

FlexReport for UWP

2018.04.10 更新

グレースィティ株式会社

目次

FlexReport for UWP の概要	2
主な機能	3
FlexReport の詳細情報	4
コンポーネントとコントロール	4
オブジェクトモデルの概要	4-6
FlexReport のセクション	6-7
クイックスタート	8
手順 1: レポート定義の作成	8
手順 2: レポートのロードとレンダリング	8-10
FlexReport の操作	11
FlexReport のデータ連結	11
外部オブジェクトを使用したデータ連結	11-14
レポートのエクスポート	14-17
印刷	17
FlexReportDesigner について	18
FlexViewer for UWP	19
主な機能	19-20
ツールバー	20-21
レポートビューの回転	21-22
FlexReport と FlexViewer の連結	22-23

FlexReport for UWP の概要

ComponentOneは、UWP アプリケーションにレポート機能を追加する **FlexReport for UWP** を導入しました。**FlexReport for UWP** は、複雑なレポートの作成からプレビュー、エクスポートまで、完全なレポートソリューションを提供する包括的なレポートツールです。優れたオブジェクトモデル、プレビューペイン、高品質なレンダリング、さらに使いやすさを備えた **FlexReport** は、基礎レベルのレポート設計者だけでなく、上級レベルのレポート設計者にとっても必須のコントロールです。

FlexReport for UWP は、**FlexReport for WinForms** と全体的に同じアプローチを使用します。つまり、レポートは、サーバーで生成されるのではなく、UWP アプリケーション内でローカルに生成されます。アプリケーションは自己完結型なので、レポートを提供するために別の Web サービスを作成する必要はありません。

主な機能

FlexReport for UWP の主要な機能は次のとおりです。

- **軽量、高速**

FlexReport は、特に小さなレポートで軽量かつ高速です。

- **レンダリングの向上**

FlexReport は、DirectWrite/DirectX 技術による最新のレンダリングを使用して、高パフォーマンスのレポートコンテンツを描画および生成します。これにより、テキスト、図形、境界線のレンダリングが向上し、精度と品質が高まります。

- **モダン UI を使用したプレビューコントロール**

FlexViewer コントロールは、FlexReport をロードおよび表示できるレポートプレビュー機能を備えています。レポートを印刷する前にレポートページ間の移動やページ設定の変更を行ったり、レポートの印刷や複数の形式へのレポートのエクスポートを行うことができます。

- **サポートされるデータプロバイダ**

FlexReport for UWP は、現在、次のデータプロバイダをサポートしています。

- SQLite
- **IC1FlexReportExternalRecordset** をサポートする外部 DLL 内のオブジェクト
- **IC1FlexReportRecordset** または IList のいずれかをサポートするオブジェクト(コード内で DataSource.RecordSet に割り当てる必要があります)

- **エクスポート機能**

UWP アプリケーションで FlexReport を PDF、HTML、DOCX、RTF、XLSX、BMP、PNG、JPEG、GIF、TIFF の各形式にエクスポートできます。

パブリックプロパティとパブリックメソッドの変更

UWP バージョンの FlexReport では、使いやすさを向上させるために、いくつかのプロパティとメソッドが変更されました。

- UWP バージョンの **C1FlexReport.BasePath** プロパティと **C1FlexReport.DefaultBasePath** プロパティは StorageFolder 型です。
- StorageFile を引数として受け取るオーバーロードメソッド **C1FlexReport.Save(...)** および **C1FlexReport.Load(...)** が追加されました。

FlexReport の制限

- **レポートのロード**

- FlexReport for UWP は、従来の C1Report レポート定義ファイル(.xml)を直接ロードできません。そのようなレポート定義を使用するには、まずそれらを FlexReport 形式に変換する必要があります(WinForms 用 FlexReport デザイナアプリケーションを使用するなど)。その後、変換された .flxr レポート定義を使用します。

- **まだサポートされていない機能**

- メタファイル(メタファイル画像が使用されているレポートでは、メタファイルが表示されない)
- チャートフィールド、RTFField、および Field.RTF=True である通常のフィールド

