カスタムの Web コントロールに MyWebSite.Web.UI.WebControls と同等なものを使用することをぜひ検討してください。

### メモ

慣例としては、開発者は通常、レイアウトとサブレイアウトをどちらも /layouts ファイル システム ディレクトリ内に保存します。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



### 2.2 ASP.NET

このセクションでは ASP.NET を使用する際の手順と考慮点について解説します。

### 2.2.1 ASP.NET のタグ プリフィックス

ASP.NET の Web フォーム (Sitecore のレイアウト) と Web ユーザー コントロール (Sitecore のサブレイアウト) はトークンとアセンブリの名前空間をマッピングするためにタグ プリフィッ クスを使用します。 デフォルトでは ASP.NET は Web コントロールをエクスポーズし、 タグ プリフィッ クス asp を System.Web アセンブリの名前空間 System.Web.UI.WebControls にマッピング します。 たとえば:

<asp:textbox runat="server" />

ASP.NET はこのコードに出会うと System.Web.UI.WebControls.TextBox クラスからオブジェクトを作成します。

web.config の ASP.NET のタグ プリフィックスの登録ディレクティブを使って、各 Web フォーム および各 Web ユーザー コントロールで、さらに追加のタグ プリフィックスを任意のアセンブリの任意 の名前空間にマッピングすることができます。 asp または sc タグ プリフィックスをマッピングする必要 はありません。

web.config の /configuration/system.web/pages/controls/add 要素を使って、すべての レイアウトとサブレイアウトでタグ プリフィックスを使用可能にすることができます。たとえば、Sitecore はデフォルトで asp と sc タグ プリフィックス に対してこのアプローチを使います。

```
---
<system.web>
<pages validateRequest="false">
<controls>
<add tagPrefix="sc" namespace="Sitecore.Web.UI.WebControls"
assembly="Sitecore.Kernel"/></a>
```

個々のレイアウトおよびサブレイアウトのタグ プレフックスを登録することができます。次のサンプルコ ードでは mws タグ プレフックスを登録することで MyWebSite.dll アセンブリで実装された名前空 間 MyWebSite.Web.UI.WebControls のクラスを使用できるようにする例を示しています。

```
<%@ Register TagPrefix="mws" Namespace="MyWebSite.Web.UI.WebControls"
Assembly="MyWebSite" %>
```

### メモ

開発者がデベロッパー センターで Web コントロールをレイアウトまたはサブレイアウトにドラッグすると、Sitecore は Web コントロール定義アイテムのタグ プリフィックスを使って適切な登録ディテク



ティブをレイアウトまたはサブレイアウトに追加します。開発者が Web コントロールを Visual Studio のツールボックスから Web フォームまたは Visual Studio の Web ユーザー コントロー ルにドラッグすると、Visual Studio はタグ プリフィックス登録ディレクティブを追加しますが、任意の タグ プリフィックスを使用します。

### 2.2.2 ASP.NET コントロール識別子 (ID)

Sitecore のレイアウトエンジンは ASP.NET を使用します。ASP.NET はページをリテラル コントロールの階層構造に構築し、それにより静的なコンテンツと動的なコンテンツを生成するサーバーコントロールが作成されます。Sitecore のレイアウト、サブレイアウト、レンダリングは ASP.NET のコントロールです。各コントロールはページ内で一意な ASP.NET コントロール識別子をもつ必要があります。

### 2.2.3 コードビハインド、コードビサイド、コード ファイル?

ASP.NET の Web フォームと Web ユーザー コントロールはコードビハインドを使ってデザインを ロジックから分離することができます。コードビハインドは .aspx または .ascx ファイルのマークアップと コントロールを別の .NET コードビハインド ファイルにあるロジックから分離します。たとえば、Web ユーザー コントロール sublayout.ascx は C# のコードビハインド ファイル sublayout.ascx.cs を 持つことができます。

レイアウトのための Page ディレクティブの CodeBehind 属性またはサブレイアウトのための Control ディレクティブはコードビハインド ファイルを参照します。たとえば:

<%@ Control Language="C#" AutoEventWireup="true" **CodeBehind="MySublayout.ascx.cs"** Inherits="Namespace.Web.UI.MySublayout" %>

ASP.NET 2.0 では部分クラスが導入されました。レイアウトやサブレイアウトを部分クラスとして 実装することでデザイナー ファイルと呼ばれる第 3 のファイルが作成されます。たとえば MySubLayout.ascx.designer.cs です。ASP.NET 2.0 では CodeBehind 属性に代わるもの として CodeFile 属性も導入されています。たとえば:

<%@ Control Language="c#" AutoEventWireup="true" Inherits="Namespace.Web.UI.MySublayout" CodeFile="/layouts/MyWebSite/MySublayout.ascx.cs" %>

CodeFile 属性を使用することによるメリットのひとつは、コードビハインド ファイルを開発者がプリコンパイルするのでなく、実行時にコンパイルできることです。

デフォルトでは Visual Studio 2008 は Page または Control ディレクティブの CodeBehind 属性を持つ部分クラスを使って Web フォームと Web ユーザー コントロールを作成します。



デフォルトではデベロッパー センターはサブレイアウトをコードビハインドなしで作成します。開発者 がサブレイアウトのためにコードビハインドを作成する場合、デベロッパー センターは部分クラスを作 成せず、CodeFile 属性を使用します。デベロッパー センターはコードビハインドでのレイアウトの作 成をサポートしません。

### ヒント

レイアウトまたはサブレイアウトをコードビハインドで作成する際に、CodeFile 属性を使ってコンパイ ルのパフォーマンスへの影響を避けることができます。これによって ASP.NET が再起動され、キャッ シュがクリアされます。必ず属性の名前を CodeFile から CodeBehind に変更してください。開発 環境からコンポーネントを移動させる前にコンパイルしてください。コード ファイルを本番環境にコピー する必要を避けるためです。



### 2.3 デベロッパー センター

デベロッパー センターは Sitecore のプレゼンテーションおよびその他のコンポーネントを取り扱う ためのブラウザ ベースのアプリケーションです。デベロッパー センターは Microsoft Visual Studio のような統合開発環境の機能を提供しますが、Windows アプリケーションとしてではなく Web ブラウザで稼働します。

### 重要

デベロッパー センターには Microsoft Internet Explorer 6 以上が必要です。Sitecore は Internet Explorer (IE) 8 を推奨します。<sup>3</sup>

### 重要

Internet Explorer とデベロッパー センターにはどちらも [ファイル] メニューがあります。特に断りのない限り、この文書での [ファイル] メニューとはデベロッパー センターの [ファイル] メニューを 指し、Internet Explorer の [ファイル] メニューではありません。

### 重要

デベロッパー センターにはテーブル (表) を扱うツールがありますが、tables、CSS、またはその両 方を使用してコンポーネントを開発することも可能です。

### メモ

デベロッパー センターまたはテキスト エディターを使用してレイアウト、サブレイアウト、XSL レンダ リングを編集することができます。Sitecore は Visual Studio 2008 で Web アプリケーション プロジェクト モデルを使った C# での開発を推奨します。

**デベロッパー センター**を使うとレイアウト、サブレイアウト、XSL レンダリングの編集を Web ブラウ ザのみを使用して行うことができます。

デベロッパーセンターはクライアントライセンスおよびクライアントのインストールを必要としません。

デベロッパー センターは Visual Studio よりも比較的容易に学ぶことができます。

デベロッパー センターから Sitecore のブラウザ ベースのデバッガーに容易にアクセスできます。

<u>http://sdn.sitecore.net/reference/Sitecore%206/IE%20Configuration%20Reference.aspx</u>から 『Internet Explorer 構成リファレンス』を参照してください。

<sup>&</sup>lt;sup>3</sup> Sitecore の Web クライアントの要件の詳細については、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese.aspx</u>から『Sitecore インスト ール ガイド』を参照してください。Internet Explorer の構成方法の詳細については、



### 2.3.1 デベロッパー センターへのアクセス方法

Sitecore デスクトップからデベロッパー センターにアクセスする方法:

- 1. Internet Explorer で Sitecore ログイン ページ (/sitecore) にアクセスします。
- 2. Internet Explorer で Sitecore ログインページの [ユーザー名] と [パスワード] のフィールド で認証情報を入力します。
- 3. Internet Explorer で Sitecore ログインページで [オプション] を Sitecore ユーザー インター フェースの選択肢が表示されるまでクリックします。
- 4. Internet Explorer で Sitecore ログインページで [デスクトップ] をダブルクリックします。 ブラウ ザで Sitecore デスクトップが表示されます。
- 5. Internet Explorer で Sitecore デスクトップで Sitecore ボタンをクリックし、次に [デベロッパ - センター] をクリックします。Sitecore デスクトップにデベロッパー センターが表示されます。.

### 2.3.2 デベロッパー センターで最近使ったアイテムにアクセスする方法

**デベロッパー センターのスタート ページ**にはレイアウト、サブレイアウト、XSL レンダリング、その 他の最近使ったアイテムを開くショートカットが用意されています。

デベロッパー センターでショートカットを使って最近使ったアイテムにアクセスする方法:

- 1. デベロッパー センターで [表示] メニューをクリックし、次に [スタート ページ] をクリックします。
- 2. デベロッパー センターで [最近使ったファイル] パネルで最近使ったアイテムをクリックします。最近 使ったアイテムはデベロッパー センターに表示されます。

### 2.3.3 デベロッパー センターからコンテンツ エディターにアクセスする方法

デベロッパー センターからコンテンツ エディターにアクセスする方法:

1. デベロッパー センターで [表示] メニューをクリックし、次に [コンテンツ エクスプローラー] をクリッ クします。デベロッパー センターにコンテンツ エクスプローラーが表示されます。

2. デベロッパー センターの [コンテンツ エクスプローラー] で、任意のアイテムをダブルクリックします。デベロッパー センターで選択されたアイテムを使ってコンテンツ エディターが表示されます。



### 2.3.4 デベロッパー センターの定型コード ファイル

レイアウト、サブレイアウト、XSL レンダリング、またはその他のタイプのコードをデベロッパー センタ ーで作成するとき、Sitecore は定義アイテムを作成し、新しいコード ファイルを作成するための定 型コード ファイルをコピーします。<sup>4</sup> Sitecore はこれらの定型コード ファイルを Web サイトのドキュ メント ルートの /sitecore/shell/templates フォルダーに保存します。

### 重要

定型コード ファイルをアップデートする前と後に、そのファイルを Visual Studio のプロジェクトと使 用中のソース コード管理システムに追加することを検討してください。 デバッグの前に、 Visual Studio のソリューション エクスプローラーでファイルをすべて非表示にすることを忘れないでください。

デベロッパー センターが使用する定型コード ファイルを更新することには次の理由があります:

• 新しいすべてのレイアウトまたはサブレイアウトが ASP.NET のタグ プリフィックスを含むよう にするため。

• 新しいすべてのレイアウトまたはサブレイアウトが共通のベース クラスまたはその他のプロパ ティから継承するようにするため。

• 新しいすべての XSL レンダリングのカスタムの名前空間を登録するため。

• 新しいすべての XSL レンダリングの \$home 変数を再定義するため、またはそれを削除 するため。

• 新しいすべての XSL レンダリングに XSL テンプレート ライブラリを含むため。

定型コード ファイルのすべての使用に当てはまらないコードを無効に (コメントに) します。たとえ ば、XSL レンダリングの <xsl:include> XSL 要素が、すべての XSL レンダリングでは必要ないか もしれない一時ライブラリのためであれば、コメントにすることができます。必要な場合には容易にコ メントを消して元に戻すことができます。

### デベロッパー センターの定型コード ファイルの編集方法

デベロッパー センターがレイアウト、サブレイアウト、XSL レンダリングを作成するために使う定型コ ード ファイルを Visual Studio またはテキスト エディターまたはデベロッパー センターで編集するこ

<sup>&</sup>lt;sup>4</sup>新しい XSL レンダリングの定型コード ファイルに関する詳細は、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u>から『プレゼンテーション コンポーネント XSL リファレンス』を参照し てください。



とができます。 定型コードをデベロッパー センターを使って編集するには [ファイル] メニューをクリックし、 [ファイルを開く] をクリックします。 ファイルを選択するダイアログが表示されます。

/sitecore/shell/templates にある定型コード ファイルは次のものを含んでいます。

- layout.aspx: レイアウト ファイルのための定型コード
- layout.aspx.cs: レイアウト コード ファイルのための定型コード
- sublayout.ascx: サブレイアウト ファイルのための定型コード
- sublayout.aspx.cs: サブレイアウト コード ファイルのための定型コード
- xsl.xslt: XSL レンダリング ファイルのための定型コード



### 2.4 Microsoft Visual Studio

このセクションでは Microsoft Visual Studio を使って Sitecore ソリューションを取り扱う開 発者のための情報を提供します。このセクションでは読者が Microsoft Visual Studio 2005 またはそれ以降のバージョンについて知識があることを前提としています。

### 重要

この文書での .NET 名前空間とクラスの名前の記述はすべて大文字と小文字を区別します。 XML、XSL、XPath も同様に大文字と小文字を区別します。

### メモ

この文書では C# と Microsoft Visual Studio 2008 について記述します。他の言語または 他の Microsoft Visual Studio のバージョンのユーザー インターフェースの手順は異なる場合が あります。

Microsoft Visual Studio は ASP.NET、XSL、CSS、JavaScript、その他のリソースを含む 全てのタイプのファイルのための統合された開発環境を提供します。

Microsoft Visual Studio はインテリセンス、自動コード生成、自動インデント、エラー表示、その他の統合開発環境機能を提供します。

Microsoft Visual Studio は .NET コードのためのデバッガーを提供します。

Microsoft Visual Studio は ソース コード管理ツールと統合されています。

### メモ

Visual Studio 2005 SP1 および Visual Studio 2008 で利用できる ASP.NET 2.0 Web アプリケーション プロジェクト モデルはほとんどの Sitecore ソリューションに適します。<sup>5</sup>

### ヒント

デベロッパー センターでレイアウト、サブレイアウト、XSL レンダリングを作成するときには、 Sitecore はウィザードを起動し、定型コード ファイルを新しい場所にコピーし、対応する定義アイ テムを作成します。このプロセスでは、Visual Studio でアイテムを作成し Sitecore で対応する 定義アイテムを手動で作成するよりも、少ない手順で済みます。

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

<sup>&</sup>lt;sup>5</sup> Sitecore ソリューションで Visual Studio Web アプリケーション プロジェクトを作成する方法の 詳細については、



### 2.4.1 Visual Studio のソリューション エクスプローラーの表示方法

Microsoft Visual Studio のソリューション エクスプローラーを表示する方法:

- 1. Visual Studio で Web アプリケーション プロジェクトを開きます。
- Visual Studio で [表示] メニューをクリックし、[ソリューション エクスプローラー] をクリックします。

## 2.4.2 Visual Studio のソリューション エクスプローラーですべてのファイルを表示 または非表示にする方法

ソリューション エクスプローラーですべてのファイルを表示する、またはプロジェクトに含まれていない ファイルを非表示にするには、Web アプリケーション プロジェクトで Visual Studio の**ソリューショ** ン エクスプローラーを表示し、すべてのファイルを表示 のトグルボタンを押します。(通常は Visual Studio の**ソリューション エクスプローラー** の一番上の左から 2 番目にあります。) Visual Studio の**ソリューション エクスプローラー**を表示する方法は「Visual Studio のソリュー ション エクスプローラーの表示方法」のセクションを参照してください。

