



# Sitecore CMS 6.1

# 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開発者向けヒントとテクニック

## 目次

Chapter 1	イントロダクション .....	3
Chapter 2	Sitecore ルール エンジン .....	4
2.1	ルール エンジンの概要 .....	5
2.1.1	スクリプト .....	5
	スクリプトの実装方法 .....	7
2.1.2	条件 .....	7
	デフォルト条件 .....	7
	条件テキスト .....	8
	条件の実装方法 .....	11
2.1.3	アクション .....	12
	デフォルトのアクション .....	12
	アクション テキスト .....	13
	アクションの実装方法 .....	16
2.1.4	ルール .....	16
	ルールの種類 .....	17
	ルールの実装方法 .....	17
Chapter 3	条件付きレンダリング .....	19
3.1	条件付きレンダリングの概要 .....	20
3.2	条件付きレンダリング例の要件 .....	21
3.3	条件付きレンダリングの実装 .....	22
3.3.1	条件付きレンダリング アクション .....	22
	デフォルトの条件付きレンダリング アクション .....	22
	条件付きレンダリング アクションの実装方法 .....	22
	例: 条件付きレンダリング アクションへのレンダリング プロパティの設定 .....	23
	例: レンダリングを追加する条件付きレンダリング アクション .....	24
3.3.2	条件付きレンダリング条件 .....	25
	条件付きレンダリング条件の実装方法 .....	25
	例: 乱数一致の条件付きレンダリング条件 .....	25
3.3.3	条件付きレンダリング ルール .....	26
	例: 乱数一致の条件付きレンダリング ルール .....	27
3.4	条件付きレンダリングの適用 .....	28
3.4.1	条件付きレンダリングを適用する方法 .....	28
3.4.2	グローバルな条件付きレンダリング ルール .....	28

# Chapter 1

## イントロダクション

Sitecore の管理者と開発者は、Sitecore ルール エンジンを使用する前に、このマニュアルをよくお読みください。ルール エンジンは、挿入オプションの動的な操作とシステム イベントの処理を行うために使用できます。また、条件付きレンダリングのためにも使用できます。条件付きレンダリングは、パーソナリゼーション、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も呼ばれます。

このマニュアルには次の章があります。

- Chapter 1 - イントロダクション
- Chapter 2 - Sitecore ルール エンジン
- Chapter 3 - 条件付きレンダリング

## Chapter 2

# Sitecore ルール エンジン

この章では、ルール エンジンの概要、スクリプト、条件、アクション、ルールなど、Sitecore ルール エンジンをサポートする概念について説明します。

この章には次のセクションがあります。

- ルール エンジンの概要
- スクリプト
- 条件
- アクション
- ルール

## 2.1 ルール エンジンの概要

Sitecore ルール エンジンは、アクションで定義されているロジックを、特定の条件が True と評価されたときに適用します。ルール エンジンには、挿入オプションおよび動的応答エンジン（条件付きレンダリングとも呼ばれる）などの Sitecore 機能を制御するためのユーザーインターフェースが用意されています<sup>1</sup>。

### 2.1.1 スクリプト

スクリプトには、特定の関数に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で構成されるか、スクリプトを実装する .NET クラスを参照します。

1. スクリプト定義アイテムの [データ] セクションには、次のフィールドがあります。
  - [種類] - .NET クラスのシグネチャ (Namespace.Class, assembly)。
  - [Code] - .NET フレームワークでサポートされる任意の言語で記述されたインライン コード。
  - [References] - 参照する .NET アセンブリのカンマ区切りのリスト (" , ")。
  - [言語] - C# に対する CSharp など .NET 言語指定子。

#### 重要

[種類] フィールドに値を入力するか、[Code]、[References]、および [言語] の各フィールドに値を入力します。4 つすべてのフィールドに入力することはしないでください。

#### 重要

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できます。アクションの詳細については、「アクション」のセクションを参照してください。

#### メモ

これらのフィールドが他のデータ テンプレートで表示される場合、このセクションで説明された目的で機能します。

---

<sup>1</sup> レイアウト エンジンの詳細については、

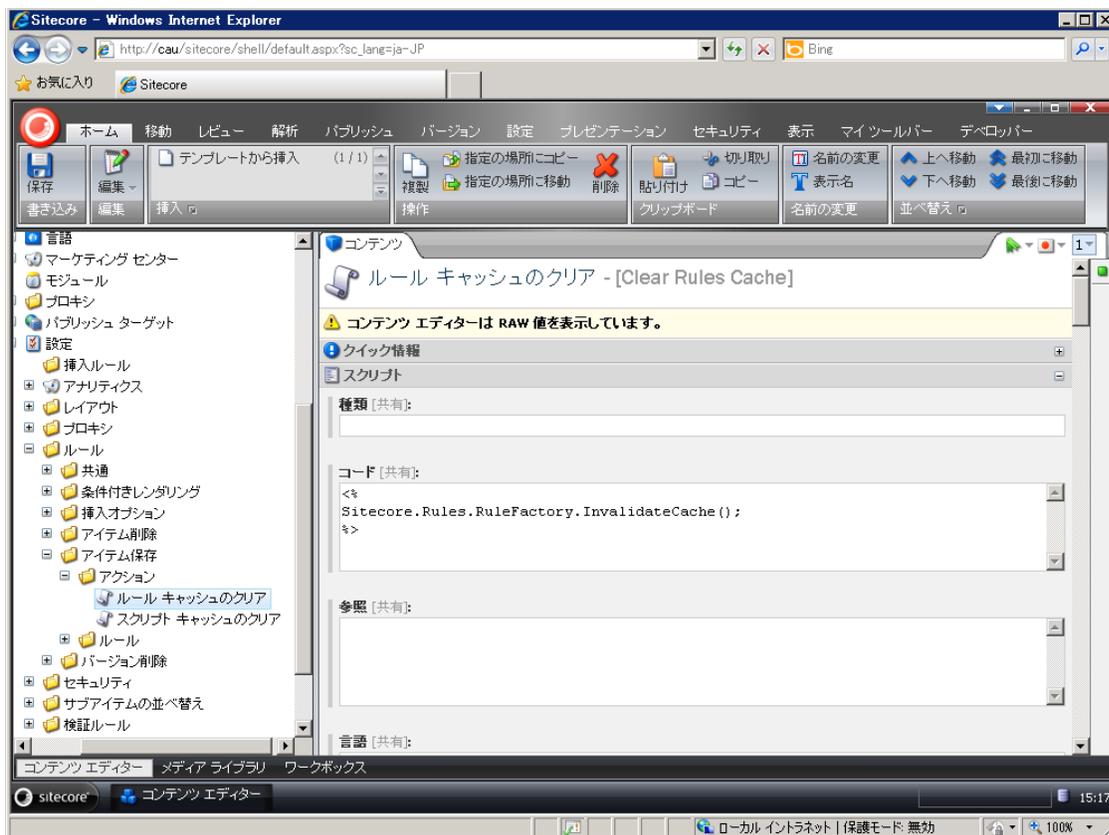
<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Component%20Reference.aspx> にある『プレゼンテーション コンポーネント リファレンス マニュアル』を参照してください。挿入オプションの詳細については、<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Sitecore%206/Data%20Definition%20Reference.aspx> にある『データ定義リファレンス マニュアル』を参照してください。