- **サポートされないデータプロバイダ**

- UWP には ADO.NET がないため、次のデータプロバイダはサポートされません。
 - OLEDB
 - ODBC
 - SQLServerCe3_5
 - SQLServerCe4_0
 - XMLFile データプロバイダ

- **レポートの出力**

- C1FlexReport.Print(...) メソッドはサポートされません。代わりに、**C1DocumentSource.ShowPrintUIAsync()** メソッドが提供されています。

FlexReport の詳細情報

FlexReport コントロールの使用や操作を開始する前に、FlexReport に付属しているコンポーネントとコントロール、FlexReport のオブジェクトモデル、FlexReport で使用できるセクションまたはテーブル領域について理解しておく必要があります。以下のセクションで、これらの詳細について説明します。

コンポーネントとコントロール

FlexReport は、以下のアセンブリで構成されています。

C1.UWP.FlexReport dll

以下のコンポーネントを介してレポート生成/レンダリング機能を提供します。

- **C1FlexReport:**
C1FlexReport は、テーブル領域型レポートを生成するレポート生成コンポーネントです。レポートをプレビューコントロールに直接レンダリングしたり、さまざまな移植可能な形式 (PDF、XLSX、HTML、DOCX など) にエクスポートすることができます。

C1.UWP.FlexViewer dll


以下のレポート表示/UI コンポーネントが含まれます。

- **C1FlexViewer:**
C1FlexViewer コントロールは、レポートのズーム、エクスポート、印刷などの機能にアクセスできる高性能の UI を備えたフル装備のレポート/ドキュメント表示コントロールです。
- **C1FlexViewerPane:**
C1FlexViewerPane は、プログラムでズーム、回転などのビュー操作が可能な必要最小限の機能を備えたプレビューコントロールです。

組み込まれているアプリケーション

レポート作成用のコンポーネントとコントロールに加えて、FlexReport には以下のスタンドアロンアプリケーションも含まれます。

- **C1FlexReportDesigner**
C1FlexReportDesigner は、**C1FlexReport** レポート定義ファイル (.FLXR) の作成および編集に使用できるデスクトップアプリケーションです。従来の **C1Report** レポート定義 (.XML) を新しい .FLXR 形式に変換することもできます。**C1FlexReportDesigner** は、製品の UWP バージョンと WinForms バージョンの両方と互換性があるレポートを設計できます。デザイナーアプリは 2 種類あります。「すべての CPU」をターゲットとして x64 システムの 64 ビットモードで実行される **C1FlexReportDesigner.4.exe** と、x86 をターゲットとして常に 32 ビットモードで実行される **C1FlexReportDesigner32.4.exe** です。これらのアプリケーションは、次のパスにあります。
C:\Program Files (x86)\ComponentOne\Apps\v4.0

 **C1FlexReportDesigner** アプリケーションは、ComponentOne for WinForms と共にインストールされています。ディレクトリは、WinForms Edition と共にインストールされるデフォルトのインストールパスを反映します。

オブジェクトモデルの概要

FlexReport は優れたオブジェクトモデルを備え、その大部分は Microsoft Access モデルに基づいています。これらのオブジェクト、コレクション、関連するプロパティやメソッドなどにより、FlexReport を柔軟かつ容易に生成することが可能です。次の表に、オブジェクトおよびそのプロパティとメソッドを一覧します。

C1FlexReport

プロパティ: **Credential, DataSource, Document, FileName, Layout, MaxPages, OnClose, OnError, OnOpen, Page,**

FlexReport for UWP

Parameters, ReportDefinition, ReportInfo, ReportName, Sections
メソッド: Evaluate, Execute, GetReportList, Load, Render, Save, Clear

Layout

プロパティ: Width, MarginLeft, MarginTop, MarginRight, MarginBottom, PaperSize, Orientation, Columns, ColumnLayout, PageHeader, PageFooter

