



Web Forms for Marketers 2.4 Sitecore CMS 7.1 以降

リファレンスガイド

管理者および開発者のためのリファレンスガイド

目次

Chapter 1	イントロダクション	4
Chapter 2	モジュールの使用	6
2.1	Web フォームの保存	7
2.2	新しいフォームの作成	9
2.3	プレースホルダー リストに表示されるプレースホルダーの制限	10
2.4	Sitecore フォームのフィールド タイプ	11
2.4.1	リスト アイテム	19
	XPath クエリを使用する	20
	サイトコア クエリを使用する	20
	高速 クエリを使用する	20
	リスト アイテムのローカライズ	20
2.5	検証	22
2.5.1	“必須” チェック ボックス フィールド検証の実装方法	24
2.6	送信アクション	26
2.6.1	フォームの検証	27
	フォームの検証のエラー メッセージの変更	27
2.6.2	保存アクション	27
2.6.3	成功	31
2.6.4	監査情報	32
2.7	レポート	33
2.7.1	サポートされているデータベース	34
2.7.2	データ リクエストのタイムアウト	35
2.7.3	概要	35
2.8	モジュールへのユーザー アクセスの設定	36
2.8.1	Web フォーマットのセキュリティ ロール	36
2.9	イベントと訪問の詳細(セッションの追跡)	40
2.10	マルチサイトの実装	43
2.11	マルチサーバー環境	45
2.11.1	コンテンツ デリバリー サーバー上にモジュールをデプロイする	45
2.12	ライブ モードでの Web フォームの使用	48
Chapter 3	Web フォームの開発者のための注意事項	49
3.1	Sitecore.forms.config	51
3.1.1	Events	51

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.

3.1.2	Pipelines	51
3.1.3	Commands	52
3.1.4	Settings	52
3.2	カスタム プロセッサーを使用したモジュールの挙動の修正	53
3.2.1	ProcessMessage パイプラインを使用した電子メール メッセージの設定方法	53
3.2.2	カスタム プロセッサーを使用した SMS/MMS の送信方法	54
3.2.3	ファイルのアップロード フィールドの大容量ファイルアップロードの回避方法	55
3.3	Web ページへの Web フォームの挿入方法	57
3.4	データ プロバイダーの設定方法	59
3.4.1	SQLite	59
3.4.2	MSSQL.....	59
	SQL Server 2008 上のモジュール用 データベースに必要なデータベース ユーザーの権限	60
3.4.3	Oracle	60
3.5	標準機能の拡張/オーバーライド方法	63
3.5.1	フォームのレンダリング	63
3.5.2	フィールド コントロール.....	64
3.5.3	フィールド アクション.....	65
3.6	モジュールで使用されるフィールドの階層構造.....	68
3.7	新しいフィールド タイプの作成方法	71
3.7.1	ビジュアル フィールド プロパティの作成	75
3.7.2	WebControl クラスに基づく新しいフィールド タイプの作成	77
3.7.3	UserControl クラス (.ascx ファイル) に基づく新しいフィールド タイプの作成.....	81
3.8	保存アクションの作成.....	82
3.8.1	ベース インターフェースとクラス	82
3.8.2	パラメーター	85
3.8.3	保存アクションの作成方法	85
3.9	アクション エディターの作成方法.....	88
3.10	フォームの検証アクションの作成方法	94
3.11	送信された Web フォーム データのアクセス方法	96
3.12	CSS テーマの使用方法	97
3.13	CSS スタイルの設定方法.....	99
3.14	ASCX へのエクスポート方法.....	101
3.15	ASCX コントロールをページに追加する方法.....	104
3.16	モジュールを再インストールする方法	106
3.17	モジュールをアンインストールする方法.....	107

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 1

イントロダクション

このドキュメントは Sitecore 管理者を対象としています。モジュールのセットアップ方法についての情報も含まれます。エンド ユーザーのための詳細な取扱説明については、『Web Form ユーザー ガイド』を参照してください。

ほとんどの Web サイト上のフォームはシンプルかつ簡潔であるため、フォームを作成し管理するプロセスもまたシンプルで簡潔であるべきです。

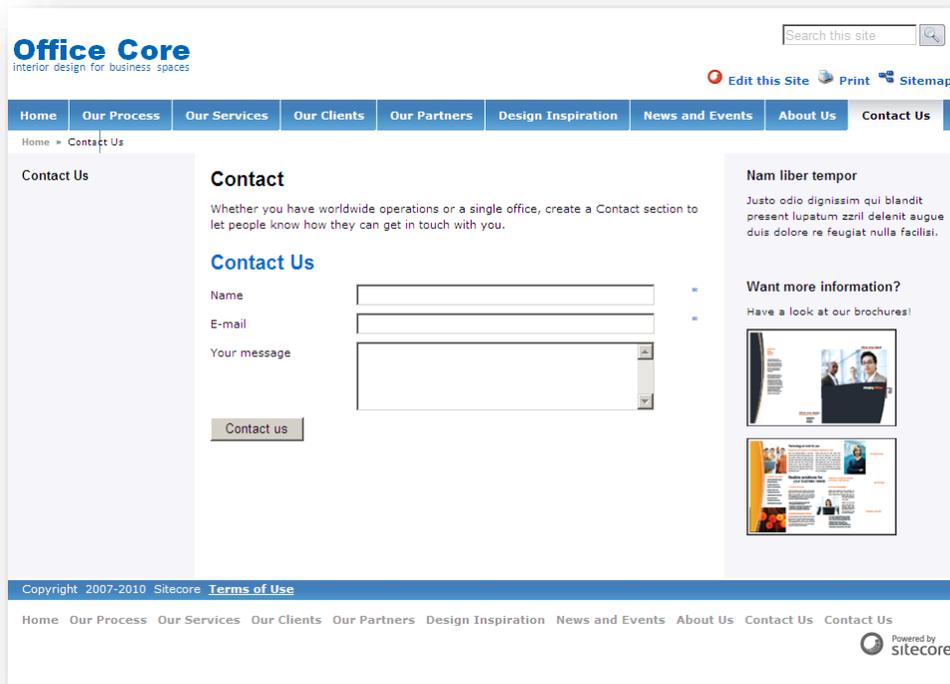
Web Forms for Marketers モジュールは、ユーザー フレンドリな方法で簡単にフォームを作成できるように設計されています。このモジュールを使って作成できるフォームは、WCAG 2.0 および XHTML 1.0 に準拠しています。Web Forms for Marketers モジュールは、ユーザーに Web アナリティクスとレポート機能を提供します。さらに、Web サイトの訪問者がフォームの送信に成功したか否かに関らず、フォームに入力されたすべての情報の記録とレポートを行います。このモジュールは、Sitecore CMS 7.1 における Sitecore エンゲージメント アナリティクスと完全に統合されています。

モジュールは少数の調整可能なパラメーターのみを持つように設計されています。それにより、分かりやすい、非常にシンプルなユーザー インターフェイスになっています。基本的なオプションには、標準的なコンテンツ編集者のニーズを満たす内容が含まれています。たとえば、テキスト ボックスやチェック ボックスなどの基本的な入力フィールドの作成、データベースへの保存や電子メール メッセージを送信などの基本的なアクションの作成、必須チェックの検証や電子メール アドレスの検証などの 基本的な検証が含まれます。

また、このモジュールによって生成されたフォームに基づいてより複雑なフォームを開発することもできます。たとえば、フォームをサブレイアウト (.ascx コントロール) にコンバートし、Visual Studio などの開発ツールを使ってサブレイアウトを編集することができます。

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.

このモジュールを使って作成されたフォームは次のようになります。



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 2

モジュールの使用

この章ではモジュールの機能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。

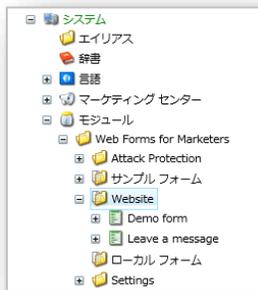
- Web フォームの保存
- 新しいフォームの作成
- プレースホルダー リストに表示されるプレースホルダーの制限
- Sitecore フォームのフィールド タイプ
- 検証
- 送信アクション
- レポート
- モジュールへのユーザー アクセスの設定
- イベントと訪問の詳細(セッションの追跡)
- マルチサイトの実装
- マルチサーバー環境
- ライブ モードでの Web フォームの

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.

2.1 Web フォームの保存

フォームのストラクチャは Sitecore CMS のコンテンツ ツリーで定義されます。訪問者が入力したフォームの情報は、Sitecore_WebForms というモジュール独自のデータベースに保存されます。このデータベースは、website/data フォルダに配置されています。このデータベースは、モジュールの旧バージョンでは Sitecore.WebForms と呼ばれていました。

Web Form for Marketers モジュールには、Sitecore.forms.config という独自の設定ファイルがあります。このファイルは、website/app_config/include フォルダに配置されています。フォームを作成するアイテムは、/sitecore/System/Modules/Web Forms for Marketers 配下の適切なフォルダに格納されています。



website¥app_config¥include¥Sitecore.forms.config ファイルの /sitecore/sites/site ノードの formsRoot 属性に新しいフォームを保存する場所を指定します。サイトで formsRoot 属性が定義されていない場合、新しいフォームは sitecore/System/Modules/Web Forms for Marketers/Local Forms フォルダに作成されます。

フォーム フォルダは、/sitecore/Templates/Web Forms for Marketers/Forms Folder テンプレートに基づきます。各フォームは、/sitecore/Templates/Web Forms for Marketers/Form テンプレートに基づき、任意の数のセクションまたはフィールドを含むことができます。



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.

すべてのフォームは、アクションを関連付けることができる [送信] ボタンを含みます。アクションはサーバー上で実行されます。

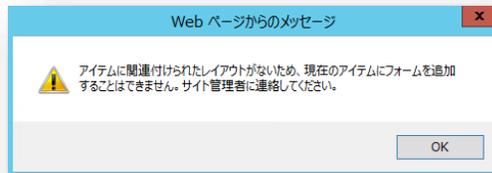
利用可能なアクションの一覧は、

/sitecore/System/Modules/Web Forms for Marketers/Settings/Actions フォルダーの配下に格納されています。

2.2 新しいフォームの作成

ユーザーは**ページ エディター**と**コンテンツ エディター**を使って新しいフォームを作成することができます。

ユーザーが**コンテンツ エディター**で [プレゼンテーション] > [フォーム] > [挿入] ボタンをクリックした場合、まずウィザードによって、現行のアイテムにフォームに割り当てられたレイアウトがあるかどうかを確認されます。この条件に満たない場合は、ウィザードは次のメッセージを表示します。



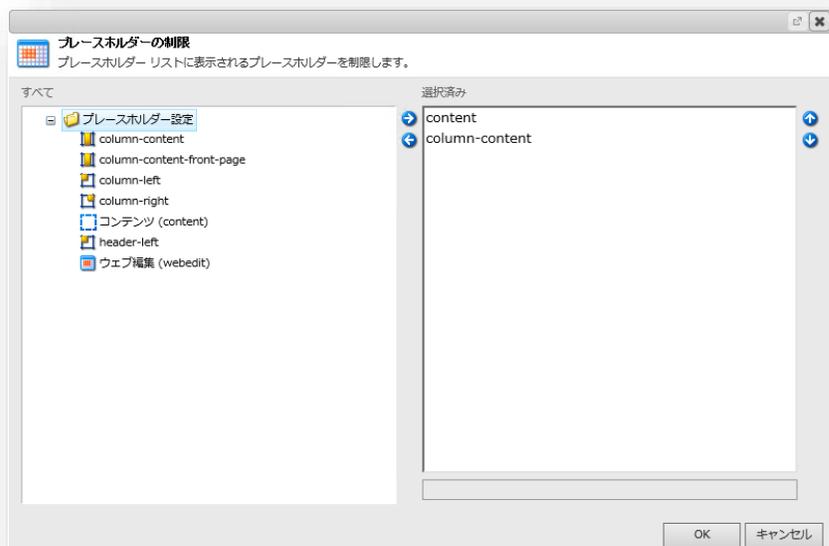
フォームはレイアウト上のプレースホルダーに配置する必要があります。Web Form for Marketers モジュールで有効なプレースホルダーについての追加情報は、「プレースホルダー リストに表示されるプレースホルダーの制限」セクションを参照してください。

プログラムから Web フォームを挿入する方法についての追加情報は、「Web ページへの Web フォームの挿入方法」セクションを参照してください。

2.3 プレースホルダー リストに表示されるプレースホルダーの制限

[**新しいフォームを作成**] ウィザードでは、“プレースホルダー設定” アイテムを持つプレースホルダーにのみフォームを追加することが可能です。新しいフォームを追加するユーザーは、“プレースホルダー リスト” でプレースホルダーを確認するために、フォームを追加するアイテムへの書き込みアクセスが必要です。これによって、開発者および Web サイトの管理者は、どのプレースホルダーにフォームを配置できるかを定義することができます。

[**プレースホルダーの制限**] ウィザードは、[**新しいフォームを作成**] ウィザードの [**プレースホルダー リスト**] に表示される、プレースホルダーのリストを制限する場合に役立ちます。[**プレースホルダーの制限**] ウィザードを開始するには、[Sitecore] > [すべてのプログラム] > [Web Forms for Marketers] > [**プレースホルダーの制限**] と順番にクリックします。このウィザードは、Web Form for Marketers モジュールのインストールが完了した場合にも表示されます。



[**選択済み**] フィールドには、ユーザーが新しいフォームを追加できるプレースホルダーがリスト表示されます。[**すべて**] リストから [**選択済み**] リストにプレースホルダーを追加するには、プレースホルダーを選択し、 をクリックします。[**OK**] をクリックすると、すべての変更が保存されます。

[**プレースホルダーの制限**] ダイアログボックスで、プレースホルダーを [**すべて**] リストに追加するには、新しいプレースホルダーアイテムを Sitecore/Layout/Placeholder Settings アイテムの配下に作成する必要があります。既にプレースホルダーがある場合は、プレースホルダー アイテム作成して、[**プレースホルダー キー**] フィールドにプレースホルダーの名前に設定します。

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.

2.4 Sitecore フォームのフィールド タイプ

Web Forms for Marketers モジュールは、フォームを構築するために使用可能な多数のフィールド タイプが含まれています。すべてのフォームのフィールド タイプは、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types アイテムの配下に保存され、/sitecore/Templates/Web Forms for Marketers/Field Type テンプレートを使用して作成されています。



Field Type テンプレートには次のフィールドが含まれます。

- **Required** — 必須チェックの検証がこのフィールド タイプに適用されるかどうかを定義するフィールドです。
- **Validation** — このフィールドに入力された値に適用する検証のリストです。
- **User Control** — ASCX コントロールへの参照を定義するフィールドです。
- **Deny Tag** — このフィールド タイプに入力された情報がアナリティクス データベースの [タグ] に保存されるかどうかを定義するフィールドです。

Web Forms for Marketers モジュールは次のフィールド タイプを含みます。

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.

単一行テキスト

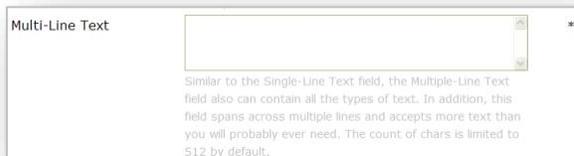
1 行のテキストを入力するために、このフィールドタイプを使います。フィールドの長さはデフォルトで 255 文字に制限されています。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Single-Line Text アイテムに含まれています。

複数行テキスト

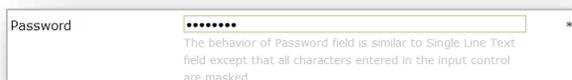
複数行のテキストを入力するために、このフィールド タイプを使います。文字数はデフォルトで 512 文字に制限されています。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Multiple-Line Text アイテムに含まれています。

パスワード

パスワードを入力するためにこのフィールドタイプを使用します。[パスワード] フィールドに入力したすべての文字はマスクされます。[パスワード] フィールドはテキスト フィールドです。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Password アイテムに含まれています。

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.

番号

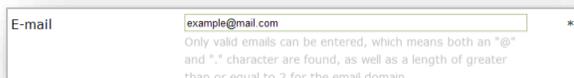
数値データを入力するためにこのフィールドタイプを使用します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Number アイテムに含まれています。

電子メール

このフィールド タイプを使って電子メール アドレスを入力します。文字 "@" および "." が検証されます。また、その電子メール サーバーのドメインの長さも検証されます。



このフィールド タイプの設定は、sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/E-mail アイテムに含まれています。

電話

このフィールド タイプを使って電話番号を入力します。またユーザーは、この数値フィールドには 記号 "+", "-", "(", ")" およびスペースを入力することができます。

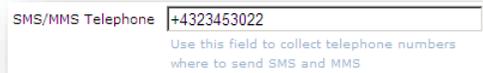


このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Telephone アイテムに含まれています。

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.

SMS/MMS の電話

このフィールド タイプを使って SMS と MMS を送信できる電話番号を入力します。このフィールドには数値と記号 "+" を入力することができます。入力データの先頭の記号は "+" のみを含むことができます。このフィールド タイプは、**SMS を送信** および **MMS を送信** 保存アクションで使用されます。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/SMS/MMS Telephone アイテムに含まれています。

日付

このフィールドを使って日付を入力します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Date アイテムに含まれています。

日付の選択

このフィールド タイプを使ってカレンダーから日付を選択します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Date Picker アイテムに含まれています。

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.

ファイルのアップロード

このフィールド タイプを使って、サーバーにアップロードするファイルのパスを表示するテキスト ボックスと [参照] ボタンを表示します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/File Upload アイテムに含まれています。

チェック ボックス

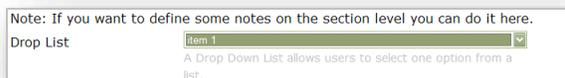
このフィールド タイプを使って、true または false を選択できるチェック ボックスを表示します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Simple Types/Checkbox アイテムに含まれています。

ドロップ リスト

このフィールド タイプを使って、ドロップダウン リストからオプションを選択します。

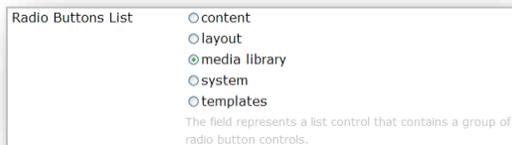


このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/List Types/Drop List アイテムに含まれています。

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.

ラジオ ボタンのリスト

このフィールド タイプを使って、オプションのグループを表示します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/List Types/Radio List アイテムに含まれています。

リスト ボックス

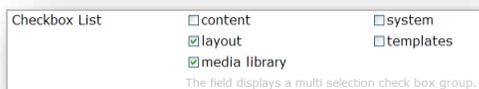
このフィールド タイプを使って、複数のアイテムを選択できるリスト ボックスを表示します。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/List Types/List Box アイテムに含まれています。

チェック ボックスのリスト

このフィールド タイプを使って、チェック ボックスのグループを表示します。チェック ボックスは複数選択することができます。



このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/List Types/Checkbox List アイテムに含まれています。

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.

