

ComponentOne for ASP.NET Web Forms ユーザーガイド

2020.04.17 更新

グレースシティ株式会社

目次

ComponentOne for ASP.NET Web Forms ユーザーガイド	2
コンポーネントをプロジェクトに組み込む方法	2
コンポーネントのランタイムファイル	2-3
エクスポートサービス	3-4
.NET Framework 4.0から 4.5.2への移行	4-6
各種オープンソースソフトウェアについて	6-7
テーマ	8
ThemeRoller for Visual Studio	8
ThemeRoller for Visual Studio クイックスタート	8
手順1:アプリケーションの作成	8
手順2:テーマのデザイン	8-11
手順3:アプリケーションの実行	11-12
ThemeRoller for Visual Studio の要素	12
[タスク]メニュー	12-13
「新しいテーマ」画面	13-14
ThemeRoller	14-17
Bootstrap for ASP.NET Web Forms クイックスタート	17-19
重要なヒント	20
クライアント側タスク別ヘルプ	21

ComponentOne for ASP.NET Web Forms ユーザーガイド

より少ないコードでより多くの成果を達成できる製品です。**ComponentOne for ASP.NET Web Forms**に収録されている40種類以上の優れたコンポーネントがAjax、jQueryに対応し、HTML5／CSS3に準拠しています。すべてのコントロールがASP.NETコントロールとMVCツールとして提供されています。

コンポーネントをプロジェクトに組み込む方法

コンポーネントの組み込み

Visual Studio では、ツールボックスにコンポーネントを追加しただけでは、プロジェクトにコンポーネントを追加したことにはなりません。プロジェクトの参照設定へ追加された時点でコンポーネントが組み込まれます。

以下のいずれかの操作を行うとプロジェクトへコンポーネントが組み込まれます。

1. フォームにコンポーネントを配置する
2. ソリューションエクスプローラ上で参照の追加を行う

プロジェクトに組み込まれているコンポーネントの一覧は、ソリューションエクスプローラで確認できます。また、各コンポーネントが使用している DLL もソリューションエクスプローラに登録される場合があります。詳細については、Visual Studio の製品ヘルプを参照してください。

本製品で使用しているアセンブリの一覧を以下に示します。

ファイル	内容
C1.C1Report.4.dll	本体アセンブリ
C1.Web.Wijmo.Controls.4.dll	本体アセンブリ
C1.Web.Wijmo.Controls.4.resources.dll	本体アセンブリのサテライトリソース
C1.Web.Wijmo.Controls.Design.4.dll	デザイナーアセンブリ
C1.Web.Wijmo.Controls.Design.4.resources.dll	デザイナーアセンブリのサテライトリソース

フォームにコンポーネントを配置する方法

以下に、コントロールをツールボックスに追加し、フォームに配置する方法を示します。これにより、コンポーネントがプロジェクトに組み込まれます。

配置手順

1. [ツール]メニューから[ツールボックス アイテムの選択]を選択します。
2. [ツールボックス アイテムの選択]ダイアログの「.NET Framework コンポーネント」タブを選択します。
3. 使用するコンポーネントのチェックボックスを ON にして(OK)ボタンをクリックしてください。ツールボックスに指定したコントロールのアイコンが表示されます。
4. ツールボックスから指定したコントロールのアイコンを選択してフォームに配置します。ソリューションエクスプローラの参照設定に指定したコントロールの名前空間が追加されます。


コンポーネントのランタイムファイル

ComponentOne for ASP.NET Web Forms のランタイムファイルは、次のファイルです。

アセンブリファイル

ComponentOne for ASP.NET Web Forms ユーザーガイド

ファイル	内容
C1.C1Report.4.dll	本体アセンブリ
C1.Web.Wijmo.Controls.4.dll	本体アセンブリ
C1.Web.Wijmo.Controls.4.resources.dll	本体アセンブリのサテライトリソース
C1ASPNETExportService.msiおよび、 このインストーラによりインストールされるファイル	Wijmoエクスポートサービス

 **注意:** 本製品に含まれているファイルのうち、上記以外のファイルは配布できません。

- C1.Web.Wijmo.Controls.Design.4.dll
- C1.Web.Wijmo.Controls.Design.4.resources.dll

エクスポートサービス

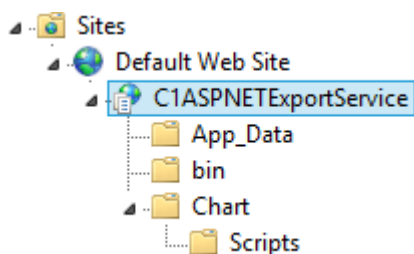
C1 ASP.NET Export Service を使用すると、複雑なエクスポートアプリケーションを作成しなくても、チャートを画像または PDF とグリッドをエクセル、CSVまたはPDFとしてにエクスポートできます。このサービスはアプリケーションサーバー上に置かれます。

C1 ASP.NET Export Service を使用する利点は次のとおりです。

- 書式設定を維持したままチャートまたはグリッドをエクスポートできます。
- エクスポートされたファイルの設定を必要に応じて変更できます。

インストール

Export Service は、インターネットインフォメーションサービス (IIS) 上に配布される Web アプリケーションです。C:\Program Files\ComponentOne\ASP.NET Web Forms フォルダにある **C1ASPNETExportService** インストーラを実行してください。次のファイルが IIS にインストールされます。