### 重要

Microsoft Visual Studio 2008 デバッガーはすべてのファイルを表示していると反応しなくなることがありますので、すべてのファイルを表示し、プロジェクトにファイルを追加し、すべてのファイルを 非表示にします。

### 2.4.3 Visual Studio の Web アプリケーション プロジェクトの作成方法

Sitecore は Sitecore ソリューションに Visual Studio の Web アプリケーション プロジェクト モデルをサポートします。

### ۶Ł

Visual Studio 2005 の初期のリリースは Web アプリケーション プロジェクト モデルを含んでいませんでした。 Visual Studio 2005 SP1 は Web アプリケーション プロジェクト モデルを含んでいます。<sup>6</sup>

<sup>&</sup>lt;sup>6</sup> Visual Studio 2005 SP1 に関する詳細は <u>http://msdn.microsoft.com/en-</u> us/vstudio/bb265237.aspxを参照してください。



### 重要

Visual Studio ソリューションを作成し、各 Sitecore ソリューションに少なくとも 1 つは Visual Studio を使うプロジェクトを作成します。新規の各 Sitecore ソリューションに対して 1 度下記の 手順に沿って対応し、それからは新しいプロジェクトは既存のソリューションに適宜追加します。

既存の Sitecore ソリューションに Visual Studio の Web アプリケーション プロジェクトを作成 する方法:

- 1. Visual Studio で [**ファイル**] メニューをクリックし、[新規作成] をクリックし、[プロジェクト] をクリ ックします。[新規プロジェクト] ダイアログが表示されます。
- [新規プロジェクト] ダイアログで [プロジェクトタイプ] のツリーで [Visual C#] を展開し、
   [Web] をクリックします。
- 3. [新規プロジェクト] ダイアログで [**テンプレート**] のリストで [ASP.NET Web アプリケーション] をクリックします。
- 4. [新規プロジェクト] ダイアログで [プロジェクト名] フィールドでプロジェクト名を入力します。これは 通常はたとえば MyWebSite などプロジェクトの名前です。 Visual Studio はこの名前をデフォル トの .NET 名前空間とアセンブリの名前として使います。
- 5. 新規プロジェクトのダイアログで [場所] フィールドで Sitecore ソリューションのドキュメント ルートを 入力します。たとえば C:¥inetpub¥sitecore¥MyWebSite¥WebSite などです。
  - 6. [ソリューションのディレクトリを作成] のチェックをオフにします。

7. [場所] のデフォルト値を承諾し、[OK] をクリックします。 Visual Studio は [場所] で 指定されたディレクトリのサブディレクトリにプロジェクトを作成し、そのプロジェクトを開きます。

Visual Studio プロジェクトを移動する方法:

- 1. Visual Studio を閉じます。
- 2. Windows のファイルシステムのエクスプローラーを使ってプロジェクトを含むディレクトリに移動しま す。たとえば C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite です。
- Properties フォルダーと .csproj ファイルと .csproj.user ファイルを Sitecore ソリューションのドキ ュメントルートに移動します。たとえば、
   C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite¥Properties と
   C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite¥MyWebSite.csproj と
   C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite¥MyWebSite.csproj.user を
   C:¥inetpub¥sitecore¥MyWebSite¥WebSite に移動します。



メモ

Visual Studio プロジェクトを作成したときに [**ソリューションのディレクトリを作成**] のチェックを オフにしなかった場合には、Properties ディレクトリと .csproj と .user ファイルを C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite¥MyWebSite ディレクトリから C:¥inetpub¥sitecore¥MyWebSite¥WebSite ディレクトリに移動します。

Visual Studio が作成したその他のファイル システム リソースを削除します。たとえば、
 C:¥inetpub¥sitecore¥MyWebSite¥WebSite¥MyWebSite フォルダー全体を削除します。

Visual Studio のスタート ページから移動したプロジェクトを削除する方法:

- Visual Studio で [表示] メニューをクリックし、 [その他のウィンドウ] をクリックし、 [スタート ページ] をクリックします。 [最近使ったプロジェクト] のリストが表示されます。
- 2. Visual Studio で [最近使ったプロジェクト] のリストでプロジェクト名をクリックします。Visual Studio は最近使ったプロジェクトのリストからエントリーを削除するかどうかたずねます。

Visual Studio Web アプリケーション プロジェクトを開く方法:

- Visual Studio で [ファイル] メニューをクリックし、[開く] をクリックし、[プロジェクト/ソリューション] をクリックします。ファイルを選択するダイアログが表示されます。
- [ファイルの選択] ダイアログで Visual Studio のプロジェクトを含むディレクトリに移動します。たと えば C:¥inetpub¥sitecore¥MyWebSite¥WebSite です。
- 3. .csproj ファイルをクリックし、次に [開く] をクリックします。たとえば、MyWebSite.csproj をクリックし、次に [開く] をクリックします。

Web アプリケーション プロジェクトの Visual Studio ソリューションを作成するには、Visual Studio を閉じます。Visual Studio がソリューション ファイルに変更をセーブするかどうかたずねます。たとえば MyWebSite.sln です。

Visual Studio Web アプリケーション プロジェクトを構成する方法::

 Visual Studio で、Web アプリケーション プロジェクトを開き、Visual Studio の**ソリューション** エクスプローラーを表示します。

2. Visual Studio のソリューション エクスプローラーで default.aspx を右クリックし、[プロ ジェクトから除外] をクリックします。このファイルは Sitecore の一部ですが、Visual Studio ソリ ューションの一部ではありません。

3. Visual Studio のソリューション エクスプローラーで [参照設定] を右クリックし、[参照の追加] をクリックします。[参照の追加] ダイアログが表示されます。



4. [参照の追加] ダイアログで [参照] タブをクリックします。ファイルを選択するダイアログが 表示されます。

5. [ファイルの選択] ダイアログで Sitecore ソリューションのドキュメントルートの /bin フォルダーに移動します。たとえば C:¥inetpub¥siotecore¥MyWebSite¥WebSite¥bin です。

6. [ファイルの選択] ダイアログで Sitecore.Kernel.dll をクリックし、[OK]をクリックします。

7. Visual Studio の**ソリューション エクスプローラー**で [参照設定] を展開し、 Sitecore.Kernel を右クリックし、[**プロパティ**] をクリックします。Visual Studio の [プロパティ] パレットが表示されます。

8. Visual Studio の [プロパティ] パレットで、Sitecore.Kernel アセンブリの参照の [ロー カルコピー] プロパティを False に設定します。

### 注意

Sitecore ソリューションのドキュメント ルートの /bin フォルダーのアセンブリへの参照の [**ローカル コピー**] プロパディが **False** でない場合には、Visual Studio が /bin フォルダーのアセンブリを削 除することができ、その場合には Sitecore ソリューションは正しく稼働しません。 Web サイトのドキ ュメントルートの /bin フォルダーのアセンブリへの各参照の [**ローカルコピー**] プロパディを False に 設定します。

### 重要

追加のアセンブリの参照は必要な場合のみ追加します。

Visual Studio の Web アプリケーション プロジェクトのアセンブリの名前とデフォルトの名前空間 を構成する方法:

- 1. Visual Studio で Web アプリケーション プロジェクトを開きます。
- 2. Visual Studio で [プロジェクト] メニューをクリックし、プロジェクトの [プロパティ] オプションをクリ ックします。たとえば、[プロジェクト] メニューをクリックし、[MyWebSite のプロパティ] をクリック します。 Visual Studio のプロジェクト プロパティ エディターが表示されます。
- 3. プロジェクト プロパティ エディターで [**アプリケーション**] タブをクリックします。
- 4. プロジェクト プロパティ エディターで、 [**アセンブリ名**] にプロジェクトで生成するアセンブリの名前 を.dll 拡張子なしで入力します。 たとえば、 MyWebSite と入力します。
- 5. プロジェクト プロパティ エディターで [既定の名前空間] にプロジェクトのデフォルトの名前空間を入 力します。たとえば、MyWebSite と入力します。



### 2.4.4 既存のファイルを Web アプリケーション プロジェクトに追加する方法

既存のレイアウト、サブレイアウト、XSL レンダリングのファイルを Visual Studio の Web アプリ ケーション プロジェクトに追加する方法:

- 1. Visual Studio で、Web アプリケーション プロジェクトを開き、Visual Studio の**ソリューション エクスプローラー**を表示します。
  - 2. Visual Studio のソリューション エクスプローラーですべてのファイルを表示します。

3. Visual Studio の**ソリューション エクスプローラー**でファイルを含んでいるディレクトリを展開します。

4. Visual Studio の**ソリューション エクスプローラー**でファイルを右クリックし、[プロジェク トに含める] をクリックします。

### 重要

Visual Studio の**ソリューション エクスプローラー**ではすべてのファイルを非表示にしてください。 そうしない場合、Visual Studio のデバッガーが反応しなくなることがあります。

### 2.4.5 Sitecore のコントロールを Visual Studio のツールボックスに追加する方 法

Sitecore のコントロールを Visual Studio のツールボックスに追加することができます。

### 重要

コントロールをツールボックスに追加するバージョン以外の Sitecore を使っている場合、 Sitecore のコントロールをツールボックスからレイアウトまたはサブディレクトリにドラッグすると、 Visual Studio はコントロールをツールボックスに追加するバージョンで /bin フォルダーの各種のア センブリを上書きすることがあります。コントロールをツールボックスに追加するバージョン以外の Sitecore を使っている場合、コントロールをツールボックスから削除し、それを再度追加します。

### メモ

アセンブリを Visual Studio の **ツールボックス** に追加したとき、Visual Studio はデフォルトで アセンブリのすべてのコントロールを追加します。多くの開発者はこれらのコントロールの多くは使いま せん。使用しないコントロールを Visual Studio の [**ツールボックス**] から削除することは Visual Studio を高速化させ、開発者がコントロールを見つけることも容易になります。

Sitecore のコントロールを Visual Studio のツールボックスに追加する方法:

1. Visual Studio ですべての開いているソリューションまたはプロジェクトを閉じます。



- 2. Visual Studio で [ツール] メニューをクリックし、次に [ツールボックスアイテムの選択] をクリック します。コントロールを選択するダイアログが表示されます。
- 3. コントロールを選択するダイアログで [参照] をクリックします。ファイルを選択するダイアログが表示されます。
- 4. [ファイルの選択] ダイアログで Web サイトのドキュメントルートの /bin フォルダーに行き、 Sitecore.Kernel.dll アセンブリをクリックします。

5. [ファイルの選択] ダイアログで [**開く**] をクリックします。[コントロールの選択] ダイアログで はアセンブリのすべてのコントロールが選択されてハイライトされています。

6. [コントロールの選択] ダイアログで、アセンブリのすべてのコントロールが選択されてハイライトされたまま、すべてのコントロールのチェック ボックスをオフにし、アセンブリのすべてのコントロールのチェック ボックスをオフにします。

7. [コントロールの選択] ダイアログで [名前空間] でソートします。

8. [コントロールの選択] ダイアログで名前空間 Sitecore.Web.UI.WebControls までスク ロールします。

9. [コントロールの選択] ダイアログで各 Sitecore コントロールのチェック ボックスをオンにし、 ツールボックスに追加します。通常は FieldRenderer (FieldRenderer Web コントロール)、 Method (method レンダリング Web コントロール)、Sublayout、WebPage (URL レンダリング Web コントロール)、XslFile (XSL レンダリング Web コントロール) です。

Sitecore コントロールを Visual Studio [ツールボックス] の 1 つのタブでグループ化する方法:

- 1. Visual Studio で Visual Studio の**ツールボックス**を表示します。
- 2. Visual Studio で Visual Studio [ツールボックス] を右クリックし、[タブの追加] をクリックしま す。タブの名前として [Sitecore] と入力します。
- Visual Studio の [ツールボックス] で [全般] タブから Sitecore コントロールを [Sitecore] タブにドラッグします。

Visual Studio ツールボックスから Sitecore コントロールを削除する方法:

- 1. Visual Studio で Visual Studio の [ツールボックス] を表示します。
- 2. Visual Studio で Visual Studio の [ツールボックス] を右クリックし、[アイテムの選択] をクリ ックします。コントロールを選択するダイアログが表示されます。
- 3. コントロールの選択ダイアログで [名前空間] でソートします。



- 4. コントロールの選択ダイアログで名前空間 Sitecore.Web.UI.WebControls までスクロールします。
- コントロールの選択ダイアログで各 Sitecore コントロールのチェック ボックスをオフにします。
   または -
- Visual Studio ですべてのプロジェクトまたはソリューションを閉じ、Visual Studio の [ツールボックス] を表示します。

2. Visual Studio で Visual Studio の [ツールボックス] を右クリックし、[ツールボックス のリセット] をクリックします。

### 2.4.6 Visual Studio を使って .NET コードをデバッグする方法

Sitecore ログファイル、管理ページ、Sitecore ブラウザ ベースのデバッガーを使って、ロジカル エ ラー、パフォーマンス上のボトルネック、その他の望ましくないコードの状態を含むコンポーネントを認 識します。それから Visual Studio のデバッガーを使って .NET のコードをデバッグします。

Visual Studio を使って .NET コードをデバッグする方法:

1. ASP.NET ワーカー プロセスがアクティブであることを確認するため、Web クライアントを使って Sitecore ソリューションから ASP.NET リソース (たとえばホームページ) を要求します。

2. Visual Studio で Visual Studio の**ソリューション エクスプローラー**を表示し、すべてのファイルを非表示にします。

3. Visual Studio で対象の .NET コード ファイルの適当な行に移動します。

4. Visual Studio でブレークポイントを作成するためには [デバッグ] メニューをクリックし、 [ブレークポイントの設定/解除] をクリックするか、または[F9] を押します。

5. Visual Studio で [デバッグ] メニューをクリックし、次に [プロセスにアタッチ] をクリックし ます (または CTRL-ALT-P)を押す)。[プロセスにアタッチ] ダイアログが表示されます。

6. [プロセスにアタッチ] ダイアログで [すべてのユーザーからのプロセスを表示する] と [す べてのセッションのプロセスを表示する] のチェック ボックスをオンにします。

[プロセスにアタッチ] ダイアログで IIS 5 (Windows XP) の場合は
 [aspnet\_wp.exe] プロセスを、IIS 6 または 7 (Windows 2003、Windows Vista、
 Windows 2008) の場合は [w3wp.exe] プロセスをクリックし、 [アタッチ] をクリックします。

8. Microsoft Internet Explorer などの Web クライアントを使ってデバッグするコードを 使う ASP.NET リソースを要求します。



9. デバッグを停止する方法:

• Microsoft Visual Studio で [デバッグ] メニューをクリックし、[デバッグの停止] をクリ ックします。

### 重要

Visual Studio の [デバッグ] メニューをクリックしたあとに [デバッグ開始] をクリックしないでくだ さい。そうすると IIS でなく、Microsoft Visual Studio がインストールした カッシーニ Web サー バーが開始されます。

### 2.4.7 Web サービス メソッドのコレクションの作成方法

Sitecore にアクセスするリソースへのすべての要求はロジカルなサイト定義 (web.config の /configuration/sitecore/sites/site) を含む Sitecore コンテキストで稼働し、構成などのリソー スへのアクセスを提供します。ファイルシステムでの Web サービス ファイル (.asmx) の場所がコン テキスト サイトを決定し、したがって Web サービス要求が実行されるコンテキストを決定します。