セクション

セクションは、他のフィールドのコンテナとして使用できる特別なフィールド タイプです。

クレジット カード

このフィールド タイプには 2 つのフィールドが含まれます。ひとつはユーザーがクレジット カードの種類を選択できるフィールドで、もうひとつはユーザーがクレジット カードの番号を入力できるフィールドです。

このフィールド タイプは、様々な種類のクレジット カードで許可されている有効な番号の範囲と組合せを使って、クレジット カードを検証します。Luhn 式、または次のクレジット カードの種類に基づく検証を指定できます: アメリカン エクスプレス、ダイナース クラブ、カルテ ブランシュ、ダイナース クラブ インターナショナル、北米およびカナダのダイナース クラブ、JCB、マエストロ、マスター カード、Solo、Switch、Visa、Visa エレクトロン。



The image shows a dropdown menu titled "Card Type". The menu is open, displaying a list of credit card brands. The "American Express" option is currently selected and highlighted in blue. The list includes: American Express, Diners Club Carte Blanche, Diners Club International, Diners Club US and Canada, JCB, Maestro, MasterCard, Solo, Switch, VISA, and Visa Electron.

このフィールド タイプの設定は、/sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Complex/Credit アイテムに含まれています。

パスワード – 確認

このフィールド タイプには、[パスワード] フィールドと [確認] フィールドが含まれます。これらはパスワードを作成するために使用されます。これらのフィールドに入力された文字はすべてマスクされます。



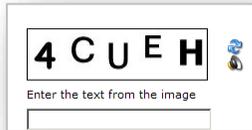
The image shows two input fields. The top field is labeled "Password" and the bottom field is labeled "Confirmation". Both fields are currently empty and have a light gray background, indicating they are masked.

このフィールド タイプの設定は、sitecore/System/Modules/Web Forms for Marketers/Settings/Field Types/Complex/Password Confirmation アイテムに含まれています。

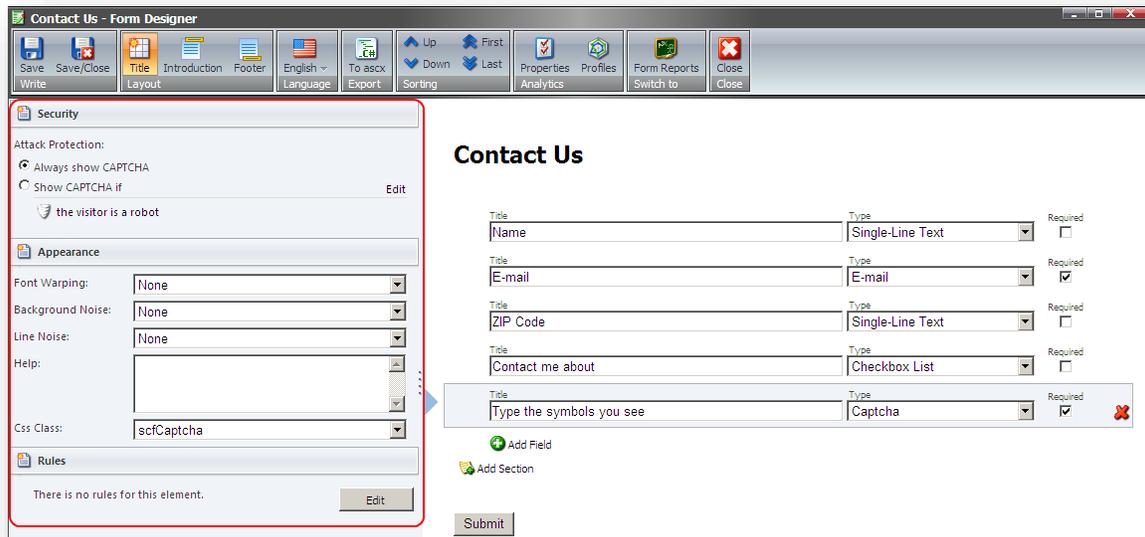
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

キャプチャ

このフィールド タイプには、画像フィールドと テキスト フィールドの 2 つのフィールドが含まれます。ユーザーは、画像フィールドからのテキストを [テキスト] フィールドに入力する必要があります。このフィールドを使用することで Web サイト上でのロボットによる送信を防ぐことができます。



フォーム フィールドのプロパティで、キャプチャ フィールドで使用されるフォント ワープ、バックグラウンド ノイズ、ライン ノイズを設定できます。



ユーザーは次の 3 つのオプションを使ってキャプチャ フィールドを設定することができます。

- Visitor are robots
- A suspicious visitor is detected
このオプションは、ユーザーがフォーム送信回数と時間(分)の閾値を入力する必要があります。

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.

- Suspicious form activity detected
このオプションは、ユーザーがフォーム送信回数と時間(分)の閾値を入力する必要があります。

最後の 2 つのオプションにおいては、ユーザーが入力できる値の範囲を設定することができます。

A suspicious visitor is detected

Sitecore.forms.config ファイルで、*WFM.SessionThreshold* パラメーターの値を変更します。デフォルトでは、値は 1 分間に 2 回から 60 分間に 100 回まで(2/1-100/60) と指定されています。

Suspicious form activity detected

Sitecore.forms.config ファイルで、*WFM.ServerThreshold* パラメーターの値を変更します。デフォルトでは、値は 1 分間に 2 回から 60 分間に 100 回まで(2/1-100/60) と指定されています。

2.4.1 リスト アイテム

リスト フィールド タイプには、次のフィールド タイプが含まれます。

- ドロップ リスト
- リスト ボックス
- ラジオ ボタンのリスト
- チェック ボックスのリスト

次を行うことによって、リストに表示されるアイテムを指定することができます。

- 手動で名前を入力する
- サイトコア アイテムを選択
- XPath クエリを使用する
- サイトコア クエリを使用する
- 高速 クエリを使用する

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.

最初の 2 つのメソッドは、主にマーケターによって使用されます。「手動で名前を入力する」と「サイトコア アイテムを選択」についての追加情報は、『[Web Forms for Marketers User Guide](#)』を参照してください。

最後の 3 つのメソッドは、特定の Sitecore アイテムを検索し、リスト フィールドの値として使用する場合に使用できません。

リスト フィールド タイプは、バリュー-テキスト のコンセプトを使用します。これはフィールドの値には、ユーザーに表示されるテキストと、データベースに保存される値が含まれることを意味します。リスト アイテムの [テキスト] フィールドは、通常フォームをローカライズしたり、ユーザー フレンドリなテキストを表示するために使用されます。

XPath クエリを使用する

この方法は、XPath クエリを使って Sitecore アイテムを選択するために使用されます。選択された Sitecore アイテムは、リスト フィールド の値として使用されます。XPath クエリについての追加情報は、『[Web page](#)』を参照してください。

サイトコア クエリを使用する

この方法は、Sitecore クエリを使って Sitecore アイテムを選択するために使用されます。選択された Sitecore アイテムはリスト フィールドの値として使用されます。Sitecore クエリについての追加情報は、『[Sitecore Developers Network](#)』を参照してください。

高速 クエリを使用する

高速 クエリを使用する 方法は、高速 クエリを使って Sitecore アイテムを選択するために使用されます。選択された Sitecore アイテムは、リスト フィールドの値として使用されます。高速クエリについての追加情報は、『[Sitecore Developers Network](#)』を参照してください。

リスト アイテムのローカライズ

リスト アイテムの定義済みの値を翻訳することができます。

クエリ を使って指定されたリスト アイテムをローカライズするには、次の手順に従います。

1. リスト アイテムとして使用する すべての Sitecore アイテムにおいて同じフィールドを翻訳します (たとえば **表示名** フィールドなど)。

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.

2. **フォーム デザイナー**で、フォームの編集を開始し、新しい言語に切り替えます。
3. **[リスト アイテム]** ウィザードで、**テキスト** の値として翻訳したフィールドを選択します。
[プレビュー] は言語に依存するため 即座に翻訳されたアイテムを確認することができます。

2.5 検証

モジュールには、ユーザーがフィールドに基本的な検証を追加可能にする定義済みの検証が含まれています。カスタムの検証も定義することもできます。これについては、『*Web Forms for Marketers User Guide*』を参照してください。

また、モジュールには、デフォルトで提供されるいくつかのフォームのフィールドタイプで使用される組込の検証が含まれます。

デフォルトの検証は、`sitecore/System/modules/Web Forms for Marketers/settings/validation` アイテムの配下に配置されます。必須チェックの検証は、`/sitecore/system/Modules/Web Forms for Marketers/Settings/System/System Validation/NotEmpty` アイテムです。

デフォルトでは次の検証があります。

検証	説明
文字をカウント	文字数を確認します。文字数の最小数と最大数を設定することができます。
日付	入力された値が日付かどうかを確認します。
電子メール	入力された値が電子メール アドレスのフォーマットかどうかを確認します。
番号	入力された値が数字かどうかを確認します (負数と整数を使用することができます)。
数字範囲	入力された値が指定された数の範囲内にあるかどうかを確認します。
正規表現パターン	入力された値が指定したルールを満たすかどうかを確認します。
クレジットカード	クレジットカードの種類に基づいて、クレジットカード番号が有効であるかを確認します。
パスワードと 確認を比較	[パスワード] と [確認] フィールドに入力された値を比較します。
キャプチャ	画像上に表示されているテキストと、ユーザーによって入力された値を比較します。

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.

モジュールは 2 つの検証用のテンプレートをインストールします。

- /sitecore/Templates/Web Forms for Marketers/Validators/BaseValidator.
- /sitecore/Templates/Web Forms for Marketers/Validators/Regular Expression Validator.

Regular Expression Validator テンプレートは、すべてのフィールドを *BaseValidator* テンプレートから継承し、さらに *Validation Expression* というフィールドを含みます。*Regular Expression Validator* テンプレートに基づく検証は、*Validation Expression* フィールドを使用します。*BaseValidator* テンプレートに基づく検証は、クラスで定義された検証を使用します。

検証アイテムは次のフィールドを含みます。

フィールド	説明
Class	検証を処理するクラスの完全名。
Assembly	クラスを含むアセンブリの名前。
Error Message	検証が失敗した場合に <i>ValidationSummary</i> コントロールに表示されるエラー メッセージのテキスト。
Text	検証が失敗した場合に検証コントロールに表示されるテキスト。
Validator Display	検証コントロールにおけるエラーメッセージの表示方法: <ul style="list-style-type: none"> • なし — 検証メッセージをインライン表示しない。 • 静的 — 検証メッセージのスペースはページレイアウトに割り当てられます。 • 動的 — 検証メッセージのスペースは、検証が失敗した場合に動的にページに追加されます。
Enable Client Script	クライアントサイドの検証が有効か否か
Parameters	検証コントロール用の追加のパラメーターです。
Inner Control	検証コントロールが追加される場所を示します。
Validation Expression	検証に使用する正規表現を設定します。(<i>Regular Expression Validator</i> から作成した検証アイテムに限ります)。

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.

2.5.1 “必須” チェック ボックス フィールド検証の実装方法

Web Forms for Marketers モジュールにおいて、チェック ボックス フィールドは“必須”の検証ルールをサポートしません。チェック ボックス フィールドを“必須”にするには、次の方法のいずれかに従います。

チェック ボックスのリスト フィールドを使用する

1. チェック ボックスのリスト フィールドをフォームに追加します。
2. 項目をひとつだけリストに追加します。
3. このチェック ボックス リストを必須としてマークします。

チェックボックスのリスト フィールドのカスタムの検証を作成する

1. System.Web.UI.WebControls.BaseValidator クラスから継承してクラスを作成します。次のコード サンプルを参照してください。

```
class CheckboxValidation : BaseValidator
{
    protected CheckBox ctrToValidate;
    protected CheckBox CheckBoxToValidate
    {
        get
        {
            if (ctrToValidate == null)
            {
                ctrToValidate = base.FindControl(ControlToValidate) as CheckBox;
            }
            return ctrToValidate;
        }
    }
    protected override bool ControlPropertiesValid()
    {
        if (base.ControlToValidate == null || base.ControlToValidate.Length == 0)
        {
            throw new HttpException(string.Format("The ControlToValidate property of '{0}' cannot be blank.", this.ID));
        }
        if (this.CheckBoxToValidate == null)
        {
```

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.

```
        throw new HttpException(string.Format("The CheckBoxValidator can only validate controls of  
type CheckBox."));  
    }  
    return true;  
}  
protected override bool EvaluateIsValid()  
{  
    this.ErrorMessage = string.Format(this.ErrorMessage, "{0}", CheckBoxToValidate.Text);  
    //Validate whether checkbox is checked  
    return this.CheckBoxToValidate.Checked == true;  
}  
}
```

2. *sitecore/system/modules/web forms for marketers/settings/validation* フォルダの配下にアイテムを作成します。このアイテムは *BaseValidator* テンプレートに基づく必要があります。
3. **[Assembly]** と **[Class]** フィールドに、カスタムのアセンブリの適切な値を入力します。
4. **[Error Message]** フィールドに、文字列 [*The {0} checkbox must be checked*] を入力します。
5. **[Text]** フィールドに、適切なメッセージを入力します。このフィールドが空白の場合は、その値は **Error Message** の値と同じになります。
6. */sitecore/system/modules/web forms for marketers/settings/field types/simple types/checkbox* アイテムを複製し、たとえば **CheckboxRequired** などに名前を変更します。
メモ: **CheckboxRequired** アイテムでは "必須" チェックボックスを選択しないでください。
7. */sitecore/system/modules/web forms for marketers/settings/field types/simple types/CheckboxRequired* アイテムの **[Validation]** フィールドで、作成したカスタムの検証を選択します。

2.6 送信アクション

Web サイトの訪問者が [送信] をクリックした場合、3 つの種類のアクションが実行されます。

- フォームの検証
- 保存アクション
- 成功

すべてのアクションは、/sitecore/System/Modules/Web Forms for Marketers/Settings/Actions アイテムの配下に保存されています。



メモ:

旧バージョンの Web Forms Marketers モジュールをアップグレードする場合、既存のアクションは [保存アクション] フォルダーに移動されます。

アクションの次のフィールドを設定することができます。

- **Class** — サーバーサイドでフォーム データを処理するクラスの完全名。
- **Assembly** — 参照されるクラスを含むアセンブリの名前。
- **Parameters** — フォームのパラメーターに無関係なアクションのパラメーター。これらはグローバルなパラメーターで、すべてのフォーム上で共通です。たとえば、[電子メール メッセージを送信] 保存アクションにおいて、このフィールドは電子メールを送信する SMTP サーバーを指定するために使用されます。

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.

- **Editor** — アクションのパラメーターを変更するために使用するアプリケーションを提供します。

2.6.1 フォームの検証

フォームの検証は、複数のフォーム フィールドに入力された値を検証します。フォームの検証が失敗した場合、訪問者はフォームに戻され、エラー メッセージが表示されます。後続のその他のフォームの検証や保存アクションが実行されることはありません。

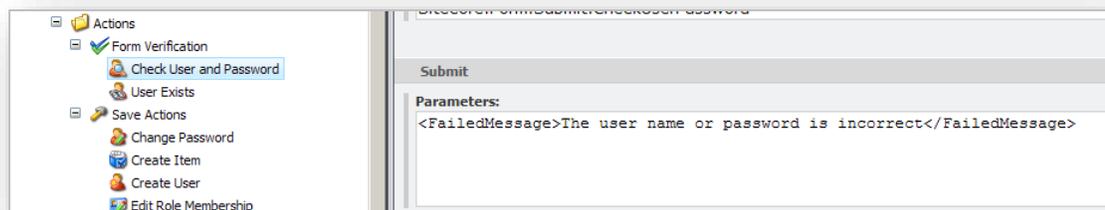
フォームの検証アクションの作成方法についての追加情報は、「フォームの検証アクションの作成方法」セクションを参照してください。

フォームの検証のエラー メッセージの変更

デフォルトおよびローカライズされたフォームの検証のエラー メッセージを設定することができます。ローカライズされたエラーメッセージの設定方法についての追加情報は、『[Web Forms for Marketers User Guide](#)』を参照してください。

デフォルトのエラー メッセージを設定するには、次の手順に従います。

1. **コンテンツ エディター**で、編集するフォームの検証アクションを選択します。
2. **[Parameters]** フィールドで、<FailedMessage> タグにエラー メッセージを入力します。



2.6.2 保存アクション

保存アクションはフォームに割り当てることができ、訪問者がフォーム上で **[送信]** をクリックした場合に実行されます。Web Forms for Marketers モジュールには、デフォルトで 18 の保存アクションがあります。

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.

保存アクションは Sitecore アイテムとして保存され、.net クラスで実装されたアクションを実行します。このクラスは、保存アクション アイテムの Class および Assembly フィールドで指定されます。保存アクション アイテムは、/sitecore/templates/Web Forms for Marketers/Actions/Submit Action テンプレートに基づきます。保存アクション アイテムは、sitecore/ system/modules/web forms for marketers/settings/actions/save actions アイテムの配下に配置されます。

保存アクション アイテムは次の設定フィールドを含みます。

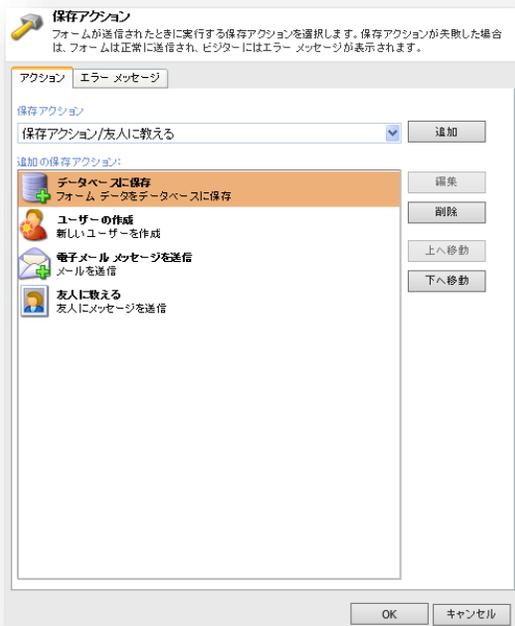
フィールド	説明	サンプルのフィールド バリュー
Assembly	関連付けられたクラスを含むアセンブリの名前。	Sitecore.Forms.Custom.dll
Class	名前空間を含む関連付けられたクラスの名前。	Sitecore.Form.Submit.SendMessage
Parameters	保存アクションのパラメーター。アクションの処理でパラメーターを使用するには、プロパティをカスタムのアクションクラスに追加し、プロパティ名をパラメーターと同じにします。	<pre><DefaultDomain>extranet</DefaultDomain> アクションクラスに DefaultDomain プロパティが定義されている場合、このプロパティは extranet で初期化されます: /// <summary> /// Gets or sets the default domain. /// </summary> /// <value>既定のドメイン</value> public string DefaultDomain { get; set; }</pre>
Client Action	このフィールドはステージング環境でのみ使用されます。このチェックボックスを選択しない場合、その保存アクションは Slave から Master サーバーへ転送され、Master サーバーで実行されます。このチェックボックスを選択した場合は、保存アクションは Slave サーバーで実行されます。	選択.

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.

フィールド	説明	サンプルのフィールド バリユー
Editor	Sitecore ユーザーがフォーム デザイナーで保存アクションのパラメーターを編集するとき使用するコントロールです。	control:Forms.MappingFields
QueryString	編集者向けの追加の設定です。	Fields=Login Login,Password Password

カスタムの保存アクションを作成することができます。保存アクションの作成についての追加情報は、「保存アクションの作成」セクションを参照してください。

フォームに複数の保存アクションがある場合、保存アクションのダイアログに表示されている順番に実行されます（上部の要素から開始）。



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.

