

# Sitecore CMS 7.1

# ルール エンジン クックブック

ルール エンジンと条件付きレンダリングのヒントと、開発者向けテクニック



# 目次

Chapter 1 イントロダクション	3
1.1 ルール エンジンの概要	4
1.1.1 条件	4
条件のテキスト	4
条件の実装方法	8
1.1.2 アクション	8
アクション テキスト	9
アクションの実装方法	10
1.1.3 スクリプト	11
スクリプトの実装方法	12
1.1.4 ルール要素の表示場所の指定	13
1.2 ルール	16
ルールの種類	16
ルールの実装方法	17
Chapter 2 条件付きレンダリング	19
2.1 条件付きレンダリング	20
2.2 条件付きレンダリングの実装	21
2.2.1 条件付きレンダリングの条件	21
2.2.2 条件付きレンダリング アクション	21
条件付きレンダリング アクションの実装方法	21
例: 条件付きレンダリングを使ったレンダリングの追加	22
2.2.3 条件付きレンダリング ルール	23
2.3 条件付きレンダリングの適用	24
2.3.1 条件付きレンダリングの適用方法	
2.3.2 グローバル 条件付きレンダリング ルール	24
個別のレンダリングの条件付きルールの作成	24
Chapter 3 条件付きレンダリング ルールの実装	26
3.1 条件付きレンダリングの模範例	27
乱数の条件付きレンダリングの条件を作成	27
条件の実装	28
条件付きレンダリング アクションの作成	29
乱数の条件付きレンダリング ルールの作成	30



# **Chapter 1**

# イントロダクション

Sitecore 管理者と開発者は、Sitecore ルール エンジンの使用を開始する前に、本ドキュメントに目を通すことをお勧めします。ルール エンジンを使用することによって、挿入オプションの動的な操作や、システム イベントの処理、アイテムの検証を行うことができ、また条件付きレンダリング(プレゼンテーション、行動ターゲティング、多変量テスト、A/B テスト、または行動マーケティングとして参照されることもあります)を活用することができます。

多変量テストについての情報は、Marketing Operations Cookbook を参照してください。 このドキュメントには次の章があります。

- Chapter 1 イントロダクション マニュアルのイントロダクションです。 この章では、ルール エンジン、スクリプト、条件、アクション、ルールの概要を含む Sitecore ルール エンジンの内容について説明します。
- Chapter 2 イントロダクション条件付きレンダリング この章では、プレゼンテーション コンポーネントの実行時の操作を可能にする条件付きレンダリングの 実装方法について説明します。
- Chapter 3 条件付きレンダリング ルールの実装 この章では、条件付きレンダリング ルールの実装方法と、コード サンプルの挿入方法について説明 します。



#### 1.1 ルール エンジンの概要

Sitecore ルール エンジンは、ルールで定義された特定の条件が真である場合、アクションで定義された ロジックを適用します。ルール エンジンにはユーザー インターフェースが備えられています。これにより、挿入 オプション、条件付きレンダリング、パーソナライゼーションを含む Sitecore 機能をコントロールすることがで きます。これらのルールはエンゲージメント プランで広範囲にわたって使用されます。

レイアウト エンジンについての情報は、<u>Presentation Component Reference</u> マニュアルを参照してください。

挿入オプションについての追加情報は、Data Definition Reference マニュアルを参照してください。

条件、アクション、ルールはすべて /sitecore/system/Settings/Rules フォルダーに保存されます。

ルール設定エディターを使用すると、これらの異なる要素 (条件とアクション) を組み合せることができます。 これによって、パーソナライゼーションやエンゲージメント プランなどで使用可能なルールを作成することができます。

#### 1.1.1 条件

それぞれの条件には、条件が真かどうかを決定するロジックが含まれています。たとえば、アイテムをロックしている場合、「条件: (ワークフロー)アイテムがあなたにロックされている場合]は真になります。

条件は、条件定義アイテムと、その条件を実装する .NET クラスによって構成されます。

#### 条件のテキスト

ユーザー インターフェースで条件を選択する際に表示されているテキストは、条件定義アイテムの [データ] セクションの [テキスト] フィールドに含まれます。条件をルールの一部として選択する場合、条件のなかに は、変更可能なパラメーターで置き換えられるトークンを含むものがあります。

選択した条件の [条件:] はリンク コマンドとなり、そのリンクをクリックすると [条件:] と [除外条件:] が切り替わります。





#### 大王

[条件:] と [除外条件:] を切り替えると、ルール エンジンは自動的に条件ロジックの判定結果を置換えます。独自の条件を実装する場合、コードの中でこの切り替えを意識する必要はありません。

条件のテキストの角括弧[]のトークンによって、条件用にパラメーターの指定が可能なリンク コマンドが有効になります。

```
    最 力 20 ム
    □ プータ
    □ は存時期
    ○ 毎月
    ② ● 周週
    ③ ● 八年の月
    ② ● 次
    ② ● 次
    ② ● 次
```

角括弧には4つのカンマで区切られたパラメーターが含まれます。

- 1. 条件を実装する .NET クラスにパラメーターをセットする際のプロパティ名
- 2. /Sitecore/System/Settings/Rules/Definitions/Macros フォルダーにある定義済み機能のアイテムの名前、または空の文字列。たとえば、ユーザーが整数を選択できるユーザー インターフェースを起動するには、[Integer] のマクロを指定します
- 3. 2 のパラメーターで指定されたマクロにより起動されたユーザー インターフェースへの URL パラメーター、または空の文字列。たとえば、/Sitecore/Content/Home アイテムを選択するツリーのルートとして指定するには、root=/sitecore/content/home と入力します
- 4. ユーザーがパラメーターの値を指定していない場合に表示されるテキスト