## ヒント

[Code] フィールドが、次の図に示されているように <% で始まり、%> で終わる場合、コードは Sitecore によって次の同等なものにラップされます。

```
namespace Sitecore
{
    public class DefaultClass
    {
        public void DefaultMethod()
        {
            // Contents of <% %>
        }
    }
}
```

たとえば、[Clear Rules Cache] スクリプトは、ルール エンジンに関連するすべてのキャッシュをクリアします。



## メモ

上記の図に示されているように、スクリプトをアクションとして使用できます。アクションの詳細については、「アクション」の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方法:

1. コンテンツ エディターで、スクリプト定義アイテムの格納先アイテムを選択します。
2. コンテンツ エディターで、System/Rules/Script データ テンプレートを使用してスクリプト定義アイテムを挿入します。スクリプト定義アイテムに、スクリプトの機能を明確かつ簡潔に示す名前を付けます。
3. スクリプト定義アイテムの [データ] セクションの [種類] フィールドに .NET クラスのシグネチャを入力するか、[Code]、[References]、および [言語] フィールド用の値を入力します。

### 2.1.2 条件

条件には、単一の条件が True であるかどうかを判断するためのロジックが含まれます。たとえば、[アイテム] 条件は、アイテムが他のアイテムの子孫である場合は True になります。

条件は、条件定義アイテムおよび条件を実装する .NET クラスで構成されます。

### デフォルト条件

Sitecore には、使用できるさまざまな条件が用意されています。

- **[アイテム]** - アイテムが特定のアイテムの子孫である場合。
- **[フィールド]** - アイテム内のフィールド値が、特定の値に等しい場合。
- **[ペルソナ]** - 訪問者のプロフィールが特定のユーザーに一致する場合。
- **[クエリ結果]** - アイテムに関連する特定の Sitecore クエリが 1 つまたは複数のアイテムを返す場合。
- **[ルール]** - 指定した条件が True と評価される場合。
- **[テンプレート]** - 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。

## 条件テキスト

Sitecore は、ユーザー インターフェイス内の条件定義アイテムの [データ] セクションの [Text] フィールドの値を使用します。ユーザー インターフェイスでは、条件を選択したり、条件パラメーターを入力したりできます。[Text] フィールド内の特定のトークンは、条件用に特定のパラメーターを指定することを可能にする機能に置換されます。

用語 `if`、`when`、および `where` の周囲には、Sitecore によってリンクが追加されます。リンクをクリックすると、条件が `[if]` と `[except if]` との間、`[when]` と `[except when]` との間、または `[where]` と `[except where]` との間で反転されます。

### メモ

条件が反転されると、それに応じて、条件ロジックの結果もルール エンジンによって自動的に反転されます。個々の条件を実装するためのコードは、ユーザーの選択に影響しません。

角かっこ ("`[]`") で囲まれたトークンを使用すると、パラメーターを指定するためのリンクを指定できます。角かっこ内には、カンマ ("`,`") で区切られた、4 つの位置指定パラメーターが含まれます。

1. 条件を実装する .NET クラス内でパラメーターが設定するプロパティの名前。
2. 事前定義されている機能を提供する `/Sitecore/System/Settings/Rules/Common/Macros` アイテムの下の子アイテムの名前、または空の文字列。たとえば、コンテンツ ツリーからアイテムを選択することを可能にするユーザー インターフェイスを起動するには、値ツリーを指定します。
3. 直前のパラメーターで指定したマクロによって起動されるユーザー インターフェイスへの URL パラメーター、または空の文字列。たとえば、選択ツリーのルートとして `/Sitecore/Content/Home` アイテムを指定するには、"`root=/sitecore/content/home`" と入力します。
4. パラメーターに値を指定していない場合に表示されるテキスト。