電子メール メッセージを送信、SMS を送信、MMS を送信 という、ユーザーに通知を送信する 3 つの保存アクションがあります。これらの保存アクションはすべて web.config ファイルで指定されたデフォルトの SMTP サーバー設定を使用します。保存アクションごとに別の SMTP サーバー設定を指定することができます。

電子メール メッセージを送信

このアクションに別の SMTP 設定を使用するには、**コンテンツ エディター**で、*Sitecore/System/Modules/Web Forms for Marketers/Settings/Actions/Save Actions/Send Email Message* に移動し、**[Parameters]** フィールドで適切な値を設定します。

たとえば、web.config ファイルで定義されたパラメーターとは異なる *Host* および *From* パラメーターを使用するには、**[Parameters]** フィールドに次のコードを入力します。



[Parameters] フィールドで SMTP サーバーの資格情報を指定することもできます。

```
<Login>login</Login> <Password>pass</Password>
```

SMS を送信

このアクションに別の SMTP 設定を使用するには、**コンテンツ エディター**で、*Sitecore/System/Modules/Web Forms for Marketers/Settings/Actions/Save Actions/Send SMS* に移動し、**[Parameters]** フィールドで適切な値を設定します。

たとえば、web.config ファイルで定義されたパラメーターとは異なる *Host* および *From* パラメーターを使用するには、**[Parameters]** フィールドに次のコードを入力します。

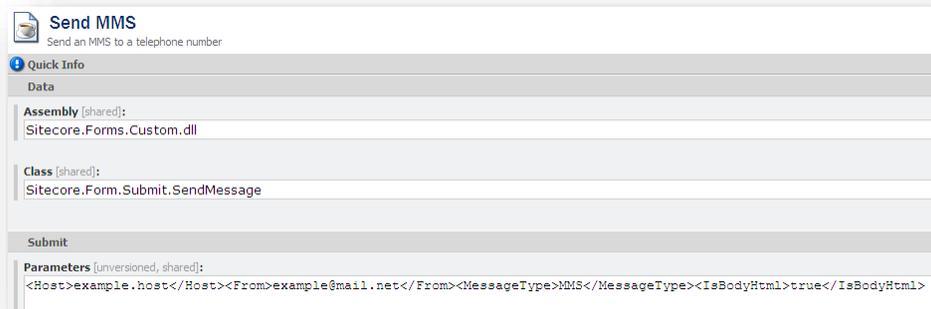
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.



MMS を送信

このアクションに別の SMTP 設定を使用するには、**コンテンツ エディター**で *Sitecore/System/Modules/Web Forms for Marketers/Settings/Actions/Save Actions/Send MMS* に移動し、**[Parameters]** フィールドで適切な値を設定します。

たとえば、Web.config ファイルで定義されたパラメーターとは異なる *Host* および *From* パラメーターを使用するには、**[Parameters]** フィールドに次のコードを入力します。



2.6.3 成功

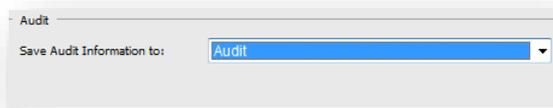
このアクションを使用することによって、Web サイトの訪問者がフォームの送信に成功した場合に表示する Web サイトのページまたはメッセージのどちらかを選択することができます。これはフォームの送信で実行される最後のアクションです。このアクションに関連するパイプラインは、successAction パイプラインと呼ばれます。

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.

2.6.4 監査情報

デフォルトの保存アクションの中には、Sitecore のセキュリティ モデル上のユーザーまたはロールを作成または編集するものがあります。これらはセキュリティ アクションと呼ばれます。これらには、[ユーザーの作成]、[ロールのメンバーシップの編集]、[ユーザー ログイン]、[パスワードを使ってユーザーがログインする] 保存アクションが含まれます。これらはすべてユーザー情報に影響を及ぼす可能性があるため、実行されたアクションを記録するためにユーザー プロファイルに監査情報を登録しておく便利です。

すべてのセキュリティ アクションには [監査情報を次の場所に保存する] があります。これは、監査情報を書き込むことができるユーザー プロファイルのフィールドを表示するドロップダウン リストです。

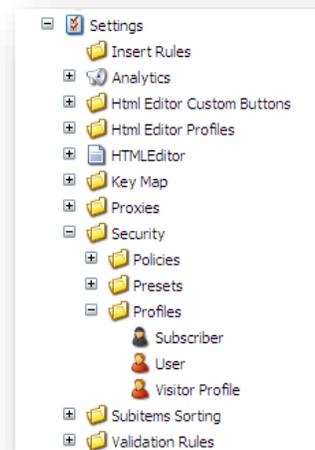


デフォルトでは、すべての Rich Text、html、text、Memo、Multi-Line Text、Single-Line Text フィールドは、監査情報を登録するために使用できます。監査情報を登録できるフィールド タイプは、Sitecore.forms.config ファイルで WFM.AuditAllowedTypes の設定を使用して構成することができます。

```
<setting name="WFM.AuditAllowedTypes" value="|Rich Text|html|text|Multi-Line Text|Single-Line Text|memo|" />
```

すべてのユーザー プロファイルは、Core データベースの `/sitecore/system/Settings/Security/Profiles` フォルダー配下のアイテムです。

Visitor profile はデフォルトで使用されるプロファイルアイテムです。各フォーム アイテムには [ユーザー プロファイル] フィールドにこのユーザー プロファイルへの参照があります。



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.

2.7 レポート

どのように Web サイトの訪問者がフォームに入力を行ったかについての情報は、Web Form モジュールに含まれる様々なレポートを使って確認することができます。

Web Forms for Marketers モジュールでは、フォームの送信に成功しなかったユーザーについての情報を含むフォームのドロップアウト レポートの作成がサポートされています。この機能は AJAX 機能を使用して、個々のフィールドごとに追跡を行い、ユーザー入力をアナリティクス データベースに記録することによって実現されています。

システム イベントはセッションの追跡で記録されます。システム イベントについての追加情報は、「イベントと訪問の詳細 (セッションの追跡)」セクションを参照してください。

フォーム レポートを開くには、次の手順に従います。

1. Sitecore デスクトップにログインします。
2. Sitecore、すべてのプログラム、Web Forms for Marketers、フォーム レポート と順番にクリックします。
3. レポートを表示するフォームを選択します。

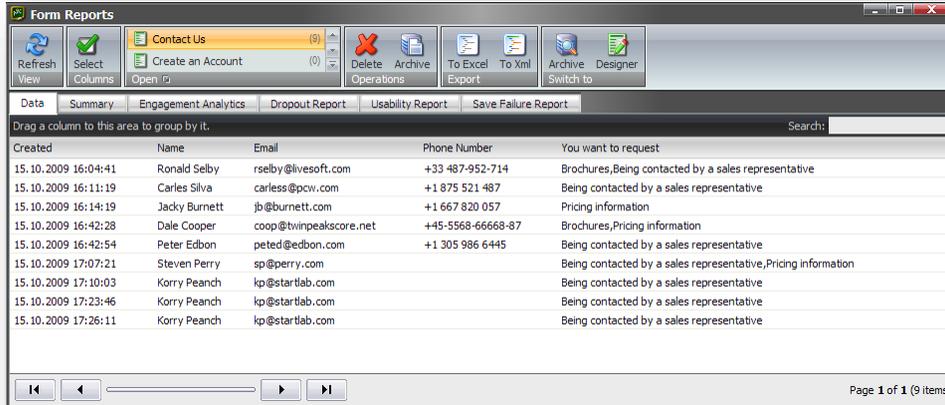
または、**コンテンツ エディター**で、/sitecore/System/Modules/Web Forms for Marketers/Website アイテムの配下のフォームを選択し、**フォーム レポート**をクリックします。

次のレポートを使用することができます。

- データ
- 概要
- エンゲージメント アナリティクス
- ドロップアウト レポート
- ユーザビリティ レポート
- 失敗のレポートを保存する

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.

フォーム レポートは、**フォーム レポート** ウィンドウの個別のタブに表示されます。



すべてのレポートはデフォルトで最新の 200 件のレコードを表示します。この数を変更するには、Web レポート デザイナーでレポートを開き、適切な SQL クエリの値を変更します。

2.7.1 サポートされているデータベース

Web Forms for Marketers モジュールのレポートは、次のデータベース サーバーをサポートします。

レポート	サポートされているデータベース
データ	MSSQL, SQLite, Oracle
概要	MSSQL, SQLite, Oracle
エンゲージメント アナリティクス	MSSQL
ドロップアウト レポートとサブ レポート	MSSQL
ユーザビリティ レポートとサブ レポート	MSSQL
失敗のレポートを保存する	MSSQL

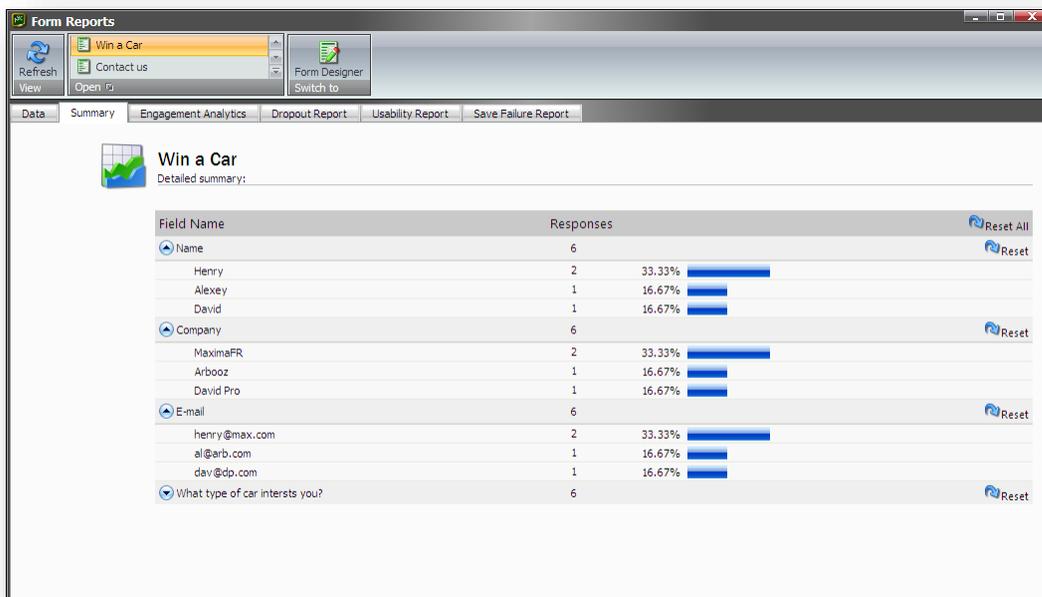
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

2.7.2 データ リクエストのタイムアウト

デフォルトでは、データ レポートのリクエスト タイムアウトは 180 秒です。この値は、`¥WebSite¥App_Config¥Include¥ Sitecore.forms.config` ファイルで `WFM.CommandTimeout` パラメーターを変更することによって修正できます。

2.7.3 概要

デフォルトでは 3 つのレコードのみが表示されます。



レポートに表示されるレコードの数を変更するには、`¥WebSite¥App_Config¥Include¥ Sitecore.forms.config` ファイルで `WFM.MostPopularApplicantCount` パラメーターを変更します。

デフォルトではフィールドの値が 80% を超過する場合は、青色のストライプ線が緑色になります。この値は、`¥WebSite¥App_Config¥Include¥ Sitecore.forms.config` ファイルで `WFM.RelevantScale` パラメーターを変更することによって修正することができます。

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.

2.8 モジュールへのユーザー アクセスの設定

ユーザーのモジュール機能へのアクセス権を設定するには、管理者は次のロールをユーザーに割り当てます。

- Sitecore Client Form Author
- Sitecore Client Developing
- Analytics Maintaining
- Analytics Reporting
- Sitecore Marketer Form Author
- Sitecore Client Securing

ユーザーにモジュールへの最小限のアクセス権を与えるには、ユーザーに *Sitecore Client Form Author* ロールを割り当てます。

ユーザーにモジュールのすべての機能へのアクセスを与えるには、次のロールをユーザーに割り当てます。

- Sitecore Marketer Form Author
- Sitecore Client Developing
- Sitecore Client Securing

2.8.1 Web フォーマットのセキュリティ ロール

Web Forms for Marketers モジュールを対象に作成されたセキュリティ ロールについて簡単に説明します。

Sitecore Client Form Author

Sitecore Client Form Author ロールは、ユーザーに Web Forms for Marketers モジュールの最小限の機能へのアクセスを提供します。その他すべてのロールはユーザーのアクセス権を拡張します。このモジュールによってユーザーは次のことを行うことができます。

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.

- 新しいフォームの挿入
- 既存のフォームの編集
- データ レポートの表示
- 概要レポートの表示

Sitecore Client Developing

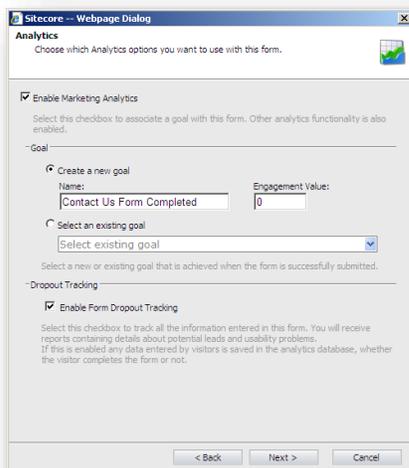
Sitecore Client Developing ロールを使用することによって、ユーザーは**フォーム デザイナー**で **[エクスポート]** の **[ascx ファイル]** ボタンを使用することができます。



Analytics Maintaining

Analytics Maintaining ロールを使用することによって、ユーザーは次のことができます。

- **[新しいフォームを作成]** ウィザードで **[アナリティクス]** ページの使用。

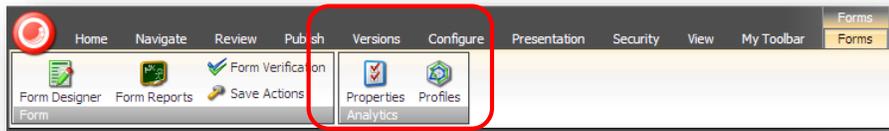


Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

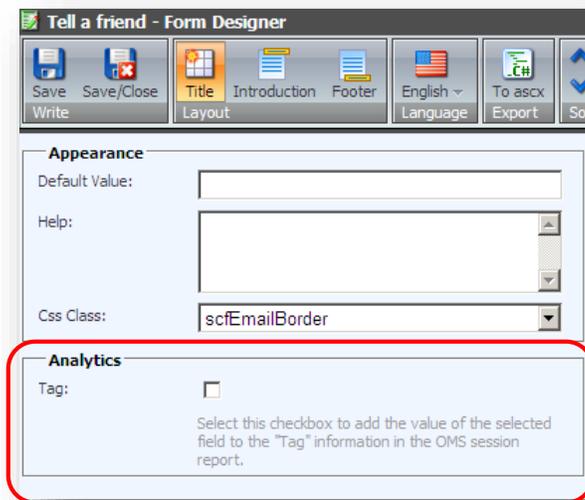
- フォーム デザイナーのリボン上で [アナリティクス] セクションの使用。



- コンテンツ エディターのリボン上で [アナリティクス] セクションの使用。



- タグの使用。



Analytics Reporting

Analytics Reporting ロールによって、ユーザーは次のことができます。

- ドロップアウト レポートの表示

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.

- ユーザビリティ レポートの表示
- 失敗したレポートの保存の表示

Sitecore Marketer Form Author

Sitecore Marketer Form Author ロールは、次のロールからアクセス権を継承します。

- Sitecore Client Form Author
- Analytics Maintaining
- Analytics Reporting

Sitecore Marketer Form Author ロールのメンバーは、これらのロールに割り当てられたすべての権限を所有します。

Sitecore Client Securing

Sitecore Client Securing ロールを使用することによって、ユーザーは次のことができます。

- ユーザーの作成 保存アクションの編集。
- ロールのメンバーシップの編集 保存アクションの編集。
- パスワードの変更 保存アクションの編集。

2.9 イベントと訪問の詳細(セッションの追跡)

訪問の詳細(セッションの追跡)は、ユーザーが Web サイト上で実行したすべてのアクティビティを記録する Sitecore エンゲージメント アナリティクス機能です。これにはユーザーがいつ、どのページを訪問したかという情報も含まれます。

訪問の詳細(セッション) レポートは、訪問者がフォームを入力した場合にトリガーされるイベントも一覧に表示します。これらのイベントは、開発の目的に使用できる標準的な Sitecore エンゲージメント アナリティクスのページ イベントです。レポートはフィールド イベントとフォーム イベントの両方をリスト表示します。

Form check action error、Form submit、Form conversion イベントを除く すべてのイベントにおいて、アナリティクス データベースの [**Datakey**] フィールドはフォームの GUID と同じです。

ユーザーがフィールドで情報を変更することによって、フォームを送信する同じ試みが発生するため、同じフィールドで複数のフィールド イベントが発生する可能性があります。

フォーム レポートにリスト表示されるイベントには次があります。

Field Completed

このイベントは、フォーム上のフィールドへの入力が完了してタブされた場合、または他の場所をクリックした場合にトリガーされます。これは AJAX を使用して行われます。

Field Not Completed

このイベントは、訪問者が必須入力フィールドを入力しなかったことが原因でフィールドの検証が失敗した場合にトリガーされます。

Field Out of Boundary

このイベントは、フィールドに定義された境界範囲を超えて値が入力されたことが原因でフィールドの検証が失敗した場合にトリガーされます。このイベントは [テキスト] および [パスワード] フィールドの 最小および最大長で使用されます。さらに [日付] および [番号] フィールドでも使用されます。

Form Check Action Error

このイベントは検証アクションが失敗した場合にトリガーされます。これは失敗ではなく、イベントです。検証アクションが失敗した場合、訪問者はフォームに戻されます。他の検証アクションまたは保存アクションが開始されることはありません。

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.