#### ヒント

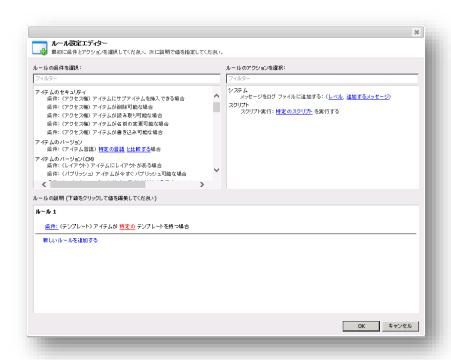
[テキスト] フィールドの構文に関する追加情報は、既存の条件定義アイテムを参照してください。

たとえば、アイテムが特定のデータ テンプレートに関連付けられている場合、[条件: (テンプレート) アイテムが特定のテンプレートを持つ場合] 条件は、真になります。以下はこの条件のテキストです。

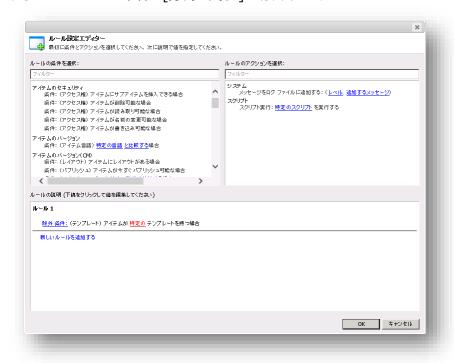
条件: (テンプレート) アイテムが [templateid, Tree, root=/sitecore/templates, 特定の] テンプレートを持つ場合



条件の選択とパラメーターの編集を行う [ルール設定エディター] で、Sitecore は [条件] と角括弧の中のトークンを翻訳し、テキストを次の画像例のようにレンダリングします。



[**ルールの説明**] フィールドで、テンプレートの選択と条件を切り替えることができます。[**条件:**] をクリック すると Sitecore は条件を「**除外 条件:**]に切り替えます。

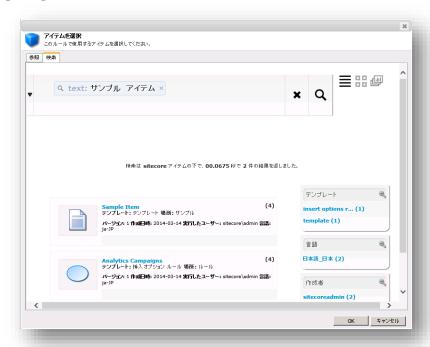




[**特定の**] をクリックするとデータ テンプレートを選択するダイアログが表示されます。



[検索] タブを使って使用するアイテムを検索し指定することもできます。



このダイアログ ボックスでの選択は、条件を実装するクラスの TemplateID プロパティをコントロールします。 条件パラメーターの定義に関する追加情報は、ルール を参照してください。



#### 条件の実装方法

条件を実装するには、次の手順に従います。

1. Visual Studio プロジェクトで、次から継承を行って条件クラスを作成します。

条件	機能
Sitecore.Rules.Conditions.WhenCondition,	ブール条件
Sitecore.Rules.Conditions.StringOperatorCondition	文字列値の比較
Sitecore.Rules.Conditions.OperatorCondition,	その他の値の種類の比較

#### ヒント

作成する手間を省くには、同様の機能を持つ既存の条件を複製します。

- 2. 条件付きクラスで、Execute()メソッドを実装し、条件の真偽を返します。
- 3. **コンテンツ エディター**で条件定義アイテムを作成する場所の親アイテムを選択します。 条件は、sitecore/system/Settings/Rules/Definitions/Elements アイテム配下に作成します。
- 4. /sitecore/templates/System/Rules/Condition データ テンプレートに基づく条件定義アイテム を挿入します。条件定義アイテムには、.NET から引き継いだ名前を付けます。
- 5. 条件のテキスト セクションの記述にあるように、条件定義アイテムの [データ] セクションの [テキスト] フィールドに、条件のテキストを入力します。
- 6. 条件定義アイテムの [スクリプト] セクションの [タイプ] フィールドに、.NET 条件クラスのシグニチャーを入力します。

これで、この条件は条件付きレンダリング ルールの作成時に使用できます。ルールの条件の設定についての追加情報は、条件付きレンダリング ルール セクションを参照してください。

## 1.1.2 アクション

ルールでひとつ以上の条件がすべて真である場合に実行されるロジックが含まれます。たとえば、「**挿入オプションの追加**」アクションは、アイテムをアイテムの挿入オプションに追加します。

アクションはアクション定義アイテムと、アクションを実装する .NET クラスによって構成されます。

#### **JE**

ユーザー インターフェースがない場合は、アクションはスクリプトとして機能することがあります。追加情報は、 スクリプト セクションを参照してください。スクリプトがアクションとして機能することもあります。



#### アクション テキスト

ユーザーがアクションの選択とアクション パラメーターの入力が可能なユーザー インターフェースで、 Sitecore はアクション定義アイテムの [データ] セクションの [テキスト] フィールドの値を使用します。 テキスト フィールドのフォーマットについての追加情報は、条件のテキスト を参照してください。