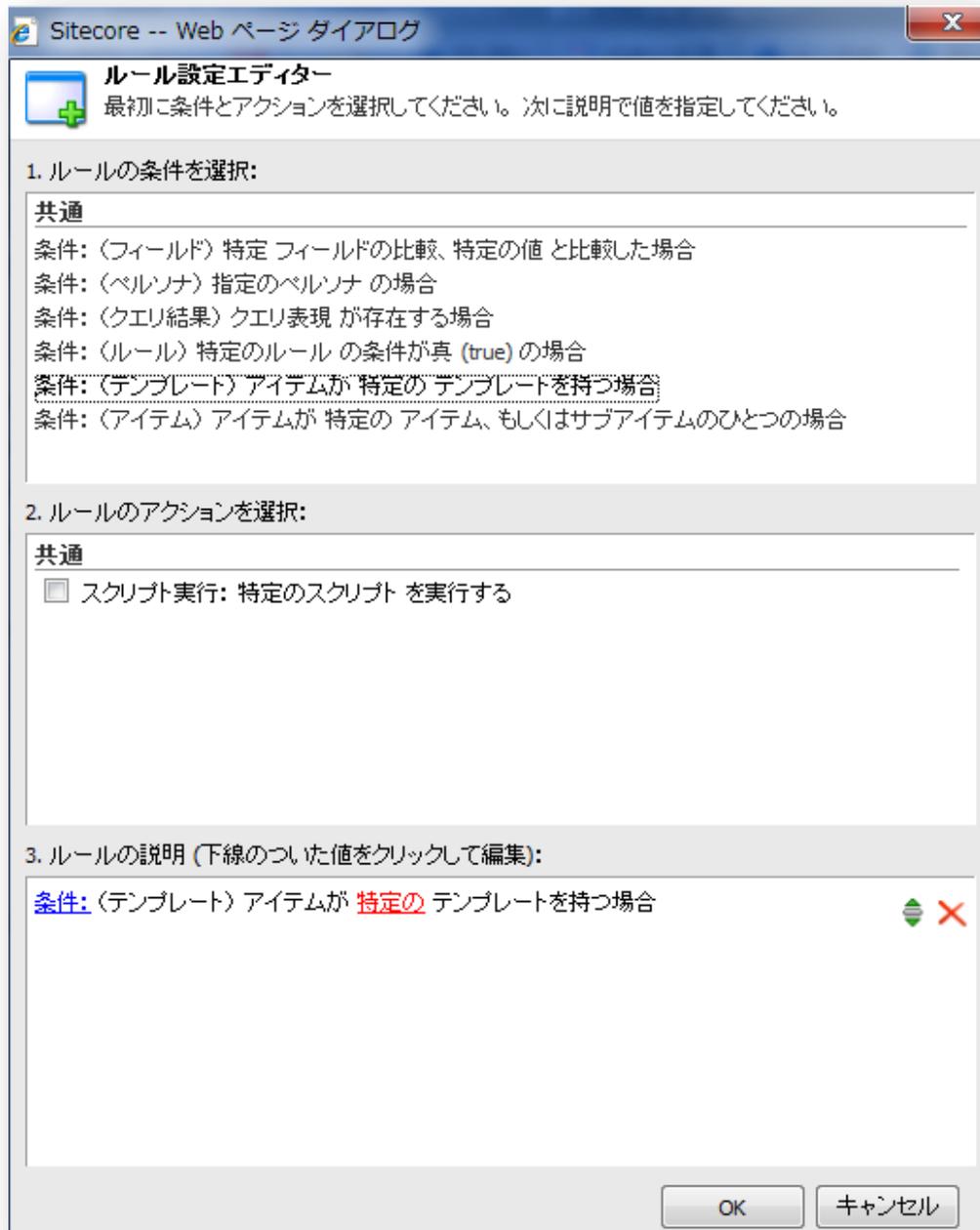
### ヒント

[Text] フィールドの構文の詳細については、デフォルトの条件定義アイテムを参照してください。

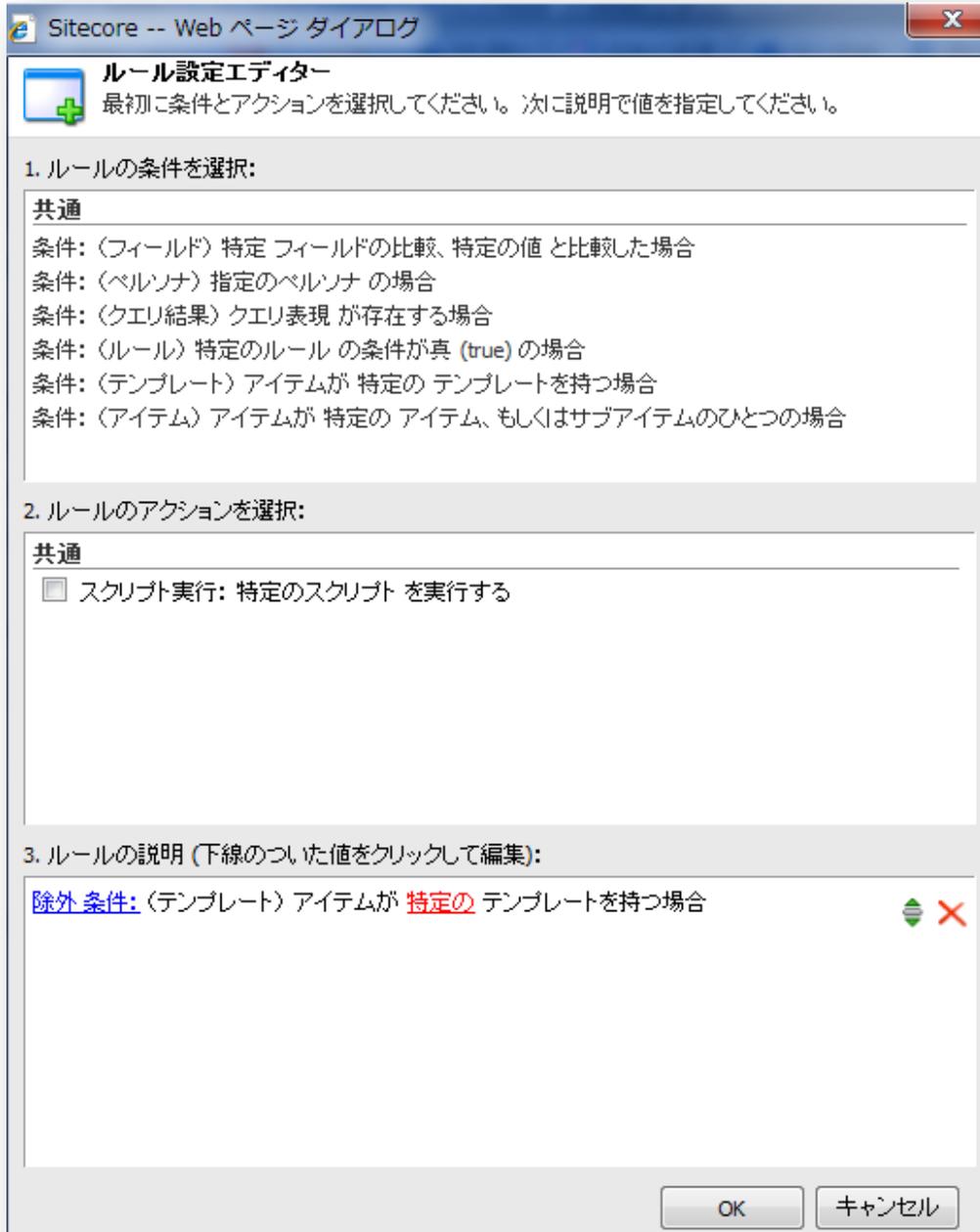
たとえば、[When Template] 条件は、アイテムが特定のデータ テンプレートに関連付けられている場合に `True` に評価されます。この条件のテキストは、次のとおりです。

```
where item has [templateid,Tree,root=/sitecore/templates,specific] template
```