Form Save Action Failure

このイベントは、保存アクションが失敗した場合にトリガーされます。これは失敗であり、イベントではありません。[エラーの発生] プロパティが有効になっています。

Form Submit

このイベントは、訪問者が [送信] ボタンをクリックした場合、または ENTER キーを押した場合にトリガーされます。これはフォームの送信が試みられたことを示します。

Invalid Field Syntax

このイベントは、フィールドに入力されたデータが特定のシンタックスのチェックに失敗したことが原因でフィールドの検証が失敗した場合にトリガーされます。このイベントは、[テキスト] および [パスワード] タイプの正規表現のフィールドの検証、[電子メール] フィールドにおけるフィールドの検証で使用されます。

Submit Success

このイベントは、送信アクションがエラーを返さなかった場合にトリガーされます。このイベントは、フォームのコンバージョン イベントと共に書き込まれ、主にフォームのドロップアウト、フォームのユーザビリティ、フォームの失敗のレポート向けに必要な SQL ステートメントを作成するために使用されます。

Form Conversion

このイベントは、にフォームの送信が成功し、Form Submit イベントがトリガーされた後にトリガーされます。これは、フォームと関連付けられたゴールの達成が成功したことも示します。

Form Begin (システムイベント)

これはレポートには表示されないシステム イベントです。Form Being はユーザーが Web フォームの使用を開始した時 (フィールドの入力または送信ボタンのクリック) にトリガーされます。

重要

Form Begin イベントはレポートにとって重要です。フォームのドロップアウト レポートに影響を与える場合があるため、このイベントをカスタムコードからトリガーすることは推奨していません。

Form Dropout (システム イベント)

これはレポートには表示されないシステム イベントです。フォームのドロップアウトは、ユーザー セッションが切れ、ユーザーが Web フォームの送信に成功しなかった場合にトリガーされます。

重要

フォームのドロップアウト イベントはレポートングにとって重要です。フォームのドロップアウト レポートングに影響を与える場合があるため、このイベントをカスタムコードからトリガーすることは推奨していません。

2.10 マルチサイトの実装

Web Forms for Marketers モジュールはマルチ サイト環境をサポートします。つまり、管理者は様々な Web サイトごとに異なるフォームのロケーションおよび設定 を定義することができます。これは、.config ファイルのサイトの定義の formsRoot 属性によって行われます。

この属性の値は、次を定義する Sitecore パッチです。

- 現行サイト上のフォームを保存するフォルダー
- 現行サイト上のフォームの外観と色の設定
- アクセス権

formsRoot パラメーターは、アイテムのパスまたはターゲット アイテムの ID のどちらかを含む必要があります。ターゲット アイテムは、/sitecore/Templates/Web Forms for Marketers/Forms Folder テンプレートに基づく必要があります。

たとえば、formsRoot パラメーターは web.config ファイルで次のように定義できます。

```
<sites>
  <site
    name="samplesite"
    virtualFolder="/"
    physicalFolder="/"
    rootPath="/sitecore/content"
    startItem="/forms" database="web" domain="extranet"
    formsRoot="/sitecore/System/modules/Web Forms for Marketers/local forms"
    ...
```

これは ID を使って Sitecore.forms.config ファイルで定義することができます。

```
<site name="website">
  <patch:attribute name="formsRoot">
    {F1F7AAB6-C8CE-422F-A214-F610C109FA63}
  </patch:attribute>
</site>
</sites>
```

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.

formsRoot パラメーターを/App_Config/Include/ Sitecore.forms.config ファイルで定義することを推奨します。このアプローチによって重複する値がないことを確認します。

Web サイトに formsRoot 属性が定義されていない場合、新しいフォームは/sitecore/System/Modules/Web Forms for Marketers/Local Forms フォルダーに保存されます。

2.11 マルチサーバー環境

マルチサーバー環境を設定することによって、Web Forms for Marketers モジュールをスケールすることができます。このような設定では、モジュールはコンテンツ デリバリー サーバーから Master サーバーへの保存アクションの実行を一部委任します。

マルチサーバー環境を設定するには、次の手順に従います。

1. Master サーバー上でモジュールをインストールします。モジュールのインストールについての追加情報は、『[SDN website](#)』を参照してください。
2. すべてのコンテンツ デリバリー サーバー上にモジュールをデプロイします。

2.11.1 コンテンツ デリバリー サーバー上にモジュールをデプロイする

Master サーバーにモジュールをインストールし、すべてのコンテンツ デリバリー サーバー上にモジュールをデプロイします。

コンテンツ デリバリー サーバー上にモジュールをデプロイするには、次の手順に従います。

1. Master データベースをすべてのコンテンツ デリバリー サーバーにパブリッシュします。Master データベースのパブリッシュについての追加情報は、『[拡張性ガイド](#)』の 3.1 リモート パブリッシュ ターゲットを構成する を参照してください。
2. モジュール ファイルをコンテンツ デリバリー サーバーにデプロイします。

コンテンツ デリバリー サーバーにクライアント インターフェースがない場合は、コンテンツ マネージメント サーバーからすべてのコンテンツ デリバリー サーバーに次のファイルとフォルダーをコピーします。

- Bin¥sitecore.forms.core.dll
- Bin¥Sitecore.Forms.Custom.dll
- Bin¥MSCaptcha.dll
- Bin¥system.data.sqlite.dll

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.

- App_Config¥Include¥Sitecore.forms.config
- App_Config¥Include¥Captcha.config
- Sitecore modules¥shell¥Web Forms for Marketers¥Themes¥*.*
- Sitecore modules¥web¥Web Forms for Marketers¥scripts¥*.*
- Sitecore modules¥web¥Web Forms for Marketers¥Tracking.aspx
- Sitecore modules¥web¥Web Forms for Marketers¥control¥*.ascx
- Sitecore modules¥web¥Web Forms for Marketers¥UI¥UserControl¥*.ascx

コンテンツ デリバリー サーバーにクライアント インターフェースがある場合、インストール ウィザードを使用してコンテンツ デリバリー サーバー上に Web Form for Marketers パッケージをインストールします。

3. 次の接続文字列をすべてのコンテンツ デリバリー サーバーの connectionstrings.config ファイルに追加します。

```
<add name="remoteWfmService"
connectionString="url=http://[masterserver]/sitecore%20modules/shell/Web%20Forms%20for%20Marketers/Staging/WfmService.asmx;user=[domain¥username];password=[password];timeout=60000" />
```

設定する値

[masterserver] – Master サーバーの IP またはホスト名。

[domain¥username] – Sitecore ユーザー (フル ネーム)。必要なアクセス権は、フォームが使用しているアイテムに依存します。

[password] – ユーザーのパスワード。

メモ

remoteWfmService 接続文字列を、すべてのコンテンツ デリバリー サーバーの connectionstrings.config ファイルに追加すると、[**Client Action**] のチェックボックスが選択されていないすべての保存アクションは、現行のサーバー上で実行されるのではなく、コンテンツ マネージメント サーバー上で順番に実行されます。その後、[**Client Action**] のチェックボックスが選択されているすべての保存アクションは、コンテンツ デリバリー サーバー上で順番に実行されます。アップロ

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.

ードされたすべてのファイルは、コンテンツ マネージメント サーバーにもアップロードされます。**Client Action** チェックボックスについての追加情報は、「保存アクション」セクションを参照してください。

2.12 ライブ モードでの Web フォームの使用

Sitecore は “ライブモード” と呼ばれる Master データベースを使用して直接 Web サイトを公開する実行方法をサポートしています。ライブ モードで動作させることによって、コンテンツをパブリッシュする必要性がなくなります。サイトコア クライアントのプレビューでサイトを表示することと似ています。ライブ モードで動作するように設定された Web サイトは、通常の Web サイトと同じように機能します。ライブ モードは、デフォルトの Web サイトと同じように、パブリッシュ制限およびワークフローを取り扱います。

ライブモードで Web Forms for Marketers モジュールを動作させるには、web.config ファイルを編集する必要があります。

1. web.config ファイルで、適切な <site> セクションを見つけ、database="web" を database="master" に変更します。
2. <modules_website> セクションを見つけ、database="web" を database="master" に変更します。
3. web.config ファイルの<sites> セクションは次のようになります。

```
<sites>
...
  <site name="modules_website"...database="master".../>

  <site name="website"...database="master".../>
...
</sites>
```

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

Web フォームの開発者のための注意事項

この章では、開発者を対象に Web Forms for Marketers モジュールのカスタマイズ方法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。

- Sitecore.forms.config
- カスタム プロセッサーを使用したモジュールの挙動の修正
- Web ページへの Web フォームの挿入方法
- データ プロバイダーの設定方法
- 標準機能の拡張/オーバーライド方法
- モジュールで使用されるフィールドの階層構造
- 新しいフィールド タイプの作成方法
- 保存アクションの作成
- アクション エディターの作成方法
- フォームの検証アクションの作成方法
- 送信された Web フォーム データのアクセス方法
- CSS テーマの使用方法

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.

- CSS スタイルの設定方法
- ASCX へのエクスポート方法
- ASCX コントロールをページに追加する方法
- モジュールを再インストールする方法
- モジュールをアンインストールする方法

3.1 Sitecore.forms.config

このセクションでは、¥Website¥App_Config¥Include フォルダに配置されている Sitecore.forms.config ファイルの主な設定パラメーターについて説明します。

3.1.1 Events

Sitecore.forms.config ファイルには、forms:save という唯一の Web フォーム特有のイベントがあります。このイベントは、Web フォームが送信されたときに、保存アクションが実行される前に開始されるイベントです。

3.1.2 Pipelines

このモジュールは次のパイプラインを使用します。

パイプライン	説明
formUploadFile	Web フォームの ファイルのアップロード フィールドを使用してファイルがアップロードされた場合に動作します。
successAction	Web フォームの送信が成功した場合に動作します。デフォルトのプロセッサは、ユーザーを成功ページにリダイレクトするか、または成功メッセージを生成します。
errorSubmit	ユーザーが Web フォームを送信する際に例外が発生した場合に動作します。
errorSave	モジュールが保存アクションを処理する際に例外が発生した場合に動作します。
errorCheck	モジュールが検証アクションを処理する際に例外が発生した場合に動作します。
exportToXml	モジュールが Web フォーム データを データ ビューアーから XML ドキュメントにエクスポートする場合に動作します。
exportToExcel	モジュールが Web フォーム データをデータ ビューアーから Microsoft Excel にエクスポートする場合に動作します。
parseAscx	モジュールが .ascx ファイルにエクスポート ダイアログ ボックスを起動する場合に動作します。このパイプラインは Web フォームを解析し、.ascx コントロールを出力します。
exportToAscx	ユーザーが .ascx ファイルにエクスポート ダイアログ ボックスで [ダウンロード] をクリックした場合に動作します。このパイプラインはダウンロード用に .ascx コントロールを生成します。

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.

パイプライン	説明
auditRender	モジュールがセキュリティに関する保存アクションを実行した場合に (たとえば ユーザーの作成 保存アクション)、監査情報をユーザー プロファイルに書き込みます。
setCustomCrmProperty	CRM 保存アクションによって CRM エンティティが作成された場合に、CRM エンティティのフィールドを変更できるようにします。CRM 保存アクションについての追加情報は、『 SDN article 』を参照してください。
setSystemCrmProperty	CRM 保存アクションによって CRM エンティティが作成された場合に、CRM エンティティのシステム フィールドを変更できるようにします。CRM 保存アクションについての追加情報は、『 SDN article 』を参照してください。
processMessage	モジュールが電子メール メッセージを送信 保存アクションの電子メール メッセージを生成する場合に動作します。

3.1.3 Commands

このセクションには Web Form for Marketers モジュールに特有の UI コマンドが含まれています。

たとえば、ユーザーがコンテンツ エディターで [プレゼンテーション] タブの [挿入] ボタンをクリックした場合に、forms:insert コマンドが実行されます。

3.1.4 Settings

このセクションには Web Forms for Marketers モジュールを設定するために使用できる設定が含まれています。各設定には、この設定がどのように機能するのかを理解するのに役立つ短い説明文が含まれます。

3.2 カスタム プロセッサを使用したモジュールの挙動の修正

このセクションには、カスタム プロセッサを使用したモジュールの動作を変更する方法を示すサンプル アクションが含まれます。

3.2.1 ProcessMessage パイプラインを使用した電子メール メッセージの設定方法

この例では、ProcessMessage パイプラインを使用した 電子メール メッセージを送信 保存アクションの電子メール メッセージの設定方法について説明します。

電子メール メッセージを設定するには、次のアクションを実行します。

1. 次のサンプル コードを使用してプロセッサ クラスを作成します。

```
using Sitecore.Form.Core.Pipelines.ProcessMessage;
// 本プロセッサは メール本文の末尾にノートを追加します
public class AddTextToBody
{
    public void Process(ProcessMessageArgs args)
    {
        string additionalText = "<p>This message was sent using the Sitecore Web Forms for Marketers
module.</p>";
        args.Mail.Append(additionalText);
        /*
        * 同様に SUBJECT, TO, CC and BCC メッセージフィールドを変更することもできます
        * args.Subject.Append(" subject text");
        * args.To.Append("; secondrecipient@mail.net");
        * args.CC.Append("; secondrecipient@mail.net");
        * args.BCC.Append("; secondrecipient@mail.net");
        * args.From = "sender@mail.net";
        */
    }
}
```

2. Sitecore.forms.config ファイルの

Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage プロセッサの前に、新しいプロセッサを登録します。

```
<processMessage>
```

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.

```
...
<processor type="YourNamespace.AddTextToBody,YourAssemblyName" />
<processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="SendEmail"/>
</processMessage>
```

3.2.2 カスタム プロセッサーを使用した SMS/MMS の送信方法

MMS を送信 および SMS を送信 保存アクションは、メッセージの送信を SMTP サーバーを介して MMS/SMS ゲートウェイに委任します。

processMessage パイプラインを使ってこの挙動を変更し、メッセージを送信することができます。たとえば、サードパーティーの有料の Web サービスを使用するように変更できます。

1. 次のサンプル コードを使用して新しいプロセッサーを作成します。

```
namespace Sitecore.Form.Core.Pipelines.ProcessMessage
{
    using System.IO;
    using System.Net;

    public class SendSMSorMMS
    {
        public void Process(ProcessMessageArgs args)
        {
            if (args.MessageType == MessageType.MMS || args.MessageType == MessageType.SMS)
            {
                WebClient wc = new WebClient();
                wc.Credentials = (NetworkCredential)args.Credentials;

                wc.QueryString.Add("sendto", args.Recipient);
                wc.QueryString.Add("message", args.Mail.ToString());
                if (!string.IsNullOrEmpty(args.From))
                {
                    wc.QueryString.Add("from", args.From);
                }
                using (Stream responseStream = wc.OpenRead("https://3rdparty.smsormms.com/"))
                {
                    using (StreamReader responseReader = new StreamReader(responseStream))
                    {
                        responseReader.ReadToEnd();
                        responseReader.Close();
                    }
                }
            }
        }
    }
}
```

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.

```
        responseStream.Close();
    }
}
}
}
}
```

2. 新しいプロセッサを Sitecore.forms.config ファイルに登録します。

```
<processMessage>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="ExpandLinks"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="ExpandTokens"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="AddHostToItemLink"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="AddHostToMediaItem"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="AddAttachments"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="BuildToFromRecipient"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.SendSMSorMMS, MyAssembly"/>
  <processor type="Sitecore.Form.Core.Pipelines.ProcessMessage.ProcessMessage,
Sitecore.Forms.Core" method="SendEmail"/>
</processMessage>
```

3.2.3 ファイルのアップロード フィールドの大容量ファイルアップロードの回避方法

この例では、ユーザーがファイルのアップロード フォーム フィールドで 10 MB 以上のファイルをアップロードすることを禁止する方法を説明します。

ファイルのアップロード フィールドで大容量のファイルをアップロードすることを禁止するには、次のアクションを実行します。

1. 次のサンプル コードを使用して新しいプロセッサ クラスを作成します。

```
using Sitecore.Form.Core.Pipelines.FormUploadFile;

public class UploadingLimitation
{
    public void Process(FormUploadFileArgs args)
```

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.

```
{
  int size = 10485760; // == 10 Mb
  if (args.File.Data.Length > size)
  {
    Sitecore.Diagnostics.Log.Info(string.Format("User {0} tried to upload a file larger than 10 Mb. The file
name is {1}",
      Sitecore.Context.User.Name,
      args.File.FileName), this);
    args.AbortPipeline();
  }
}
```

2. 新しいプロセッサを Sitecore.forms.config ファイルに登録します。

```
<formUploadFile>
  <processor type="YourNamespace. UploadingLimitation, YourAssemblyName"/>
  ...
</formUploadFile>
```

このソリューションを実装した後、ユーザーがファイルのアップロード フォーム フィールドで 10 MB 以上のファイルのアップロードを試みた場合、モジュールはユーザーによるアップロードを回避し、対応するメッセージがログ ファイルに保存されます。

3.3 Web ページへの Web フォームの挿入方法

ユーザーは ページ エディターを使って Web フォームを Web ページに追加することができます。開発者と管理者は、次の方法で Web フォームを Web ページに追加することができます。

- Web フォームを Sitecore の標準的なレンダリングとして挿入します。
- Web フォームを Web コントロールとして挿入します。
- コードビハインド クラスを使用して Web フォームを挿入します。

Web フォームを Sitecore レンダリングとして挿入する

Web フォームは Sitecore レンダリングであるため、コンテンツ エディターで挿入することができます。

1. コンテンツ エディターで、Web フォームを追加するアイテムを選択します。
2. リボンの [プレゼンテーション] タブで、[詳細] をクリックします。
3. [レイアウト詳細] ダイアログ ボックスで、適切なデバイスの [編集] をクリックします。
4. [デバイス エディター] ダイアログ ボックスで、新しいコントロールを追加します: *Renderings/Modules/Web Forms for Marketers/Form*
5. 追加されたコントロールを編集します。[FormID] フィールドで、適切な Web フォームを選択します。

Web フォームを Web コントロールとして挿入する

Web フォームをレイアウトに静的に追加することができます。

1. [*.aspx] または [*.ascx] ファイルを開きます。
2. Web フォームの名前空間のタグ プレフィックスを登録します。

```
<%@ Register TagPrefix="wffm" Namespace="Sitecore.Form.Core.Renderings"  
Assembly="Sitecore.Forms.Core" %>
```

3. FormRendererer タグを追加します。

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.

```
<wffm:FormRender FormID="<id of the form item>" runat="server"/>
```

Web フォームをコード ビハインド クラスを使用して挿入する

コードビハインド クラスを使用して、様々な条件に基づく様々な Web フォームを追加することができます。コードビハインド クラスで次のサンプル コードを使用します。

```
FormRender fr = new FormRender();  
fr.FormID = "5D9E85F3-5E03-49A7-A136-93269DEA22A7";//form item id  
Sitecore.Context.Page.GetPlaceholder("main").Controls.Add(fr);
```

このサンプル コードは、ID で指定された Web フォームを Main プレースホルダーに挿入します。

3.4 データプロバイダーの設定方法

モジュールは次のデータベースをサポートします。

- SQLite
- MSSQL
- Oracle

デフォルトでは、モジュールは MSSQL データベースを使用します。

3.4.1 SQLite

SQLite を使用する場合、[site root]¥website¥App_Config¥include¥Sitecore.forms.config ファイルの次の SQLite データ プロバイダーの定義のコメントを解除するか追加します。

```
<formsDataProvider
type="Sitecore.Forms.Data.DataProviders.SQLite.SQLiteWFMDDataProvider,Sitecore.Forms.Core">
  <param desc="connection string">Data
Source=/data/sitecore_webforms.db;version=3;BinaryGUID=true</param>
</formsDataProvider>
```

3.4.2 MSSQL

MSSQL を使用している場合、フォーム レポートが正しく機能するように、モジュールとデータベース間の接続を設定する必要があります。

MSSQL データベースは、[site root]¥data フォルダの配下に配置されます。

- sitecore.webforms.mdf – MSSQL データベース
- sitecore.webforms.bak – 空の MSSQL データベースのバックアップ ファイル
- sitecore.webforms_log.ldf – MSSQL ログ ファイル

データベースは MSSQL サーバー上にデプロイする必要があります。

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.