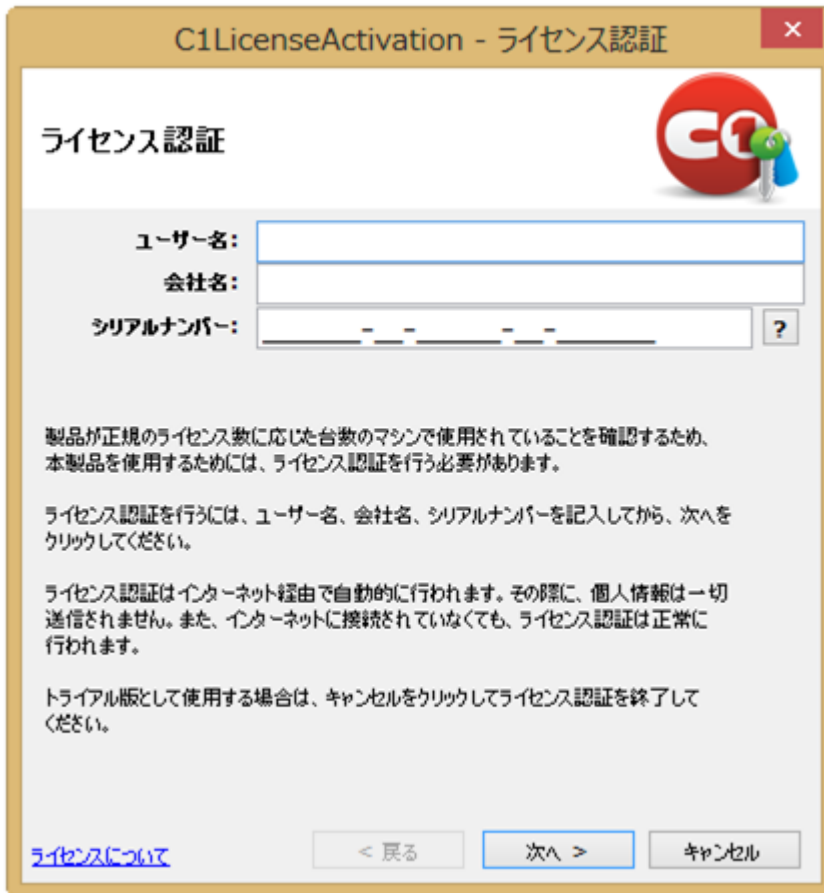
これらのファイルは、次の場所にも格納されています。

C:\ProgramData\ComponentOne\C1ASPNET\C1APNETExportService

ライセンス認証

Webサーバーでエクスポートサービスを実行するには、Webサーバー環境でライセンス認証を行う必要があります。ライセンス認証を行わないとトライアル版として実行され、エクスポートされたファイルにトライアル版であることを示すウォーターマークが埋め込まれます。トライアル版の使用期間 (30日間) を過ぎるとエラーが発生します。

ライセンス認証はエクスポートサービスのインストール中に自動的に実行されますので、ユーザー名、会社名、ComponentOne のシリアルナンバーを入力してください。




インストール後にライセンス認証を行う場合は、下記の場所にインストールされるライセンス認証ツールを実行してください。

C:\Program Files\Common Files\ComponentOne Shared\C1LicenseActivation.exe

システム要件

サービスホストのシステム要件は次のとおりです。

- Microsoft Windows 7 以上。
- ASP.NET 4.0 以上 (.NET Framework 4.0) を含む IIS 7.0 以上。
- サービスホスト上に IE9 以上。

 **Microsoft Windows 7** または **Microsoft Windows Server 2008 R2** では .Net Framework 4.0 をアップデートできません。詳細については、「<http://support.microsoft.com/kb/2468871>」を参照してください。

.NET Framework 4.0から 4.5.2への移行

ASP.NET Web Formsアプリケーションを.NET Framework 4.0から4.5.2に移行するには、次の手順に従います。

ターゲットフレームワークを変更するには


アプリケーションのターゲットフレームワーク設定を変更した後、.NET Framework 4.5.2をサポートするアセンブリ参照を更新する必要があります。参照を更新するには、次の手順に従います。

1. ソリューションエクスプローラから、[参照]フォルダを右クリックして、[参照の追加]オプションを選択します。
2. 表示されるプロパティウィンドウの[アプリケーション]タブにある[ターゲットフレームワーク]リストからフレームワークを「4.0」から「4.5.2」に変更します。
3. [ターゲットフレームワークの変更]ダイアログボックスで、「はい」をクリックして変更を保存します。

ComponentOne for ASP.NET Web Forms ユーザーガイド

C1アセンブリ参照を更新するには

1. ソリューションエクスプローラから、「参照」フォルダを右クリックして、「参照の追加」オプションを選択します。
2. 表示される[参照の追加]ダイアログで.NET Framework 4.5.2をサポートするC1アセンブリを追加します。

 **メモ:** デフォルトでは、C1アセンブリファイルは(C:\Program Files (x86)\ComponentOne\ASP.NET Web Forms Edition\Bin\v4)フォルダーにインストールされています。

license.licxファイルを更新するには

1. ソリューションエクスプローラにある「license.licx」ファイルをダブルクリックして開きます。
2. license.licxファイルで、すべてのアセンブリの記述を以下のように変更します。

変更前:

C1.Web.Wijmo.Controls.4 , Version=X.0.XXXXX.XXX

変更後:

C1.Web.Wijmo.Controls.45 , Version=X.5.XXXXX.XXX

注意:

- Versionは記載がある場合のみ変更が必要です。
- WinForms のアセンブリファイル(C1.C1Report.4.dll、C1.Win.4.dll、C1.C1Zip.4.dllなど)のバージョン記述を「4.5.2」バージョンに更新する必要がありません。

3. ファイルを保存します。

Web.configファイルを更新するには

1. ソリューションエクスプローラで、**Web.config**ファイルをダブルクリックして開きます。
2. ファイル全体で、すべてのアセンブリの記述を以下のように変更します。

変更前:

C1.Web.Wijmo.Controls.4 , Version=X.0.XXXXX.XXX

変更後:

C1.Web.Wijmo.Controls.45 , Version=X.5.XXXXX.XXX

注意:

- Versionは記載がある場合のみ変更が必要です。
- WinForms のアセンブリファイル(C1.C1Report.4.dll、C1.Win.4.dll、C1.C1Zip.4.dllなど)のバージョン記述を「4.5.2」バージョンに更新する必要がありません。