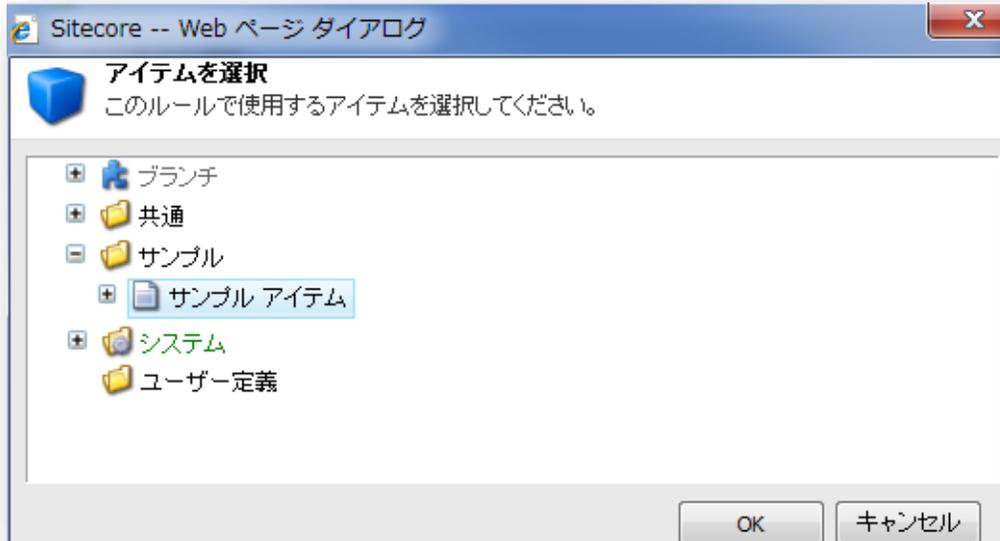
条件パラメーターを入力できるユーザー インターフェイスでは、[条件] および角かっこ ("`[]`") 内のトークンは変換され、ルール定義ユーザー インターフェイスを示す次の図の 3 番目のフィールドに示されるようにテキストが表示されます。条件パラメーターを定義する方法の詳細については、「ルール」のセクションを参照してください。



このダイアログ ボックスでは、データ テンプレートを選択し、オプションで条件を反転できます。[条件] をクリックすると、条件が反転します。



[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ように求められます。



このダイアログ ボックスでの選択により、条件を実装するクラスの `TemplateID` プロパティが制御されます。

## 条件の実装方法

条件を実装する方法:

1. Visual Studio プロジェクトで、`Sitecore.Rules.Conditions.OperatorCondition`、`Sitecore.Rules.Conditions.WhenCondition`、または `Sitecore.Rules.Conditions.StringOperatorCondition` から継承することによって、条件クラスを作成します。

### メモ

ブール型の条件を実装するには、`WhenCondition` から継承します。文字列値を比較するには、`StringOperationCondition` から継承します。他の種類の値を比較するには、`OperatorCondition` から継承します。

### ヒント

同じ機能の場合には、既存の条件を複製します。

2. 条件クラスで `Execute()` を実装し、条件が `true` または `false` のいずれであるかを示します。
3. コンテンツ エディターで、条件定義アイテムの格納先アイテムを選択します。

#### メモ

条件付きレンダリング固有の条件定義アイテムは、/Sitecore/System/Settings/Rules/Conditional Renderings/Conditions アイテムの下に挿入します。他の種類の条件定義アイテムは、/Sitecore/System/Settings/Rules アイテムの適切な子孫の下に挿入します。

4. コンテンツ エディターで、System/Rules/Condition データ テンプレートを使用して条件定義アイテムを挿入します。条件を実装する .NET クラスにちなんで、条件定義アイテムに名前を付けます。
5. コンテンツ エディターで、条件定義アイテムの [データ] セクションの [Text] フィールドに、「条件テキスト」のセクションで説明されている手順に従って、テキストを入力します。
6. コンテンツ エディターで、条件定義アイテムの [Script] セクションの [種類] フィールドに、.NET 条件クラスの種類シグネチャを入力します。
7. 「条件付きレンダリング ルール」のセクションで説明されている手順に従って、1 つまたは複数の条件付きレンダリング ルールに条件を構成します。

### 2.1.3 アクション

アクションは、1 つまたは複数の条件が True である場合に実装するロジックを含みます。たとえば、[挿入オプションの追加] アクションは、アイテムの効率的な挿入オプションにアイテムを追加します。

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 クラスで構成されます。

#### メモ

ユーザー インターフェイスが表示されない場合、アクションは「スクリプト」のセクションで説明されているスクリプトとして機能できることがあります。また、スクリプトは、このセクションで説明されているようにアクションとして機能できることがあります。