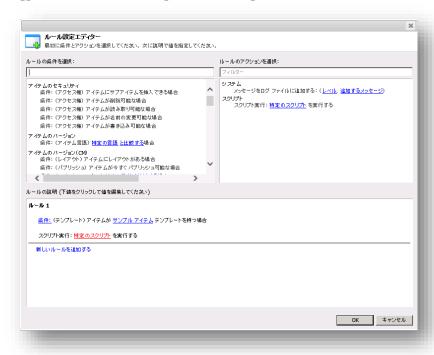
#### ヒント

[テキスト] フィールドの構文についての追加情報は、既存のアクション定義アイテムを確認してください。

たとえば、スクリプト実行アクションは次のスクリプトを起動します。

スクリプト実行: [scriptid, Script, , 特定のスクリプト] を実行する

ユーザーがアクション パラメーターを入力することが可能なユーザー インターフェースで、Sitecore は角括弧[]内のトークンを翻訳し、[**ルールの説明**] フィールドに表示されているテキストをレンダリングします。





ルール パラメーターの定義についての追加情報は、ルール を参照してください。[特定のスクリプト] リンクコマンドをクリックすると、スクリプトを選択できる [**スクリプトを選択**] ダイアログボックスが開きます。



スクリプトを選択すると、このダイアログはアクションを実装するクラスの ScriptID プロパティをコントロールします。

#### 大王

スクリプトをアクションとして設定できます。

スクリプトについての追加情報は、スクリプトセクションを参照してください。

#### アクションの実装方法

アクションを実装するには、次の手順に従います。

- 1. Visual Studio プロジェクトで、Sitecore.Rules.Actions.RuleAction<T> クラスから継承するクラスを作成し、Apply()メソッドを実装します。
- 2. コンテンツ エディターで、/sitecore/system/Settings/Rules/Definitions/Elements フォルダーに 移動し、アクション定義アイテムを保存する適切な親要素アイテムを選択します。



#### 大王

アクションは、sitecore/system/Settings/Rules/Definitions/Elements アイテムの配下に作成する必要があります。

- 3. System/Rules/Action データ テンプレートを使って、アクション定義アイテムを挿入します。アクション定義アイテムには、アクションを実装する.NET クラスから引き継いだ名前をつけます。
- 4. アクション定義アイテムの [データ] セクションの [テキスト] フィールドに、アクションのテキストを入力します。

アクション テキストについての追加情報は、アクション テキスト セクションを参照してください。

- 5. アクション定義アイテムの [スクリプト] セクションの [タイプ] フィールドに、アクションを実装する .NET クラスのシグニチャーを入力します。
- 6. 条件付きレンダリング ルール セクションの記述にあるように、ひとつ以上のルール定義アイテムを設定して、アクション定義アイテムを参照します。

これで様々な種類のルールでこのアクションを使用することができます。

## 1.1.3 スクリプト

スクリプトには特定の機能を実装するロジックが含まれます。スクリプトは、インライン コードを含むスクリプト 定義アイテムによって構成されます。または、そのスクリプトを実装する .NET クラスを参照します。

スクリプト定義アイテムのスクリプトセクションには次のフィールドが含まれます。

- **タイプ** .NET クラス シグネチャー (Namespace.Class, assembly)
- **コード** .NET フレームワークがサポートする全ての言語でのインライン コード
- 参照情報 (リファレンス) カンマ (",") によって区切られた、参照用の .NET 構築のリスト
- **言語** .NET 言語の指定子。たとえば C# の CSharp など。

#### 重要

値を [タイプ] フィールドに入力するか、または [コード]、[参照情報 (リファレンス)]、[言語] フィールド に入力します。値を4つのフィールドすべてに入力することはできません。

ほとんどの場合、スクリプトの代わりにアクションを使用することができます。

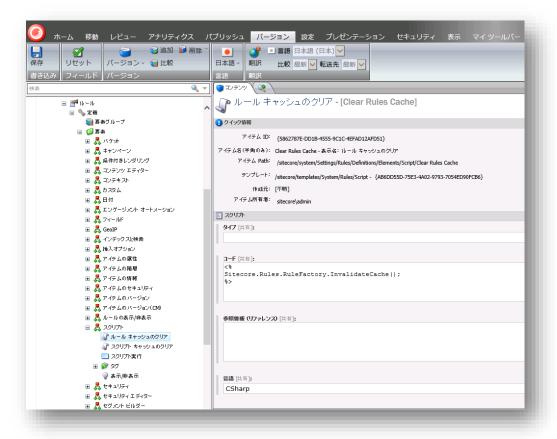
これらのフィールドが別のデータテンプレートに表示されると、このセクションで説明した機能が提供されます。

[コード] フィールドの値が <% で始まり %>で終わる場合は、Sitecore ではそのコードを次のようにワードラップします。



```
namespace Sitecore
{
   public class DefaultClass
   {
     public void DefaultMethod()
     {
        // <% %> の中身をここに記述します
     }
   }
}
```

たとえば、[**ルール キャッシュのクリア**] スクリプトはルール エンジンに関連付けられたすべてのキャッシュをクリアします。



アクションについての追加情報は、アクションセクションを参照してください。

#### スクリプトの実装方法

スクリプトを実装するには、次の手順に従います。

- 1. **コンテンツ エディター**で、/sitecore/system/Settings/Rules/Definitions/Elements フォルダー に移動し、スクリプト定義アイテムを保存する場所のアイテムを選択します。
  - スクリプト定義アイテムはこのフォルダーに保存する必要があります。
- 2. System/Rules/Script データ テンプレートを使ってスクリプト定義アイテムを挿入します。
- 3. スクリプトには、そのスクリプトの機能が明確に分かる名前を付けます。