3. ファイルを保存します。

aspxファイルを更新するには

1. ソリューションエクスプローラで、ダブルクリックして**aspx**ファイルを開きます。
2. アセンブリの記述を以下のように変更します。

変更前:

```
<%@ Register assembly="C1.Web.Wijmo.Controls.4" namespace="C1.Web.Wijmo.Controls.XXXXXXXX" tagprefix="wijmo" %>
```

変更後:

```
<%@ Register assembly="C1.Web.Wijmo.Controls.45" namespace="C1.Web.Wijmo.Controls.XXXXXXXX" tagprefix="wijmo" %>
```

3. ファイルを保存します。

[ビルド] メニュー から [ソリューションのリビルド] を選択します。

各種オープンソースソフトウェアについて

当社では、各種オープンソースソフトウェア（以下、「OSS」といいます）の使用許諾契約および使用方法に基づき、複数の種類の OSS プログラムを使用しています。

本ソフトウェアの一部には以下の OSS が含まれます。

- **globalize**

Copyright JS Foundation and other contributors, <https://js.foundation>
(MIT license)

- **bgiframe**

Copyright 2013, Brandon Aaron (<http://brandonaaron.net/>)
(MIT license)

- **jQuery Mouse Wheel Plugin**

Copyright OpenJS Foundation and other contributors, <https://openjsf.org/>
(MIT license)

- **JavaScript Cookie**

Copyright (c) 2018 Copyright 2018 Klaus Hartl, Fagner Brack, GitHub Contributors
(MIT license)

- **Raphaël**

Copyright (c) 2008-2010 Dmitry Baranovskiy
(MIT license)

- **jQuery UI**

Copyright jQuery Foundation and other contributors, <https://jquery.org/>
(MIT license)

- **Bootstrap**

Copyright (c) 2011-2020 Twitter, Inc.
Copyright (c) 2011-2020 The Bootstrap Authors
(MIT license)

- **Breeze**

Copyright (c) 2012-2019 IdeaBlade
(MIT license)

- **Knockout**

Copyright (c) 2010 Steven Sanderson, the Knockout.js team, and other contributors

<https://knockoutjs.com/>

(MIT license)

- **Q**

Copyright 2009–2018 Kristopher Michael Kowal. All rights reserved.

(MIT license)

- **SWFObject**

Copyright (c) 2007-2015 The SWFObject team

(MIT license)

テーマ

CSSの知識が無くても、Wijmoコントロールでは容易にテーマを設定することができます。Wijmo では、**ThemeRoller for Visual Studio** コンポーネントが含まれ、**Bootstrap**テーマも対応されています。デザイン環境を離せず美しいカスタムテーマを作成してASP.NET Web Formsコントロールに適用するか、1クリックで**Bootstrap**テーマを組み込みます。

ThemeRoller for Visual Studio

ThemeRoller for Visual Studio は、ComponentOne for ASP.NET Web Forms コントロールに使用する美しいテーマを簡単に作成できるようにします。ThemeRoller for Visual Studio を使用して、アプリケーション全体のテーマを ComponentOne for ASP.NET Web Forms に適用できます。アプリケーションに **ComponentOne for ASP.NET Web Forms** コントロールを追加したら、コントロールのスマートタグから **ThemeRoller for Visual Studio** にアクセスすることができます。ThemeRoller は Visual Studio 内に開きます。**ThemeRoller** を使用することで、CSS やコントロールの知識がなくても、テーマをカスタマイズすることができます。

ThemeRoller for Visual Studio クイックスタート

このクイックスタートでは、**ComponentOne for ASP.NET Web Forms** コントロールを Visual Studio アプリケーションに追加する方法について説明します。ここでは、**スマートタグ**を使用して**[タスク]**メニューを開く、**ThemeRoller for Visual Studio** を開く、**ThemeRoller for Visual Studio** を使用してテーマをデザインするなどの操作を実行します。

 **メモ:**ここでは C1Calendar を例として説明します。なお、チャートとゲージを除く、他の **ComponentOne for ASP.NET Web Forms** コントロールでも、同様の手順で **ThemeRoller for Visual Studio** を使用することができます。

手順1: アプリケーションの作成

この手順では、Visual Studio でアプリケーションを作成し、適切なアセンブリ参照を追加して、ComponentOne for ASP.NET Web Forms コントロールをアプリケーションに追加します。


次の手順に従います。

1. Visual Studio の**[ファイル]**メニューから、**[新規作成]**→**[プロジェクト]**を選択します。**[新しいプロジェクト]**ダイアログボックスが表示されます。
2. **[新しいプロジェクト]**ダイアログボックスの左ペインで、言語を展開し、**[Web]**を選択します。右ペインで、**[ASP.NET 空の Web アプリケーション]**を選択し、アプリケーションの名前を入力して、**[OK]**を選択します。新しいアプリケーションが作成されます。
3. ソリューションエクスプローラで、プロジェクトを右クリックし、**[参照の追加]**を選択します。
4. **[参照の追加]**ダイアログボックスで、**C1.Web.Wijmo.Controls.4** アセンブリと **C1.Web.Wijmo.Controls.Design.4** アセンブリを見つけて選択し、**[OK]**をクリックします。参照が追加されます。
5. デザインビューで Visual Studio のツールボックスに移動し、**[C1Calendar]**アイコンをダブルクリックして、**C1Calendar** コントロールをページに追加します。

Visual Studio ASP.NET アプリケーションを作成し、適切なアセンブリ参照を追加し、**C1Calendar** コントロールをアプリケーションに追加しました。次の手順では、**ThemeRoller for Visual Studio** を使用して **C1Calendar** コントロールをカスタマイズします。