### デフォルトのアクション

デフォルトで提供されるアクションは、次のとおりです。

- **[挿入オプションの追加]** - アイテムを効率的な挿入オプションに追加します。
- **[スクリプト実行]** - 特定のスクリプトを起動します。

## アクション テキスト

Sitecore は、ユーザー インターフェイス内のアクション定義アイテムの [データ] セクションの [Text] フィールドの値を使用します。ユーザー インターフェイスでは、アクションを選択したり、アクション パラメーターを入力したりできます。[Text] フィールドのフォーマットの詳細については、「条件テキスト」のセクションを参照してください。

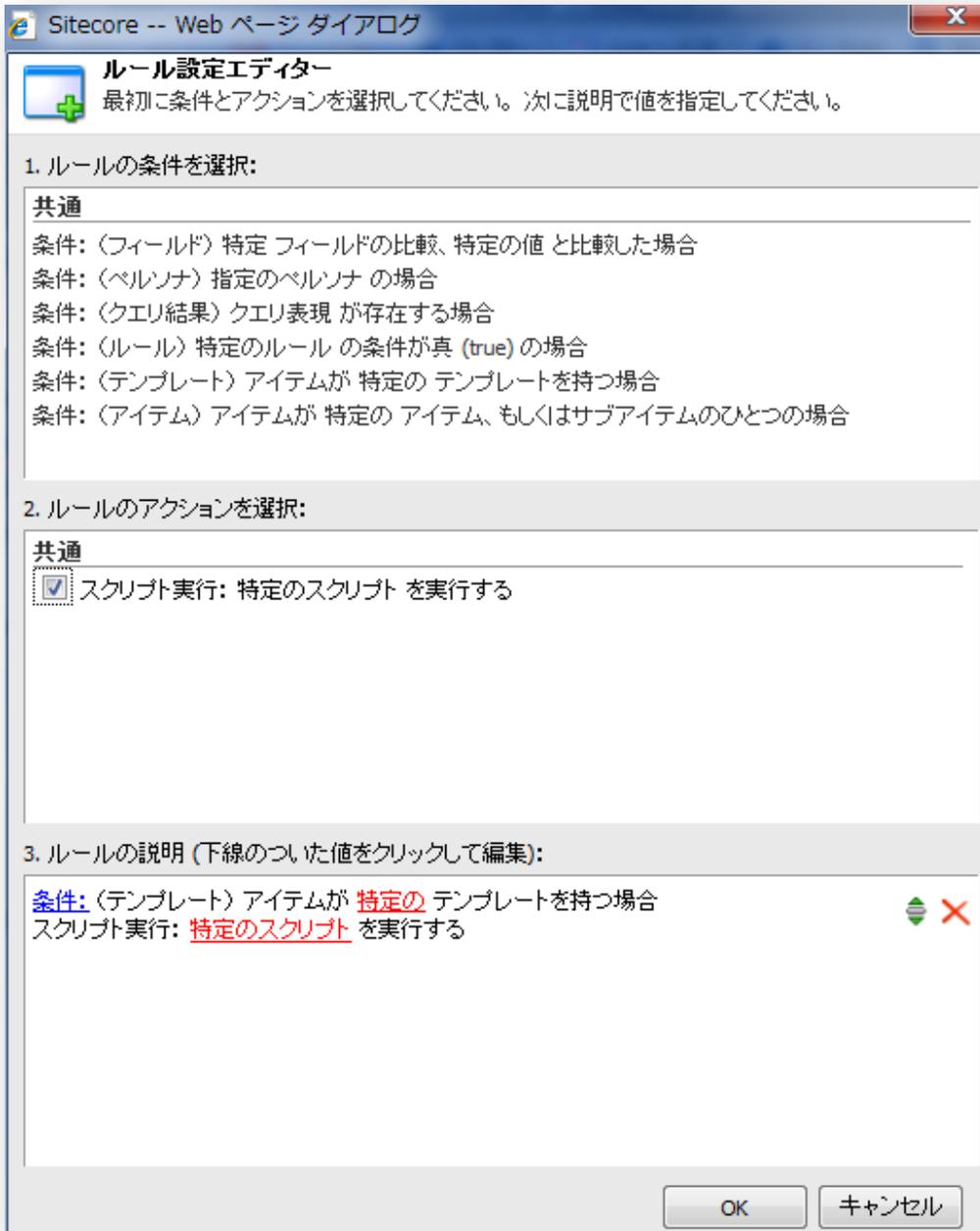
### ヒント

[Text] フィールドの構文の詳細については、デフォルトのアクション定義アイテムを参照してください。

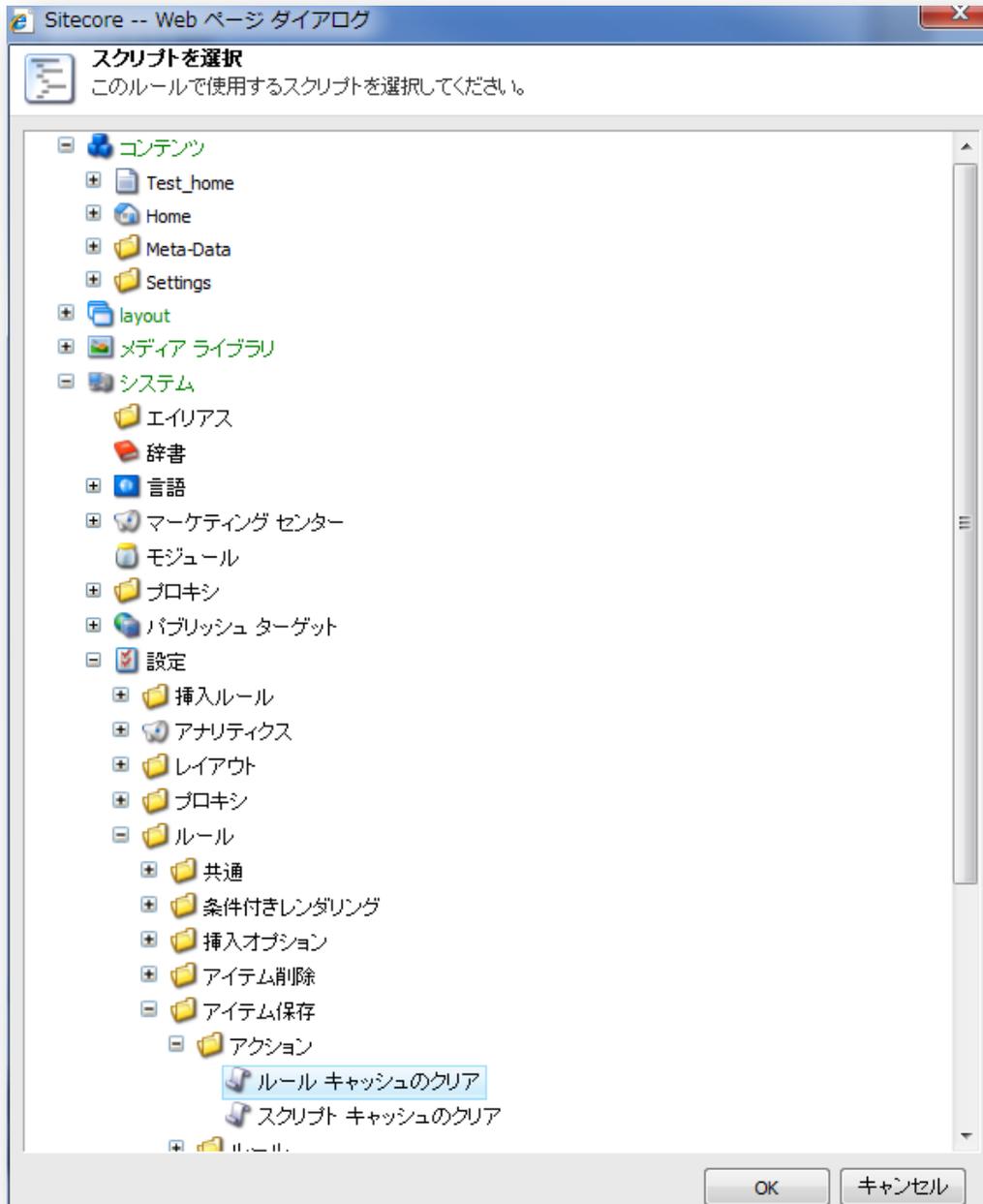
たとえば、[スクリプト実行] アクションは、スクリプトを起動します。このアクションのテキストは、次のとおりです。

```
run [scriptid,Script,,specific] script
```

アクション パラメーターを入力できるユーザー インターフェイスでは、角かっこ ("[]") 内のトークンは変換され、次の図の 3 番目のフィールドの 2 行目に示されるようにテキストが表示されます。ルール パラメーターを定義する方法の詳細については、「ルール」のセクションを参照してください。



このダイアログ ボックスでは、スクリプトを選択できます。[特定のスクリプト] をクリックすると、スクリプトを選択するように求められます。



このダイアログ ボックスでの選択により、アクションを実装するクラスの `ScriptID` プロパティが制御されます。

#### メモ

スクリプトは、アクションとして構成できます。スクリプトの詳細については、「スクリプト」の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方法:

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、アクション定義アイテムの格納先アイテムを選択します。

### メモ

条件付きレンダリング固有のアクション定義アイテムは、`/Sitecore/System/Settings/Rules/Conditional Renderings/Actions` アイテムの下に挿入します。他の種類のアクション定義アイテムは、`/Sitecore/System/Settings/Rules` アイテムの適切な子孫の下に挿入します。

3. コンテンツ エディターで、`System/Rules/Action` データ テンプレートを使用してアクション定義アイテムを挿入します。アクションを実装する .NET クラスにちなんで、アクション定義アイテムに名前を付けます。