[site root]¥website¥App_Config¥include¥Sitecore.forms.config ファイルの 次の MSSQL データ プロバイダーの定義のコメントを解除するか追加します。

```
<formsDataProvider type="Sitecore.Forms.Data.DataProviders.WFMDDataProvider,Sitecore.Forms.Core">
  <param desc="connection string">Database=(database);Data Source=(server);user
id=(user);password=(password);Connect Timeout=30</param>
</formsDataProvider>
```

(database) は MSSQL データベースの名前です。

(server) は MSSQL サーバーの名前です。

(user) はデータベースのユーザー名です。

(password) はデータベース ユーザーのパスワードです。

SQL Server 2008 上のモジュール用 データベースに必要なデータベース ユーザーの権限

データベース ユーザーはデータベース上で 実行 権限を所有している必要があり、*db_datareader* および *db_datawriter* のメンバーシップ ロールに割り当てられている必要があります。

上記の権限の設定方法についての追加情報は、SDN の『[Create Database Account](#)』を参照してください。

3.4.3 Oracle

Oracle server 10g 以降を使用している場合は、Sitecore CMS 6.4 インスタンス以降に Web Forms for Marketers version 2.2.0 rev.110303 以降をインストールします。

アップデート パッケージをインストールした後、次のことを行います。

1. [site root]¥data¥web forms oracle フォルダを Oracle サーバーが動作しているコンピューターにコピーします。
2. Oracle サーバーが動作しているコンピューター上で、コンピューターの **システムのプロパティ** 画面を表示し、[**詳細設定**] タブを選択し、[**環境変数**] をクリックします。ユーザー環境変数を次のように設定します。

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.

- *DATABASE* = Sitecore データをホストする Oracle データベース インスタンスの名前 (コンピューター上にセットアップされるため)。
 - *TBS* = Sitecore データのテーブルスペースの名前。
 - *PASSWORD* = Oracle インスタンスのシステム ユーザーのパスワード。
 - *PREFIX* = ユーザーのスキーマ名およびパスワードのプレフィックス。
3. Oracle サーバーが動作中のコンピューター上で、[site root]¥data¥web forms oracle フォルダーから CreateData.bat ファイルを実行します。

インストール スクリプトは %PREFIX%wffm ユーザー スキーマを設定します。ユーザー パスワードはユーザー スキーマと同じです。

4. Oracle サーバーが動作中のコンピューター上から、*DATABASE*、*TBS*、*PASSWORD*、*PREFIX* 環境変数を削除します。
5. [site root]¥website¥App_Config¥include¥Sitecore.forms.config ファイルを編集します。

configuration/sitecore セクションで、MS SQL および SQLite データ プロバイダーの定義をコメントアウトします (どちらか一方は既にコメントアウトされています)。

configuration/sitecore セクションで、次の Oracle データ プロバイダーの定義をコメント解除するか追加します。

```
<!-- Oracle -->
  <formsDataProvider
type="Sitecore.Forms.Data.DataProviders.Oracle.OracleWFMDDataProvider,Sitecore.Forms.Core">
  <param desc="connection string">user id=scwffm;password=scwffm;Data Source=sitecore</param>
</formsDataProvider>
```

適切な値を接続文字列の変数に割り当てます。

```
user id=<prefix>wffm
password=<prefix>wffm
Data Source=<database>
```

<prefix>、<password>、<database>は、手順 2 で環境変数に設定した値です。

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.

重要

デフォルトでは、formsDataProvider の type 属性は Sitecore.Forms.Oracle アセンブリを参照します。参照を Sitecore.Forms.Core アセンブリに変更したことを確認してください。

6. C:\Windows\Assembly フォルダー配下の グローバルアセンブリキャッシュに移動します。

Oracle クライアント のバージョンを確認するために Oracle.DataAccess.dll を選択し、右クリックして [プロパティ] を選択します。[バージョン] ([アセンブリのバージョン]) の値をコピーします。これは次のフォーマットのどちらかになります。

- * 10.2.x.x (10.2.0.2 以前の Oracle クライアント バージョン)
- * 2.102.x.x or 2.111.x.x (10.2.0.2 以降の Oracle クライアント バージョン)

7. web.config ファイルの configuration/runtime/assemblyBinding セクションに、次の定義を追加します。

```
<dependentAssembly>
  <assemblyIdentity name="Oracle.DataAccess" publicKeyToken="89b483f429c47342"
xmlns="urn:schemas-microsoft-com:asm.v1" />
  <bindingRedirect oldVersion="9.2.0.401" newVersion="[アセンブリのバージョン]" xmlns="urn:schemas-
microsoft-com:asm.v1" />
</dependentAssembly>
```

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.

3.5 標準機能の拡張/オーバーライド方法

Web Forms for Marketers を使用することによって、既存の機能を拡張することができます。

3.5.1 フォームのレンダリング

モジュールは、Form レンダリングを使用して Web サイト上に Web フォームを表示します。Form レンダリングは、`/sitecore/layout/Renderings/Modules/Web Forms for Marketers/` アイテムの配下にあります。ユーザーがコンテンツエディターから [挿入] をクリックして Web フォームを追加する場合、モジュールは自動的に Form レンダリングを現行のアイテムのプレゼンテーションに追加します。

Form アイテムはいくつかの設定フィールドを含みます。たとえば次のようなものがあります。

- タグ – ロジックを定義するレンダリング クラス。
- 名前空間 – クラスの名前空間。
- アセンブリ – クラスのアセンブリ。
- パラメーター – デフォルトでは、このフィールドは Web フォームを表す ASP.NET ユーザー コントロール ファイル (`*.ascx`) へのパスを指定します: `FormTemplate=/sitecore modules/web/Web Forms for Marketers/Control/SitecoreSimpleFormAscx.ascx`



Web フォームのレイアウトを変更し、クライアント スクリプトを追加し、Web フォームのレンダリングとライフサイクルに影響を与えることができます。Web サイト上のすべての Web フォームに影響するようなカスタマイズを実行するには、`SitecoreSimpleFormAscx.ascx` ファイルを編集します。

グローバルな Web フォーム ロジックを変更するには、次の手順に従います。

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.

- Sitecore.Form.Web.UI.Controls.SitecoreSimpleFormAscx クラスから継承される新しいクラスを作成します。
- 適切なメソッドをオーバーライドします。
- SitecoreSimpleFormAscx.ascx コントロールの Inherits 属性で、標準のクラスを作成したクラスで置換えます。

重要

標準機能をオーバーライドする他の方法がない場合にのみ、SitecoreSimpleFormAscx.ascx ファイルを編集することを推奨します。この方法は、モジュールをアップデートする場合に問題を引き起こす可能性があります。

例. Web サイト上のすべての Web フォームにロゴを追加する方法

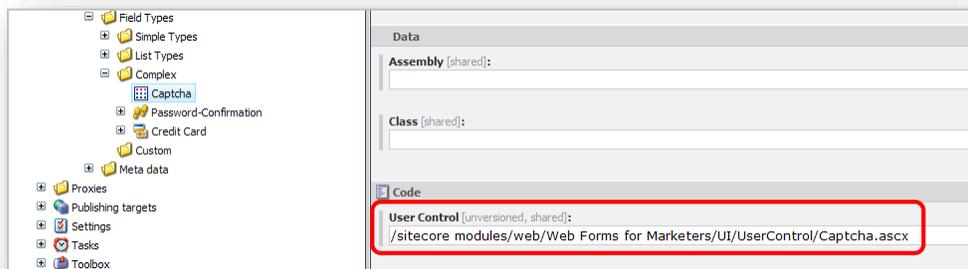
`/sitecore modules/web/Web Forms for Marketers/Control/SitecoreSimpleFormAscx.ascx` ファイルにおいて、`<wfm:FormTitle>` タグの前に、ロゴ イメージを示す `` タグを追加します。たとえば、`/sitecore` など。

```
....  
  
<wfm:FormTitle ID="title" runat="server"/>  
....
```

これで Web サイト上のすべての Web フォームにロゴが含まれます。

3.5.2 フィールド コントロール

Web Form for Marketers モジュールは、ascx フィールド コントロールをサポートします。.ascx コントロールへのパスを入力し、フォームで使用します。



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.

次の .ascx フィールド コントロールの外観を変更することができます。

- パスワード - 確認
- クレジット カード
- キャプチャ

これらは、sitecore¥modules¥Web¥Web Forms for Marketers¥UI フォルダーに配置されています。

3.5.3 フィールド アクション

Web Forms for Marketers では、指定された条件が満たされた場合に実行されるアクションを設定することによって、Web フォームをパーソナライズすることができます。条件と対応するアクションの一式をルールと呼びます。モジュールには定義済みのアクションの一式が含まれます。カスタムのフィールド アクションを実装することができます。

たとえば、フィールドへの入力を無効にするアクションを実装したいとします。このアクションは、Web フォームがユーザー プロファイルや注文を更新する用に設計されている場合に役立ちます。Web フォームは値をフィールドに表示しますが、ユーザー ID や オーダー ID などの変更できないため、編集できないようにします。

フィールドの入力を無効にするフィールド アクションを実装するには、次の手順に従います。

1. Visual Studio で新しいプロジェクトを作成します。
2. Sitecore.Kernel.dll および Sitecore.Forms.Core.dll アセンブリへの参照を追加します。
3. Sitecore.Rules.Actions.RuleAction<T> where T : Sitecore.Forms.Core.Rules.ConditionalRuleContext を継承する新しいクラスを作成します。
4. Apply(T ruleContext)メソッドをオーバーライドし実装します。このメソッドがアクションの実行を行います。
5. ruleContext.Control は、フォーム デザイナーで選択されたフィールドまたはセクションを含みます。

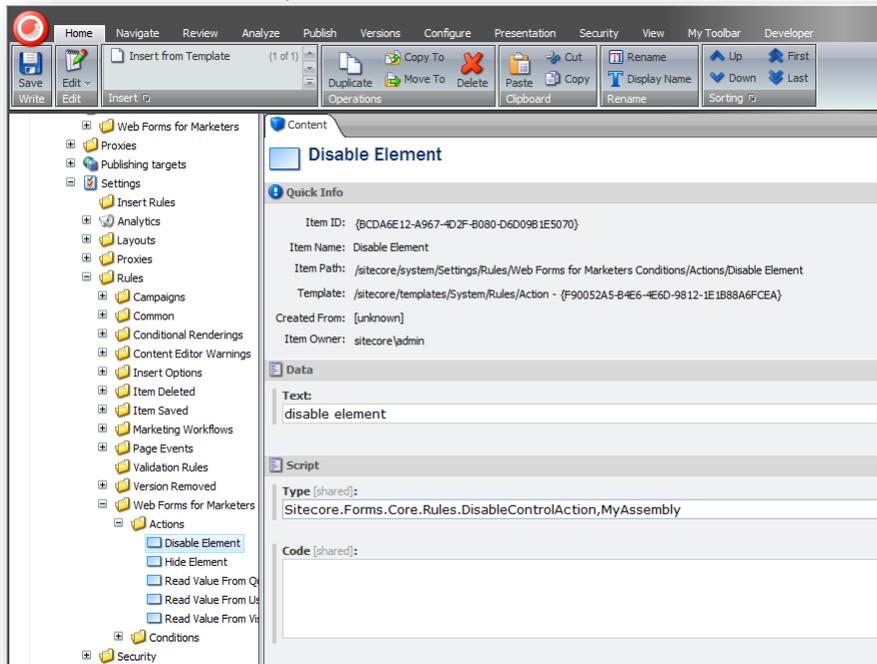
```
namespace Sitecore.Forms.Core.Rules
{
    using System.Web.UI.WebControls;
    using Sitecore.Diagnostics;
    using Sitecore.Rules.Actions;
```

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.

```
/// <summary>
/// The disable control action
/// </summary>
/// <typeparam name="T"></typeparam>
public class DisableControlAction<T> : RuleAction<T> where T : ConditionalRuleContext
{
    /// <summary>
    /// Applies the specified rule context.
    /// </summary>
    /// <param name="ruleContext">The rule context.</param>
    public override void Apply(T ruleContext)
    {
        Assert.ArgumentNotNull(ruleContext, "ruleContext");

        if (ruleContext.Control != null && ruleContext.Control is WebControl)
        {
            ((WebControl)ruleContext.Control).Enabled = false;
        }
    }
}
```

6. コンテンツ エディターで、アクション テンプレートを使って sitecore/system/Settings/Rules/Web Forms for Marketers Conditions/Actions フォルダーに新しいアクション アイテムを作成します。

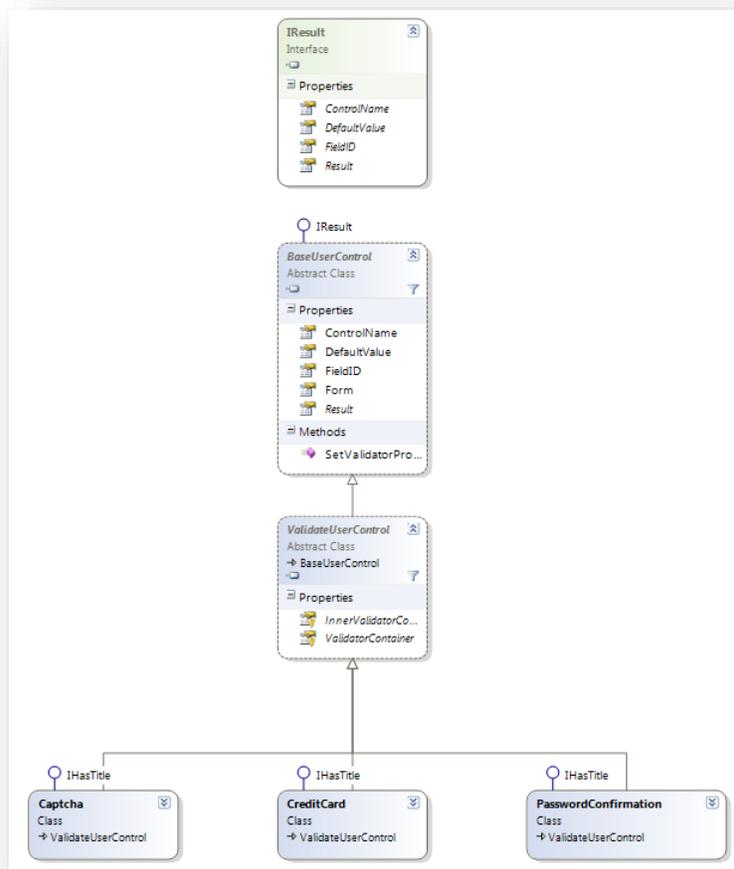


7. [テキスト] と [タイプ] フィールドを入力します。

3.6 モジュールで使用されるフィールドの階層構造

Web Forms for Marketers モジュールには 2 つのフィールドのグループがあります。

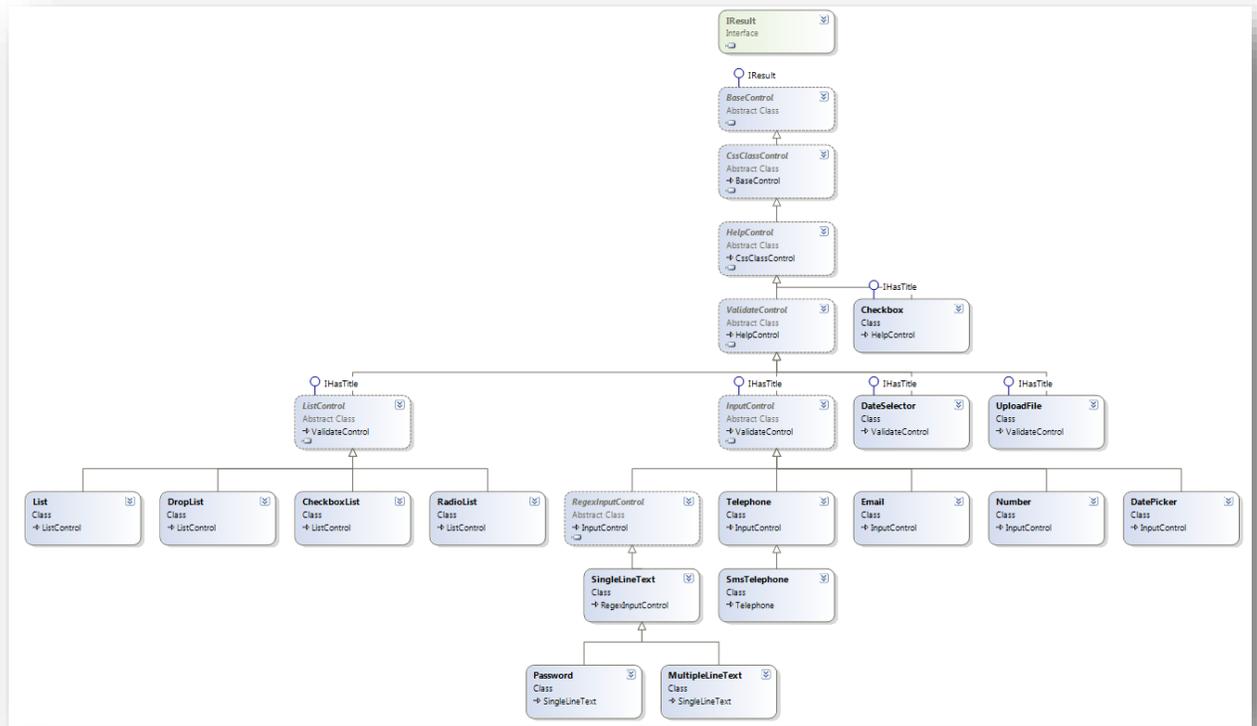
- System.Web.UI.UserControl クラスを継承するフィールド。これらのフィールドのマークアップは、分離 .ascx ファイルの中で使用されます。これらのフィールドの外観は、アセンブリを再コンパイルすることなく変更することができます。Sitecore.Form.UI.UserControls 名前空間には、このグループの次の 3 つのフィールドがあります。
 - キャプチャ (Captcha)
 - クレジットカード (CreditCard)
 - パスワード-確認 (PasswordConfirmation)



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.

- System.Web.UI.WebControls.WebControl クラスを継承するフィールド。通常これらのフィールドは、アセンブリの再コンパイル後に修正を必要としないシンプルなストラクチャを持ちます。このグループのフィールドは、System.Web.UI.UserControl クラスを継承するフィールドよりも簡単に再利用できます。これらのフィールドは、Sitecore.Form.Web.UI.Controls 名前空間に定義されます。
 - チェック ボックス(Checkbox)
 - チェック ボックスのリスト(CheckboxList)
 - 日付の選択(DatePicker)
 - 日付(DateSelector)
 - ドロップ リスト(DropList)
 - 電子メール(Email)
 - ラベル(Label)
 - リスト(List)
 - 複数行テキスト(MultiLineText)
 - 番号(Number)
 - パスワード>Password)
 - ラジオ リスト(RadioList)
 - 単一行テキスト(SingleLineText)
 - SMS の電話(SmsTelephone)
 - 電話(Telephone)
 - ファイルのアップロード(UploadFile)

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.



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.