手順2: テーマのデザイン

この手順では、ThemeRoller for Visual Studio を使用してテーマをデザインします。

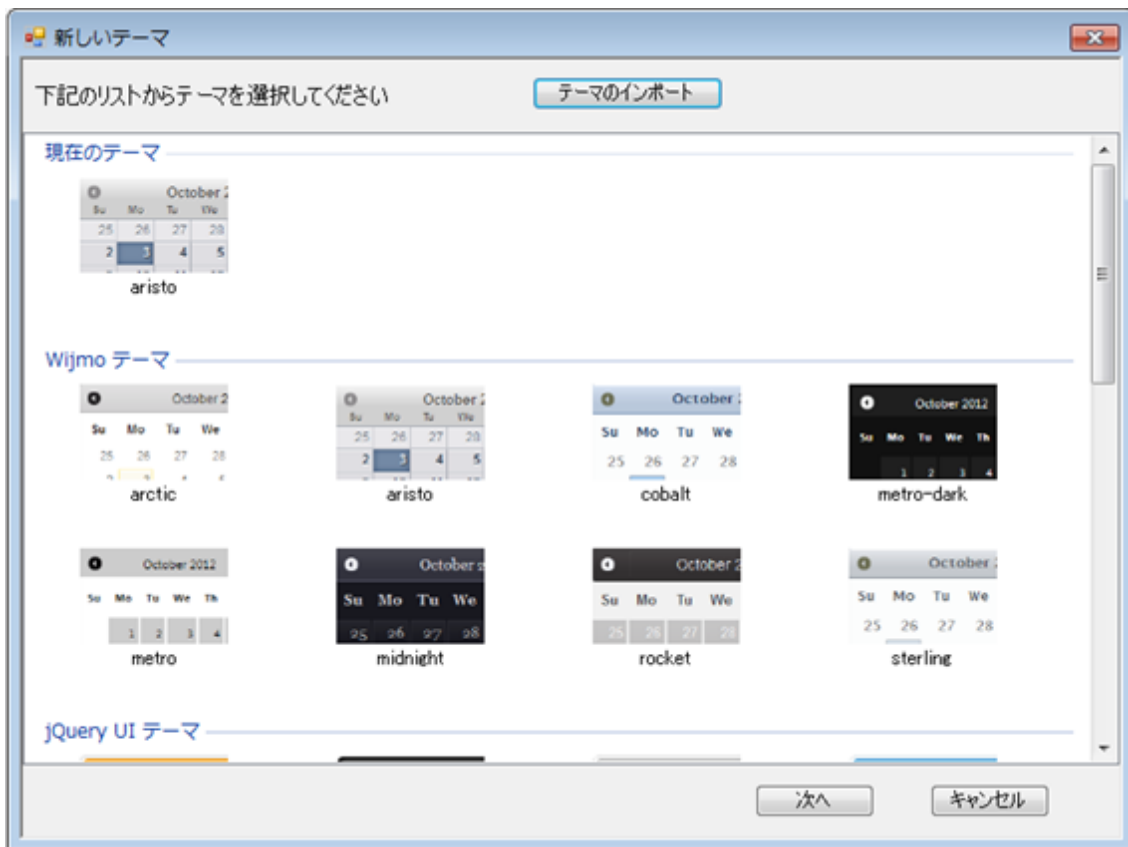
1. アプリケーション内の **C1Calendar** コントロールを選択し、スマートタグをクリックして**[C1Calendar のタスク]**メニューを開きます。



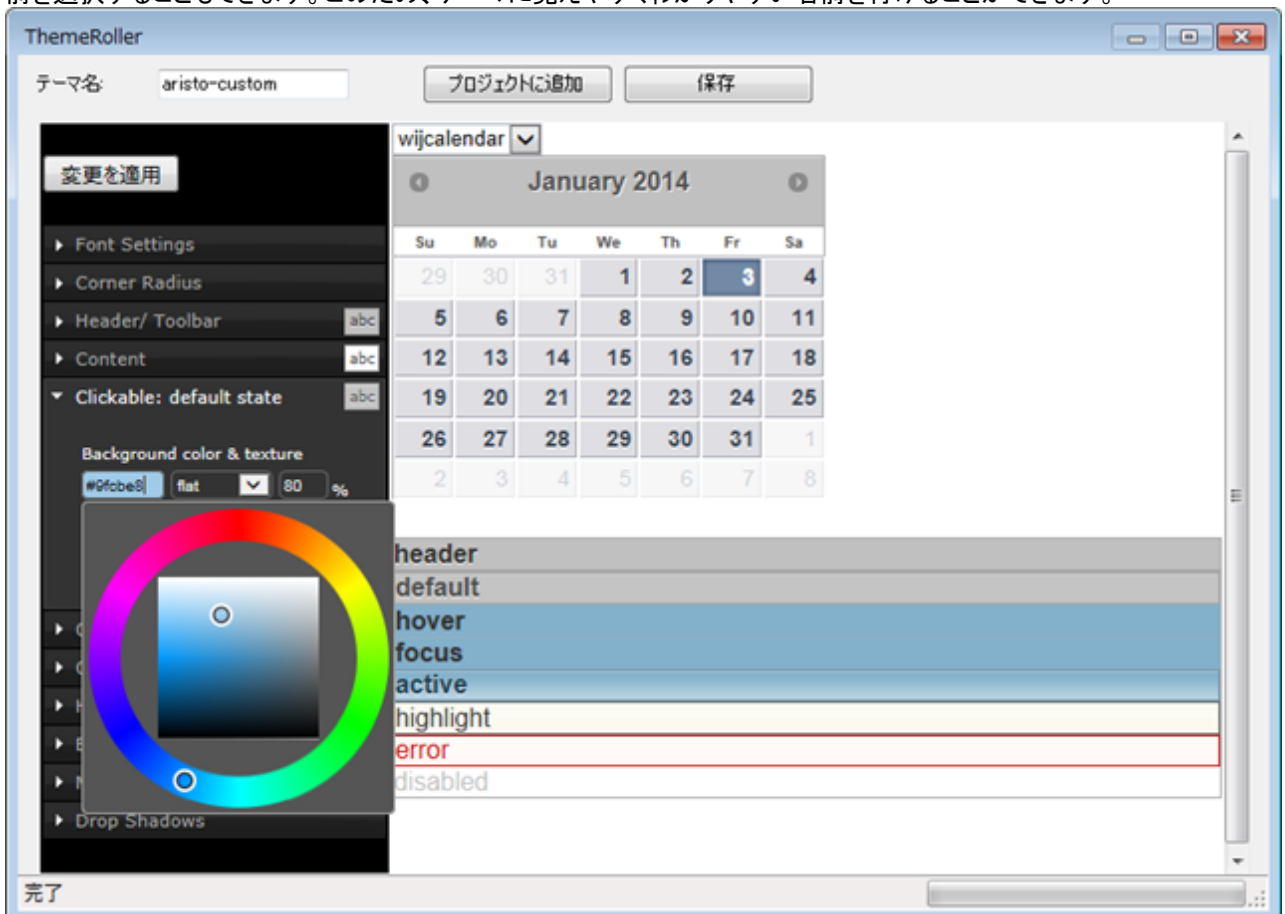
2. [C1Calendar のタスク]メニューから[新しいテーマの作成]を選択して、ThemeRoller for Visual Studio を起動します。