4. コンテンツ エディターで、アクション定義アイテムの [データ] セクションの [Text] フィールドに、「アクション テキスト」のセクションで説明されている手順に従って、テキストを入力します。
5. コンテンツ エディターで、アクション定義アイテムの [Script] セクションの [種類] フィールドに、アクションを実装する .NET クラスのシングネチャを入力します。
6. 「条件付きレンダリング ルール」のセクションで説明されている手順に従って、アクション定義アイテムを参照するための 1 つまたは複数のルール定義アイテムを構成します。
7. 「条件付きレンダリングの適用」のセクションで説明されている手順に従って、1 つまたは複数のアイテム用に条件付きレンダリング ルールを構成します。

### 2.1.4 ルール

ルールは、アクションと条件の両方のパラメーター値を含めて、1 つまたは複数のアクションを 1 つまたは複数の条件に関連付けます。たとえば、[条件付きレンダリング] ルールは、多変数テストが有効にされた場合 (条件) に、多変数テスト (アクション) を処理します。この場合には非論理的となる可能性もありますが、条件を反転できます (多変数テストが有効でない場合に、多変数テストを処理する)。

`and` および `or` などの論理演算子を使用して、条件の特定の組み合わせの場合だけにアクションを起動するように構成できます。

## ルールの種類

挿入オプション、挿入ルール、および `uiGetMasters` パイプラインに加えて、実行時に効率的な挿入オプションを制御する挿入ルールを実装できます<sup>2</sup>。挿入オプション ルールは、`/Sitecore/System/Settings/Rules/Insert Options/Rules` アイテムの下に構成します。

### メモ

Sitecore では、イベント ハンドラーを使用してルールが起動されます。イベントに対してルールが実行される条件の詳細については、対応するイベント ハンドラーを参照してください。

### ヒント

特定のアイテムに対して挿入オプション 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 ロジックに対しては、`uiGetMasters` パイプラインを使用します。挿入オプション ルールを使用すると、管理者は、可変ロジックを使用して挿入オプションを構成できます。

`item:deleting` イベント、`item:deleted` イベント、および `uiDeleteItems` パイプラインに加えて、アイテム削除済みルールを実装して、特定の条件に一致するアイテムをユーザーが削除した後に起動するアクションを構成できます。アイテム削除済みルールは、`/Sitecore/System/Settings/Rules/Item Deleted/Rules` アイテムの下に構成します。