3.7 新しいフィールド タイプの作成方法

Web Form for Marketers モジュールで新しいフィールド タイプを作成するには、次の手順に従います。

1. Visual Studio で、既存の Sitecore ソリューションの新しい Web アプリケーション プロジェクトを作成します。

新しいプロジェクトの作成についての追加情報は、下記リンクの 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」 セクションを参照してください。

<http://sdn.sitecore.net/Reference/References%20in%20Japanese/Presentation%20Component%20Cookbook.aspx>

2. Sitecore.Forms.Core アセンブリへの参照を追加します。
3. 当該プロジェクトに **クラス** ファイル テンプレートに基づく新しい項目を追加します。
4. モジュールにはフィールドの 2 つのグループがあります。新しいフィールドが関連するグループを分析します。フィールドの階級構造についての追加情報は、「モジュールで 사용되는フィールドの階層構造」セクションを参照してください。次のクラスのグループのどちらかを継承します。
 - Sitecore.Form.Web.UI.Controls.IResult および System.Web.UI.WebControls.WebControl クラス。または、既に両方のクラスの必要なパラメーターを含む Sitecore.Form.Web.UI.Controls.ValidateControl クラスを継承します。
 - Sitecore.Form.Web.UI.Controls.IResul および System.Web.UI.UserControl クラス。または、既に両方のクラスの必要なパラメーターを含む Sitecore.Form.Web.UI.Controls.ValidateUserControl クラス。このグループのフィールドは、.ascx ファイルに基づきます。

Sitecore.Form.Web.UI.Controls.IResult インターフェイスを使用することによって、モジュールはフィールドの値を参照することができます。

```
public interface IResult
{
    ControlResult Result { get; }

    string ControlName { get; set; }

    string FieldID { get; set; }
```

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.

```
string DefaultValue { set; }  
}
```

Result — `ControlResult` クラスのオブジェクト型のフィールド値を返します。

ControlName — フィールド タイプのシステムアイテム名を取得または設定します。

FieldID — フィールド タイプのアイテムの ID。

DefaultValue – デフォルトのフィールドの値。

返された値が文字列でない場合、*Result* バリューのアダプターを設定する必要があります。アダプターを設定するには、クラス定義の前に `Adapter` 属性を使用します。

```
namespace Sitecore.Form.Web.UI.Controls  
{  
    [Adapter(typeof (DataAdapter))]  
    [ValidationProperty("Value")]  
    public class CustomField : ValidateControl
```

この属性のタイプは、`Sitecore.Form.Core.Client.Submit.Adapter` 抽象クラスを継承する必要があります。

```
public class DateAdapter : Sitecore.Form.Core.Client.Submit.Adapter  
{  
    public DateAdapter () { }  
    // Convert object value to string  
    public override string AdaptResult(object value)  
    {  
        return value.ToString();  
    }  
    // Convert value from database to friendly value. Using in the Form Data Viewer  
    public override string AdaptToFriendlyValue(string value)  
    {  
        return value;  
    }  
    // Convert list value from database to friendly value. Using in the Form Data  
    Viewer  
    public override IEnumerable<string> AdaptToFriendlyListValues(string value)  
    {  
        return new List<string>(new[]{value});  
    }  
}
```

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.

```
}  
}
```

5. フォーム デザイナーでユーザーにフィールド タイトルを入力させたい場合は、`Sitecore.Form.Web.UI.Controls.IHasTitle` インターフェースを実装します。

```
public interface IHasTitle  
{  
    string Title { set; get; }  
}
```

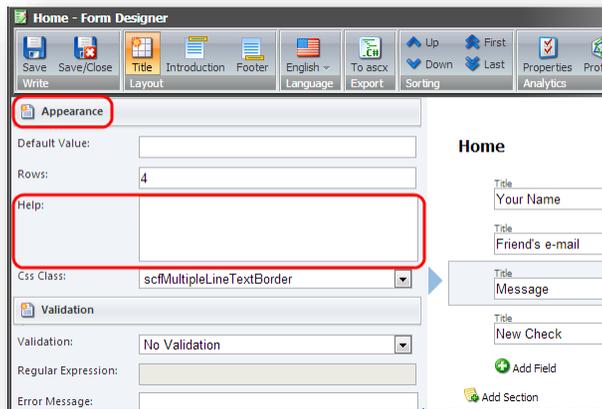
これはタイトル プロパティのみを含みます。モジュールはこのプロパティを使ってフィールド タイトルを設定します。

6. フォーム デザイナーでユーザーがコントロールのプロパティを利用できるようにする場合は、次の属性を使用します。

- *VisualProperty* — フォーム デザイナーでプロパティを表示します。
- *VisualCategory* — カテゴリの名前を設定します。
- *VisualFieldType* — プロパティの入力コントロールのタイプを定義します。属性が指定されていない場合、プロパティは `EditField` 入力タイプになります。利用可能な Visual Field Type のリストは、`Sitecore.Form.Core.Visual` 名前空間の配下で定義されています。
- *DefaultValue* — フォーム デザイナーでデフォルト値を設定します。
- *Localize* — プロパティが複数言語をサポートするか否かを定義します。

たとえば、フォーム デザイナーで [アピアランス] セクションの中に 500 の並べ替え順序を持つ ヘルプ プロパティを利用できるようにするには、次のコードを使用することができます。

```
[VisualProperty("Help:", 500)]  
[VisualCategory("Appearance")]  
[VisualFieldType(typeof(TextAreaField)), Localize]  
public string Information  
{  
    ...  
}
```

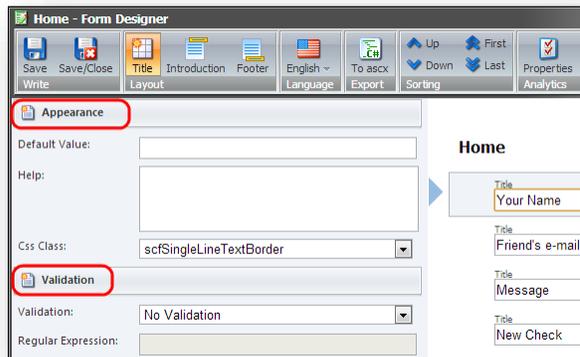


7. プロジェクトをビルドし、コンパイルされた DLL ファイルを Sitecore ソリューションの `bin` フォルダに配置します。
8. **コンテンツ ツリー**の `/sitecore/system/Modules/Web Forms for Marketers/Settings/Field Types/Custom` フォルダに、`/sitecore/templates/Web Forms for Marketers/Field` タイプの テンプレートに基づくアイテムを作成します。
 - a. フィールド タイプが `System.Web.UI.UserControl` クラスを継承する場合、フィールド タイプ アイテムの **[User Control]** フィールドに、対応する `ascx` ファイルへのパスを入力します。
 - b. フィールド タイプが `System.Web.UI.WebControls.WebControl` クラスを継承する場合、フィールド タイプ アイテムの **[Assmby]** フィールドに、DLL ファイルの名前を入力し、**[Class]** フィールドに対応するクラスの名前を入力します。

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.

3.7.1 ビジュアル フィールド プロパティの作成

このセクションでは、フォーム デザイナーに表示されるフィールド タイプのプロパティ セクションの作成方法について説明します。



この例では、`Sitecore.Form.Web.UI.Controls.SingleLineText` クラスから継承され、`Disabled` と `Visible` の 2 つのビジュアルプロパティを持つ、新しいフィールド タイプを作成します。

1. Visual Studio で、新しいプロジェクトを作成し、名前を付けます。たとえば `CustomSingleLineText` など。
2. `Sitecore.Forms.Core` および `Sitecore.Forms.Custom` アセンブリへの新しい参照を追加します。
3. `Sitecore.Form.Web.UI.Controls.SingleLineText` クラスを継承する新しいクラスを作成し、`Disabled` と `Visible` という名前のビジュアルプロパティを追加します。

```
namespace Sitecore.Custom.Form.Web.UI.Controls
{
    public class CustomSingleLineText : SingleLineText
    {
        // Set field enabled or disabled
        [DefaultValue("No"), VisualFieldType(typeof(BooleanField)), VisualProperty("Disabled:", 100),
        VisualCategory("Properties")]
        public string IsDisabled
        {
            get
            {
                return base.textbox.Enabled.ToString();
            }
        }
    }
}
```

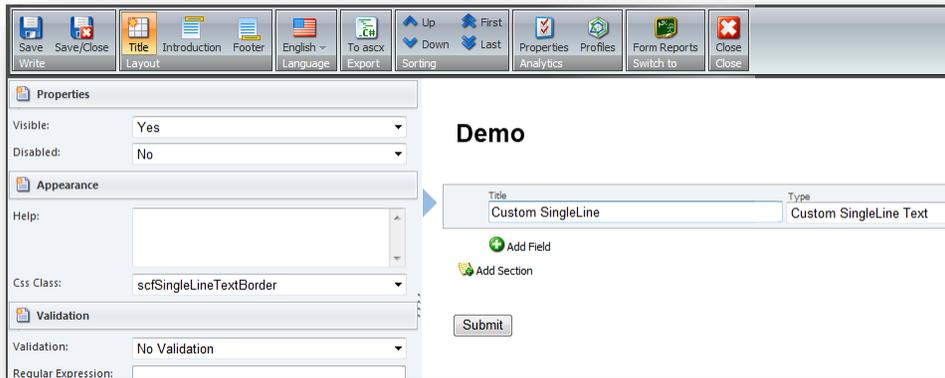
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

```
set
{
    base.textbox.Enabled = value == "No";
}
}

//// Set field visible or not
[DefaultValue("Yes"), VisualFieldType(typeof(BooleanField)), VisualProperty("Visible:", 100),
VisualCategory("Properties")]
public string IsVisible
{
    get
    {
        return base.textbox.ToString();
    }
    set
    {
        base.textbox.Visible = value == "Yes";
        base.title.Visible = value == "Yes";
    }
}
}
```

4. プロジェクトをビルドし、コンパイルされた CustomSingleLineText.dll ファイルを¥website¥bin¥ フォルダに配置します。
5. **コンテンツ エディター**で、/sitecore/system/Modules/Web Forms for Marketers/Settings/Field Types/Custom フォルダに/sitecore/templates/Web Forms for Marketers/Field Type テンプレートに基づく新しい **Custom SingleLineText** アイテムを作成します。
6. **Custom SingleLineText** アイテムの **[Assembly]** フィールドに [CustomSingleLineText] と入力し、**[Class]** フィールドに [Sitecore.Custom.Form.Web.UI.Controls.CustomSingleLineText] と入力します。

次のように、新しいフィールド タイプとそのプロパティはフォーム デザイナーでレンダリングされます。



3.7.2 WebControl クラスに基づく新しいフィールド タイプの作成

このセクションでは、日付の選択 フィールドを例として、*WebControl* クラスに基づく新しいフィールド タイプの作成方法について説明します。

日付の選択 フィールド タイプを作成するには、次の手順に従います。

1. Visual Studio で、新しいプロジェクトを作成し、名前をつけます。たとえば、*Demo.Examples* など。
2. Sitecore.Forms.Core アセンブリへの新しい参照を追加します。
3. Sitecore.Form.Web.UI.Controls.ValidateControl を継承する新しいクラスを作成します。

```
namespace Demo.Examples
{
    using System;
    using System.Web.UI;
    using System.Web.UI.WebControls;
    using Sitecore.Form.Core.Attributes;
    using Sitecore.Form.Core.Controls.Data;
    using Sitecore.Form.Core.Visual;
    using Sitecore.Form.Web.UI.Controls;

    /// <summary>
    /// The date picker control
```

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.

```
/// </summary>
public class DatePicker : ValidateControl, IHasTitle
{
    /// Here we define the controls that make the structure of the field
    /// <summary>
    /// The control container
    /// </summary>
    protected Panel container = new Panel();

    /// <summary>
    /// The help notes of the control
    /// </summary>
    protected Label helpNotes = new Label();

    /// <summary>
    /// The input for the control
    /// </summary>
    protected TextBox input = new TextBox();

    /// <summary>
    /// The title of the control
    /// </summary>
    protected Label title = new Label();

    ///the constructor that defines the default values of the field properties
    /// <summary>
    /// Initializes a new instance of the <see cref="DatePicker"/> class.
    /// </summary>
    public DatePicker()
        : base(HtmlTextWriterTag.Div)
    {
        this.CssClass = "scfDatePickerBorder";
    }

    /// Implementation of the ihasTitle interface to get the title that is entered in the Form Designer
    /// <summary>
    /// Gets or sets the title of the control.
    /// </summary>
    /// <value>The title.</value>
    public string Title
    {
        get
        {
```

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.

```
        return this.title.Text;
    }

    set
    {
        this.title.Text = value;
    }
}
/// Implementation of the property that is set in the Form Designer
/// <summary>
/// Gets or sets the help notes for the control.
/// </summary>
/// <value>The text for help notes.</value>
[VisualProperty("Help:", 500), VisualCategory("Appearance"), VisualFieldType(typeof(TextAreaField)),
Localize]
public string HelpNotes
{
    get
    {
        return this.helpNotes.Text;
    }

    set
    {
        this.helpNotes.Text = value ?? string.Empty;
    }
}
///The result of the control that will be used in the save actions and verification actions later
/// <summary>
/// Gets the result of clients' inputs.
/// </summary>
/// <value></value>
public override ControlResult Result
{
    get
    {
        return new ControlResult(this.ControlName, this.input.Text, null);
    }
}

///Creating the structure of the field
/// <summary>
/// Builds the control structure
/// </summary>
```

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.

```
/// <param name="e">An <see cref="T:System.EventArgs"/> object that contains the event
data.</param>
protected override void OnInit(EventArgs e)
{
    this.input.CssClass = "scfDatePickerTextBox";
    this.helpNotes.CssClass = "scfDatePickerUsefulInfo";
    this.container.CssClass = "scfDatePickerGeneralPanel";
    this.title.CssClass = "scfDatePickerLabel";

    this.input.TextMode = TextBoxMode.SingleLine;

    this.Controls.AddAt(0, this.container);
    this.Controls.AddAt(0, this.title);

    this.container.Controls.AddAt(0, this.helpNotes);
    this.container.Controls.AddAt(0, this.input);
}

/// <summary>
/// Attaches jscripsts to the page
/// </summary>
/// <param name="e">The e.</param>
protected override void OnPreRender(EventArgs e)
{
    string script = "$(document).ready(function() { $('#" + this.input.ClientID + ").datepicker()";

    this.Page.ClientScript.RegisterClientScriptInclude("jquery", "/sitecore modules/web/web forms for
marketers/scripts/jquery.js");
    this.Page.ClientScript.RegisterClientScriptInclude("jquery.ui", "/sitecore modules/web/web forms for
marketers/scripts/jquery-ui.min.js");
    this.Page.ClientScript.RegisterStartupScript(this.GetType(), script, script, true);

    base.OnPreRender(e);
}
}
}
```

4. プロジェクトをビルドし、コンパイルされた Custom DatePicker.dll ファイルを ¥website¥bin¥ フォルダに配置します。
5. **コンテンツ エディター**で、/sitecore/templates/Web Forms for Marketers/Field Type テンプレートを使って /sitecore/system/Modules/Web Forms for Marketers/Settings/Field Types/Custom フォルダに **Custom DatePicker** アイテムを追加します。

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.

6. **Custom DatePicker** アイテムの **[Assembly]** フィールドに `[Demo.Example]` と入力し、**[Class]** フィールドに `[DatePicker]` と入力します。

3.7.3 UserControl クラス (.ascx ファイル) に基づく新しいフィールド タイプの作成

UserControl クラス (.ascx ファイル) に基づく新しいフィールド タイプを作成するには、次の手順に従います。

1. Visual Studio で、新しいプロジェクトを作成し、名前を付けます。たとえば、`UserControlField` など。
2. Sitecore.Forms.Core アセンブリへの新しい参照を追加します。
3. Sitecore.Form.Web.UI.Controls.ValidateUserControl を継承する新しいクラスを作成します。新しいフィールドにタイトルをつけるには、クラスを IHasTitle インターフェースから継承します。

```
public class CreditCard : ValidateUserControl, IHasTitle
{
    ...
}
```

4. `UserControlField.ascx` ユーザー コントロール ファイルを `¥Website¥sitecore modules¥Web¥Web Forms for Marketers¥UI¥UserControl` フォルダに作成します。
5. プロジェクトをビルドし、コンパイルされた `Custom UserControlField.dll` ファイルを `¥website¥bin¥` フォルダに配置します。
6. **コンテンツ エディター**で、`/sitecore/system/Modules/Web Forms for Marketers/Settings/Field Types/Custom` フォルダに、`/sitecore/templates/Web Forms for Marketers/Field Type` テンプレートに基づく新しい **Custom UserControlField** アイテムを作成します。
7. **Custom UserControlField** アイテムの **[User Control]** フィールドに、対応する ascx ファイルへのパスを入力します: `/sitecore modules/web/Web Forms for Marketers/UI/UserControl/UserControlField.ascx`

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.

3.8 保存アクションの作成

カスタムの保存アクションを作成することができます。新しいクラスを作成する場合は、ベース インターフェイスとクラスを使用します。

3.8.1 ベース インターフェイスとクラス

カスタムの保存アクションを作成する場合は、`Sitecore.Form.Submit.ISaveAction` インターフェイスを継承するクラスを作成する必要があります。このインターフェイスは、新しいクラスに実装する必要がある `Execute` メソッドを含みます。これは、Web フォームに割り当てられたすべての保存アクションによって呼び出されるためです。このメソッドを使って保存アクションのロジックを実装します。

`Execute` メソッドは次の 3 つの変数を受け入れます。

- `ID formid` – アクションが割り当てられた Web フォーム アイテムの ID。
- `AdaptedResultList` フィールド – `AdaptedResult` クラスのリスト。リストの各アイテムは、送信された Web フォーム フィールドについての情報を提供します (値、フィールド ID、フィールド名、パラメーター)。
- `params object[]` データ – 配列の先頭の要素にアナリティクスのセッション ID を含みます。

Web Forms for Marketers モジュールは、カスタムのクラス から継承できる複数のクラスを提供します。

- `Sitecore.Form.Submit.UserBaseAction` クラス
- `Sitecore.Form.Core.Submit.AuditSaveAction` クラス

Sitecore.Form.Submit.UserBaseAction クラス

`Sitecore.Form.Submit.UserBaseAction` クラスを使用することによって、ユーザーのアカウントを操作することができます。

`Sitecore.Form.Submit.UserBaseAction` クラスは次のメソッドを含みます。

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.

メソッド	説明
public static string Escape(string userName)	ユーザー名を書式設定します (“.”を “_dot_” と、“@”を “_at_” と置換えます)。
public static string GetFullUserName(string domainName, string userName)	ユーザー名にドメイン名が含まれるようにします。
protected string GetProfileProperty(UserProfile profile, string profileproperty)	指定されたプロファイル プロパティの値を取得します。
protected string GetUserNameIfExist(string preUserName)	指定された名前を持つユーザーが存在する場合はユーザー名を返します。そうでなければ null を返します。
public static string GetValidUserName(string domainName, string userName)	指定されたユーザー名に基づく有効なユーザー名を返します。
protected virtual string ProcessBaseOperations(ID formId, AdaptedResultList fields, bool createIfNotExist)	ユーザーが存在するかまたはメソッドによって作成された場合 (bool createIfNotExist 引数値に基づく)、ユーザー名を返します。
protected virtual void UpdateEmail(string userName, string mail)	指定された電子メールが有効な場合はユーザーのメールを設定します。
protected virtual void UpdateGlobalSession(string userName)	指定されたユーザーでアナリティクス グローバル セッションを更新します。
protected virtual void UpdatePassword(ID formID, string userName, string password)	ユーザーが存在しない場合は指定された名前およびパスワードでユーザーを作成します。PasswordField プロパティの値が blankPassword である場合、そのユーザーは空のパスワードで作成されます。値が randomPassword である場合は、ランダムなパスワードが生成されます。また、このメソッドは作成されたユーザーのプロファイル アイテムの ID も設定します。
protected void UpdateProfileProperty(UserProfile profile, string profileproperty, string propertyValue)	プロファイル プロパティの値を設定します。

Sitecore.Form.Submit.UserBaseAction クラスは、次のプロパティを含みます。

プロパティ	説明
public bool AssociateUserWithVisitor { get; set; }	アナリティクス グローバル セッションをユーザー情報を使用して更新する必要があるかどうかを示します。このプロパティは、UpdateGlobalSession メソッドで確認されます。
public string AuditField { get; set; }	派生型でのみ使用されます。
public string DomainField { get; set; }	ドメイン名を指定します。
public string PasswordField { get; set; }	パスワード フィールドの名前が特別な値を含むことができるかを指定します。
public string ProfileItemId { get; set; }	ユーザー プロファイル アイテムの ID を設定または取得します。
public string UserNameField { get; set; }	ユーザー名を含む Web フォーム フィールドの名前を設定または取得します。
public string UserNameIsEmpty { get; set; }	ユーザー名が空の場合に表示されるエラー メッセージを設定または取得します。

Sitecore.Form.Core.Submit.AuditSaveAction クラス

Sitecore.Form.Core.Submit.AuditSaveAction クラスを使用することによって、保存アクションが実行された場合に監査メッセージを収集することができます。これは UserBaseAction クラスのベース クラスです。

Sitecore.Form.Core.Submit.AuditSaveAction クラスは、次のメソッドを含みます。

メソッド	説明
public void AuditMessage(string message)	メッセージ コレクション (クラス メンバー) にメッセージを追加します。
public void AuditSkippedField(string takenFrom, string insertTo, string value)	スキップされたコレクション (クラス メンバー) にエントリーを追加します。このメソッドは、上書きされなかったプロフィール フィールドを追跡するために使用することができます。

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.