3. ThemeRoller for Visual Studio が起動したら、「新しいテーマ」画面から、最終的な製品のテーマによく似たテーマを選択します。このテーマから作業を開始することで、テーマのデザインにかかる時間を節約することができます。




4. [次へ]をクリックすると、ThemeRoller が起動します。ThemeRoller では、フォント、背景色、境界線、テキスト色などを変更することができます。ThemeRoller の右ペインに、デザイン中のテーマのプレビューが表示されます。テーマの名前を選択することもできます。このため、テーマに覚えやすくわかりやすい名前を付けることができます。



ComponentOne for ASP.NET Web Forms ユーザーガイド

5. テーマのカスタマイズが完了したら、[プロジェクトに追加]をクリックします。ThemeRoller は、自動的にテーマをプロジェクトに追加して参照します。プロジェクトにテーマが表示されたら、ThemeRoller を閉じます。

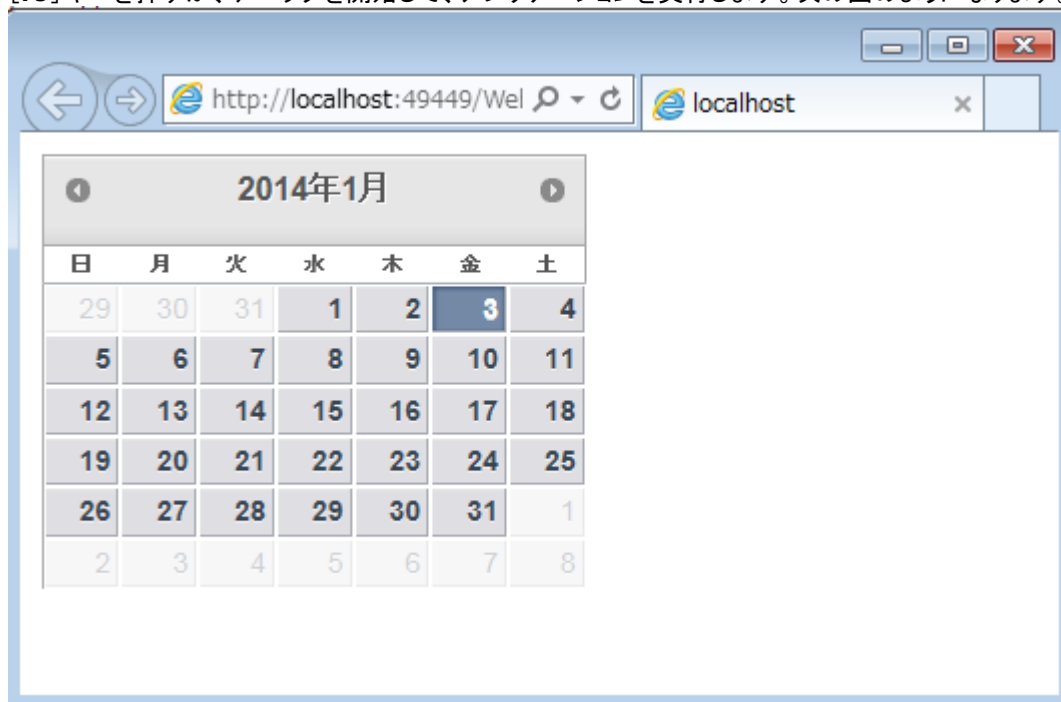
 **メモ:**プロジェクトに追加されたカスタムテーマは ~/Content/themes/ に保存されます。

この手順では、ThemeRoller テーマをカスタマイズしてプロジェクトに追加しました。次の手順では、このアプリケーションを実行します。

手順3:アプリケーションの実行

この手順ではアプリケーションを実行します。

1. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。次の図のようになります。



テーマに加えた変更を確認してください。

テーマをさらに変更する場合は、コントロールのスマートタグをクリックします。次の図に、ThemeRoller テーマを適用した後の**スマートタグ**の例を示します。

ThemeRoller で作成したテーマが[テーマ]ドロップダウンに表示され、[C1Calendar のタスク]メニューに[テーマのカスタマイズ]項目が追加されていることがわかります。この項目をクリックすることで、作成したテーマを編集したり、さらにカスタマイズすることができます。

ThemeRoller for Visual Studio の要素

以下のトピックでは、ThemeRoller for Visual Studio の構成要素について説明します。

メモ:ここでは C1Calendar を例として説明します。なお、チャートとゲージを除く、他の **ComponentOne for ASP.NET Web Forms** コントロールでも、同様の手順で **ThemeRoller for Visual Studio** を使用することができます。

[タスク]メニュー

ThemeRoller for Visual Studio には、任意の **ComponentOne for ASP.NET Web Forms** コントロールの [タスク]メニューおよび[新しいテーマの作成]オプションからアクセスできます。次の図にこのオプションを示します。



C1Calendar タスク

- 他の月日の表示
- 曜日の表示
- 週番号の表示
- 曜日の書式
- プレビューを有効にする
- テーマ
- [新しいテーマの作成...](#)
- CDN の使用
- CDN パス
- Bootstrap の使用
- Mobile Mode
- [バージョン情報...](#)