DataSource

プロパティ: CalculatedFields, ConnectionString, Filter, RecordSource, SortDefinitions

DataSourceCollection

プロパティ: Report
メソッド: Add, RemoveAt

SortDefinition

プロパティ: Direction, Expression, Owner

SortDefinitionCollection

プロパティ: Owner, Report

CalculatedField

プロパティ: DataSource, Expression, Type

CalculatedFieldCollection

プロパティ: Owner, Report

Group

プロパティ: GroupBy, KeepTogether, SectionHeader, SectionFooter, Sort, SortExpression

GroupCollection

プロパティ: Report
メソッド: Add, Clear, RemoveAt

ReportParameter

プロパティ: AllowedValuesDefinition, DisplayText, ParentReport
メソッド: SetName

ReportParameterCollection

プロパティ: Item, Report
メソッド: InsertItem, RemoveItem, SetItem

AllowedValuesDefinition

プロパティ: Binding, Values
メソッド: AssignFrom

Section

プロパティ: Calculated, Fields, Height, SplitBehavior, SubSections

SectionCollection

プロパティ: Detail, Footer, Header, PageFooter, PageHeader

SubSection

プロパティ: Calculated, Fields, Height, ParentReport, ParentSection, SplitBehavior, Visible
SubSectionCollection
プロパティ: Report メソッド: Add, Remove, RemoveAt
FieldBase
プロパティ: Anchor, Height, ForcePageBreak, MarginBottom, MarginLeft, MarginRight, MarginTop, Section, SplitHorzBehavior, SplitVertBehavior
FieldCollection
メソッド: Add, Remove, RemoveAt
BarCodeField
プロパティ: BarCode, BarCodeOptions, Font, Text
CheckBoxField
プロパティ: CheckAlign, CheckMark, Text, ThreeState, Value
DataField
プロパティ: Calculated, Name, Type, Value
ImageField
プロパティ: Picture, PictureAlign, PictureScale
ShapeField
プロパティ: Line, Shape, ShapeBackground, ShapeType
SubreportField
プロパティ: ParameterValues, Subreport, SubreportFilter
TextField
プロパティ: Format, Text
VisualReportObject
プロパティ: Background, Border, BordersSplitHorzMode, BordersSplitVertMode, OutlineLabel
BehaviorOptions
プロパティ: AllowHorizontalSplitting, EnableAggregatesOnReportFields, IgnoreInvisibleFieldsInGrowShrinkSections メソッド: AssignFrom, Reset

FlexReport のセクション


すべてのレポートは、次の 5 つの基本セクションで構成されます。

セクション	説明
詳細	詳細セクションには、ソースレコードセット内の各レコードごとに一度ずつレンダリングされるフィールドが含まれます。
ヘッダー	レポートのヘッダーセクションは、レポートの最初にレンダリングされます。

セクション	説明
フッター	レポートのフッターセクションは、レポートの最後にレンダリングされます。
ページヘッダー	ページヘッダーセクションは、各ページの上部にレンダリングされます(オプションで、レポートヘッダーを含むページを除外することもできます)。
ページフッター	ページフッターセクションは、各ページの下部にレンダリングされます。

グループヘッダーと**グループフッター**は、グループごとに追加される 2 つのセクションです。たとえば、3 つのグループレベルを持つレポートには 11 のセクションがあることになります。

各セクションはいくつかのサブセクションで構成され、ここに実際にレポートが表示されます。セクションには常に少なくとも 1 つのサブセクションが含まれます。さらにサブセクションを追加することで、機能を高めることができます。たとえば、何らかの条件に基づいて、サブセクションの表示/非表示をスクリプトで切り替えることができます。

 これらのセクションを非表示にすることはできますが、グループを追加または削除しない限り、セクションを追加または削除することはできません。

すべてのセクションの詳細については、「[FlexReport セクション](#)」を参照してください。

クイックスタート

C1FlexReport は、デスクトップや Web 上のさまざまな目的で使用できますが、どの場合でも主な手順は同じです。

1. レポート定義を作成する

これには、**FlexReportDesigner** アプリケーションを使用し、一からレポートを設計するか、既存の Microsoft Access Reports や Crystal Reports からレポートをインポートします。コードを使用して作成することもできます。この場合は、オブジェクトモデルを使用してグループおよびフィールドを追加するか、カスタム FLXR ファイルを記述します。