`item:saving` イベント、`item:saved` イベント、および `saveUI` パイプラインに加えて、アイテム保存済みルールを実装して、特定の条件に一致するアイテムをユーザーが保存した後に起動するアクションを構成できます。アイテム削除済みルールは、`/Sitecore/System/Settings/Rules/Item Saved/Rules` アイテムの下に構成します。

`item:versionRemoving` イベント、`item:versionRemoved` イベント、および `deleteVersionsUI` パイプラインに加えて、バージョン削除済みルールを実装して、特定の条件に一致するアイテムのバージョンをユーザーが削除した後に起動するアクションを構成できます。バージョン削除済みルールは、`/Sitecore/System/Settings/Rules/Version Removed/Rules` アイテムの下に構成します。

Sitecore では、条件付きレンダリングに対して異なるデータ テンプレートが使用されますが、同じ概念が適用されます。条件付きレンダリングの詳細については、「Chapter 3 条件付きレンダリング」を参照してください。

## ルールの実装方法

ルールを実装する方法:

1. コンテンツ エディターで、ルール定義アイテムの格納先アイテムを選択します。

---

<sup>2</sup> 挿入オプションの詳細については、

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

**メモ**

削除を処理するルール定義は、`/Sitecore/System/Settings/Rules/Item Deleted/Rules` アイテムの子として挿入します。保存を処理するルール定義は、`/Sitecore/System/Settings/Rules/Item Saved/Rules` アイテムの子として挿入します。バージョン削除を処理するルール定義は、`/Sitecore/System/Settings/Rules/Version Removed/Rules` アイテムの子として挿入します。条件付きレンダリング固有のルール定義アイテムは、`/Sitecore/System/Marketing Center/Personalization/Rules` アイテムの下に挿入します。グローバルな条件付きレンダリング ルールは、`/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules` アイテムの下に挿入します。他の種類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挿入します。

2. コンテンツ エディターで、`System/Rules/Rule` データ テンプレートをを使用してルール定義アイテムを挿入します。

**メモ**

条件付きレンダリング ルールに対しては、`System/Rules/Conditional Rendering Rule` データ テンプレートを 사용합니다。挿入オプション ルールに対しては、`System/Rules/Insert Options Rule` データ テンプレートを 사용합니다。他のすべてのルールに対しては、`System/Rules/Rule` データ テンプレートを 사용합니다。

3. コンテンツ エディターで、ルール定義アイテムの [データ] セクションの [Name] フィールドに、ユーザー インターフェイスに表示されるルールの名前を入力します。
4. コンテンツ エディターのルール定義アイテムの [データ] セクションの [Rule] フィールドで、条件付きレンダリングの条件とアクションを選択し、条件とアクションの両方のパラメーターを入力します。
5. 「条件付きレンダリングの適用」のセクションで説明されている手順に従って、グローバルではない条件付きレンダリング ルールを適用します。

## Chapter 3

### 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操作を可能にする条件付きレンダリングを実装するための固有の手順について説明します。Sitecore 開発者用に、条件付きレンダリングの完全な例が用意されています。

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の概要
- 条件付きレンダリング例の要件
- 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

### 3.1 条件付きレンダリングの概要

条件付きレンダリングという用語は、HTTP 要求を処理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するための Sitecore レイアウト エンジンの機能を指します。この機能には、レンダリングの包含または除外、プレースホルダーの制御、データ ソース および他のプロパティの設定、多変数条件の処理、他のロジックなどがあります。

## 3.2 条件付きレンダリング例の要件

このマニュアルでは、次の要件を満たす条件付きレンダリング コンポーネントの例について説明します。

- ユーザーがページにアクセスしたときにコンピューターによって生成された乱数が特定の数値に一致した場合、レンダリングはユーザーが勝利したことを示します。
- 乱数の下限は、1 です。
- CMS ユーザーは、乱数の上限を指定する必要があります。
- CMS ユーザーは、乱数に一致する数値を指定する必要があります。

## 3.3 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について説明します。

### 3.3.1 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やプロパティの内の 1 つの設定など、条件付きレンダリングの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 アクションは、「アクション」のセクションで説明されているアクションです。レイアウト詳細は、条件付きレンダリング アクションを指す条件付きレンダリング ルールを指します。

#### デフォルトの条件付きレンダリング アクション

Sitecore には、使用できるさまざまなデフォルトの条件付きレンダリング アクションが用意されています。

- [レンダリングを非表示にする] - レイアウト エンジンにレンダリングが含まれないようにします。
- [データソースを特定のアイテムに設定する] - レンダリングのデータ ソースを設定します。
- [パラメーターを特定の値に設定する] - レンダリングのパラメーターまたはプロパティを設定します。
- [プレースホルダーを特定の値に設定する] - レンダリングのバインド先となるプレースホルダーを設定します。

#### 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方法:

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/Conditional Renderings/Actions` アイテムを選択します。
3. コンテンツ エディターで、`/Sitecore/System/Settings/Rules/Conditional Renderings/Actions` アイテムの下に、`System/Rules/Action` データ テンプレートを使用してアクション定義アイテムを挿入します。

#### ヒント

条件付きレンダリング アクションを実装する .NET クラスにちなんで、条件付きアクション定義アイテムに名前を付けます。

4. 条件付きアクション定義アイテムの [データ] セクションの [Text] フィールドに、適切なパターンを入力します。[Text] フィールドの詳細については、「アクション テキスト」のセクションを参照してください。
5. 条件付きアクション定義アイテムの [Script] セクションの [種類] 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 クラスのシグネチャを入力します。
6. 「条件付きレンダリング ルール」のセクションで説明されている手順に従って、1 つのまたは複数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 アクションを構成します。
7. 「条件付きレンダリングの適用」のセクションで説明されている手順に従って、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 アクションへのレンダリング プロパティの設定

条件付きレンダリング アクションを、レンダリングの `IsWinner` プロパティを設定する次の例に基づいて実装できます。

```
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);
        }
    }
}
```

```

    }
  }
}