[新しいテーマの作成]オプションには、テーマを作成した後もアクセスできます。テーマの作成後に何らかの変更を加える場合は、[テーマのカスタマイズ]オプションを選択します。



C1Calendar タスク

- 他の月日の表示
- 曜日の表示
- 週番号の表示
- 曜日の書式
- プレビューを有効にする
- テーマ
- [テーマのカスタマイズ...](#)
- [新しいテーマの作成...](#)
- CDN の使用
- CDN パス
- Bootstrap の使用
- Mobile Mode
- [バージョン情報...](#)

[テーマのカスタマイズ]を選択すると、ThemeRoller ウィンドウが再度開きます。このウィンドウで新しいテーマをさらにカスタマイズすることができます。

「新しいテーマ」画面

コントロールの [タスク] メニューから [新しいテーマの作成] を選択すると、「新しいテーマ」画面が開きます。「新しいテーマ」画

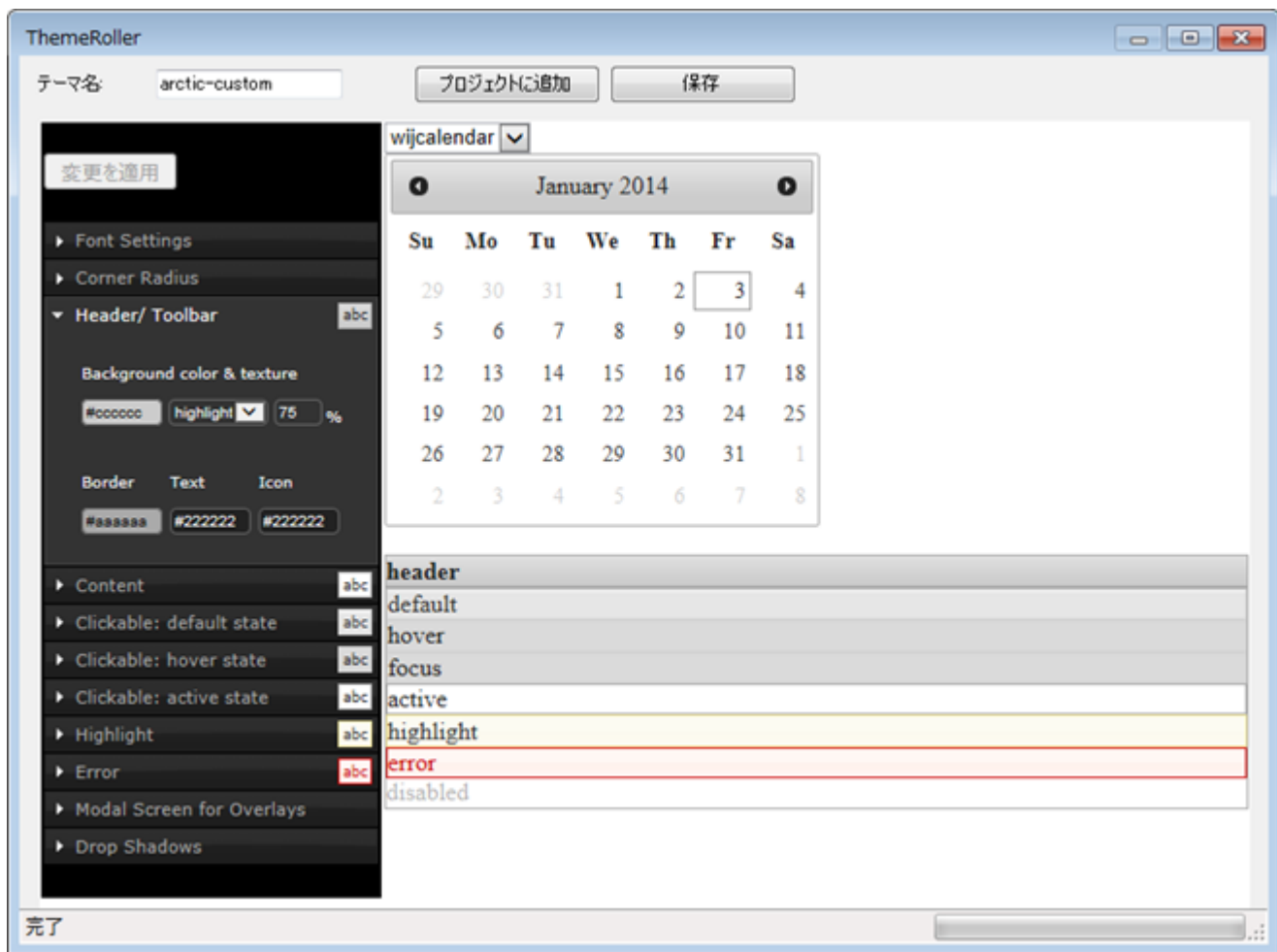
面では、用意されている Wijmo テーマまたは jQuery UI テーマのいずれかを基本テーマとして選択することができます。



以前にカスタマイズしたテーマも【ユーザーテーマ】セクションに表示され、他のテーマと同様にアクセスして編集することができます。【テーマのインポート】ボタンを使用してテーマをインポートすることで、外部のテーマをカスタマイズすることもできます。【次へ】をクリックすると、ThemeRoller が開きます。