3.8.2 パラメーター

保存アクションは任意の数のパラメーターを持つことができます。保存アクションのクラスでは、パラメーターは文字列型のクラス プロパティとして表されます。

```
public string MyProperty{ get;set;}
```

次のフォーマットのように、パラメーターの値を保存アクション アイテムの [パラメーター] フィールドまたは [Localized parameters] フィールドで指定することができます。

```
<parameter name>value</parameter name>
```

特定の Web フォーム パラメーターは Web フォームのアイテムに保存され、それらをフォーム デザイナーのアクション エディターで編集することができます。アクション エディターについての追加情報は、「アクション エディターの作成方法」セクションを参照してください。

保存アクション アイテムのパラメーターは、Web フォーム アイテムで指定されたパラメーターをオーバーライドします。

3.8.3 保存アクションの作成方法

ログイン アクションなどの保存アクションを作成するには、次の手順に従います。

1. 新しいプロジェクトを作成します。たとえば、Sitecore.Forms.Sample など。
2. 新しい参照を Sitecore.Forms.Core アセンブリに追加します。
3. Sitecore.Form.Submit.ISaveAction インターフェイスを継承する新しいクラスを作成します。

アクション コード:

```
using Sitecore.Data;  
using Sitecore.Diagnostics;  
using Sitecore.Form.Core.Client.Data.Submit;  
using Sitecore.Form.Core.Controls.Data;  
using Sitecore.Form.Submit;  
using Sitecore.Security.Authentication;  
  
namespace Sitecore.Forms.Sample
```

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.

```
{
  /// <summary>
  /// Login action
  /// </summary>
  public class LoginAction : ISaveAction
  {
    #region Methods

    /// <summary>
    /// Initializes a new instance of the <see cref="LoginAction"/> class.
    /// </summary>
    public LoginAction()
    {
      this.DefaultDomain = "sitecore";
    }

    /// <summary>
    /// Executes the login action.
    /// </summary>
    /// <param name="formid">The form id.</param>
    /// <param name="fields">The fields of the form.</param>
    /// <param name="data">The custom data.</param>
    public void Execute(ID formid, AdaptedResultList fields, params object[] data)
    {
      AdaptedControlResult login = fields.GetEntry(this.Login, "Login");
      AdaptedControlResult password = fields.GetEntry(this.Password, "Password");

      Assert.ArgumentNotNull(login, "You should point the login field.");
      Assert.ArgumentNotNull(password, "You should point the password field.");

      string userName = login.Value;
      Assert.ArgumentNotNullOrEmpty(userName, "Login can't be empty");

      if (!userName.Contains("¥¥"))
      {
        userName = string.Join("¥¥", new[] { this.DefaultDomain, userName });
      }

      AuthenticationManager.Login(userName, password.Value, true);
    }

    #endregion

    #region Properties
```

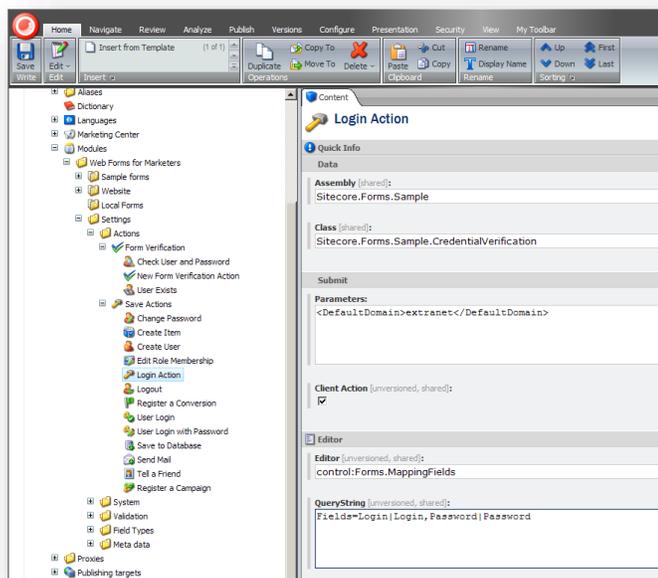
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

```
/// <summary>
/// Gets or sets the default domain.
/// </summary>
/// <value>The default domain.</value>
public string DefaultDomain { get; set; }

/// <summary>
/// Gets or sets the login field.
/// </summary>
public string Login { get; set; }

/// <summary>
/// Gets or sets the password field.
/// </summary>
public string Password { get; set; }
#endregion
}
}
```

4. **コンテンツ エディター**で、/sitecore/System/Modules/Web Forms for Marketers/Settings/Actions/Save Actions の配下に新しいアイテムを作成し、そのパラメーターを指定します。



保存アクションのパラメーターについての追加情報は、「保存アクション」セクション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.

3.9 アクション エディターの作成方法

アクション エディターは、ユーザーに保存アクションのパラメーターを設定させるダイアログ ボックスです。カスタム保存アクション用にアクション エディターを作成することができます。

アクション エディターを作成する前に、xml コントロールに関する追加情報について次の記事を参照してください。

- <http://sdn.sitecore.net/Articles/XML%20Sheer%20UI/Beginning%20with%20XML%20controls.aspx>
- <http://sdn.sitecore.net/SDN5/Articles/XML%20Sheer%20UI/My%20first%20XML%20application.aspx>

アクション エディターを使用するロジックは次のようになります。

- OnLoad メソッドで保存アクションのパラメーターを読み込みます。
- ユーザーに保存アクションのパラメーターを表示します。
- ユーザーがパラメーターを編集した後、OnOk メソッドを使ってそれらを Web フォームの **保存アクション** フィールドに保存します。

アクション エディターを実装する前に、次のコードチップスを分析します。

- 現行の保存アクションのパラメーターを取得するには、次を実行します。

```
string params=HttpContext.Current.Session[Sitecore.Web.WebUtil.GetQueryString("params")] as string;
NameValueCollection nvParams=ParametersUtil.XmlToNameValueCollection(params);
```

- パラメーターの値を保存するには、OnOk メソッドをオーバーライドします。

```
protected override void OnOK(object sender, EventArgs args)
{
    string str3 = ParametersUtil.NameValueCollectionToXml(this.nvParams ?? new NameValueCollection());
    if (str3.Length == 0)
    {
        str3 = "-";
    }
    SheerResponse.SetDialogValue(str3);
    base.OnOK(sender, args);
}
```

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.

```
}
```

- 現在の Web フォーム アイテム ID を取得するには、次を実行します。

```
Sitecore.Web.WebUtil.GetQueryString("id");
```

- 現在の言語を取得するには、次を実行します。

```
Sitecore.Web.WebUtil.GetQueryString("la", "en");
```

- 現在の Sitecore データベースを取得するには、次を実行します。

```
Sitecore.Web.WebUtil.GetQueryString("db");
```

アクション エディターを作成するには、次のアクションを実行します。

1. ダイアログのレイアウトの *.xml ファイルを作成します。

```
<?xml version="1.0" encoding="utf-8" ?>
<control xmlns:def="Definition" xmlns="http://schemas.sitecore.net/Visual-Studio-Intellisense">
  <SimpleEditor>
    <Stylesheet>
      .scfContent {
        padding-top : 15px;
      }

      .scfFieldScope{
        width:100%; margin:7px;
      }

      .scfFieldLabel {
        width:40%;
      }

      .scfFieldSelect {
        width:58%;
        position:absolute;
        right:0;
        margin-right:20px;
      }
    </Stylesheet>
```

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.

```
<FormDialog ID="Dialog" Icon="Software/32x32/step_new.png">
  <CodeBeside Type="Sitecore.Forms.Sample.SimpleEditor,Sitecore.Forms.Sample"/>
  <Border Class="scfContent" Width="100%" Align="left" Style="overflow:none;">
    <Literal ID="name" Text="Login:" Class="scfFieldLabel"/>
    <Combobox runat="server" ID="login"/>
  </Border>
  <Border Class="scfContent" Width="100%" Align="left" Style="overflow:none;">
    <Literal ID="pass" Text="Password:" Class="scfFieldLabel"/>
    <Combobox runat="server" ID="password"/>
  </Border>
  <Border Class="scfContent" Width="100%" Align="left" Style="overflow:none;">
    <Literal ID="domLiteral" Text="Default domain:" Class="scfFieldLabel"/>
    <Combobox runat="server" ID="domain"/>
  </Border>
</FormDialog>
</SimpleEditor>
</control>
```

2. Sitecore.Web.UI.Pages.DialogForm クラスをベースクラスとして アクション エディター クラスを作成します。作成されたクラスを*.xml ファイルで CodeBeside として指定します。以下はアクション エディター クラスのサンプルコードです。

```
public class SimpleEditor : DialogForm
{
    protected Combobox domain;
    protected Combobox login;
    protected Combobox password;
    private NameValueCollection nvParams;
    protected override void OnLoad(EventArgs e)
    {
        base.OnLoad(e);
        if (!Context.ClientPage.IsEvent)
        {
            foreach (Domain d in DomainManager.GetDomains())
            {
                Sitecore.Web.UI.HtmlControls.ListItem item = new Web.UI.HtmlControls.ListItem();
                item.Value = d.Name;
                item.Header = d.Name;
                domain.Controls.Add(item);
            }
            foreach (FieldItem f in this.CurrentForm.Fields)
            {
                Sitecore.Web.UI.HtmlControls.ListItem item = new Web.UI.HtmlControls.ListItem();
```

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.

```
        item.Value = f.ID.ToString();
        item.Header = f.DisplayName;
        login.Controls.Add(item);
        item = new Web.UI.HtmlControls.ListItem();
        item.Value = f.ID.ToString();
        item.Header = f.DisplayName;
        password.Controls.Add(item);
    }
    domain.Value = this.GetValueByKey("DefaultDomain") ?? string.Empty;
    login.Value = this.GetValueByKey("Login") ?? string.Empty;
    password.Value=this.GetValueByKey("Password") ??string.Empty;
}
}
protected void SaveValues()
{
    this.SetValue("DefaultDomain", domain.Value);
    this.SetValue("Login", login.Value);
    this.SetValue("Password", password.Value);
}
protected override void OnOK(object sender, EventArgs args)
{
    this.SaveValues();
    string str3 = ParametersUtil.NameValueCollectionToXml(this.nvParams ?? new NameValueCollection());
    if (str3.Length == 0)
    {
        str3 = "-";
    }
    SheerResponse.SetDialogValue(str3);
    base.OnOK(sender, args);
}

public void SetValue(string key, string value)
{
    if (this.nvParams == null)
    {
        this.nvParams = ParametersUtil.XmlToNameValueCollection(this.Params);
    }
    this.nvParams[key] = value;
}

public string GetValueByKey(string key)
{
    if (this.nvParams == null)
    {
```

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.

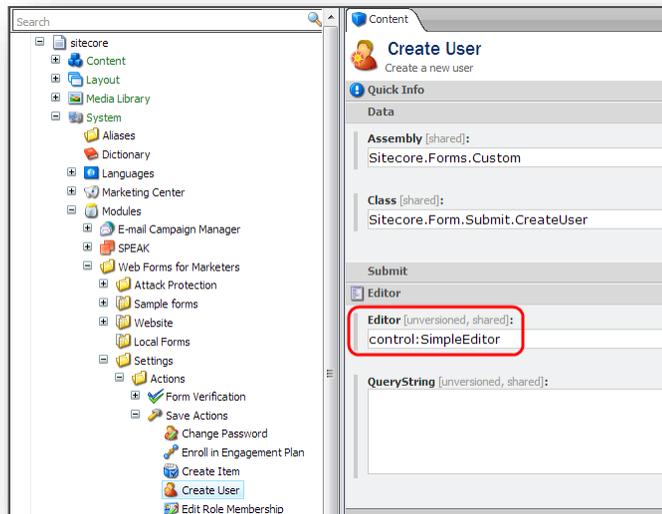
```
        this.nvParams = ParametersUtil.XmlToNameValueCollection(this.Params);
    }
    return (this.nvParams[key] ?? string.Empty);
}
public string Params
{
    get
    {
        return (HttpContext.Current.Session[Sitecore.Web.WebUtil.GetQueryString("params")] as string);
    }
}

public string CurrentID
{
    get
    {
        return Sitecore.Web.WebUtil.GetQueryString("id");
    }
}
public FormItem CurrentForm
{
    get
    {
        if (!string.IsNullOrEmpty(this.CurrentID))
        {
            Item innerItem = this.CurrentDatabase.GetItem(this.CurrentID, this.CurrentLanguage);
            if (innerItem != null)
            {
                return new FormItem(innerItem);
            }
        }
        return null;
    }
}
public virtual Database CurrentDatabase
{
    get
    {
        return Factory.GetDatabase(Sitecore.Web.WebUtil.GetQueryString("db"));
    }
}
public virtual Language CurrentLanguage
{
    get
```

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.

```
{  
    return Language.Parse(Sitecore.Web.WebUtil.GetQueryString("la", "en"));  
}  
}  
}
```

3. 適切な保存アクション アイテムの **[Editor]** フィールドで、作成されたエディターを指定します。

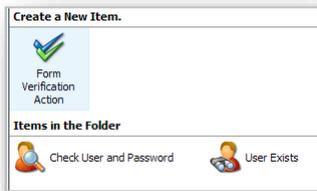


Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

3.10 フォームの検証アクションの作成方法

フォームの検証アクションを作成するには、次の手順に従います。

1. **コンテンツ エディター**で、sitecore/content/system/modules/web forms for marketers/settings/actions/form verification アイテムを選択します。
2. 新しいフォームの検証アクション アイテムを作成し、適切なパラメーターを追加します。



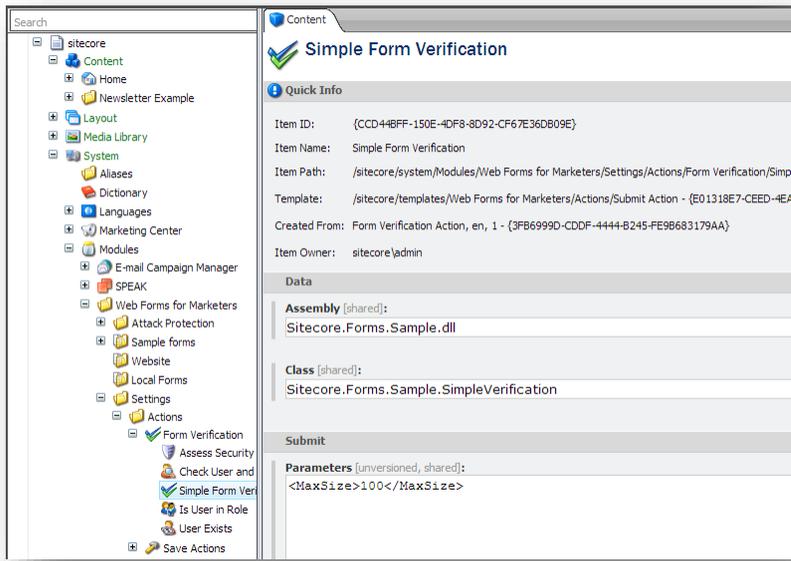
3. 新しいフォームの検証アクションの作成方法は、新しい保存アクションの作成方法と似ています。保存アクションの作成方法についての追加情報は、「保存アクションの作成」セクションを参照してください。
4. 唯一の違いは、検証アクションには Sitecore.Form.Core.Submit.BaseCheckAction クラスをベース クラスとして使用する必要があることです。検証が失敗したことを示すには、モジュールはアクションのコードによって生成された例外を生成する必要があります。モジュールはこの例外を処理し、ユーザーにエラー メッセージを表示します。次の例で、**ファイルのアップロード** フィールドを使って、アップロードされたファイルのサイズが制限されたサイズ以下であるかどうかを確認する検証アクションについて示します。

```
namespace Sitecore.Forms.Sample
{
    class SimpleVerification:BaseCheckAction
    {
        public override void Execute(Sitecore.Data.ID formid,
IEnumerable<Form.Core.Controls.Data.ControlResult> fields)
        {
            foreach (ControlResult cr in fields)
            {
                PostedFile file= cr.Value as PostedFile;
                if(file!=null)
                {
                    if (file.Data.Length > int.Parse(MaxSize))
                    {
```

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.

```
        throw new Exception("File size should be less than " + MaxSize+" bytes");
    }
}
}
}
}
public string MaxSize { get; set; }
}
}
```

この例では、アクションの定義アイテムは次のように表示されます。



The screenshot displays the Sitecore Content Editor interface. On the left is a navigation tree showing the hierarchy: Content > Modules > Web Forms for Marketers > Settings > Actions > Form Verification. The main pane shows the details for the 'Simple Form Verification' action. The 'Quick Info' section includes: Item ID: {CCD44BFF-150E-4DF8-8D92-CF67E36DB09E}, Item Name: Simple Form Verification, Item Path: /sitecore/system/Modules/Web Forms for Marketers/Settings/Actions/Form Verification/Simple Form Verification, Template: /sitecore/templates/Web Forms for Marketers/Actions/Submit Action - (E01318E7-CEED-4EAB-8000-000000000000), Created From: Form Verification Action, en, 1 - {3FB6999D-CDDF-4444-B245-FE9B683179AA}, and Item Owner: sitecore\admin. The 'Data' section shows: Assembly [shared]: Sitecore.Forms.Sample.dll and Class [shared]: Sitecore.Forms.Sample.SimpleVerification. The 'Submit' section shows: Parameters [unversioned, shared]: <MaxSize>100</MaxSize>.

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.