```

条件付きレンダリング アクション定義アイテムの [データ] セクションの [Text] フィールドには、乱数に一致した場合の `IsWinner` プロパティの値を指定できます。

```
set parameters to (true or false) [IsWinner,Text,,value]
```

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。これらのパラメーターが使用されている状況を示す図については、「例: 乱数一致の条件付きレンダリング ルール」のセクションを参照してください。

## 例: レンダリングを追加する条件付きレンダリング アクション

レンダリングをブレースホルダーに追加する次の例に基づいて、条件付きレンダリング アクションを実装できます。

```

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);
        }
    }
}

```

ユーザーがレンダリングを選択し、プレースホルダー キーを入力できるようにするために、アクション定義アイテムで、次のテキストを入力します。

```
add the [RenderingID,Tree,root=/sitecore/layout/renderings,specific] rendering
to the [Placeholder,Text,,specific] placeholder
```

### 3.3.2 条件付きレンダリング条件

条件付きレンダリング条件には、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ために条件が True であるかどうかを確認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条件を指す条件付きレンダリング ルールを指します。

#### ヒント

「デフォルト条件」のセクションで説明されている条件を、条件レンダリング条件として使用できます。

#### 条件付きレンダリング条件の実装方法

条件付きレンダリング条件を実装するには、「条件」のセクションで説明されている手順に従います。

#### 例: 乱数一致の条件付きレンダリング条件

乱数が特定のパラメーター値に一致した場合に true と評価する次の例に基づいて、条件付きレンダリング条件を実装できます。

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

    return false;
}
}
}

```

条件付きレンダリング条件を実装するには、「条件の実装方法」のセクションで説明されている手順に従います。ただし、System/Rules/Conditional Rendering Rule データ テンプレートを System/Rules/Rule データ テンプレートの代わりに使用します。コンテンツ エディターで、条件付きレンダリング定義アイテムの [データ] セクションの [Text] フィールドに、次のテキストを入力します。これにより、ユーザーは、条件付きレンダリング条件の UpperLimit パラメーターと MatchNumber パラメーターを設定し、等値演算子 ("=") などの演算子を選択できるようになります。

```

when a random number between 1 and [UpperLimit,,,upper limit]
[operatorid,Operator,,,compares to] [MatchNumber,,,number]

```

条件レンダリング条件の基本クラス (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クラス (Sitecore.Sharedsource.Rules.Conditions.RandomNumberCondition) は、UpperLimit プロパティと MatchNumber プロパティを定義します。これらのパラメーターが使用されている状況を示す図については、「例: 乱数一致の条件付きレンダリング ルール」のセクションを参照してください。

#### メモ

この例で示されている要件を厳密に満たすために、ユーザーに演算子を選択させるのではなく開発時に比較ロジックをハードコードして組み込むことができます。

### 3.3.3 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 ロジックを制御します。条件付きレンダリング ルールは、アクションと条件の両方のパラメーター値を含めて、1 つまたは複数の条件付きレンダリング アクションを 1 つまたは複数の条件付きレンダリング条件に関連付けます。条件付きレンダリング ルールは、「ルール」のセクションで説明されているルールですが、

System/Rules/Conditional Rendering Rule データ テンプレートを System/Rules/Rule データ テンプレートの代わりにベースにします。

## 例: 乱数一致の条件付きレンダリング ルール

例の要件を満たす条件付きレンダリング ルールを実装する方法:

1. 「ルールの実装方法」のセクションで説明されている手順に従って、Pick a Random Winner という名前の条件付きレンダリング ルール定義アイテムを挿入します。
2. 条件付きレンダリング ルール定義アイテムの [データ] セクションの [Rule] フィールドの [ルールの条件を選択] の下の [when a random number between 1 and the upper limit compares to number] をクリックして、条件をルールに追加します。
3. 条件付きレンダリング ルール定義アイテムの [データ] セクションの [Rule] フィールドの [ルールの条件を選択] の下の [user wins] チェック ボックスをオンにして、指定した条件が True に評価された場合に実行されるアクションのリストにそのアクションを追加します。
4. 条件付きレンダリング ルール定義アイテムの [データ] セクションの [Rule] フィールドの [ルールの説明] の下の [upper limit] をクリックし、乱数生成時の上限を入力します。
5. 条件付きレンダリング ルール定義アイテムの [データ] セクションの [Rule] フィールドの [ルールの説明] の下の [compares to] をクリックし、[is equal to] 演算子を選択します。
6. 条件付きレンダリング ルール定義アイテムの [データ] セクションの [Rule] フィールドの [ルールの説明] の下の [number] をクリックし、乱数に一致する数値を入力します。
7. 「条件付きレンダリングを適用する方法」のセクションで説明されている手順に従って、条件付きレンダリング ルールを適用します。

## 3.4 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を適用する方法について説明します。

### 3.4.1 条件付きレンダリングを適用する方法

データ テンプレートの標準値のレイアウト詳細で、条件付きレンダリングを構成します<sup>3</sup>。

### 3.4.2 グローバルな条件付きレンダリング ルール

各レンダリングでは、レイアウト詳細で選択した条件付きレンダリング ルールの評価に加えて、`/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules` アイテムの下で定義されるグローバルな条件付きレンダリング ルールも Sitecore によって評価されます。

#### 重要

グローバルな条件付きレンダリング ルールは各レンダリングで実行されるため、特にレイアウト詳細で指定するレンダリングの個数が増加する場合は、パフォーマンスを最適化する必要があります。

グローバルな条件付きレンダリング ルールを定義するには、`/Sitecore/System/Settings/Rules/Conditional Renderings/Global Rules` アイテムの下に条件付きレンダリング ルールを実装します。これを行うには、「[ルールの実装方法](#)」のセクションで説明されている手順に従いますが、`System/Rules/Conditional Rendering Rule` データ テンプレートを使用します。

---

<sup>3</sup> レイアウト詳細の詳細については、

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Component%20Reference.aspx> にある『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。