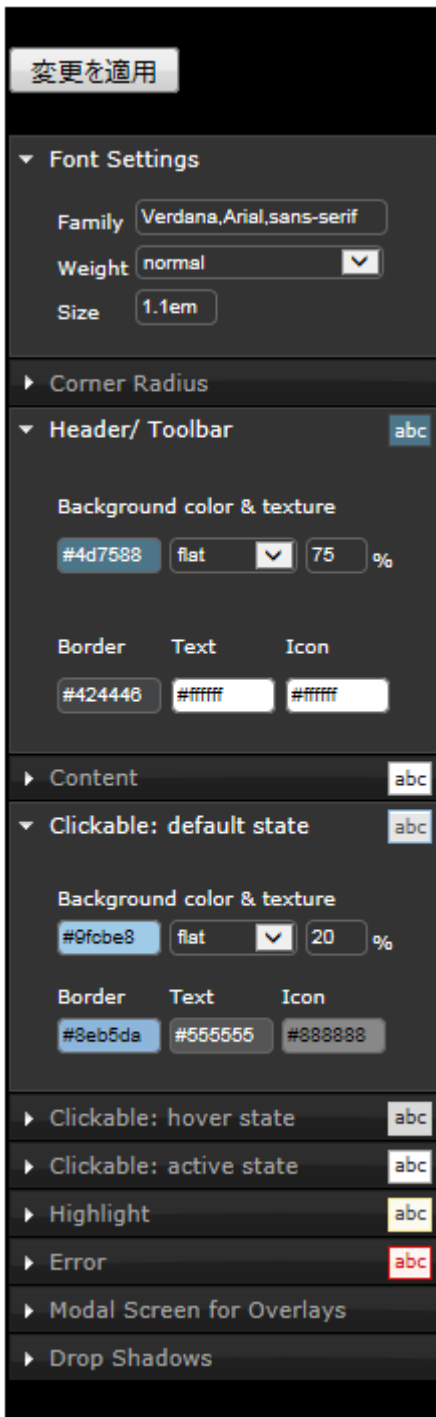
ThemeRoller

ThemeRoller for Visual Studio を使用して、使用するコントロールの外観のほとんどの要素をカスタマイズすることができます。別のテーマと間違えないように、自分のテーマには独自の名前を付けてください。カスタマイズのベースとして選択したテーマの名前に「-custom」を付けたものが、初期のテーマ名になります。つまり、**arctic** テーマを選択した場合、初期のテーマ名は「**arctic-custom**」になります。

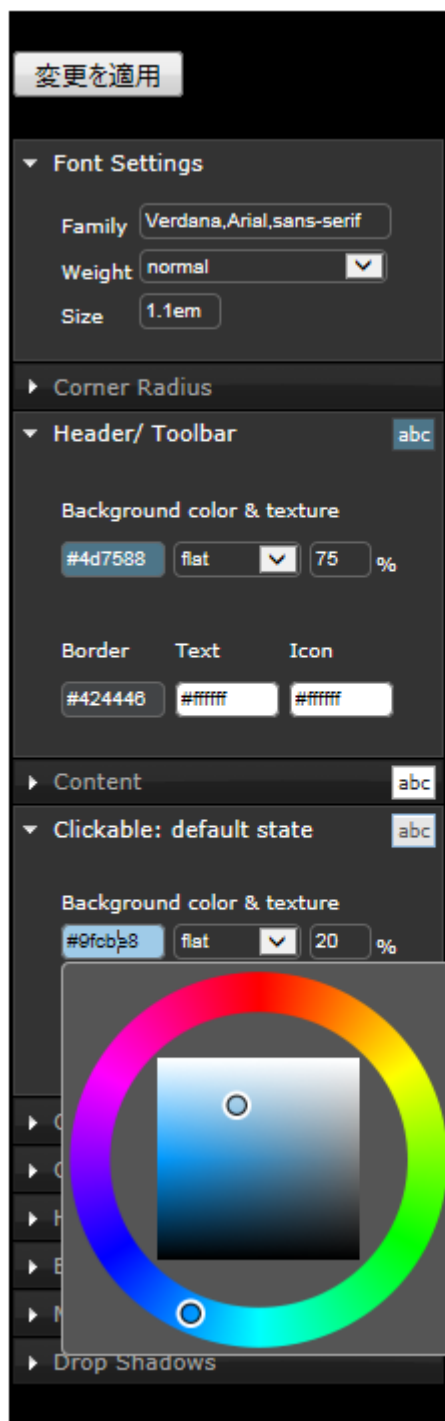


[ThemeRoller]ウィンドウには、ウィンドウの左側に設定パネル、設定パネルの右側にコントロールのプレビュー、コントロールのプレビューの下にプレビュースウォッチが表示されます。

ThemeRoller の左側の設定パネルでは、フォント、フォントスタイル、背景色、さまざまなインタラクティブ操作状態、オーバーレイ、シャドウなどを変更することができます。設定パネル内で変更を加えると、コントロールのプレビューとプレビュースウォッチに自動的に適用されます。