3.11 送信された Web フォーム データのアクセス方法

「データベースに保存」保存アクションを Web フォームに割り当てた場合、Web サイトの訪問者によって入力されたすべてのデータはモジュールのデータベースに保存されます。プログラムによって保存されたデータにアクセスし修正することができます。

この例では、次のアクションを実行します。

- 特定の Web フォームの送信をすべて取得します。
- 各 Web フォームの **Age** フィールドの値を取得します。
- **Age** フィールドの値が *Undefined* である場合は、*The age is not defined* に変更します。

```
string formId = "{C797CBE2-B5B9-4C5B-9B60-50438A1783A8}"; // replace the value with ID of your
form item
List<GridFilter> args = new List<Sitecore.Web.UI.Grids.GridFilter>();
args.Add(new GridFilter("storageName", string.Empty, GridFilter.FilterOperator.Contains));
args.Add(new GridFilter("dataKey", formId, GridFilter.FilterOperator.Contains));
var submits = DataManager.GetForms().GetPage(new PageCriteria(0, 0x7fffffff), null, args);

foreach (IForm submit in submits)
{
    foreach (IField field in submit.Field)
    {
        if (field.FieldName == "Age" && field.Value == "undefined")
        {
            field.Value = "The age is not defined";
            DataManager.UpdateForm(submit); // Call to save the changed field values
        }
    }
}
```

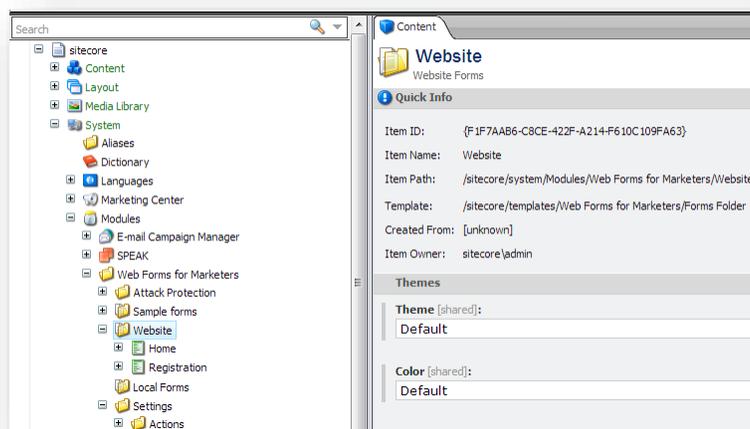
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

3.12 CSS テーマの使用方法

モジュールには Web フォームに適用可能ないくつかの CSS テーマが含まれます。テーマを変更するには、コンテンツ エディターで、Web フォームが保存されているフォルダーに移動します。このフォルダーは Sitecore.forms.config ファイルの Web サイトの formsRoot パラメーターで定義されます。このフォルダーは、/sitecore/Templates/Web Forms for Marketers/Forms Folder テンプレートに基づきます。

Web フォームのフォルダーには次のフィールドが含まれます。

- **Theme** — すべてのフォームで使用される外観のテーマを設定します。
- **Color** — すべてのフォームで使用される色のテーマを設定します。

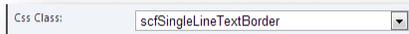


現行の Web サイトの すべての Web フォームは同じテーマと色を持ちます。これは、これらのパラメーターは Web フォームのフォルダーに設定するためです。

利用できるテーマのリストを拡張する場合、新しいテーマを登録する必要があります。新しいアイテムは、/sitecore/System/Modules/Web Forms for Marketers/Settings/Meta data/Themes フォルダー、または/sitecore/System/Modules/Web Forms for Marketers/Settings/Meta data/Colors フォルダーの配下に作成する必要があります。作成するアイテムの名前は、CSS スタイルを定義したファイルの名前と一致する必要があります。このファイルは、sitecore modules/shell/web forms for marketers/themes または sitecore modules/shell/web forms for marketers/themes/colors フォルダーにある必要があります。

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.

また、フィールド レベルで CSS クラスを変更することもできます。**フォーム デザイナー**で、すべてのフィールドで使用できる CSS のクラス プロパティを使用することができます。



新しい CSS クラスをリストに展開または追加するには、次の手順に従います。

1. CSS クラスの定義を Web サイトの %sitecore modules%Shell%Web Forms for Marketers%Themes%Custom.css ファイルに追加します。
2. /sitecore/System/Modules/Web Forms for Marketers/Settings/Meta data/Css Classes フォルダーの配下に、/sitecore/Templates/Web Forms for Marketers/Meta Data/Extended List Item テンプレートに基づく新しいアイテムを作成します。
3. 新しいアイテムの [バリュー] フィールドに、CSS クラスの名前を追加します。

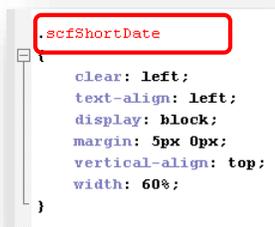
Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

3.13 CSS スタイルの設定方法

モジュールにはフォーム フィールドに適用可能な少数の CSS スタイルが含まれます。すべてのフィールド タイプには適用されるデフォルトの CSS スタイルがあります。CSS スタイルをカスタマイズすることができます。

新しい CSS スタイルを追加するには、次の手順に従います。

1. Website¥sitecore modules¥Shell¥Web Forms for Marketers¥Themes¥フォルダーの配下に配置された custom.css ファイルで、新しい CSS スタイルを定義します。



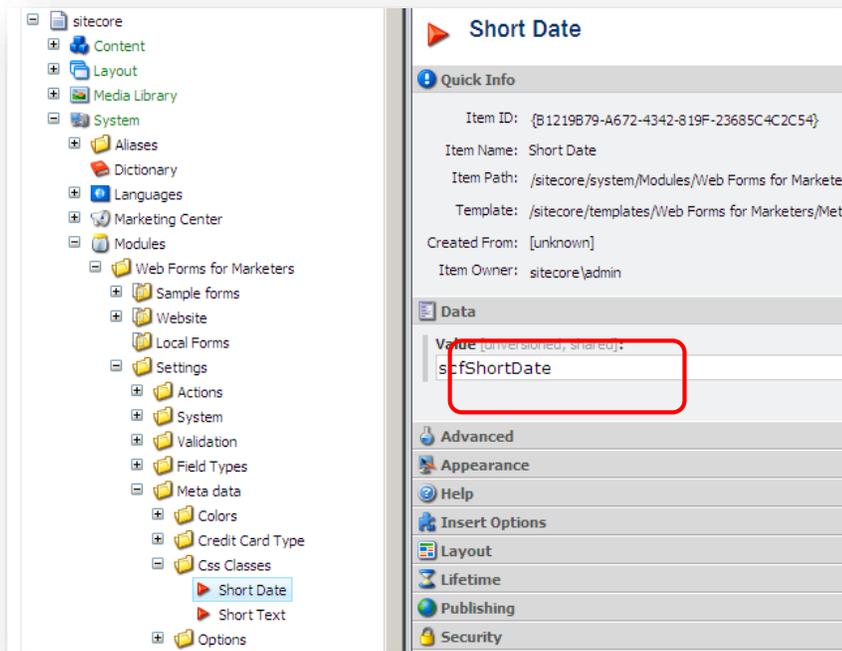
```
.scfShortDate
{
  clear: left;
  text-align: left;
  display: block;
  margin: 5px 0px;
  vertical-align: top;
  width: 60%;
}
```

2. **コンテンツ エディター**で、Sitecore/System/Modules/Web Forms for Marketers/Settings/Meta data/CSS Classes フォルダーに移動します。
3. 新しい **Extended List item** を作成します。

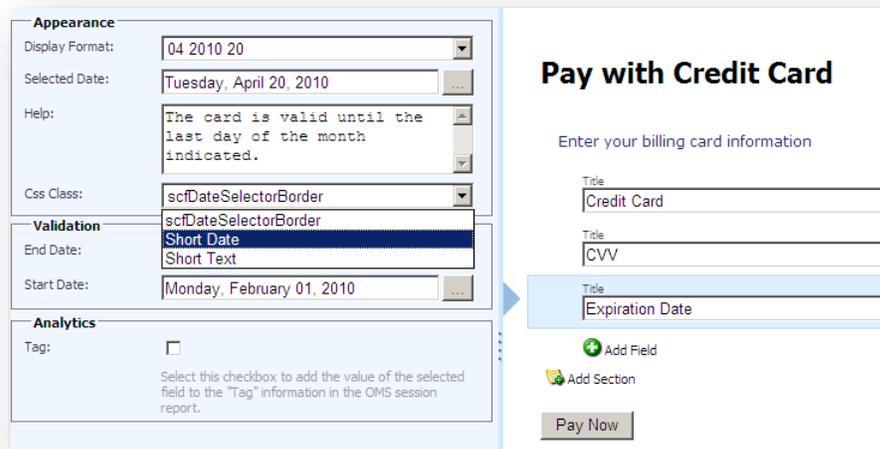


Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

4. 新しいアイテムの [バリュー] フィールドに、CSS スタイルの名前を入力します。



これでカスタムの CSS スタイルがシステムに追加されます。**フォーム デザイナー**で任意のフォーム フィールドに適用することができます。

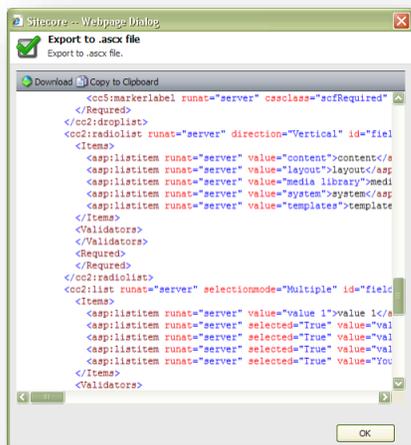


Sitecore® は登録商標です。その他の記載されているブランドや製品名はそれぞれ各社の所有物です。この文書の内容はサイトコアの所有物です。Copyright © 2001-2014 Sitecore. All rights reserved.

3.14 ASCX へのエクスポート方法

Web フォームを.ascx ファイルにコンバートすることは、他の Web フォームに影響を与えることなく特定の Web フォームの動作または外観を変更する場合に適したソリューションです。

フォーム デザイナーで、**[ascx ファイル]** をクリックし、フォームを.ascx ファイルにコンバートします。コードは **[.ascx ファイルにエクスポート]** ダイアログボックスに表示されます。



[ダウンロード] をクリックし、コードを.ascx ファイルとして保存します。

メモ

Web フォームを .ascx ファイルにコンバートした後、コンテンツ ツリーで Web フォーム アイテムを削除しないでください。このアイテムはアナリティクス レポートで必要です。

エクスポートされたフォームをアイテムに追加する方法

静的または動的にフォームをアイテムに追加することができます。静的にフォームをアイテムに追加する場合は、フォームをアイテムのプレゼンテーション レイアウトまたはそのサブレイアウトに追加します。

静的にフォームをレイアウトに追加するには、次の手順に従います。

1. **コンテンツ エディター**で、アイテムのレイアウトの定義アイテムを Sitecore/layout/layouts フォルダーから選択します。

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.

2. レイアウトの定義アイテムの [データ] セクションの [Path] フィールドの値を表示します。これはレイアウトの定義アイテムに対応するファイル システムの .aspx ファイルへのパスです。
3. ファイル システム上の.aspx ファイルを選択し、chapter 3.13「ASCX コントロールをページに追加する方法」の手順 3 に従って編集します。

静的にフォームをサブレイアウトに追加するには、次の手順に従います。

1. フォームを追加するアイテムに割り当てられたサブレイアウトを選択します。
2. **コンテンツ エディター**で、サブレイアウトの定義アイテムを Sitecore/layout/sublayouts フォルダから選択します。
3. このサブレイアウト アイテムの定義の Ascx ファイル フィールドで定義された .ascx ファイルを選択します。
4. chapter 3.13「ASCX コントロールをページに追加する方法」の手順 3 に従って、必要なタグを .ascx ファイルに追加します。

動的にフォームを追加するには、フォームをアイテムのプレゼンテーションに追加します。

1. **コンテンツ エディター**で、新しいサブレイアウト アイテムを作成します。
2. フォームの.ascx ファイルにバインドします (サブレイアウト アイテムの Ascx ファイル フィールドの値を実際の.ascx ファイルのパスに設定します)。
3. サブレイアウトを必要なアイテムのプレゼンテーションのプレースホルダーのコンテンツ アイテムに追加します (**コンテンツ エディター – プレゼンテーション – レイアウト – 詳細**)。

別のフィールドの値に基づいてフィールドを表示または非表示にする方法

この例では、.ascx ファイルへのエクスポート方法について説明します。たとえば、別のフィールドの値に基づいてフィールドを表示または非表示にする Web フォームが必要だとします。これを実装するには、次のアクションを実行します。

1. 単一行 テキスト フィールドとチェック ボックス フィールドを含む Web フォームを作成します。
2. Web フォームを .ascx ファイルへエクスポートします。

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.

3. エクスポートされた Web フォームを Web ページに追加します。これについての追加情報は、「ASCX コントロールをページに追加する方法」セクションを参照してください。
4. Web コントロール ファイルで、テキストとチェックボックス フィールドの <input id> タグを探します。たとえば:

```
テキスト フィールド ID <input  
id="WebUserControl1_field_5339820707D14FAC88D66DCC8F81EB01"
```

```
チェックボックス フィールド ID <input  
id="WebUserControl1_field_F2ACADD39B3C46A4A3036C63C0D60C3C"
```

5. チェックボックスの値に基づいてテキスト フィールドを表示または非表示にするクライアント スクリプトを作成します。

```
<script type="text/javascript">  
  var checkbox =  
document.getElementById("WebUserControl1_field_F2ACADD39B3C46A4A3036C63C0D60C3C");  
  var textbox =  
document.getElementById("WebUserControl1_field_5339820707D14FAC88D66DCC8F81EB01").parentNode.parentNode;  
  
  textbox.style.display = "none";  
  checkbox.onclick = function () {  
    if (checkbox.checked) {  
      textbox.style.display = "block";  
    }  
    else {  
      textbox.style.display = "none";  
    }  
  }  
</script>
```

6. このコードを .ascx ファイルの最後に配置します。

3.15 ASCX コントロールをページに追加する方法

1. レイアウト フォルダーで、新しい default.aspx ページを作成します。

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    </form>
</body>
</html>
```

2. [ascx ファイル] をクリックし、必要なフォームを layouts¥ フォルダーにエクスポートします。ファイルの名前には forms.ascx とします。

3. 次の変更を default.aspx ページに追加します。

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Register Src="form.ascx" TagName="SimpleForm" TagPrefix="uc1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <uc1:SimpleForm ID="WebUserControl1" runat="server" />
    </div>
  </form>
</body>
</html>
```

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.

ブラウザのアドレスバーに <http://localhost/layouts/default.aspx> と入力します。

エクスポートした .ascx ファイルに任意の変更を行った場合、モジュールはフォームが正しく動作するかの検証を行います。

3.16 モジュールを再インストールする方法

時々 モジュールの ファイルおよびアイテムがすべて正しいことを確実にするために Web Forms for Marketers モジュールを再インストールすることがあります。ファイルおよびアイテムは、アップグレード プロセスが成功しなかった場合に損なわれる可能性があります。

メモ

モジュールを再インストールする場合、すべての保存アクションのパラメーターはデフォルト値にリセットされます。

モジュールを再インストールし作成された Web フォームをすべて保存するには、次の手順に従います。

1. 作成された Web フォームのアイテムをすべて含む Sitecore パッケージを作成します。
2. モジュールのデータベースをバックアップします。
3. Sitecore.forms.config ファイルを修正した場合、このファイルをバックアップします。
4. Web Forms for Marketers のインストール パッケージをインストールします。
 - ダイアログが表示された場合は **[すべて上書き]** を選択します。
 - ダイアログが表示された場合は **[常に続行する]** を選択します。
5. インストールが終了した場合、Web フォームのアイテムを持つパッケージをインストールします。
6. モジュール データベースを復元します。
7. Sitecore.forms.config ファイルを復元します。
8. 保存アクションのパラメーターはデフォルト値にリセットされるため、これらのパラメーターを設定します。たとえば、電子メール メッセージを送信 保存アクションでは、host パラメーターを設定します。

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.

3.17 モジュールをアンインストールする方法

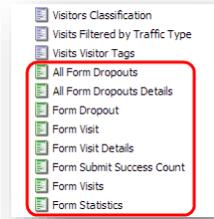
関連するファイルおよびアイテムをすべて削除することによって、モジュールをアンインストールすることができます。

Web Forms for Marketers モジュールをアンインストールするには、次のアクションを実行します。

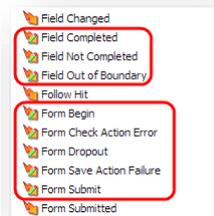
1. Master データベースで、次のアイテムを削除します。
 - /sitecore/layout/Renderings/Modules/Web Forms for Marketers
 - /sitecore/media library/Web Forms for Marketers
 - /sitecore/system/Modules/Web Forms for Marketers
 - /sitecore/system/Marketing Center/Goals/Leave a message
 - /sitecore/system/Marketing Center/Goals/Tell a Friend
 - /sitecore/system/Settings/Rules/Web Forms for Marketers Conditions
 - /sitecore/system/Settings/Analytics/Reports/Subreports/Web Forms for Marketers Reports
 - /sitecore/system/Settings/Analytics/Reports/Subreports/Web Forms for Marketers Detailed Reports
 - /sitecore/templates/Branches/Web Forms for Marketers
 - /sitecore/templates/Web Forms for Marketers

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.

- `/sitecore/system/Settings/Analytics/Reports SQL Queries` フォルダで  アイコンを持つ Web フォームに関連するすべてのレポート アイテム



- `/sitecore/system/Settings/Analytics/Page Events` フォルダで  アイコンを持つ Web フォームに関連するすべてのページ イベント アイテム。



- `/sitecore/system/Marketing Center/Goals` フォルダで Web フォームに関連するすべてのゴール アイテム。

2. Core データベースで次のアイテムを削除します。

- `/sitecore/content/Applications/Content Editor/Ribbons/Chunks/Forms`
- `/sitecore/content/Applications/Content Editor/Ribbons/Contextual Ribbons/Forms`
- `/sitecore/content/Applications/Content Editor/Ribbons/Strips/Presentation/Forms`
- `/sitecore/content/Applications/Modules/Web Forms for Marketers`
- `/sitecore/content/Applications/WebEdit/Custom Experience Buttons/Edit Form`

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.

- /sitecore/content/Documents and settings/All users/Start menu/Programs/Web Forms for Marketers

3. ファイル システムで、次のファイルを削除します。

- %App_Config%Include%Captcha.config
- %App_Config%Include%Sitecore.forms.config
- %bin% MSCaptcha.dll
- %bin% Sitecore.Forms.Core.dll
- %bin% Sitecore.Forms.Custom.dll
- %bin% System.Data.SQLite.dll
- %bin_x64% System.Data.SQLite.dll
- WFFM databases in the %Data folder
- %layouts%system% VisitorIdentificationExtension.aspx
- %sitecore%shell%Applications%Modules%Web Forms for Marketers
- %sitecore%shell%Themes%Standard%Default%WFM
- %sitecore%shell%Themes%Standard%Default%FormBuilder.css
- %sitecore%shell%Themes%Standard%Default%FormDataViewer.css
- %sitecore%shell%Themes%Standard%Default%MultiTreeView.css
- %sitecore%shell%Themes%Standard%Default%Placeholder.css
- %sitecore%shell%Themes%Standard%Firefox%WFM
- %sitecore%shell%Themes%Standard%Firefox% FormBuilder.css

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.

- ¥sitecore modules¥Shell¥Web Forms for Marketers
- ¥sitecore modules¥Web¥Web Forms for Marketers

4. Web サイトをパブリッシュします。