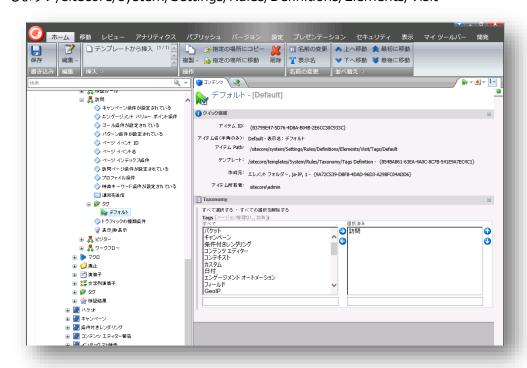
4. スクリプト定義アイテムの [**タイプ**] フィールドに、.NET クラス シグネチャーを入力します。または、 [**コード**]、[**参照情報 (リファレンス)**]、[**言語**] フィールドに値を入力します。

#### 1.1.4 ルール要素の表示場所の指定

条件またはアクションを作成する場合は、その表示場所を決める必要があります。[**ルール設定エディタ**ー]には、各要素を適切なコンテキストにのみ表示する必要があります。

要素が条件付きレンダリングに関連する場合は、その要素は適切な場所にだけ表示される必要があります。同様に、訪問に関連する条件またはアクションの場合は、Web サイトの訪問に関するルールを作成するために、このルール要素を使用するのに適したコンテキストにのみ表示させます。

ルール要素の構築を容易にするために、Sitecore ではルール要素を要素の種類ごとにコンテンツ ツリー に分類します。たとえば、訪問者に関するすべてのアクションと条件は、コンテンツツリーの次の場所に分類します:/sitecore/system/Settings/Rules/Definitions/Elements/Visit



上記の例では、[訪問] タグは要素カテゴリーのデフォルト タグです。これは、訪問を評価するルールを設定する場合、[ルール設定エディター] にこれらのルール要素が表示されることを意味します。

ルールを作成するために使われる要素 (アクションと条件) を作成した後、それらの表示場所を指定する 必要があります。

表示場所を指定するには、次の操作が可能です。

既存のタグを使用する

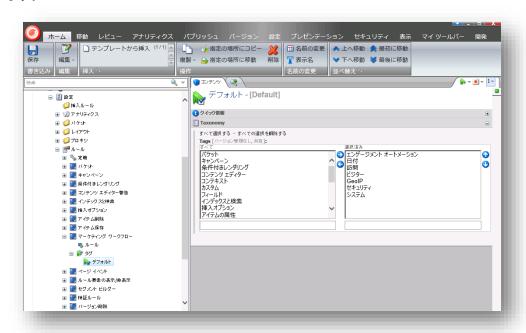


新しいタグを作成する

既存のタグを使用するには、適切な要素フォルダーの下にルール要素を保存します。

新しいタグを作成して、ルールに割り当てるには、次の手順に従います。

- 1. /sitecore/system/Settings/Rules/Definitions/Tags アイテムに移動します。
- 2. [ホーム] タブの [挿入] グループで、「タグ] をクリックして、新しいタグに適切な名前をつけます。
- 3. /sitecore/system/Settings/Rules/Definitions/Elements フォルダーで、このタグを新しいルール用に作成したルール要素カテゴリーに追加するか、または現存するルール カテゴリーに追加します。
- 4. /sitecore/system/Settings/Rules フォルダーで、適切なルールのコンテキスト フォルダーを選択します。



5. このルール コンテキスト アイテムの Tags/Default アイテムで、新しいタグを [選択済み] フィールドに追加します。

これで新しいタグはルールにリンクされました。

追加条件を満たす場合にのみルール要素を表示するように、表示のルールを実装することもできます。 表示ルールは、ルール タグとルール要素カテゴリーに割り当てることができます。

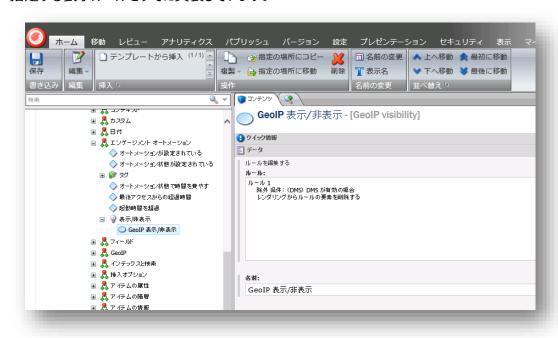
ルール要素カテゴリーに表示ルールを実装するには、次の手順に従います。

- 1. /sitecore/system/Settings/Rules/Definitions/アイテムに移動します。
- 2. 適切なルール要素カテゴリーを展開し、[表示/非表示] アイテムを選択します。



- 3. [ホーム] タブの [挿入] グループで、[Rules Element Visibility Rule]をクリックし、ルールに適切な名前をつけます。
- 4. Rules Element Visibility Rule アイテムを選択し、[データ] セクションの [ルール] フィールドで、 [ルールを編集する] をクリックします。
- 5. [ルール設定エディター]で、実装する表示ルールを作成します。

たとえば、Sitecore インストレーションでアナリティクスが有効な場合にのみ特定のルールを表示するように 指定する表示ルールをすでに実装しています。





#### 1.2 ルール

ルールを使用することによって、ひとつ以上のアクションをひとつ以上の条件に割り当てることができます。アクションと条件の両方のパラメーター バリューも含まれます。たとえば、条件付きレンダリング ルールは、多変量テストが有効な場合 (条件)、多変量テストの処理を行います (アクション)。 つまり、多変量テストが有効な限り、多変量テストの処理を行います。

and と or などの演算子を使って、特定の条件の組み合せによりアクションを起動することができます。

#### ルールの種類

挿入オプション、挿入ルール、uiGetMasters パイプラインに加え、挿入ルールを実装して有効な挿入オプションを実行時にコントロールすることができます。挿入オプション ルールは、