デフォルトの Sitecore の Web サービス ライブラリの URL は

/sitecore/shell/webservice/service.asmx です。これらの Web サービス メソッドは web.config で name が shell である /configuration/sitecore/sites/site 要素のサイトのコ ンテキストで稼働します。この要求コンテキストは Sitecore.Context.Database を Core データベ ースに、Sitecore.Context.ContentDatabase を Master データベースに設定します。デフォルト の Web サービス メソッドの多くは、構成のみに依存するのでなく、サービスがアクセスするデータベ ースの名前を示すパラメーターを受け入れます。

/sitecore modules/web ディレクトリの ASP.NET リソースの要求は modules\_website という名前の論理サイトのコンテキストで稼働します。これはデフォルトで

Sitecore.Context.Database を Web データベースに設定し、

Sitecore.Context.ContentDatabase を null のままにします。コンテキスト データベースを Web サービスに必須とするディレクトリに Web サービス ファイルを配置するか、

Sitecore.Context.Database を使わずにデータベース名を Web サービスにパラメーターとして渡します。

Web サービス メソッドのコレクションを Visual Studio の ASP.NET Web アプリケーション プロジェクトに追加する方法:

1. Visual Studio の Web アプリケーション プロジェクトの Visual Studio の**ソリューション エクス** プローラーで、フォルダーを作成するか、フォルダーをプロジェクトに追加するか、プロジェクトにすでに 存在しているフォルダーを見つけます。



- Visual Studio の ソリューション エクスプローラーで前のステップで言及したフォルダーを右クリックし、[追加]をクリックし、[新しい項目]をクリックします。[新しい項目の追加]ダイアログが表示されます。
- 3. [新しい項目の追加] ダイアログで [テンプレート] リストで [Web サービス] をクリックします。
- 4. [新しい項目の追加] ダイアログの [名前] で Web サービス メソッドを含めるファイルの名前を入 力し [追加] をクリックします。 Visual Studio は Web サービス ファイルを作成します。
- 5. Microsoft Visual Studio の Web サービス ファイルで Web サービス メソッドを作成します。

### 2.4.8 Visual Studio のパフォーマンスを最適化する方法

このセクションでは Visual Studio のパフォーマンスを改善するためのヒントを解説します。

Visual Studio を閉じる前に、次回アプリケーションを開始するときに開く必要のないユーザー インターフェース コンポーネントを閉じます。

Visual Studio の [オプション] ダイアログにアクセスする方法:

1. Visual Studio で [**ツール**] メニューをクリックし、[オプション] をクリックします。

Visual Studio の RSS フィードを無効にする方法:

1. Visual Studio の [オプション] ダイアログで [環境] を展開し、[スタートアップ] をクリックしま す。

2. Visual Studio の [オプション] ダイアログで [ニュース チャンネルのスタート ページ] を クリアします。

3. Visual Studio の [オプション] ダイアログで [**コンテンツをダウンロードする間隔**] のチェ ックをオフにします。

スタート ベージを無効にする方法:

- 1. Visual Studio の [オプション] ダイアログで [環境] を展開し、[スタートアップ] をクリックしま す。
- Visual Studio の [オプション] ダイアログで [スタートアップ時] のドロップダウンで [空の環境の 表示] を選択します。

۶Ł

Visual Studio のスタート ページを表示するには、Visual Studio で [表示] メニューをクリック し、 [その他のウィンドウ] をクリックし、 [スタート ページ] をクリックします。

Visual Studio のスプラッシュ画面を無効にする方法:



- 1. Windows デスクトップで Visual Studio を起動するショートカットを右クリックし、[プロパティ] をクリックします。ショートカットの [プロパティ] ダイアログが表示されます。
- 2. ショートカットの [プロパティ] ダイアログで [**ショートカット**] タブをクリックします。
- 3. ショートカットの [プロパティ] ダイアログで、[ショートカット] タブで /nosplash コマンドライン オプショ ンを [**ターゲット**] に追加します。たとえば:

"C: ¥Program Files ¥Microsoft Visual Studio 9.0 ¥Common7 ¥IDE ¥devenv.exe" /nosplash.

アニメーションを無効にする方法:

1. Visual Studio の [オプション] ダイアログで [環境] をクリックします。

2. Visual Studio の [オプション] ダイアログで [リッチクライアントの視覚的効果を有効 にする] のチェックをオフにします。

変更の履歴を無効にする方法:

- 1. Visual Studio の [オプション] ダイアログで [テキスト エディター] をクリックします。
- Visual Studio の [オプション] ダイアログで [変更履歴を記録する] のチェックをオフにします。
   アクティブな項目のトラッキングをオフにする方法:
- Visual Studio の [オプション] ダイアログで [プロジェクトおよびソリューション] をクリックします。
- Visual Studio の [オプション] ダイアログで [アクティブな項目をソリューション エクスプローラ ーで選択された状態にする] のチェックをオフにします。

AutoToolboxPopulate をオフにする方法:

- Visual Studio の [オプション] ダイアログで [Windows フォーム デザイナー] をクリックします。
- Visual Studio の [オプション] ダイアログで [ツールボックス環境] の下で、
   [AutoToolboxPopulate] を False に設定します。

Visual Studio がレイアウトとサブレイアウトをデフォルトでデザイン ビューでなくソース コード ビューで開くようにする方法:

- 1. Visual Studio の [オプション] ダイアログで [HTML デザイナー] をクリックします。
- 2. Visual Studio の [オプション] ダイアログで [ページを開始するビュー] で [ソース ビュー] を選択します。



もし Jetbrains ReSharper を使用している場合には、CTRL-F12を押すと編集ウィンドウの 上にメソッドとフィールドの更新リストが表示されます。<sup>7</sup> Visual Studio のナビゲーション バーが 提供する類似の機能を無効にするには、または Visual Studio のナビゲーション バーが不要な 場合方法:

- 1. Visual Studio の [オプション] ダイアログで [テキスト エディター] を展開し、[C#] をクリックし ます。
- 2. Visual Studio の [オプション] ダイアログで [ナビゲーション バー] のチェックをオフにします。

<sup>7</sup> JetBrains ReSharper に関する詳細は <u>http://www.jetbrains.com/resharper</u>を参照してください。

Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。 この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



# **Chapter 3**

# レイアウト詳細

この章ではレイアウト詳細を使う方法について解説します。

この章には次のセクションがあります。

- レイアウト詳細の使い方
- レイアウト詳細をスタンダードバリューにリセットする方法
- レイアウト詳細のコピー方法
- 使用されるプレゼンテーション コンポーネントを判定方法
- デバイスを使う
- レイアウト プリセット

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



### 3.1 レイアウト詳細の使い方

このセクションではレイアウト詳細を使う方法について解説します。

### አደ

ページ エディターを使用してレイアウト詳細を編集することも可能です。

### 重要

Sitecore は他のデータ テンプレートが継承する標準テンプレートで定義されたフィールドにレイア ウト詳細を保存します。すべてのフィールド値と同様に、個々のアイテムで定義されたレイアウト詳 細はそのアイテムに関連付けられたデータ テンプレートのスタンダード バリューで定義されたレイアウ ト詳細を上書きします。データの重複と管理を低減するためには、レイアウト詳細を個々のアイテム でなくスタンダード バリューに関連付けします。共通のデータ構造に基づく別のアイテムが別のレイア ウト詳細を必要とする場合には、既存のデータ テンプレートから継承する新しいデータ テンプレート のスタンダード バリューを使ってレイアウト詳細を定義することをぜひ検討してください。

### 重要

以前のバージョンとの後方互換性のため、アイテムまたはそのデータ テンプレートに関連付けられ たスタンダード バリュー アイテムにレイアウト詳細がない場合、Sitecore はデータ テンプレート定 義アイテムで定義されたレイアウト詳細を適用します。レイアウト詳細を、データ テンプレート定義ア イテム自体ではなく、データ テンプレート定義アイテムが付けられたスタンダード バリュー アイテムに 定義します。

### 重要

必ずデフォルトのデバイス用のレイアウト詳細を定義します。それを Sitecore は代替デバイスを 指定していないすべての受信 HTTP 要求に対してデフォルトで起動します。

### ۶Ł

コントロールという語はサブレイアウトとすべてのタイプのレンダリングを含みます。

### 3.1.1 デバイス エディター

デバイス エディターを使うとレイアウト詳細を入力するときにデバイスを選択することができます。

### デバイス エディターへのアクセス方法

デバイス エディターにアクセスする方法:

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 バリュー アイテムまたは個々 の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 [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] タブの、 [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、レイアウト詳細を構成したいデバイスの下で [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。

### レイアウトの選択方法

デバイスに使用するレイアウトを選択する方法:

- 1. デバイス エディターで、 [レイアウト] タブをクリックします。 デバイス エディターにアクセスする方法 は、「デバイス エディターへのアクセス方法」のセクションを参照してください。
- 2. デバイス エディターで、[レイアウト] タブを選択し、レイアウトのドロップダウンからレイアウトを選択 します。

### コントロールの追加方法

レイアウト詳細でコントロールを追加する方法:

- 1. デバイス エディターで [コントロール] タブをクリックし、次に [追加] をクリックします。 [レンダリング の選択] ダイアログが表示されます。 デバイス エディターにアクセスする方法は、「デバイス エディタ ーへのアクセス方法」のセクションを参照してください。
- 2. [レンダリングの選択] ダイアログでレンダリングを選択します。
- レンダリングをレイアウト詳細に追加した後でレンダリングのプロパティを開くには、[レンダリングの選 択] ダイアログで [このダイアログを閉じた後、プロパティを開く] を選択します。レンダリングのプロ パティに関する詳細は、「コントロール」のセクションを参照してください。
- 4. [レンダリングの選択] ダイアログで [選択] をクリックします。

### コントロールの順序の変更方法

レイアウト詳細でコントロールの順序を変更する方法:

- 1. デバイス エディターで [コントロール] タブをクリックし、次に [コントロール] をクリックして選択しま す。デバイス エディターにアクセスする方法は、「デバイス エディターへのアクセス方法」のセクション を参照してください。
- デバイス エディターの [コントロール] タブでコントロールを選択し、 [上へ移動] または [下へ移動] をクリックし、コントロールの順序を変更します。



### 重要

レイアウト詳細でのプレゼンテーション コンポーネントの順序は、レイアウト エンジンが作成する順 序とそれらをコントロールの階層構造にバインドする順序を制御します。プレースホルダーを含むサブ レイアウトを、それらのプレースホルダーにバインドするコントロールよりも前に配置します。

### メモ

複数のプレゼンテーション コンポーネントを 1 つのプレースホルダーにバインドする場合、レイアウト 詳細における順序が出力されるマークアップの順序を制御します。

コントロールの削除方法

レイアウト詳細からコントロールを削除する方法:

- 1. デバイス エディターで [コントロール] タブをクリックし、次に [コントロール] をクリックして選択しま す。デバイス エディターにアクセスする方法は、「デバイス エディターへのアクセス方法」のセクション を参照してください。
- 2. デバイス エディターで、[コントロール] タブでコントロールを選択したまま [削除] をクリックしま す。Sitecore はそのレンダリングをレイアウト詳細から削除します。

### コントロールの置換方法

既存のレンダリング プロパティに関連付けられたレンダリングをそのプロパティを変更することなく置 換する方法:

- 1. デバイス エディターで [コントロール] タブをクリックし、次に [コントロール] をクリックして選択しま す。デバイス エディターにアクセスする方法は、「デバイス エディターへのアクセス方法」のセクション を参照してください。
- 2. デバイス エディターで、[コントロール] タブでコントロールを選択したまま [変更] をクリックしま す。[レンダリングの選択] ダイアログが表示されます。
- 3. [**レンダリングの選択**] ダイアログで [レンダリング] をクリックして選択し、次に [**選択**] をクリックしま す。Sitecore はレイアウト詳細のそのレンダリングを置換します。



### 3.2 レイアウト詳細をスタンダード バリューにリセットする方法

アイテムのレイアウト詳細をアイテムに関連付けられたデータ テンプレートのスタンダード バリュー ア イテムで定義されたものにリセットする方法:

- 1. **コンテンツ エディター**で、レイアウト詳細をアイテムに関連付けられたデータ テンプレートのスタンダ ード バリュー アイテムで定義されたものにリセットするアイテムを選択します。
- 2. **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. **コンテンツ エディター**で、[プレゼンテーション] タブの [レイアウト] グループで [リセット] コマンド をクリックします。Sitecore は確認画面を表示したあと、選択したアイテムのレイアウト詳細フィール ドをそのスタンダード バリューにリセットします。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



### 3.3 レイアウト詳細のコピー方法

レイアウト詳細を1つのアイテムから別のアイテムにコピーする方法:

- 1. コンテンツ エディターで、レイアウト詳細のコピー元のソースアイテムを選択します。
- 2. **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. **コンテンツ エディター**で、[プレゼンテーション] タブの [詳細] グループで [レイアウト] 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] ダイアログで任意のデバイスの下で [指定の場所にコピー] リンクをクリックしま す。[コピー デバイス] ダイアログが表示されます。
- 5. [**コピー デバイス**] ダイアログの [**ターゲット デバイス**] で、レイアウト詳細のコピー先のデバイスを 選択します。

6. [**コピー デバイス**] ダイアログの [**ターゲット アイテム**] で、レイアウト詳細をコピーする先の アイテムをクリックし、[**コピー**] をクリックします。 Sitecore は選択したデバイスのレイアウト詳細をコ ピー元のアイテムからコピー先のアイテムへコピーします。

レイアウト詳細を1 つのソース アイテムから複数の対象アイテムにコピーする方法:

- 1. コンテンツ エディターでソース アイテムを選択します。
- 2. コンテンツ エディターで標準テンプレート フィールドと RAW 値を表示します。<sup>8</sup>
- 3. コンテンツ エディターでソース アイテムを選択したまま、[レイアウト] セクションで [レンダリング] フ ィールドの値をトリプルクリックして値を選択し、その値を Windows のクリップボードにコピーします。
- 4. コンテンツ エディターでそれぞれの対象アイテムについて、[レイアウト] セクションで [レンダリング] フィールドの値をトリプルクリックして値を選択し、その値を Windows のクリップボードの中身で上 書きします。
- 5. コンテンツ エディターで RAW フィールド値と標準テンプレート フィールドを非表示にします。

<sup>&</sup>lt;sup>®</sup>標準テンプレート フィールドと RAW 値の表示と非表示を切り替える方法については、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Client%20Configuration%2</u> <u>0Cookbook.aspx</u> から『クライアント構成クックブック』を参照してください。



### 3.4 使用されるプレゼンテーション コンポーネントを判定方法

Sitecore デバッガーを使って HTTP 要求に応えるプレゼンテーション コンポーネントを判定する ことができます。

### 重要

Sitecore のデバッガーはデフォルトでは Web データベースにアクセスします。デバッグを行う前 に、レイアウト詳細の変更をパブリッシュします。

ページをレンダリングするために使用するプレゼンテーション コンポーネントを判定するため方法:

- 1. Sitecore デバッガーで、リボンを表示します。
- 2. Sitecore デバッガーの [レンダリング] グループで [レンダリング] を選択します。
- 3. Sitecore デバッガーの [**トレース**] グループで、[アクティベート] コマンドをクリックします。
- 4. Sitecore デバッガーで緑の三角の [情報] アイコンの上でマウスをホバーし、ページで使用されて いる各種のコントロールの詳細を調査します。
- 5. Sitecore デバッガーで [Sitecore トレース] セクションまでスクロールし、無効なレイアウト詳細 から生じているエラーを探します。たとえば、ID の参照に対応するレンダリング定義アイテムがない、 指定されたプレースホルダー キーに合致するプレースホルダーがない、などです。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



### 3.5 デバイスを使う

このセクションでは Sitecore でデバイスを使う方法について解説します。

### 3.5.1 デバイスの作成方法

デバイスを作成する方法:

- 1. コンテンツ エディターで、/Sitecore/Layout/Devices アイテムを選択します。
- コンテンツ エディターで、/Sitecore/Layout/Devices アイテムを選択したまま、
   /System/Layout/Device データ テンプレートを使ってデバイス定義アイテムを挿入します。

### 重要

アイテム ID は Sitecore CMS 内で使用されるため、/Sitecore/Layout/Devices/Default device アイテムは削除しないでください。

- 3. 新規のデバイスがフォールバック デバイスである場合には、コンテンツ エディターの、デバイス定義ア イテムの [データ] セクションの [フォールバック デバイス] フィールドで、フォールバック デバイスを選 択します。
- 4. デバイスのアクティベーションの基準は、「デバイスのアクティベーションの基準の定義方法」のセクションに記載されているように定義します。

### 重要

デバイス アイテムのデータ セクションでは、デフォルトのチェック ボックスをオンにしないでください。

### 3.5.2 デバイスのアクティベーションの基準の定義方法

デバイスを作成した後に、Sitecore が受信する HTTP 要求に対してデバイスをアクティベートす るための基準を定義します。Sitecore は 特定のクエリ文字列パラメーターを含む HTTP 要求, または特定の Web クライアント ユーザー エージェント文字列に合致する HTTP 要求に対してに 対して、デバイスをアクティベートすることができます。各サイトで別のデバイスを使用するように構成 することができます。デバイスをアクティベートするために httpRequestBegin 要求処理パイプライン を実装することができます。

デバイスをアクティベートするクエリ文字列パラメーターを構成するには、コンテンツ エディターのデ バイス定義アイテムで、 [判定] セクションの [クエリ文字列] にクエリ文字列 key=value を入力 します。たとえば、URL が値 1 をもつクエリ文字列パラメーター x を含む場合にデバイスをアクティ ベートするには、x=1 と入力します。


デバイスをアクティベートするために特定の Web クライアント ユーザー エージェントを構成するに は、**コンテンツ エディター**の、デバイス定義アイテムで、[**判定**] セクションの、[**ブラウザ エージェン** ト] に、合致するユーザー エージェントを入力します。たとえば、すべての Blackberry クライアント にデバイスをアクティベートするためには、blackberry と入力します。比較は大文字と小文字を区 別しません。

特定の管理対象 Web サイトに関連付けられた HTTP 要求に対してデバイスをアクティベート するようにレイアウトエンジンを構成するには、 web.config の /configuration/sitecore/sites/site 要素に device 属性を設定し、そのサイトにデバイスの名前 を定義します。<sup>9</sup>

デバイスをアクティベートするためにその他の基準を使うためには、httpRequestBegin 要求処理 パイプラインを実装し、 Sitecore.Pipelines.HttpRequest.DeviceResolver を置換または追跡 し、Sitecore.Context.Device を設定します。<sup>10</sup>

<sup>10</sup>.NET API を使ったデバイスのアクティベートに関する詳細については、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20API%20Cookbook.aspx</u>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>&</sup>lt;sup>9</sup>複数の論理サイトの構成に関する詳細は、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Articles/Administration/Configuring%20Multiple%20Sites.aspx</u>を参照してください。



# **3.6** レイアウト プリセット

このセクションではレイアウトプリセットを使用する手順について説明します。

#### 重要

レイアウト プリセットに代わる方法としては、追加のデータ テンプレートの作製、コマンド テンプレートの作製、その他の方法があります。レイアウト プリセットは多くのアイテムのレイアウト詳細を複製します。

## 3.6.1 レイアウト プリセットの作成方法

レイアウト プリセット定義アイテムを作製する場合には、適切な名前をつけます。その名前はレイ アウト プリセットをアイテムに適用する際にリボンのプリセット グループに表示されます。

既存のソース アイテムからレイアウト詳細をコピーしてレイアウト プリセットを作成する手順:

- 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 スタンダード バリュー アイテムまたは 個々のアイテムを編集します。
  - 2. アイテムのレイアウト詳細を構成してテストします。

#### አደ

コンテンツ アイテムのレイアウトを設定する場合は、アイテムに定義済みのプレゼンテーションを持 つスタンダード バリューがないことを確認します。

3. 「スタンダード フィールドを表示または非表示にする」のセクションに記載されている方法に したがって、スタンダード フィールドを表示します。

4. 「RAW 値を表示または非表示にする」のセクションに記載されている方法にしたがって、 RAW 値を表示します。

5. [**レイアウト**] セクションの [**レンダリング**] 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。

Select /Sitecore/System/Settings/Layouts/Presets

6. /System/Layout/Layout Preset データ テンプレートを使用してレイアウト プリセット定 義アイテムを挿入します。

7. [データ] セクションの [レイアウト] フィールドで値を Windows クリップボードから貼り付けます。

8. スタンダード バリューと RAW 値を非表示にします。



# 3.6.2 レイアウト プリセットをデータ テンプレートまたはアイテムと関連付ける方法

デフォルトでは、レイアウト 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 Designing、Sitecore Client Developing、Sitecore Client Maintaining ロールのいず れかのメンバーである必要があります。

プリセット フィールドを使用してレイアウト プリセットを選択します。 レイアウト プリセットはデータ テンプレートまたはコンテンツ アイテムと関連付けることが可能です。

レイアウト プリセットをデータ テンプレートまたはアイテムと関連付ける手順:

- 1. コンテンツ エディターでレイアウト プリセットを関連付けるアイテムを選択します。
- 2. 選択したアイテムで [プリセット] フィールドまでスクロールします。

#### УŦ

アイテムでレイアウト グループとプリセット フィールドを表示するためにはスタンダード フィールドの表示を有効化する必要があります。

3. [プリセット] フィールドでレイアウト プリセットを選択し、矢印キーを使用して [選択済み] フィールドに移動します。変更を保存します。



## 3.6.3 ページ エディターを使用してレイアウト プリセットを適用する

レイアウト プリセットとアイテムまたはテンプレートの関連付けを完了すると、ページ エディターを使 用してレイアウト プリセットを現在のアイテムに適用することが可能となります。

レイアウト プリセットをコンテンツ アイテムに適用する手順:



- 1. ページ エディターの [ホーム] タブで [プリセット] をクリックします。アイテムと関連付けたプリセット レイアウトは [レイアウトプリセットを選択してください] ダイアログ ボックスに表示されます。
- 2. [**レイアウト プリセットを選択してください**] ダイアログ ボックスでこのアイテムで利用可能なすべて のレイアウト プリセットの画像のサムネイルが表示されます。

Preview - Component Pr	resets Insert Lock Workbox	Publish R Log Off		
de New	Edit	Publish Close		
Partner-Services	Go 🛐 Edit			
		P Sitecore Webnage Dialog	1	
	Office Care	Select a Layout Pre	acet	
	interior design for business spares	Select the layout preset	that will determine the design of the page.	
				Ed
		Office Core		
	Home Our Process Our	Terrer Dar Practice Dar Services Part	Office Co	atio
		Office design services for effect	interior design for busines:	
		Office Care designs and construits physical business productivity, and actively contribute to a positive con-		
		Our Clemes Inspirations Snow our founding back in 1995, Every office is	Home Our Proce	
		Office Core has provided design - reflects the pr existence for even 500 companies - who work the in 25 countries		
		My Custom Layout	Sample Layout	
	Home > Partner Services			
	Partner Services		Select Cance	
		Provide your main text here		

- 3. レイアウト詳細を選択し [選択] をクリックしてください。
- 4. 変更を保存します。

### メモ

サムネール作成ツールを使用してレイアウトプリセットのための画像を作成することも可能です。



# **Chapter 4**

# コントロール

この章ではコントロールを使う方法について解説します。この文書ではコントロールという用語 は、プレースホルダー、サブレイアウト、XSL レンダリング、Web コントロール、URL レンダリン グ、メソッド レンダリング、FieldRenderer Web コントロールを含みます。

この章には次のセクションがあります。

- コントロールの出力の表示方法
- レンダリング定義アイテム
- コントロール プロパティのアクセス方法
- Visual Studio を使ったコントロール プロパティの設定方法
- 共通のコントロール プロパティ
- プレースホルダー
- サブレイアウト
- FieldRenderer Web コントロール
- XSL レンダリング
- Web コントロール
- メソッド レンダリング
- URL レンダリング

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



# 4.1 コントロールの出力の表示方法

Sitecore デバッガーを使って1つのコントロールの出力を表示する方法:

- 1. Sitecore デバッガーで、リボンを表示します。
- 2. Sitecore デバッガーの [**レンダリング**] グループで [**レンダリング**] を選択します。
- 3. Sitecore デバッガーでレンダリングを表している緑の三角の [情報] アイコンの上にマウスをホバー します。 レンダリング情報パネルが表示されます。
- 4. レンダリング情報パネルで、[**出力**] タブをクリックします。レンダリングの出力がレンダリング情報パネ ルに表示されます。



# 4.2 レンダリング定義アイテム

デベロッパー センター、デバイス エディター、ページ エディターを含む Sitecore のユーザー インタ ーフェースでレンダリングを使用する前に、レンダリング定義アイテムを挿入する必要があります。各 種のレンダリング定義アイテムを挿入する方法については、「サブレイアウト」、「XSL レンダリング」、 「Web コントロール」、「メソッド レンダリング」、「URL レンダリング」のセクションを参照してください。

#### 重要

各レンダリング定義アイテムの [データ] セクションの [説明] フィールドを使ってレンダリングに関す る情報をユーザーに提供することができます。コンポーネントの出力の説明や出力例などを入力す ることができます。キャッシュをサポートするコンポーネントには、キャッシュ オプションを設定するように 注意を含めます。 URL を指定しない URL レンダリングや、メソッドを指定しないメソッド レンダリン グには、ユーザーにこれらのプロパティを入力するように注意を含めます。 もしくは、 URL またはメソッ ドの名前を入力します。

#### 重要

レンダリング定義アイテムを挿入する前に、コンテンツ エディターと Windows のファイル エクスプ ローラーを使って、Sitecore のフォルダーとファイルシステムのディレクトリを作成し、定義アイテムと 定義アイテムが参照するファイルを含めます。

ヒント

プレゼンテーション コンポーネントを使う際に Visual Studio のみで扱い、Sitecore ユーザー インターフェースでは扱わない場合には、そのコンポーネントを Sitecore で登録する必要はありま せん。



# 4.3 コントロール プロパティのアクセス方法

レイアウトまたはサブレイアウトに静的にバインドされたコントロールのレンダリング プロパティ ダイア ログに**デベロッパー センター**を使ってアクセスする方法:

- 1. デベロッパー センターで、レイアウトまたはサブレイアウトを開きます。
- 2. デベロッパー センターのレイアウトまたはサブレイアウトで、[デザイン] タブをクリックします。
- 3. デベロッパー センターのレイアウトまたはサブレイアウトで、[コントロール] をタブルクリックします。[レ ンダリング プロパティ] ダイアログが表示されます。

レイアウト詳細を使ってプレースホルダーに動的にバインドされたコントロールの [コントロール プロ パティ] ダイアログにアクセスする方法:

1. デバイス エディターで [コントロール] タブをクリックし、次に [コントロール] をクリックして選択しま す。デバイス エディターにアクセスする方法は、「デバイス エディターへのアクセス方法」のセクション を参照してください。

2. **デバイス エディター**で、[**編集**] タブをクリックします。[**コントロール プロパティ**] ダイアログ が表示されます。



# 4.4 Visual Studio を使ったコントロール プロパティの設定方法

レイアウトまたはサブレイアウトに静的にバインドされたコントロールのコントロール プロパティを Visual Studio を使って設定する方法:

- 1. Visual Studio の Web アプリケーション プロジェクトで、レイアウトまたはサブレイアウトを開きま す。
- 2. Visual Studio のレイアウトまたはサブレイアウトで、[コントロール] を右クリックし、[プロパティ] を クリックします。 Visual Studio の [プロパティ] パレットが表示されます。
- 3. Visual Studio の [プロパティ] パレットを使ってプロパティを設定します。または [**ソース**] タブをク リックしてコントロール定義要素の属性と値を入力します。



## 4.5 共通のコントロール プロパティ

このセクションは各種のコントロールに共通なプロパティを説明します。

#### 重要

レイアウトまたはサブレイアウトに静的にバインドした各コントロールについて、ASP.NET コントロー ル識別子として id 属性を定義します。ASP.NET コントロール識別子に関する詳細は 「ASP.NET コントロール識別子 (ID)」のセクションを参照してください。

#### 4.5.1 コントロール キャッシュ オプションの構成方法

このセクションは各コントロールにキャッシュを構成する方法を説明します。

#### 重要

コントロールを使用する際には、コントロールを動的にレイアウトまたはサブレイアウトにバインドする 場合でも、動的にプレースホルダーにバインドする場合でも、必ずキャッシュ オプションを構成しま す。

#### 重要

レンダリング定義アイテムのキャッシュ オプションはレイアウト詳細を使って動的にバインドされたレン ダリングには適用されません。レイアウト詳細の各コントロールにキャッシュ オプションを定義します。

#### 重要

デベロッパー センターを使ってレイアウトまたはサブレイアウトにレンダリングを追加した場合、 Sitecore はキャッシュ オプションを、レンダリング定義アイテムからレイアウトまたはサブレイアウトのコ ントロールのプロパティにコピーします。各レンダリング定義アイテムでキャッシュ オプションを定義しま す。

#### 重要

レンダリング定義アイテムを更新した場合、Sitecore はそのレンダリングを静的にバインドしたレイ アウトまたはサブレイアウトを更新しません。レンダリング定義アイテムでキャッシュオプションを更新し た後で、必ずそのレンダリングを静的にバインドしたレイアウトまたはサブレイアウトのコントロールのキ ャッシュ オプションを更新します。

#### 重要

不要なメモリの消費を避けるため、同じ出力を2回キャッシュすることは避けてください。たとえば、 サブレイアウトの出力をキャッシュした場合、そのサブレイアウトに使われている各レンダリングの出力 をキャッシュする必要はありません。



[コントロール プロパティ] ダイアログを使ってキャッシュ オプションを構成する方法:

• [**コントロール プロパティ**] ダイアログの [**キャッシュ**] セクションで [キャッシュ オプション] を 選択します。

[レンダリング プロパティ] ダイアログを使ってキャッシュ オプションを構成する方法:

 レンダリング プロパティ ダイアログで [キャッシュ] タブをクリックし、[キャッシュ オプション] を 選択します。

Visual Studio を使ってキャッシュ オプションを構成する方法:

- Visual Studio のレイアウトまたはサブレイアウトで、[デザイン] タブをクリックし、次に [コントロール] をダブルクリックし、[プロパティ] をクリックします。Visual Studio の [プロパティ] パレットが表示されます。
- 2. Visual Studio の [プロパティ] パレットで、キャッシュ オプションを構成します。

## 4.5.2 コントロールのデータ ソースの構成方法

[コントロール プロパティ] ダイアログを使ってデータ ソース アイテムを渡すため方法:

• [一般] セクションの [データ ソース] フィールドで、データ ソース アイテムへの完全なパス を入力します。

- または -

• フィールドの上の [リンクを挿入] コマンドをクリックし、次にアイテムを選択します。11

Visual Studio を使ってデータ ソース アイテムをコントロールに渡すため方法:

 Visual Studio の Web アプリケーション プロジェクトで、レイアウトまたはサブレイアウトで、[デザ イン] タブをクリックし、次にコントロールをダブルクリックし、[プロパティ] をクリックします。Visual Studio の [プロパティ] パレットが表示されます。

2. Visual Studio の [プロパティ] パレットの [データ ソース] プロパティで、アイテムへの完 全なパスを入力します。

## 4.5.3 レンダリング パラメーターのコントロールへの渡し方

[コントロール プロパティ] ダイアログを使ってパラメーターをコントロールに渡す方法:

<sup>11</sup> サブレイアウトに渡されたデータ ソースにアクセスするには、

<sup>&</sup>lt;u>http://trac.sitecore.net/SublayoutParameterHelper</u>を参照してください。



• [**コントロール プロパティ**] ダイアログで、コントロール プロパティを入力します。

[レンダリング プロパティ] ダイアログを使ってパラメーターをコントロールに渡す方法:

• [レンダリング プロパティ] ダイアログで [**パラメーター**] タブをクリックし、名前付きパラメータ ーを入力します。<sup>12</sup>

## 重要

FieldRenderer Web コントロールの FieldName プロパティを含め、[レンダリング プロパティ] ダイアログの Web コントロールのプロパティを設定するには、[**属性**] タブでなく、[**パラメーター**] タ ブを使います。Parameters という名前のパラメーターを使って、Web コントロールの Parameters プロパティを、URL エンコードされた一連の key=value の組に設定することができます。

Visual Studio を使ってパラメーターをコントロールに渡すため方法:

- Visual Studio の Web アプリケーション プロジェクトで、レイアウトまたはサブレイアウトで、[デザ イン] タブをクリックし、次に [コントロール] をダブルクリックし、[プロパティ] をクリックします。
   Visual Studio の [プロパティ] パレットが表示されます。
- Visual Studio の [プロパティ] パレットの [パラメーター] プロパティで、URL エスケープされた key=value の組を使い、アンパサンド文字 ("&") で区切られた、名前付きパラメーターを入力し ます。

<sup>12</sup> サブレイアウトに渡されたパラメーターにアクセスするには、

<sup>&</sup>lt;u>http://trac.sitecore.net/SublayoutParameterHelper</u>を参照してください。

Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。 この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



# 4.6 プレースホルダー

プレースホルダーに特有な唯一のプロパティはプレースホルダー キーです。

#### メモ

Sitecore のプレースホルダーは Web コントロールのインスタンスです。デベロッパー センター は すべての Web コントロールのプロパティを設定するために、プレースホルダーを含め、同じダイアログ を使用します。プレースホルダーには、[レンダリング プロパティ] ダイアログの [プレースホルダー] フ ィールドは無意味です。レンダリング プロパティ ダイアログで、[パラメーター] タブにあるプレースホル ダーのキーを key という名前のパラメーターとして入力します。レイアウト詳細では各コントロール に、[一般] セクションの [プレースホルダー] フィールドで、プレースホルダー キー、またはコントロー ルがバインドされる完全修飾プレースホルダー キーを入力します。

#### 重要

レイアウトまたはサブレイアウトでプレースホルダー キーをアサインするとき、必ず content のような 単一のプレースホルダー キーを使用します。 /main/content のような完全修飾プレースホルダー キーは使用しません。



# 4.7 サブレイアウト

<sc:sublayout> コントロールの path 属性には、Web ユーザー コントロール ファイルへのパス を、Web サイトのドキュメント ルートへの相対パスで入力します。13

<sup>&</sup>lt;sup>13</sup> Web コントロールを含むプレゼンテーション コンポーネントで使用できる API に関する詳細については、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20API%20Cookbook.aspx</u>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。

Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。 この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



# 4.8 FieldRenderer Web コントロール

FieldRenderer Web コントロールの FieldName プロパティには、コントロールが処理するフィ ールドの名前を入力します。

## メモ

FieldRenderer Web コントロールはデフォルトでは指定されたフィールドの値をコンテキスト アイ テムから読み出します。データソースを FieldRenderer Web コントロールに渡して、特定のアイ テムから値を読み出します。

## メモ

FieldRenderer Web コントロールは出力キャッシュ オプションをサポートしません。このコントロ ールの出力キャッシュをサポートするためには、Sitecore.Web.UI.WebControls.FieldRenderer から継承する Web コントロールを作成します。GetCachingID() メソッドを上書きし、コントロール のキャッシュ キーを返します。たとえば、アイテムの GUID とフィールド定義アイテムの GUID を含 む文字列です。キャッシュ コンテキストを識別するための情報を追加します。デフォルトの FieldRenderer Web コントロールでなく、この Web コントロールを使用します。



# 4.9 XSL レンダリング

このセクションではデベロッパー センター と Visual Studio で XSL レンダリングを使うための方 法を説明します。<sup>14</sup>

XSL レンダリング定義アイテムの [データ] セクションで、 [Path] フィールドは XSL レンダリング フ ァイルへのパス (Web サイトのドキュメント ルートへの相対パス) を含んでいます。

#### 重要

カスタムの .NET XSL 拡張を使用するために、インラインで .NET コードを XSL レンダリングにおくのは避けてください。

### メモ

システムは XSL レンダリングの変換を Web クライアント上でなく、サーバー上で起動します。

## 4.9.1 新規の XSL レンダリングの作成方法

XSL レンダリングを作成する方法:

- 1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。
  - 2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [レンダリング] をクリックします。

3. [新しいファイル] ダイアログで [テンプレート] リストで [XSLT File] をクリックし、次に [作成] をクリックします。

4. [新しいファイル] ダイアログの [名前] で XSLT レンダリングの名前を入力し、 [次へ] を クリックします。 Sitecore はこの名前を XSLT レンダリング定義アイテムと XSLT レンダリング ファ イルの両方に使います。

5. [新しいファイル] ダイアログのコンテンツ ツリーで、XSL レンダリング定義アイテムを含める アイテムをクリックし、[次へ] をクリックします。

6. [新しいファイル] ダイアログのファイル システム ツリーで、XSL レンダリング ファイルを含め るディレクトリをクリックし、 [作成] をクリックします。 デベロッパー センターに XSL レンダリングが表 示されます。

<sup>&</sup>lt;sup>14</sup> XSL レンダリングに関する詳細は、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u>から『プレゼンテーション コンポーネント XSL リファレンス』を参照し てください。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



## 4.9.2 XSL レンダリングの出力の表示方法

開発者は Sitecore デバッガーまたはデベロッパー センターのプレビュー ウィンドウを使って、 個々の XSL レンダリングの出力を分析することができます。

デベロッパー センターで XSL レンダリングの出力を表示するため方法:

- 1. デベロッパー センターで、既存の XSL レンダリングを開きます。
- 2. デベロッパー センターの編集ウィンドウの最上部で [プレビュー] を有効にします。
- デベロッパー センターで編集ウィンドウの下のプレビュー ウィンドウで、ドロップダウン リストからアイテムを選択します。デベロッパー センターはこのアイテムをレンダリング (\$sc\_item と \$sc\_currentitem)のデータ ソースとして XSL 変換エンジンに渡します。
- デベロッパー センターでプレビュー ウィンドウの右上にある 再読み込み ボタンをクリックします。選択したアイテムをデータ ソースとして XSL 変換を起動した結果がプレビュー ウィンドウに表示されます。

### 4.9.3 XSL レンダリングの定型コード ファイル

デベロッパー センターを使って新規の XSL レンダリングを作成した場合、Sitecore は XSL レン ダリングの定型コード ファイルを複製します。これは新しいコードのためのベースになります。<sup>15</sup>

## メインの XSL テンプレート ブロック

XSL レンダリングを作成した後に、コードをメインの XSL テンプレートに追加します。

```
<xsl:template match="*" mode="main">
<!--//TODO:enter XSL rendering code here-->
</xsl:template>
```

#### メモ

XSL では、XSL 要素 <xsl:template>に囲まれたコード ブロックは XSL テンプレートと呼ばれます。

<sup>&</sup>lt;sup>15</sup> 新しい XSL レンダリングに使われる定型コード ファイルに関する詳細は、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u>から『プレゼンテーション コンポーネント XSL リファレンス』を参照し てください。

Sitecore®は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。 この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



## 4.9.4 カスタムの XSL テンプレート ライブラリ

XSL テンプレートは XSL 要素 <xsl:template> に囲まれた XSL コードのブロックです。これは 他のプログラミング言語でのメソッド、関数、プロシージャに類似した機能です。<sup>16</sup>

開発者は XSL テンプレートに再利用可能な XSL コードのブロックを含めることができます。開発 者は XSL テンプレートを、名前で起動することによって手続き的に使用したり、XPath 合致パター ンで起動することによって宣言的に使用したりすることができます。

各 XSL テンプレートのコンテキスト要素は、XSL 変換エンジンが XSL テンプレートを起動した 時点でコンテキスト要素であった要素です。

XSL テンプレートは XSL 要素 <xsl:param> と <xsl:with-param> を使って、さまざまな名 前付きパラメーターを受け入れることができます。

### XSL テンプレート ライブラリの作成方法

XSL テンプレート ライブラリを作成する方法:

1. Windows ファイル システムのエクスプローラーまたは Visual Studio を使って、XSL テンプレート ライブラリのコード ファイルを含めるファイル システムのディレクトリ (たとえば /xsl/mywebsite ) を作成するか、またはそこに移動します。

2. 新規に .xslt ファイル (たとえば library.xslt) を作成します。次のプロトタイプを使用します:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:sc="http://www.sitecore.net/sc"
xmlns:dot="http://www.sitecore.net/dot"
exclude-result-prefixes="dot sc">
<xsl:template="template"
<xsl:template name="TemplateName">
<!--//TODO:logic-->
</xsl:template>
</xsl:template>
```

3. <xsl:param> と <xsl:with-param> を使ってパラメーターを XSL テンプレート ブロック に渡します。

<xsl:template name="TemplateName" />

<sup>16</sup> XSL テンプレート ライブラリに関する詳細は、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u> から『プレゼンテーション コンポーネント XSL リファレンス』を参照し てください。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



```
<xsl:param name="ParamName" select="'DefaultValue'" />
<!--logic->
</xsl:template
...
<xsl:call-template name="TemplateName">
<xsl:call-template name="ParamName" select="'ParamValue'" />
</xsl:call-template>
```

#### 4. 変数を使って XSL テンプレート ブロックの結果を含めます:

```
<xsl:template>
<xsl:choose>
<xsl:when test="//TODO:logic">
Variable Value
</xsl:when>
<xsl:otherwise>
Default Value
</xsl:otherwise>
</xsl:choose>
</xsl:choose>
</xsl:template>
<xsl:variable name="VariableName">
<xsl:call-template name="TemplateName" />
</xsl:variable>
```

## XSL レンダリングの XSL テンプレート ライブラリを参照する方法

XSL レンダリングの XSL テンプレート ライブラリを参照する方法:

- 1. XSL レンダリングを編集します。
  - 2. 既存の <xsl:output> 要素の下に次のような 1 行を追加します:

<xsl:include href="/xsl/mywebsite/library.xslt" />

3. library.xslt を XSL テンプレート ライブラリ ファイルの URL で置換します。

#### メモ

<xsl:include> 要素の href 属性は URL への参照です。この URL は IIS Web サイトのド キュメント ルートからの XSL テンプレート ライブラリ ファイルへの完全なパス、または相対パス (たと えば library.xslt や ../library.xslt)、またはホストネームを含む完全修飾 URL です。

開発者が作成するレンダリングにこの参照を容易に追加することができるようにするためには、 「XSL レンダリングの定型コード ファイル」のセクションを参照してください。新規の XSL レンダリング に使われる定型コード ファイルに、href 属性に適切な値を使用してコメント化されたライブラリへの 参照を追加します。

<!--<xsl:include href="library.xslt" />-->



#### メモ

Sitecore 6 以降では XSL 拡張コントロールをサポートします。たとえば XSL テンプレート ライ ブラリの <sc:text> などです。XSL テンプレート ライブラリのコードを sc:fld() などの XSL 拡張メ ソッドに限定する必要はありません。

#### 重要

パフォーマンスが非常に重要である場合、または XSL と .NET レンダリングの両方で同じロジック が必要である場合には、XSL テンプレート ライブラリよりも .NET XSL 拡張ライブラリのほうがより 適切です。

## 4.9.5 カスタムの XSL 拡張メソッド

XSL が適さなかったり、パフォーマンスが問題であったり、またはその他の問題が生じたり、要件を満たさない場合には、.NET で書かれた XSL 拡張メソッドが代替的なソリューションとなる場合があります。<sup>17</sup> 下記のような場合には XSL 拡張ライブラリを検討してください:

• 外部ソースに保存されたデータ (たとえば、Sitecore データベース以外のデータベースやア プリケーションなどに) アクセスする。

- .NET API にアクセスする。
- 多くのリソースを要する操作を行う。
- 複雑な操作を行うコードの可読性を向上させる。

#### メモ

他のレンダリング技術に対する XSL のメリットとデメリットに関するより詳細な解説については、『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。

### ヒント

.NET で XSL 拡張を実装する前に、Sitecore の提供しているデフォルトの XSL 拡張を調査し、必要な機能がすでに存在していないかどうか確認してください。

<sup>&</sup>lt;sup>17</sup> XSL 拡張メソッドに関する詳細は、

<sup>&</sup>lt;u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20XSL%20Reference.aspx</u>から『プレゼンテーション コンポーネント XSL リファレンス』を参照し てください。

Sitecore®は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。 この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



#### メモ

XSL 拡張メソッドに加え、Sitecore は XSL 拡張コントロールをサポートします。この文書には XSL 拡張コントロールの解説は含みません。XSL 拡張メソッドは XSL 拡張コントロールよりも柔 軟です。XSL 拡張コントロールは下記のような山括弧の構文を使います:

<sc:text field="FieldName">.

XSL 拡張メソッドのみを使った同等の構文は:

<xsl:value-of select="sc:field('FieldName',.)" disable-output-escaping="yes" />

## XSL 拡張メソッド ライブラリ クラスの作成方法

XSL 拡張ライブラリ クラスを作成するには、Visual Studio の Web アプリケーション プロジェク トで、カスタムの XSL 拡張メソッドのそれぞれを表すメソッドを含むクラスを作成します。このクラスは パラメーターを受け付けないコンストラクターを持つ必要があります。XSL 拡張メソッドは普通は文 字列または System.Xml.XPath.XPathNodeIterator タイプのオブジェクトを返します。これは通 常はデータベースのアイテム、または外部システムから読み出されたデータ、または動的に生成され たデータを表します。

#### カスタムの XSL 拡張メソッド ライブラリの登録方法

任意のクラスをカスタムの XSL 拡張メソッド ライブラリとして登録することができます。このクラスは インターフェースを実装したり、特定のベース クラスから継承したりする必要はありません。

.NET クラスを XSL 拡張ライブラリとして登録する方法:

- 1. web.config で /configuration/sitecore/xslExtensions 要素に移動します。
- 2. <xslExtensions> 要素の中に下記を元にして新しい行を挿入します

<extension mode="on" type="Namespace.Class, Assembly" namespace=http://domain.tld/class singleInstance="true"/>

3. type と namespace 属性の値を適切なクラス シグネチャーと URL で置換します。

#### メモ

namespace 属性の値は有効で一意な URL である必要がありますが、有効な Web ページである必要はありません。

XSL レンダリングの新しい拡張と XSL レンダリングに使われる定型コード ファイルを使うためには、名前空間を URL とレンダリングの exclude-result-prefixes 属性にマッピングします。

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```



xmlns:sc="http://www.sitecore.net/sc" xmlns:dot="http://www.sitecore.net/dot" xmlns:namespace="http://www.domain.tld/class" exclude-result-prefixes="dot sc namespace">

## ヒント

名前空間の定義を XSL レンダリングに使う定型コード ファイルに追加することを検討してください。 定型コード ファイルに関する詳細は、「デベロッパー センターの定型コード ファイル」のセクション を参照してください。

## .NET XSL 拡張ライブラリの使い方

.NET XSL 拡張ライブラリを使う方法:

1. XSL レンダリング ファイルを編集します。

2. < xsl:stylesheet> 要素で、名前空間を拡張に関連付けられている URL にマッピングする属性を追加します。名前空間を exclude-result-prefixes 属性に追加します。これはホワイトスペースで区切られた名前空間のリストを含んでいます。たとえば:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:sc="http://www.sitecore.net/sc"
xmlns:dot="http://www.sitecore.net/dot"
xmlns:my="http://mydomain.tld/myclass"
exclude-result-prefixes="dot sc my">
```