任意の設定の色入力領域をクリックすると、カラーピッカーが開きます。



変更を加えると、[ThemeRoller] ウィンドウのコントロールのプレビューと、コントロールの下のスウォッチに自動的に適用されます。

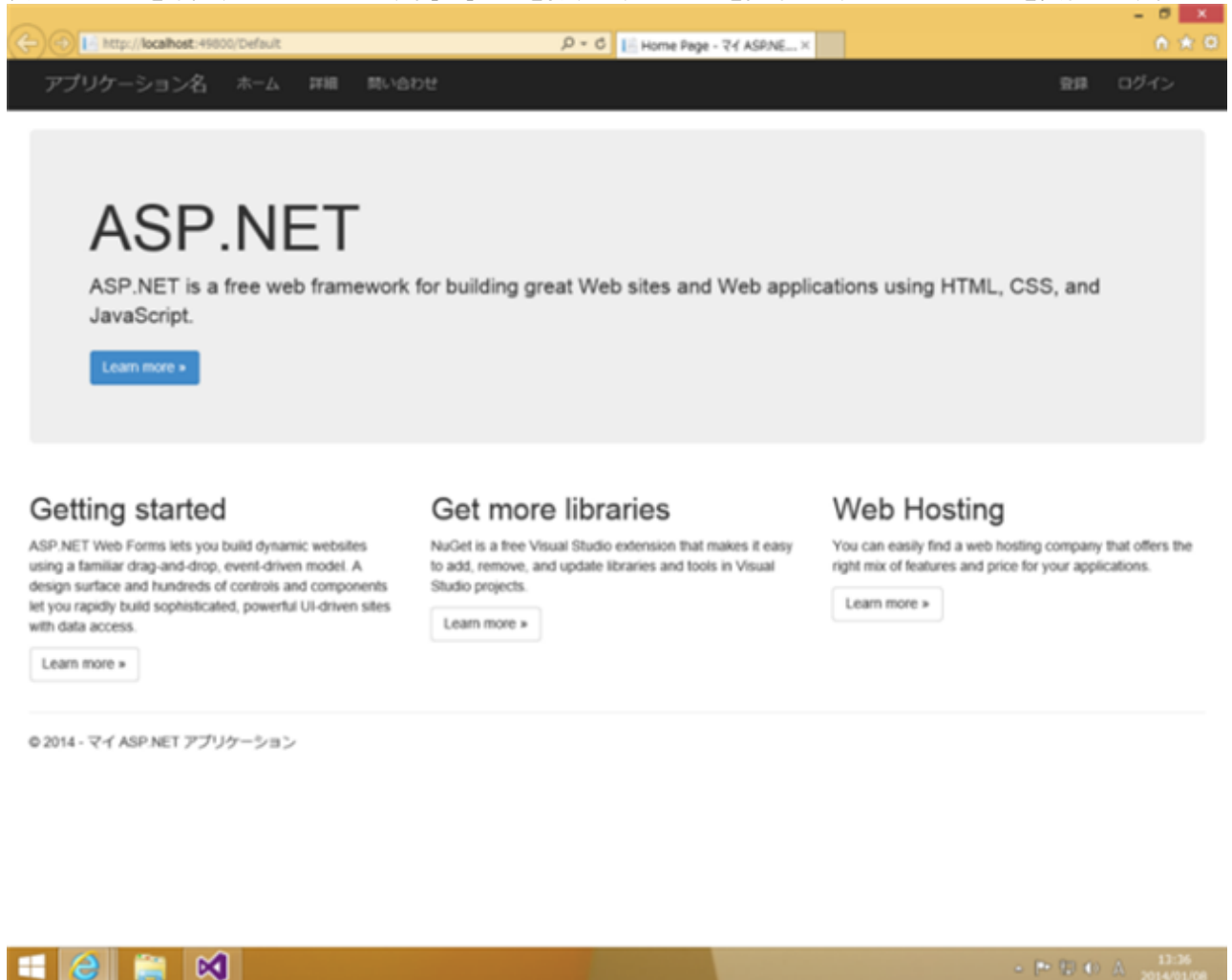
Bootstrap for ASP.NET Web Forms クイックスタート


Visual Studio 2013 では、Bootstrap テーマフレームワークを使用した新しいプロジェクトテンプレートが導入されました。**ComponentOne for ASP.NET Web Forms** も Bootstrap を正式にサポートするようになりました。

次の手順に従って、Bootstrap を **ComponentOne for ASP.NET Web Forms** コントロールに簡単に組み込めることを確認してください。

 **メモ:** ここでは C1Calendar を例として説明します。なお、チャートとゲージを除く、他の **ComponentOne for ASP.NET Web Forms** コントロールでも、同様の手順で **BootStrap** を使用することができます。

1. [ファイル]メニューを選択します。[ファイル]メニューのリストから、[新規作成]→[プロジェクト]を選択します。
2. [新しいプロジェクト]ダイアログボックスの左ペインで、[テンプレート]からいずれかの言語を選択します。
3. [新規 ASP.NET プロジェクト]ダイアログボックスで、使用可能なテンプレートから[Web フォーム]を選択し、[プロジェクトの作成]を選択します。
4. **Default.aspx** ファイルをダブルクリックして開きます。
5. Bootstrap は既にテンプレートに含まれているため、アプリケーションを実行して、Bootstrap がデフォルトでどのように表示されるかを確認することができます。[F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。



6. デバッグを停止して、デザインビューに切り替えます。
7. Visual Studio ツールボックスで **C1Calendar** コントロールを見つけます。コントロールをデザインサーフェスにドラッグしてアプリケーションに追加します。
8. コントロールを選択し、スマートタグ  をクリックして、[C1Calendar のタスク]メニューを開きます。



9. [Bootstrap の使用] オプションを選択して、Bootstrap をアプリケーションに組み込みます。このようにとても簡単です。
10. [F5] キーを押すか、デバッグを開始して、アプリケーションを実行します。C1Calendar は次の図のように表示されます。



おめでとうございます。これで、「Bootstrap for ASP.NET Web Forms クイックスタート」は終了です。このトピックでは、新しい Visual Studio 2013 ASP.NET Web フォームアプリケーションを作成し、ComponentOne for ASP.NET Web Forms コントロールをアプリケーションに追加し、Bootstrap テーマを追加しました。

重要なヒント

次のヒントは **ComponentOne for ASP.NET Web Forms** で作業する場合の問題解決に役立ちます。

ヒント1:メタタグを編集してレンダリングを修正することにより、Quirks モードでの質の悪いページレンダリングを防止する。

ユーザーのブラウザが Quirks モードでページをレンダリングしている場合、ウィジェットとコントロールがページ上に正しく表示されない場合があります。これは、アドレスバーに壊れたページアイコンが表示されることでわかります。互換表示では、ブラウザは古いレンダリングエンジンを使用します。




ユーザーがこの表示を設定して問題が発生した可能性があります。Quirks モードでのレンダリングを防止するために、ページが最新のブラウザでレンダリングされるように設定できます。以下のメタタグをページのヘッダーに追加します。

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
```

クライアント側タスク別ヘルプ

クライアント側のタスク別ヘルプは、「[MVC Tools ウィジェット](#)」の内容を参考にしてください。また、クライアント側リファレンスについては、下記のリンクを参照してください。

- [jQuery と jQuery UI の概要](#)
- [Wijmo API](#)

 **注意:** 上記リンク先のサンプルコードに記載されている `$("#accordion").wijaccordion` のような箇所を、Wijmo コントロールの場合には `$("#<%=C1Accordion1.ClientID%>").c1accordion` のように置き換える必要があります。