/Sitecore/System/Settings/Rules/Insert Options/Rules アイテムの下に設定します。

挿入オプションについての追加情報は、Data Definition Reference マニュアルを参照してください。

#### УŦ

Sitecore ではルールを起動するためにイベント ハンドラを使用します。イベントのルールをいつ実行するかを決定するために、対応するイベント ハンドラを確認します。

- ◆ 特定のアイテムに対して挿入オプションのロジックを選択する必要がある場合は、挿入ルールを使用します。
- すべてのアイテムに適用する挿入オプション ロジックに uiGetMasters パイプラインを使用します。
- 管理者による可変ロジックを使った挿入オプションの設定を可能にするには、挿入オプション ルールを 使用します。

item:deleting、item:deleted イベント、uiDeleteItems パイプラインに加え、アイテム削除ルールを実装して、ユーザーが特定の基準とマッチングするアイテムを削除した後に起動するアクションを設定することができます。アイテム削除ルールは、/Sitecore/System/Settings/Rules/Item Deleted/Rules アイテムの下に設定します。

item:saving、item:saved イベント、saveUI パイプラインに加え、アイテム保存ルールを実装して、ユーザーが特定の基準にマッチングするアイテムを保存した後に起動するアクションを設定することができます。アイテム保存ルールは、/Sitecore/System/Settings/Rules/Item Saved/Rules アイテムの下に設定します。

item:versionRemoving、item:versionRemoved イベント、deleteVersionsUI パイプラインに加え、バージョン削除ルールを実装して、ユーザーが特定の基準にマッチングするアイテムのバージョンを削



除した後に起動するアクションを設定することができます。バージョン削除ルールは、 /Sitecore/System/Settings/Rules/Version Removed/Rules アイテムの下に設定します。

/sitecore/system/Settings/Validation Rules アイテムの下に保存されている検証ルールに加え、Sitecore Rules Engine を使ってアイテムを検証する検証アイテム ルールを実装することができます。 検証アイテム ルールは、/sitecore/system/Settings/Rules/Validation Rules/Rules アイテムの下に設定します。

Sitecore では条件付きレンダリングに異なるデータ テンプレートを使用しますが、同じ原則に従います。 条件付きレンダリングについての追加情報は、条件付きレンダリング セクションを参照してください。

#### ルールの実装方法

ルールを実装するには、次の手順に従います。

1. コンテンツ エディターで、配下にルール定義アイテムを保存するアイテムを選択します。

削除の処理を行うルール定義を、/Sitecore/System/Settings/Rules/Item Deleted/Rules アイテムの下に保存します。

保存の処理を行うルール定義を、 /Sitecore/System/Settings/Rules/Item Saved/Rules アイテムの下に保存します。

バージョン削除の処理を行うルール定義を、/Sitecore/System/Settings/Rules/Version Removed/Rules アイテムの下に保存します。

アイテムの検証を行うルール定義を、/sitecore/system/Settings/Rules/Validation Rules/Rules アイテムの下に保存します。

条件付きレンダリングの処理を行うルール定義を、/Sitecore/System/Marketing Center/Personalization/Rules アイテムの下に保存します。

グローバル条件付きレンダリング ルールを、/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules アイテムの下に保存します。

その他のルール定義アイテムは、/Sitecore/System/Settings/Rules アイテムの適切な子孫の下に保存します。

コンテンツ エディターで、適切なテンプレートに基づくルール定義アイテムを挿入します。
 次のテンプレートに基づくルールを作成することができます。



ルール	Template テンプレート
条件付きレンダリング ルール	/sitecore/templates/System/Rules/ Conditional Rendering Rule データ テンプレート
Content Editor Warning rule	/sitecore/templates/System/Rules/Content Editor Warning Rule データテンプレート
挿入オプション ルール	/sitecore/templates/System/Rules/Insert Options Rule データ テンプレート
検証ルール	/sitecore/templates/System/Rules/Validation Rule データ テンプレート
その他	/sitecore/templates/System/Rules/Rule データ テンプレート

- 3. ルール定義アイテムの [データ] セクションの [名前] フィールドに、ユーザー インターフェースに表示するルールの名前を入力します。
- 4. ルール定義アイテムの [データ] セクションの [ルール] フィールドで、条件付きレンダリング条件とアクションを選択します。条件とアクションの両方にパラメーターを入力します。



# Chapter 2 条件付きレンダリング

この章では、実行時のプレゼンテーションコンポーネントの操作を可能にする条件付きレンダリングの実装方法について説明します。この章には条件付きレンダリングの模範例が含まれています。

この章には次のセクションがあります。

- 条件付きレンダリング
- 条件付きレンダリングの実装
- 条件付きレンダリングの適用



## 2.1 条件付きレンダリング

条件付きレンダリングとは、Sitecore レイアウト エンジンの機能を意味し、HTTP リクエストを提供するために使用されるプレゼンテーション コントロールを操作します。

- レンダリングの組込みと除外
- プレースホルダーの制御
- データ ソースとその他プロパティの設定
- 多変量条件の処理
- その他ロジック

Sitecore では実行時のパーソナライゼーションをサポートします。これによって、条件付きレンダリングを使って、シングルセッション中に Web サイトの訪問者に表示するコンテンツをパーソナライズすることができます。

#### 重要

実行時のパーソナライゼーションを使用するために DMS を作動させる必要はありません。



# 2.2 条件付きレンダリングの実装

このセクションでは、条件付きレンダリングの実装方法について説明します。

## 2.2.1 条件付きレンダリングの条件

条件付きレンダリングの条件には、条件が真かどうかを決定し、それによって条件付きレンダリング アクションを起動するかどうかを決定するロジックが含まれます。レイアウト詳細は条件付きレンダリングの条件を参照して、条件付きレンダリングルールを参照できます。