### 重要

名前空間を exclude-result-prefixes 属性の値に追加しない場合、XSL 変換エンジンは HTML の規格では有効でない属性を出力する場合があります。

3. この名前空間を使って XSL 拡張ライブラリのメソッドを起動します。たとえば、クラスが string パラメーターを受け入れて string を返すメソッド MyMethod() を含む場合、そのメソッドは XSL 変数にデータを投入するために使うことができます:

<xsl:variable name="myvariable" select="my:MyMethod('MyParameterValue')" />

または、レンダリングはその文字列を出力のストリームに直接書くこともできます:

<xsl:value-of select="my:MyMethod('MyParameterValue')" />

### ヒント

カスタムの名前空間の定義を XSL レンダリングに使う定型コード ファイルに追加することを検討 してください。これは「XSL レンダリングの定型コード ファイル」のセクションで解説されています。



XSL レンダリングの新しい拡張と XSL レンダリングに使われる定型コード ファイルを使うためには、名前空間を URL とレンダリングの exclude-result-prefixes 属性にマッピングします。

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:sc="http://www.sitecore.net/sc" xmlns:dot="http://www.sitecore.net/dot" xmlns:namespace="http://www.domain.tld/class" exclude-result-prefixes="dot sc **namespace**">

#### 重要

名前空間を exclude-result-prefixes 属性に追加します。そうしない場合、生成されたマークアップが名前空間を含む場合があります。

#### ヒント

名前空間の定義を XSL レンダリングに使う定型コード ファイルに追加することを検討してください。

## メソッドを名前空間 sc に追加する方法

メソッドを名前空間 sc に追加するには、Sitecore.Xml.Xsl.XslHelper を上書きします:

- 1. Sitecore.Xml.Xsl.XslHelperから継承するクラスを作成します。
  - 2. web.config の namespace が http://www.sitecore.net/sc である

/configuration/sitecore/xslExtensions/extension 要素で、type 属性の値をそのクラスのシ グネチャーで置換します。名前空間 sc はそのクラスのメソッドを、Sitecore.Xml.Xsl.XslHelper ベ ース クラスのメソッドと並び、エクスポーズします。

<extension mode="on" type="**Namespace.Class,Assembly**" namespace="http://www.sitecore.net/sc" singleInstance="true" />

## XSL 拡張クラスのプロパティへのアクセス方法

XSL 拡張クラス ライブラリのオブジェクトのプロパティにアクセスするには、明示的に get\_Property() と set\_Property() メソッドを使います。たとえば:

```
<xsl:if test="get_PropertyName()">
<xsl:value-of select="set_PropertyName('PropertyValue')" />
</xsl:if>
```

この場合、XSL 要素 <xsl:value-of> はプロパティを設定しますが、出力は生成しません。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



## 4.9.6 XSL 拡張メソッドの例

このセクションではカスタム .NET XSL 拡張メソッドの例を示します。

## GetHome():Sitecore.Data.Items.Item を返す

XSL レンダリングの定型コード ファイルは XPath ステートメントを使って \$home という名前の変数を定義します。/Sitecore/Content/Home を開始アイテムとしていないサイトで XSL レンダリン グを使う場合には、この変数は無効です。パスをハードコードするのでなく、XSL 拡張メソッドによってロジックを使ってホーム アイテムを決定することができます。

まずサイトのホームアイテムを決定します。次に Sitecore.Configuration.Factory.GetItemNavigator() メソッドを使って Sitecore.Data.Items.Item を XSL レンダリングに使われる

System.Xml.XPath.XPathNodeIterator 表現に変換します。

```
namespace Namespace.Xml.Xsl
 private Sitecore.Data.Items.Item GetHomeItem()
 {
  Sitecore.Data.Database db = Sitecore.Context.Database;
  Sitecore.Data.Items.Item home = database.GetItem(Sitecore.Context.Site.StartPath);
  return(home);
 }
 public class XslHelper
 {
  public Sitecore.Xml.XPath.ItemNavigator GetHome()
  {
    return(Sitecore.Configuration.Factory.CreateItemNavigator(GetHomeItem()));
  }
  public string GetHomeID()
   {
    return(GetHomeItem().ID.ToString());
  }
 }
}
```

XSL レンダリングと、新規の XSL レンダリングに使用される定型コード ファイルの\$home 変数 定義を更新します。

<xsl:variable name="home" select="namespace:GetHome()" />

### ΧŦ

一般に、XML 構造体を処理するより文字列を処理するほうがより効率的です。可能な場合 は、アイテムを System.Xml.XPath.XPathNavigator として返すのでなく、ID を string として返 すメソッドを使います。たとえば、すでに \$home 変数を使っていて、その変数を定義するロジックを



更新したいだけでない限り、\$home 変数を定義するのは避けます。可能な場合は、GetHome() でなく、GetHomeID() メソッドを使用します。上述のように XSL レンダリングの定型コード ファイル を更新する場合は、この変数の宣言をコメント アウトし、不要なオーバーヘッドを避けるようにしま す。開発者はこの変数が必要な場合には、この行のコメントを消して元に戻すことができます。

## GetRandomSiblings():XML を使って複数の値を返す

XSL 拡張から区切り文字または XML を使ってリストを返すことができます。この技法を使ってア イテムの ID のリストを返すことができます。これは XSL 拡張メソッド sc:Split() で使われるものと 類似の XSL コードを使って処理できます。

たとえば、レンダリングはコンテキスト アイテムの 5 つのランダムなシブリング (兄弟) へのリンクを生成する必要があるが、コンテキスト アイテム自身に対しては必要なく、同じシブリングには 2 つのリンクは生成しないとします。次の拡張ライブラリ クラスは Sitecore.Xml.Xsl.XslHelper クラスから継承し、その GetItem() メソッドを使って System.Xml.XPath.XPathNodeIterator に対応する Sitecore.Data.Items.Item を読み出します。

```
namespace Namespace.Xml.Xsl
{
 public class XslHelper : Sitecore.Xml.Xsl.XslHelper
 {
  public XPathNodeIterator GetRandomSiblings(XPathNodeIterator iterator,int max)
   Sitecore.Xml.Packet packet = new Sitecore.Xml.Packet("values","");
    iterator.MoveNext();
    Sitecore.Data.Items.Item item = GetItem(iterator);
    if(item != null )
    {
     Sitecore.Collections.ChildList children = item.Parent.Children;
     if(children.Count>1)
     {
      if(max>children.Count-1)
      {
        max = children.Count-1;
      }
      List<Sitecore.Data.ID> ids = new List<Sitecore.Data.ID>();
      Random rand = new Random();
      while(ids.Count<max)
      {
        int index = rand.Next(children.Count);
        if(children[index].ID!=item.ID && !ids.Contains(children[index].ID))
         packet.AddElement("value",children[index].ID.ToString());
         ids.Add(children[index].ID);
        }
      }
```



```
}
}
XPathNavigator navigator = packet.XmlDocument.CreateNavigator();
if (navigator == null)
{
    navigator = new XmlDocument().CreateNavigator();
}
navigator.MoveToRoot();
navigator.MoveToRoot();
return navigator.SelectChildren(XPathNodeType.Element);
}
```

このコードは下記のような XML 構造体を返します。

```
<values>
<value>{ID}</value>
...
<value>{ID}</value>
</value>
```

```
下記のようなコードを使ってこの構造体を処理することができます。
```

```
<xsl:for-each select="namespace:GetRandomSiblings(.,5)">
<xsl:for-each select="sc:item(text(),$sc_currentitem)">
<sc:link>
<xsl:value-of select="@name" />
<br />
</sc:link>
</xsl:for-each>
</xsl:for-each>
```

## メモ

}

このコードは例示用のみに提供されており、少数のシブリングの処理には非効率です。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



# 4.10 Web コントロール

このセクションでは Web コントロールを使う方法について解説します。18

Web コントロール定義アイテムの名前には Web コントロール クラスの名前を使います。

Web コントロールはいくつかの必須パラメーターに依存します。

Web コントロールの [**タグ**] プロパティには、Web コントロール クラスの名前 (たとえば MyWebControl) を入力します。

Web コントロールの [**タグ プリフィックス**] プロパティには、Web コントロールを含む名前空間の ASP.NET タグ プリフィックス (たとえば mws) を入力します。

Web コントロールの [名前空間] プロパティには、Web コントロール クラスを含む名前空間 (た とえば MyWebSite.Web.UI.WebControls) を入力します

[**アセンブリ**] プロパティには、Web コントロール クラスを含むアセンブリの名前を .dll 拡張子なし で (たとえば MyWebSite) 入力します。

# 4.10.1 Web コントロール クラスの作成方法

Web コントロール クラスを作成する方法:

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を使ってクラスを作成しま す:

```
Namespace.Web.UI.WebControls
{
    public class ClassName : Sitecore.Web.UI.WebControl
    {
        protected override void DoRender(HtmlTextWriter output)
        {
            //TODO: write to output
        }
        protected override string GetCachingID()
        {
            return GetType().ToString();
        }
    }
}
```

<sup>18</sup> Web コントロールを含むプレゼンテーション コンポーネントで使用できる API に関する詳細については、

<u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Compone</u> <u>nt%20API%20Cookbook.aspx</u>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。



}

2. このクラスで Namespace.Web.UI.WebControls を、クラスを含む適当な名前空間で 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 //TODO: write to output を、output HtmlTextWriter に書き出すロジッ クで置換します。

5. このクラスで GetType().ToString()を、コントロールのキャッシュ キーを返すロジックで置換します。

## 4.10.2 Web コントロールの登録方法

デベロッパー センター のウィザードを使って、またはコンテンツ エディターを使って、Web コントロ ールを登録することができます。

ヒント

最初の Web コントロールを名前空間に登録するにはウィザードを使います。同じ名前空間にさらに Web コントロールを登録するには、コンテンツ エディターを使って既存の Web コントロール 定義アイテムを複製し、新しいアイテムの値を更新します。

デベロッパー センター を使って Web コントロールを登録する方法:

- 1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。
  - 2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [レンダリング] を展開します。
  - 3. **[新しいファイル]** ダイアログで **[テンプレート]** リストで **[Web Control]** をクリックし、次 に **[作成]** をクリックします。 **[Web コントロール作成]** ウィザード が表示されます。

4. [Web コントロール作成] ウィザードで、Web コントロール プロパティを入力します。

5. **[Web コントロール作成]** ウィザードで [テスト] をクリックし、デベロッパー センター がう まく Web コントロールを見つけるまで、問題を解決します。 [次へ] をクリックします。

6. [Web コントロール作成] ウィザードで Web コントロール定義アイテムを含めるフォルダ ーをクリックし、次に [作成] をクリックします。Web コントロール定義アイテムがデベロッパー センタ ーに表示されます。

コンテンツ エディター を使って Web コントロールを登録する方法:



1. **コンテンツ エディター**で、Web コントロール定義アイテムを含める /Sitecore/Layout/Renderings アイテムの下のプロジェクト固有のフォルダーに移動します。

2. **コンテンツ エディター**で、/System/Layout/Renderings/Webcontrol データ テンプレートを使って Web コントロ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、Web コントロール定義アイテムで、Web コントロールのプロパティを入力します。

## 4.10.3 Web コントロールへのプロパティの追加方法

Visual Studio を使って Web コントロールにプロパティを追加する方法:

1. Visual Studio の Web アプリケーション プロジェクトで、Web コントロール クラスを開きます。

2. このクラスでプロパティを作成します。たとえば、PropertyName という名前の string プロパティを作成する方法:

```
public string PropertyName
{
  get { throw new NotImplementedException() };
  set { throw new NotImplementedException() };
}
```

## ۶Ł

Web コントロール プロパティを設定する方法は、「レンダリング パラメーターのコントロールへの渡 し方」のセクションを参照してください。



# 4.11 メソッド レンダリング

このセクションではメソッドレンダリングを使う方法について解説します。

メソッド レンダリング定義アイテムはメソッドを指定する場合も、しない場合もあります。メソッド レンダリング定義アイテムがメソッドを指定しない場合、メソッド レンダリングをレイアウト詳細のプレース ホルダーにバインドしたユーザー、またはそれをデベロッパー センターでレイアウトまたはサブレイアウトの上にドラッグしたユーザーは、メソッドを指定する必要があります。デフォルトのメソッドを別のメソッ ドレンダリングに設定するには、別のメソッドを指定する複数のメソッドレンダリング定義アイテムを 挿入する必要があります。

メソッド レンダリング定義アイテムの名前には、定義アイテムがメソッドを指定する場合には、名前 空間とクラス名を含めたメソッドの名前を使います。または

/Sitecore/Layout/Renderings/System アイテムの下にメソッド レンダリング定義アイテムを作成し、名前 Method Rendering を使用します。

[メソッド] には、メソッドの名前を入力します。

[クラス]には、メソッドを含むクラスの名前を入力します。

[アセンブリ]には、クラスを含むアセンブリの名前を.dll 拡張子なしで入力します。

#### 重要

メソッド レンダリングの使用より、Web コントロールを使ってメソッドを囲むことを検討してください。

#### ۶Ŧ

メソッド レンダリング Web コントロールは出力キャッシュをサポートしません。メソッド レンダリングの 出力キャッシュをサポートするには、Sitecore.Web.UI.WebControls.Method から継承する Web コントロールを作成し、GetCachingID() メソッドを上書きしてコントロールのキャッシュ キーを 返します。たとえば、アセンブリ、名前空間、クラス、メソッド名を含む文字列と、キャッシュ コンテキス トを識別する追加の基準とを返すことができ、この Web コントロールをデフォルトのメソッド レンダリ ング Web コントロールに代えて使うことができます。または、メソッドを起動する Web コントロール を作成し、その Web コントロールで GetCachingID() メソッドを定義します。