#### ヒント

条件付きレンダリング セクションで説明した条件を、条件付きレンダリング条件として使用することができます。

条件付きレンダリングの条件についての追加情報は、条件 を参照してください。

### 2.2.2 条件付きレンダリング アクション

条件付きレンダリング アクションには、レンダリングの除外またはプロパティの設定など、条件付きレンダリング機能を実装するロジックが含まれます。条件付きレンダリング アクションとは、アクション セクションで説明 したようなアクションです。レイアウト詳細は条件付きレンダリングを参照しますが、同様に条件付きレンダリング アクションを参照します。

条件付きレンダリング アクションは、/sitecore/system/Settings/Rules/Definitions/Elements フォルダーの適切な要素の下に保存されます。

#### 条件付きレンダリング アクションの実装方法

条件付きレンダリング アクションを実装するには、次の手順に従います。

- 1. Visual Studio プロジェクトで、Sitecore.Rules.Actions.RuleAction<T>クラスから継承するクラスを作成し、Apply()メソッドを実装します。
- コンテンツ エディターで、
   /sitecore/system/Settings/Rules/Definitions/Elements/Conditional Renderings/フォルダーに移動します。
- 3. /sitecore/templates/System/Rules/Action データ テンプレートに基づくアクション定義アイテムを作成します。

条件付きアクションの定義アイテムには、条件付きレンダリング アクションを実装する .NET クラスから引き継いだ名前をつけます。



4. 条件付きアクションの定義アイテムの [データ] セクションの [テキスト] フィールドに、適切なパターンを入力します。

テキストフィールドについての追加情報は、アクションテキストを参照してください。

- 5. 条件付きアクションの定義アイテムの [スクリプト] セクションの [タイプ] フィールドに、条件付きレン ダリング アクションを実装する .NET クラスのシグニチャーを入力します。
- 6. 条件付きレンダリング ルール セクションの記述にあるように、ひとつ以上の条件付きレンダリング ルールに条件付きレンダリング アクションを設定します。
- 7. 条件付きレンダリングの適用 セクションの記述にあるように、条件付きレンダリング ルールを適用します。

#### 例: 条件付きレンダリングを使ったレンダリングの追加

レンダリングをプレースホルダーに追加する次の例に基づいた条件付きレンダリング アクションを実装することができます。

```
namespace Sitecore.Sharedsource.Rules.ConditionalRenderings
  using System;
  public class AddRenderingAction<T> : Sitecore.Rules.Actions.RuleAction<T>
   where T: Sitecore.Rules.ConditionalRenderings.ConditionalRenderingsRuleContext
   public string Placeholder
      get;
      set;
   public string RenderingID
      get;
      set;
   public override void Apply(T ruleContext)
      Sitecore.Diagnostics.Assert.IsNotNullOrEmpty(
        this.RenderingID,
        "RenderingID");
      Sitecore.Data.Items.Item rendering =
        Sitecore.Context.Database.GetItem(this.RenderingID);
      Sitecore.Diagnostics.Assert.IsNotNull(rendering, "rendering");
      Sitecore.Layouts.RenderingReference rendRef
        new Sitecore.Layouts.RenderingReference(rendering);
      if (!String.IsNullOrEmpty(this.Placeholder)
        && String.Compare(this.Placeholder, "specific", true) != 0)
        rendRef.Placeholder = this.Placeholder;
      ruleContext.References.Add(rendRef);
```



アクション定義アイテムに、以下のテキストを入力して、ユーザーによるレンダリングの選択とプレースホルダー キーの入力を可能にします。

[RenderingID, Tree, root=/sitecore/layout/renderings, 特定の] レンダリングに [Placeholder, Text, , 特定の] プレースホルダーを追加する

## 2.2.3 条件付きレンダリング ルール

Sitecore では条件付きレンダリング ルールを起動して、条件付きレンダリング ロジックをコントロールします。条件付きレンダリング ルールは、ひとつ以上の条件付きレンダリング アクションをひとつ以上のレンダリング条件に関連付けます。アクションと条件の両方のパラメーター バリューが含まれます。

条件付きレンダリング ルールは、System/Rules/Conditional Rendering Rule データ テンプレートに基づきます。



## 2.3 条件付きレンダリングの適用

このセクションでは、条件付きレンダリングの適用方法について説明します。

## 2.3.1 条件付きレンダリングの適用方法

条件付きレンダリングを、データテンプレートのスタンダードバリューのレイアウト詳細に設定します。

レイアウト詳細についての追加情報は、*Presentation Component Reference* マニュアルを参照してください。

#### 2.3.2 グローバル 条件付きレンダリング ルール

各レンダリングで、レイアウト詳細で選択された条件付きレンダリング ルールを評価することに加えて、
Sitecore では/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules アイテム
の下で定義されたグローバル条件付きレンダリング ルールの評価も行います。

#### 重要

レンダリングとサブレイアウトをそれぞれ実行するため、特にレイアウト詳細で指定されたレンダリングの数が 増加するにつれて、グローバル条件付きレンダリング ルールのパフォーマンスを最適化することが重要になり ます。

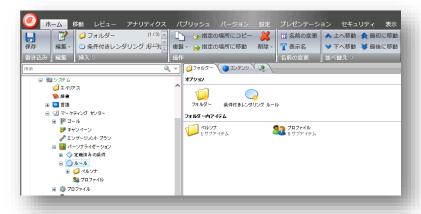
ルールの実装方法 セクションの記述にあるように、グローバル条件付きレンダリング ルールを定義するには、 条件付きレンダリング ルールを/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules アイテムの下に実装しますが、System/Rules/Conditional Rendering Rule データテンプレート を使用します。