## 4.11.1 メソッド レンダリング クラスとメソッドの作成方法

メソッド レンダリング クラスとメソッドを作成する方法:

 Visual Studio の Web アプリケーション プロジェクトで、メソッド レンダリングを含むクラスがすでに 存在している場合、そのクラスにメソッドを追加することを検討してください。または新規のクラスを作 成します。



2. そのクラスで、次のシグネチャーをもつメソッドを作成します。

public string MethodName()

3. そのクラスで、メソッドの本体を完成させます。

## 4.11.2 メソッド レンダリングの登録方法

デベロッパー センター または コンテンツ エディターを使ってメソッド レンダリングを登録することが できます。

デベロッパー センター を使ってメソッド レンダリングを登録する方法:

- 1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。
  - 2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [レンダリング] をクリックします。

3. [新しいファイル] ダイアログで [テンプレート] リストで [メソッド レンダリング] をクリック し、次に [作成] をクリックします。 [メソッド レンダリング作成] ウィザード が表示されます。

4. [メソッド レンダリング作成] ウィザードで、メソッド レンダリング プロパティを入力します。

5. [メソッド レンダリング作成] ウィザードで [テスト] をクリックし、デベロッパー センター が メソッドを見つけるまで、問題を解決します。[次へ] をクリックします。

6. [メソッド レンダリング作成] ウィザードで メソッド レンダリング定義アイテムを含めるアイテム をクリックし、次に [**作成**] をクリックします。

**コンテンツ エディター**を使ってメソッド レンダリングを登録する方法:

## 1. **コンテンツ エディター**で、メソッド レンダリング定義アイテムを含める /Sitecore/Layout/Renderings の中のアイテムを選択します。

2. **コンテンツ エディター**で、/System/Layout/Renderings/Method Rendering データテンプレートを使ってメソッド レンダリング定義アイテムを挿入します。

3. コンテンツ エディターで、メソッド レンダリング定義アイテムで、メソッド レンダリングのプロパティを入力します。



## 4.12 URL レンダリング

このセクションでは URL レンダリングを使う方法について解説します。

URL レンダリング定義アイテムは URL を参照する場合も、しない場合もあります。URL レンダリ ング定義アイテムが URL を指定しない場合、URL レンダリングをレイアウト詳細のプレースホルダ ーにバインドしたユーザー、またはそれをデベロッパー センターでレイアウトまたはサブレイアウトの上に ドラッグしたユーザーは、URL を指定する必要があります。デフォルトの URL を別の URL レンダリ ングに設定するには、別の URL を指定する複数の URL レンダリング定義アイテムを挿入する必 要があります。

URL レンダリング定義アイテムの名前には、定義アイテムが URL を指定する場合には、URL を 識別できる名前を使います。または /Sitecore/Layout/Renderings/System フォルダーの下に URL レンダリング定義アイテムを作成し、名前 URL Rendering を使用します。

URL プロパティには処理する URL を入力します。

#### УŦ

レイアウト エンジンは URL レンダリングを Web サーバー上で起動します。 Web クライアントは URL レンダリングを直接起動しません。

#### 4.12.1 URL レンダリングの登録方法

デベロッパー センター を使って URL レンダリングを登録する方法:

- 1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。
- 2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [レンダリング] をクリックします。
- [新しいファイル] ダイアログで [テンプレート] リストで [URL レンダリング] をクリックし、次に [作 成] をクリックします。[URL レンダリング作成] ウィザード が表示されます。
- 4. [URL レンダリング作成] ウィザードで、URL レンダリング プロパティを入力します。
- 5. [**URL レンダリング作成**] ウィザードで [**テスト**] をクリックし、URL に関する問題を解決します。 [次へ] をクリックします。
- [URL レンダリング作成] ウィザードで URL レンダリング定義アイテムを含めるアイテムをクリック し、次に [作成] をクリックします。

コンテンツ エディター を使って URL レンダリングを登録する方法:



- 1. **コンテンツ エディター**で、URL レンダリング定義アイテムを含める /Sitecore/Layout/Renderings アイテムの子孫を選択します。
- 2. **コンテンツ エディター**で、/System/Layout/Renderings/Url Rendering データ テンプレートを 使って URL レンダリング定義アイテムを挿入します。
- 3. **コンテンツ エディター**で、URL レンダリング定義アイテムで、URL レンダリングのプロパティを入力します。



# 4.13 レンダリング設定データ テンプレートを実装する方法

レンダリング設定データ テンプレートを実装する方法:

- 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、レンダリング設定データ テンプレート定義 を含める /Sitecore/Templates アイテムの子孫を選択します。
- テンプレートマネージャーまたはコンテンツ エディターでレンダリング設定データ テンプレート定義ア イテムを挿入します。レンダリング設定データ テンプレートの名前としてレンダリングの名前を使うこと を検討してください。System/Layout/Rendering Parameters/Standard Rendering Parameters データ テンプレートをベース テンプレートとして選択します。
- 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、レンダリング設定データ テンプレート定義 アイテムで、.NET コントロールのプロパティに対応する名前をもつフィールド、または XSL レンダリン グのパラメーターを構成します。

#### 重要

プロパティとパラメーターの名前にはスペースおよびその他の特殊文字を含むことはできません。

4. コンテンツ エディターで、プロパティを適用するレンダリング定義アイテムを選択します。

5. コンテンツ エディターのレンダリング定義アイテムで、[エディター オプション] セクションの [パラメーター テンプレート] フィールドで、レンダリング設定データ テンプレート定義アイテムを選 択します。

6. **コンテンツ エディター**で、レンダリングを使用しているアイテムのレイアウト詳細で、プロパティまたはパラメーターの値を入力します。



# **Chapter 5**

# レイアウトとサブレイアウト

この章ではレイアウトとサブレイアウトを使う方法について解説します。

この章には次のセクションがあります。

- レイアウトの作成
- サブレイアウトの作成
- コントロールをレイアウトまたはサブレイアウトに追加する
- コードビサイドをレイアウトまたはサブレイアウトに追加する
- レイアウトまたはサブレイアウトの部分クラス ファイルを追加しコード ファイルをコードビ ハインドで置換する方法

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



# 5.1 レイアウトの作成

デベロッパー センター または Visual Studio または任意のテキスト エディターを使ってレイアウトを作成することができます。

## 5.1.1 デベロッパー センターを使ったレイアウトの作成方法

デベロッパー センター を使って レイアウトを作成する方法:

1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。

2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [Layout] をクリックします。

3. [新しいファイル] ダイアログで [テンプレート] リストで [レイアウト] をクリックし、次に [作 成] をクリックします。[レイアウト作成] ウィザードが表示されます。

4. [**レイアウト作成**] ウィザードの [**名前**] でレイアウトの名前を入力し、[**次へ**] をクリックし ます。ウィザードはこの名前をレイアウト定義アイテムとレイアウト ファイルの両方に使います。

5. [**レイアウト作成**] ウィザードでレイアウト定義アイテムを含めるフォルダーをクリックし、次に [次へ] をクリックします。

6. [レイアウト作成] ウィザードでレイアウト ファイルを含めるディレクトリをクリックし、次に [作 成] をクリックします。デベロッパー センターに レイアウトが表示されます。

# 5.1.2 Web フォームをレイアウトとして登録する方法

Web フォームをレイアウトとして登録する方法:

 Visual Studio の Web アプリケーション プロジェクトで、Web サイトのドキュメント ルートの /layouts ディレクトリのサブディレクトリに Web フォームを作成します。たとえば、 /layouts/MyWebSite/MyWebLayout.aspx などです。

2. **コンテンツ エディター**で Web フォーム ファイルの場所に対応する /Sitecore/Layout/Layouts アイテムの子孫に移動します。たとえば /Sitecore/Layout/Layouts/MyWebSite などです。

3. コンテンツ エディターで/System/Layout/Layout データ テンプレートを使ってレイアウト 定義アイテムを挿入します。レイアウト定義アイテムの名前には、Web フォーム ファイルの名前 を .aspx 拡張子なしで入力します。たとえば MyWebLayout などです。

4. コンテンツ エディターで、レイアウト定義アイテムで、[グリッド デザイナー] タブの例外を 無視します。[コンテンツ] タブをクリックします。


5. **コンテンツ エディター**の [**コンテンツ**] タブの [**データ**] セクションの [**Path**] フィールド で、レイアウト ファイルのパスと名前を、Web サイトのドキュメント ルートへの相対で .aspx 拡張子 を含めて入力します。たとえば、/layouts/MyWebSite/MyWebLayout.aspx です。

### 重要

/Sitecore/Layout/Layouts アイテムの挿入オプションにある レイアウト コマンド テンプレートを 使って、既存の Web フォームを Sitecore レイアウトとして登録することはできません。レイアウト コマンド テンプレートはレイアウト定義アイテムを挿入して対応するレイアウト ファイルを作成するた めに使います。



## 5.2 サブレイアウトの作成

デベロッパー センターを使って、サブレイアウトを作成したり、Visual Studio で作成した Web ユーザー コントロールをサブレイアウトとして登録することができます。

### 5.2.1 デベロッパー センターを使ったサブレイアウトの作成方法

デベロッパー センター を使って サブレイアウトを作成する方法:

- 1. デベロッパー センターで [ファイル] メニューをクリックし、次に [新規作成] をクリックします。[新し いファイル] ダイアログが表示されます。
  - 2. [新しいファイル] ダイアログで [カテゴリー] ツリーで [Rendering] をクリックします。.

3. [新しいファイル] ダイアログで [テンプレート] リストで [Sublayout] をクリックし、次に [作成] をクリックします。 [サブレイアウト作成] ウィザードが表示されます。

4. [サブレイアウト作成] ウィザードの [名前] でサブレイアウトの名前を入力し、[次へ] を クリックします。デベロッパー センターはこの名前をサブレイアウト定義アイテムとサブレイアウト ファイ ルの両方に使います。

5. [サブレイアウト作成] ウィザードでサブレイアウト定義アイテムを含めるアイテムをクリックし、[次へ] をクリックします。

6. [**サブレイアウト作成**] ウィザードでサブレイアウト ファイルを含めるディレクトリをクリックしま す。

7. [サブレイアウト作成] ウィザードで [新しい関連付けされた C# コード ファイルを作成] を選択し、サブレイアウトのコード ファイルを作成します。

8. [作成]をクリックします。デベロッパーセンターにサブレイアウトが表示されます。

### 5.2.2 Web ユーザー コントロールをサブレイアウトとして登録する方法

Web ユーザー コントロールをサブレイアウトとして登録する方法:

- 1. Visual Studio の Web アプリケーション プロジェクトで、Web サイトのドキュメント ルートの /layouts ディレクトリのサブディレクトリに Web ユーザー コントロールを作成します。たとえば、 /layouts/MyWebSite/MySublayout.aspx などです。
- 2. コンテンツ エディターでサブレイアウトを含むプロジェクト固有のサブディレクトリに対応する /Sitecore/Layout/Sublayouts アイテムの子孫に移動します。たとえば /Sitecore/Layout/Sublayouts/MyWebSite などです。



3. コンテンツ エディターで/System/Layout/Sublayout データ テンプレートを使ってサブレイアウト 定義アイテムを挿入します。サブレイアウト定義アイテムの名前には、Web ユーザー コントロール ファイルの名前を .ascx 拡張子なしで入力します。たとえば MySublayout などです。

4. コンテンツ エディターで、サブレイアウト定義アイテムで、[グリッド デザイナー] タブの例外を無視します。[コンテンツ] タブをクリックします。

5. コンテンツ エディターのサブレイアウト定義アイテムで、[コンテンツ] タブの [データ] セク ションの [Path] フィールドで、レイアウト ファイルのパスと名前を、Web サイトのドキュメント ルー トへの相対で .ascx 拡張子を含めて入力します。たとえば、 /layouts/MyWebSite/Sublayouts/MySublayout.aspx です。

### 重要

/Sitecore/Layout/Sublayouts アイテムの挿入オプションにある サブレイアウト コマンド テンプ レートを使って、既存の Web フォームを Sitecore サブレイアウトとして登録することはできませ ん。サブレイアウト コマンド テンプレートはサブレイアウト定義アイテムを挿入して対応するサブレイ アウト ファイルを作成するために使います。



# 5.3 コントロールをレイアウトまたはサブレイアウトに追加する

このセクションではレイアウトとサブレイアウトにコントロールを追加する方法について解説します。これらの手順を使って、レイアウトまたはサブレイアウトに、プレースホルダーを追加したり、サブレイアウトやレンダリングを静的にバインドすることができます。

### 重要

サブレイアウトが静的または動的に自身の他のインスタンスにバインドしたり、自身の子孫であるコ ントロールにバインドしないようにしてください。そのようなバインドは無限再帰を生じる場合がありま す。

# 5.3.1 デベロッパー センターを使ってコントロールをレイアウトまたはサブレイアウトに 追加する方法

デベロッパー センターを使ってコントロールをレイアウトまたはサブレイアウトに追加する方法:

- 1. デベロッパー センターで、レイアウトまたはサブレイアウトを開きます。
  - 2. デベロッパー センターのレイアウトまたはサブレイアウトで、[デザイン] タブをクリックします。
- 3. デベロッパー センターで [表示] メニューをクリックし、次に [ツールボックス] をクリックします。

4. デベロッパー センターで [ツールボックス] からコントロールをレイアウトまたはサブレイアウトの上にドラッグします。

5. デベロッパー センターのレイアウトまたはサブレイアウトで、[コントロール] をタブルクリックします。[レンダリング プロパティ] ダイアログが表示されます。

6. [レンダリング プロパティ] ダイアログで、コントロール プロパティを適用します。

# 5.3.2 Visual Studio を使ってコントロールをレイアウトまたはサブレイアウトに追加する方法

Visual Studio を使ってコントロールをレイアウトまたはサブレイアウトに追加する方法:

- 1. Visual Studio の Web アプリケーション プロジェクトで [表示] メニューをクリックし、次に [ツー ルボックス] をクリックします。
  - 2. Visual Studio でコントロールを含むレイアウトまたはサブレイアウトを開きます。
  - 3. Visual Studio のレイアウトまたはサブレイアウトで、[デザイン] タブをクリックします。



4. Visual Studio でコントロールをツールボックスからレイアウトまたはサブレイアウトの上にド ラッグします。またはコントロール マークアップを挿入します。プレースホルダーには次のマークアップを プロトタイプとして使います:

<sc:placeholder key="" id="" runat="server" />

### 5. サブレイアウトには次のマークアップをプロトタイプとして使います:

<sc:sublayout path="/layouts/path/to/file.ascx" id="" runat="server" />

### 6. XSL レンダリングには次のマークアップをプロトタイプとして使います:

<sc:xslfile path="/xsl/path/to/file.xslt" id="" runat="server" />

7. FieldRenderer Web コントロールには次のマークアップをプロトタイプとして使います:

<sc:fieldrenderer fieldname="FieldName" id="" runat="server" />

8. Web コントロールにはこの文書の他で記載されているようにタグ プリフィックスを登録し、次のマークアップをプロトタイプとして使います:

<tagprefix:classname id="" runat="server" />

9. メソッド レンダリングには次のマークアップをプロトタイプとして使います:

<sc:method methodname="MethodName" assemblyname="AssemblyName" classname="Namespace.Class" id="" runat="server" />

10. URL レンダリングには次のマークアップをプロトタイプとして使います:

<sc:webpage id="" runat="server" url="" />

11. Visual Studio のレイアウトまたはサブレイアウトで、[コントロール] を右クリックし、[プロ パティ] をクリックします。 Visual Studio の [プロパティ] パレットが表示されます。

12. Visual Studio の [プロパティ] パレットで、コントロール プロパティを適用します。

### メモ

静的にバインドされたサブレイアウトで Sitecore の出力キャッシュをサポートするには、ASP.NET を使って Web ユーザー コントロールを起動するよりも、Sitecore サブレイアウト Web コントロー ル (<sc:sublayout>) を使います。

### 重要

名前付きコントロール プロパティを使ってパラメーターを静的にバインドされたサブレイアウトに渡す ことはできません。それに代えて、Sitecore サブレイアウト Web コントロール (<sc:sublayout>)の Parameters プロパティを使います。Parameters プロパティに関する 詳細は、「レンダリング パラメーターのコントロールへの渡し方」のセクションを参照してください。



# 5.4 コードビサイドをレイアウトまたはサブレイアウトに追加する

デベロッパー センターでレイアウトまたはサブレイアウトを作成するとき、Sitecore はデフォルトで はコードビサイド ファイルを作成しません。2 つの方法でコードビサイド ファイルを既存のレイアウトま たはサブレイアウトに追加することができます。既存のレイアウトまたはサブレイアウト ファイルのテキス トをコピーして、次に元のファイルを削除し、次にそのファイルを Visual Studio で再作成し、テキ ストを新しいファイルに貼り付けます。または、コードビサイドとデザイナー ファイルを手動で作成する こともできます。

# 5.4.1 既存のファイルを削除してコードビサイドをレイアウトまたはサブレイアウトに追加する方法

既存のファイルを削除してコードビサイド ファイルを既存のレイアウトまたはサブレイアウトに追加す る方法:

1. Visual Studio の Web アプリケーション プロジェクトで、レイアウトまたはサブレイアウトを開きま す。レイアウトまたはサブレイアウトの正確な名前と場所を記録しておきます。

2. Visual Studio のレイアウトまたはサブレイアウトで、[**ソース**] タブをクリックします。

3. Visual Studio のレイアウトまたはサブレイアウトの [**ソース**] タブで、レイアウトまたはサブ レイアウト ファイルのすべてのコードを選択します。ただしレイアウト ファイルの最上部の Page ディレ クティブ、またはサブレイアウト ファイルの最上部の Control ディレクティブを除きます。

4. Visual Studio のレイアウトまたはサブレイアウトで、選択したテキストを Windows のク リップボードにコピーします。

5. Visual Studio の**ソリューション エクスプローラー**でレイアウトまたはサブレイアウトを削除します。

6. Visual Studio の**ソリューション エクスプローラー**で新規の Web フォーム (レイアウト) または Web ユーザー コントロール (サブレイアウト) を元のレイアウトまたはサブレイアウトのファイル 名とディレクトリを使って追加します。

7. Visual Studio のレイアウトまたはサブレイアウト ファイルで、[**ソース**] タブをクリックしま す。

8. Visual Studio のレイアウトまたはサブレイアウトですべてのテキストを選択します。ただし ファイルの最上部の Page または Control ディレクティブを除きます。

9. Windows クリップボードの中身を貼り付けて選択したテキストを置換します。



# 5.4.2 ファイルを作成してコードビサイドをレイアウトまたはサブレイアウトに追加する 方法

コードビサイドとデザイナー ファイルを作成してコードビサイド ファイルを既存のレイアウトまたはサブ レイアウトに追加する方法:

1. Visual Studio の Web アプリケーション プロジェクトで、レイアウトまたはサブレイアウトの名前を 記録しておきます。

ヒント

レイアウトまたはサブレイアウトのファイルの名前を Windows のクリップボードにコピーするには、 Visual Studio **Solution Explorer** でファイルをクリックし、ファイルの名前を変更するため F2 を押し、次に CTRL-C を押してファイル名を Windows のクリップボードにコピーし、次に ESC を 押してファイルの名前の変更の操作をキャンセルします。

Visual Studio の ソリューション エクスプローラー でレイアウトまたはサブレイアウトを含むフォルダーを右クリックし、次に [追加] をクリックし、 [クラス] をクリックします。 [新しい項目の追加] ダイアログが表示されます。

3. [新しい項目の追加] ダイアログの [名前] でレイアウトまたはサブレイアウトのファイルの名前を.cs 拡張子を含めて入力します。たとえば、MyLayout.aspx.cs または MySublayout.ascx.cs などです。次に [追加] をクリックします。 Visual Studio は新しいクラス を既存のレイアウトまたはサブレイアウトの子として作成します。

4. Visual Studio のクラスで、適当な名前空間とクラス名を入力します。

5. Visual Studio でレイアウトまたはサブレイアウトを開き、[**ソース**] タブをクリックします。

6. Visual Studio のレイアウトまたはサブレイアウトで、既存の Page (レイアウト) または Control (サブレイアウト) ディレクティブを置換します。レイアウトには下記のようなコードを使いま す:

<%@ Page language="C#" autoeventwireup="true" inherits="Namespace.Class" codebehind="MyLayout.aspx.cs" %>

サブレイアウトには下記のようなコードを使います:

<%@ control language="C#" autoeventwireup="true" inherits="Namespace.Class" codebehind="MySubalyout.aspx.cs" %>

7. Visual Studio のレイアウトまたはサブレイアウトで、inherits 属性で名前空間とクラス 名を入力します。codebehind 属性には、コードビハインド ファイルの名前を入力します。



8. 次の「レイアウトまたはサブレイアウトの部分クラス ファイルを追加しコード ファイルをコードビ ハインドで置換する方法」のセクションを参照してください。



# 5.5 レイアウトまたはサブレイアウトの部分クラス ファイルを追加しコード ファ イルをコードビハインドで置換する方法

レイアウトまたはサブレイアウトの部分クラスファイルを追加し、CodeFile を CodeBehind で置換 するため方法:

- 1. Visual Studio で Web アプリケーション プロジェクトを開きます。
- 2. Visual Studio のソリューション エクスプローラーで .aspx または .ascx ファイルを右クリックし、 [プロジェクトに含める] をクリックします。
- Visual Studio のソリューション エクスプローラーで .aspx または .ascx ファイルを右クリックし、 [Web アプリケーションに変換] をクリックします。 Visual Studio は部分クラス ファイルを作成 し、それをプロジェクトに追加し、.aspx または .ascx ファイルの CodeFile 属性を CodeBehind 属性で置換します。



# **Chapter 6**

# RSS 機能

この章では Sitecore の Web サイト管理のための RSS (Really Simple Syndication) 機能の概要について説明します。<sup>19</sup>

この章ではまず Sitecore の Web サイト管理のための RSS 機能の概要を説明し、次に RSS の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 の RSS の概要
- RSS の構成

<sup>&</sup>lt;sup>19</sup> RSS に関する詳細は <u>http://en.wikipedia.org/wiki/RSS (file\_format)</u>を参照してください。 Sitecore の RSS 機能についての詳細は、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Content%20Author%27s%</u> <u>20Cookbook.aspx</u> 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。Sitecore クライアント の CMS フィードについての詳細は、 <u>http://sdn.sitecore.net/Reference/References%20in%20Japanese/Client%20Configuration%2</u> 0Cookbook.aspx から『クライアント構成クックブック』を参照してください。



### 6.1 Sitecore の RSS の概要

Sitecore のアイテムを RSS (Really Simple Syndication) の XML フォーマットを使って配 信することができます。RSS フィードの数はいくつでも定義できます。各フィードは配信アイテムをいく つでも含むことができます。各フィード定義アイテムがフィードに含める配信アイテムを指定します。配 信アイテムに関する詳細は、「配信アイテム」のセクションを参照してください。

### 6.1.1 フィード定義アイテム

各 RSS フィードは System/Feeds/RSS Feed データ テンプレートに基づくフィード定義アイテム からできています。

### ヒント

あるアイテムのデータ テンプレートが System/Feeds/RSS Feed データ テンプレートから継承し ている場合、そのアイテムは RSS フィードを提供できます。

各フィード定義アイテムのデフォルト デバイスのレイアウト詳細が System/Feed Delivery Layout レイアウトを指定します。System/Feed Delivery Layout レイアウトのベース クラスが、 RSS フィードとしてフィード定義アイテムによって指定された配信アイテムのフォーマットを整形しま す。

### RSS フィードの作成方法

RSS フィードを作成する方法:

1. **コンテンツ エディター**で System/Feeds/RSS Feed データ テンプレートを使ってフィード定義アイ テムを挿入します。

コンテンツ エディターのフィード定義アイテムで、[データ] セクションの [アイテム] フィールドでアイテムを選択するか、または query: プリフィックスを使って Sitecore クエリを入力します。
[アイテム] フィールドはデータ テンプレートのフィールド定義のソース プロパティのように機能します。
<sup>20</sup> アイテムを選択した場合、RSS フィードはそのアイテムの子を配信アイテムとして含みます。クエリを入力した場合、RSS フィードはそのクエリに合致するアイテムを配信アイテムとして含みます。

3. **コンテンツ エディター**のフィード定義アイテムで、[データ] セクションの [タイトル] フィール ドで RSS フィードのタイトルを入力します。

<sup>20</sup> データ テンプレートのフィールドのソース プロパティと Sitecore クエリに関する詳細は、 http://sdn.sitecore.net/Reference/References%20in%20Japanese/Data%20Definition%20Ref erence.aspx から『データ定義リファレンス』を参照してください。



4. **コンテンツ エディター**のフィード定義アイテムで、[データ] セクションの [説明] フィールドで RSS フィードの説明を入力します。

#### メモ

フィード定義アイテムの [データ] セクションの [リンク] フィールドの値を入力することによって、 RSS フィードを代替 URL に関連付けることができます。[リンク] フィールドに値を指定しなかった 場合、Sitecore はフィード定義アイテムのデフォルト URL をフィードの URL として使います。ユー ザーが RSS フィードのタイトルをクリックすると、RSS クライアントは [リンク] というフィールドで指定 された URL をロードします。または [リンク] フィールドが空の場合には、RSS フィードのデフォルト URL をロードします。

5. **コンテンツ エディター**でフィード定義アイテムの [追加のメタデータ] セクションで、フィード に関する追加のメタデータを入力することもできます。

 Sitecore にフィードをキャッシュさせる場合には、コンテンツ エディターでフィード定義アイ テムの [キャッシュ] セクションで [キャッシュ可能] のチェック ボックスをオンにします。[キャッシュ間
7イールドでキャッシュ エントリーの強制排出を制御する値を入力します。

#### 注意

認証や承認に依存する RSS フィードをキャッシュしないでください。

ΧŦ

[キャッシュ間隔] フィールドでは .NET が System.TimeSpan.Parse() メソッドを使って解析で きる値を入力することができます。<sup>21</sup>

### 6.1.2 配信アイテム

各フィードは配信アイテムをいくつでも含むことができます。各配信アイテムは Feed デバイスのレイ アウト詳細を定義します。System/Feed Delivery Layout レイアウトは Feed デバイスのレイアウ ト詳細を使用して各配信アイテムを整形します。

### メモ

Sitecore は各コンテンツ アイテムを整形するために Feed デバイスを使用しません。デフォルトでは Feed デバイスをアクティベートする HTTP 要求はありません。Sitecore は Feed デバイスを使

<sup>&</sup>lt;sup>21</sup> System.TimeSpan.Parse() メソッドに関する詳細については、

<sup>&</sup>lt;u>http://msdn.microsoft.com/en-us/library/se73z7b9(VS.71).aspx</u>を参照してください。



用して配信アイテムのプレゼンテーションを構成します。System/Feed Delivery Layout はこれらのレイアウト詳細を使って各配信アイテムを整形します。

### アイテムを配信可能にする方法

アイテムを RSS フィードを通して配信可能にするため方法:

- 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、配信するアイテムのタイプのためのスタン ダード バリュー アイテムを選択するか、または配信する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、[フィード] グループで[デザイン] コマンドをクリックします。[このフィードのプレゼンテーション] ダイアログが表 示されます。
- 4. [**このフィードのプレゼンテーション**] ダイアログで [**タイトル**] フィールドで、そのタイプのアイテムのフ ィードで表示されるタイトルを含むアイテムに関連付けられたデータ テンプレートのフィールドを選択 します。
- 5. [**このフィードのプレゼンテーション**] ダイアログで [本文] フィールドで、そのタイプのアイテムのフィー ドで表示される本文テキストを含むアイテムに関連付けられたデータ テンプレートのフィールドを選 択します。
- 6. [**このフィードのプレゼンテーション**] ダイアログで [**日付**] フィールドで、そのタイプのアイテムのフィードで表示される日付を含むアイテムに関連付けられたデータ テンプレートのフィールドを選択します。

### 重要

必ずスタンダード バリューを使って配信エントリーのオプションを管理します。個々のアイテムの配信 エントリーのオプションを更新するためにレイアウト詳細を使用しないでください。レイアウト詳細に関 する詳細は「Chapter 3 レイアウト詳細」を参照してください。

### メモ

[このフィードのプレゼンテーション] ダイアログは、選択されたアイテムに関連付けられたデータテンプレートのスタンダード バリューの配信エントリーのオプションを自動的に更新します。

レンダリング パラメーターを使って追加の配信エントリープロパティを設定することができます。追加 の配信プロパティを設定するには、Feed デバイスのレイアウト詳細を編集し、次に FeedRenderer Web コントロールのレンダリング パラメーターを設定します。レンダリング パラメー



ターに関する詳細は、「レンダリング パラメーターのコントロールへの渡し方」のセクションを参照してく ださい。

[Author] パラメーターは配信エントリーの編集者の名前を含むデータ テンプレートのフィールドの 名前を指定します。[Enclosure] パラメーターは配信エントリーに関連付けられたメディア ライブラ リ アイテムへのリンクを含むデータ テンプレートのフィールドの名前を指定します。

Sitecore CMS 6.4 以降 プレゼンテーション コンポーネント クックブック



## 6.2 RSS の構成

このセクションでは RSS の構成の設定を説明します。

### 6.2.1 Feeds.MaximumItemsInFeed 設定

web.config の/configuration/sitecore/settings/setting エレメントの name Feeds.MaximumItemsInFeed の value 属性が、管理される Web サイトの RSS フィードに表 れるフィード アイテムの最大数を制御します。

### 6.2.2 Feeds.ItemExpiration 設定

web.config の/configuration/sitecore/settings/setting エレメントの nameFeeds.ItemExpiration の value 属性が、配信アイテムがフィードに表れる期間を制御し ます。Feeds.ItemExpiration にゼロ ("0")を設定すると、配信アイテムは期限が切れることがあ りません。この Feeds.ItemExpiration 設定が正の整数の場合、[日付] フィールド の値にこの日 数を足した数が、現在のシステム日付を超える配信エントリーを RSS フィードから除外します。[日 付] フィールドに関する詳細は「アイテムを配信可能にする方法」のセクションを参照してください。