#### 個別のレンダリングの条件付きルールの作成

独自のレンダリングに条件付きレンダリングルールを作成するには、次の手順に従います。



1. コンテンツ エディターで、/sitecore/system/Marketing Center/Personalization/Rules アイテム に移動します。

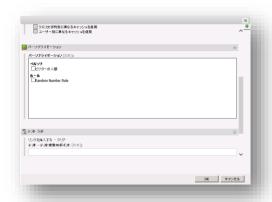


- 2. [条件付きレンダリング ルール] をクリックして、条件付きレンダリング ルールに適切な名前を付けます。
- 3. 条件付きルール アイテムの [ルール] フィールドで、[**ルールを編集する**] をクリックして、このルールで使用する条件とアクションを選択します。

条件付きレンダリングルールを独自のレンダリングに割り当てるには、次の手順に従います。

- 1. コンテンツ エディターの左上端にある [Sitecore]、[アプリケーション オプション] をクリックします。
- 2. [**アプリケーション オプション**] ダイアログ ボックスで、[表示] タブをクリックし、[コントロール プロパティ] セクションで次のチェックボックスを選択します。
- パーソナライゼーション セクションの表示
- テスト ラボ セクションの表示

[レイアウト詳細] ダイアログボックスを開き、独自のレンダリングの [プロパティ] ダイアログ ボックスを開くと、次のフィールドが表示されます。



3. パーソナライゼーション ルールまたは使用する条件付きレンダリングを選択します。



# **Chapter 3**

# 条件付きレンダリング ルールの実装

この章には、条件付きレンダリングの実装方法についての模範例が含まれます。

条件付きレンダリング ルールの適用方法はもちろん、条件、アクション、ルールの作成方法についても説明します。

この章には次のセクションがあります。

• 条件付きレンダリングの模範例



# 3.1 条件付きレンダリングの模範例

模範例を用いて条件付きレンダリングルールの作成方法を説明します。

この模範例では、ユーザーが特定のページを訪れるたびに、コンピューターによって乱数が生成されます。その乱数が定義済みの数字とマッチングした場合、そのユーザーは当選する仕組みです。

#### 乱数の条件付きレンダリングの条件を作成

この模範例を作成するには、次のことを行います。

- 乱数の上限を指定します。
- 乱数と一致する当選番号を指定します。

乱数が指定されたパラメーターバリューと一致する場合は、次の例は真になります。

```
namespace Sitecore.Sharedsource.Rules.Conditions
 using System;
  [UsedImplicitly]
 public class RandomNumberCondition<T> :
   Sitecore.Rules.Conditions.OperatorCondition<T>
   where T : Sitecore.Rules.RuleContext
   private int upperLimit;
   public int MatchNumber
     get;
     set;
   public int UpperLimit
     get
        Sitecore.Diagnostics.Assert.IsTrue(this.upperLimit > 1, "upper limit");
       return this.upperLimit;
      set
       Sitecore.Diagnostics.Assert.IsTrue(value > 1, "value");
       this.upperLimit = value;
   protected override bool Execute(T ruleContext)
      Sitecore.Diagnostics.Assert.ArgumentNotNull(ruleContext, "ruleContext");
     int random = new Random(DateTime.Now.Millisecond).Next(this.UpperLimit + 1);
      switch (this.GetOperator())
        case Sitecore.Rules.Conditions.ConditionOperator.Equal:
         return random == this.MatchNumber:
       case Sitecore.Rules.Conditions.ConditionOperator.GreaterThanOrEqual:
          return random >= this.MatchNumber;
        case Sitecore.Rules.Conditions.ConditionOperator.GreaterThan:
          return random > this.MatchNumber;
        case Sitecore.Rules.Conditions.ConditionOperator.LessThanOrEqual:
         return random <= this.MatchNumber;</pre>
```



```
case Sitecore.Rules.Conditions.ConditionOperator.LessThan:
    return random < this.MatchNumber;
    case Sitecore.Rules.Conditions.ConditionOperator.NotEqual:
        return random != this.MatchNumber;
}

return false;
}
</pre>
```

乱数の条件付きレンダリングの条件を作成したので、次は実装を行います。

#### 条件の実装

条件付きレンダリング条件を実装するには、次の手順に従います。

1. Visual Studio プロジェクトで、次から継承を行って条件付きクラスを作成します。

条件	機能
Sitecore.Rules.Conditions.WhenCondition,	ブール条件
Sitecore.Rules.Conditions.StringOperatorCondition	文字列の比較
Sitecore.Rules.Conditions.OperatorCondition,	その他の値の種 類の比較

#### ヒント

作成の手間を省くには、同様の機能を持つ現存する条件を複製します。

- 2. 条件クラスで、Execute()メソッドを実行して、条件の真偽を示します。
- 3. コンテンツ エディターで、配下に条件定義アイテムを保存するアイテムを選択します。 条件は sitecore/system/Settings/Rules/Definitions/Elements アイテムの適切な子孫の下に保存します。
- 4. /sitecore/templates/System/Rules/Condition データ テンプレートに基づく条件付き定義アイテムを挿入します。条件定義アイテムには、条件を実装する .NET クラスから引き継いだ名前をつけます。
- 5. 条件のテキスト セクションの記述にあるように、条件付き定義アイテムの [データ] セクションの [テキスト] フィールドに、テキストを入力します。
- 6. 条件定義アイテムの [スクリプト] セクションの [タイプ] フィールドに、.NET 条件クラスのシグニチャーを入力します。

これで作成するルールでこの条件付きレンダリングを使用することができます。ルールの条件の設定についての追加情報は、条件付きレンダリング ルール セクションを参照してください。



コンテンツ エディターで、条件付きレンダリングの定義アイテムの [データ] セクションの [テキスト] フィールド に、次のテキストを入力します。これによってユーザーは、条件付きレンダリング条件の UpperLimit と MatchNumber パラメーターを設定し、等号演算子 ("=") などの演算子が選択できるようになります。

```
[条件] 1 から [UpperLimit,,,最大値] までのランダムな数が
[MatchNumber,,,特定の数] と [operatorid,Operator,,比較] する場合
```

条件付きレンダリング条件のベース クラス(Sitecore.Rules.Conditions.OperatorCondition<T>) は、 OperatorId プロパティを定義します。条件付きレンダリング条件のクラス

(Sitecore.Sharedsource.Rules.Conditions.RandomNumberCondition) は、UpperLimit と MatchNumber プロパティを定義します。

#### メモ

この模範例で説明する厳格な要求を満たすには、ユーザーによるオペレータの選択を許容するのではなく、 比較ロジックをハードコードする必要があります。

#### この条件付きレンダリング条件を

/sitecore/system/Settings/Rules/Definitions/Elements/Conditional Renderings フォルダーに保存します。 こうすることで、タグ付けが正しく行われることを確実にします。

#### 条件付きレンダリング アクションの作成

レンダリングの IsWinner プロパティを設定する次の模範例に基づいて、条件付きレンダリング アクションを 実装することができます。

```
namespace Sitecore.Sharedsource.Rules.ConditionalRenderings
 using System;
 using Sitecore.Collections;
  [UsedImplicitly]
 public class SetWinner<T> : Sitecore.Rules.Actions.RuleAction<T>
   where T : Sitecore.Rules.ConditionalRenderings.ConditionalRenderingsRuleContext
   public bool IsWinner
     get;
     set;
   public override void Apply(T ruleContext)
     string key = "winner";
     if (String.IsNullOrEmpty(ruleContext.Reference.Settings.Parameters))
        ruleContext.Reference.Settings.Parameters =
         key + "=" + this.IsWinner.ToString();
        return;
      SafeDictionary<string> parameters = Sitecore.Web.WebUtil.ParseQueryString(
       ruleContext.Reference.Settings.Parameters);
      foreach (string check in parameters. Keys)
        if (String.Compare(check, key, true) == 0)
```



```
{
    key = check;
    break;
}

parameters[key] = this.IsWinner.ToString();
ruleContext.Reference.Settings.Parameters =
    Sitecore.Web.WebUtil.BuildQueryString(parameters, false);
}
}
```

乱数がマッチングした場合は、条件付きレンダリング アクションの定義アイテムの [テキスト] フィールドの [データ] セクションで、ユーザーが IsWinner プロパティの値を指定できるようにします。

```
値を [IsWinner,Text,,値] に設定する (true または false)
```

#### 条件付きレンダリング アクション クラス

(Sitecore.Sharedsource.Rules.ConditionalRenderings.SetWinner) は、IsWinner プロパティを定 義します。

#### この条件付きレンダリング アクションを

/sitecore/system/Settings/Rules/Definitions/Elements/Conditional Renderings フォルダーに保存します。 こうすることで、タグ付けが正しく行われることを確実にします。

#### 乱数の条件付きレンダリング ルールの作成

乱数を生成する条件付きレンダリング ルールを挿入する必要があります。ルール定義アイテムを Pick a Random Winner とします。

 グローバル条件付きレンダリング ルールを作成するには、コンテンツエディターで、 /Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules アイテムを選択します。

独自の条件付きレンダリング ルールを作成するには、コンテンツ エディターで、 /Sitecore/System/Settings/ Marketing Center/Personalization/Rules アイテムを選択しま す。

- 2. [ホーム] タブの [挿入] グループで、[条件付きレンダリング ルール] をクリックして、
  /sitecore/templates/System/Rules/Conditional Rendering Rule データ テンプレートに基づ
  くルールを作成します。
- 3. ルール定義アイテムの [データ] セクションの [ルール] フィールドで、[**ルールを編集する**] をクリックします。
- 4. [ルール設定エディター] の [ルールの条件を選択]フィールドで、[パラメーターを 特定の値 に設定する] 条件を選択します。



- 5. [**ルールのアクションを選択**] フィールドで、「値を [IsWinner,Text,,値] に設定する (true または false)」 アクションを選択します。
- 6. [**ルールの説明**] フィールドで、この条件付きレンダリング ルールに実装する条件とアクションのパラメーターを選択します。
- 7. 条件付きレンダリング ルールの定義アイテムの [データ] セクションの [ルール] フィールドで [ルール のアクションを選択] の下から、[user wins] を選択して、指定された条件が真の場合に実行されるアクション リストにアクションを追加します。

グローバル条件付きレンダリング ルールを作成中の場合は、自動的にすべてのアイテムに割り当てられます。

独自の条件付きレンダリング ルールを作成中の場合は、このルールを適切なレンダリングに割り当てる必要があります。

8. 独自の条件付きレンダリング ルールを割り当てるには、適切なデータ テンプレートのスタンダード バリューのレイアウト詳細を開いて、ルールを割り当てます。

これで、訪問者がこのページを見るたびに乱数が生成されます。生成された乱数が、定義済みの 当選番号と一致するかどうかが評価されます。一致する場合は、当選になります。