2. C1FlexReport コンポーネントにレポート定義をロードする

上記の手段1で作成した.FLXRレポート定義ファイルをリソースファイルまたは埋め込みリソースとして使用できるようにします。実行時に、C1FlexReportの **Load**または **LoadAsync**メソッドを使用して、レポート定義を **C1FlexReport**コンポーネントにロードします。

3. レポートをレンダリングする

レポートをプレビューするには、アプリケーションにC1FlexViewerコントロールを追加し、実行時にC1FlexReportコンポーネントをビューアの**DocumentSource**プロパティに割り当てます。これにより、レポートが生成されてC1FlexViewerコントロールに表示されます。

詳細な手順は、以下のトピックで説明します。レポート定義の作成方法、レポート定義のロード方法、レポートのレンダリング方法、そのエクスポート方法を示します。

手順 1: レポート定義の作成

FlexReportDesigner アプリケーションまたはコードでレポート定義を作成します。既存の定義をロードして、**FlexViewer** コントロールでレンダリングするだけです。レポート定義を作成する最も簡単な方法は、**FlexReport** に付属するスタンドアロン **C1FlexReportDesigner** デスクトップアプリケーションを使用することです。

64 ビットプラットフォーム用の **C1FlexReportDesigner.4.exe** と 32 ビットプラットフォーム用の **C1FlexReportDesigner32.4.exe** がコンピュータの **C:\Program Files (x86)\ComponentOne\Apps\v4.0** フォルダにあります。

レポート定義は、**ComponentOne for WinForms** に付属する **FlexReportDesigner** アプリを使用して作成できます。レポート定義を作成するための手順は次のとおりです。

1. **FlexReportDesigner** アプリを実行し、**[ファイル]**メニューから**[新規]**を選択します。
2. デザイナの左端にある**[レポート]**タブの**[新しいレポート]**ドロップダウンをクリックし、**[レポートウィザード]**をクリックします。
3. **[データプロバイダ]**ドロップダウンから**[SQLite データプロバイダ]**を選択し、**[接続文字列]**テキストボックスの隣にある省略符ボタンをクリックして、**C1NWind.db** ファイルを選択します。
4. **[データソース]**タブからテーブルを選択し、**[次へ]**をクリックします。この例では、**Products** テーブルを選択します。
5. レポートをデータソースに接続した後に、フィールド、レイアウト、スタイルを選択し、レポートに適切な名前を付け、**[完了]**をクリックします。



?(SpecialFolder.SystemDefault) 構文を使用して **C1NWind.db** のパスを指定します。FlexReport for UWP の場合、FlexReport 接続文字列は、Application.Current.LocalFolder を指す特殊な構文をサポートします。この構文を使用した場合、レポート内の接続文字列は次のようになります。

```
Data Source=?(SpecialFolder.SystemDefault)\C1NWind.db
```

手順 2: レポートのロードとレンダリング

1. Visual Studio で新しい UWP プロジェクトを作成し、**[空のアプリケーション(ユニバーサル Windows)]**を選択します。
2. **C1FlexViewer** コントロールをメインページに追加します。

3. **C1.UWP.FlexReport** への参照を追加します。
4. **FlexReport.SQLite プロジェクト**をダウンロードし、ソリューションに追加します。
UWP Edition がシステムにインストールされている場合は、**Documents\ComponentOne Samples\UWP\C1.UWP.FlexReport\CS** フォルダにも FlexReport.SQLite プロジェクトがあります。
5. プロジェクトを追加したら、プロジェクトをリビルドします。
6. **ソリューションエクスプローラー**で既存のプロジェクトを右クリックし、**[追加]**→**[参照]**を選択します。**[参照マネージャー]**が開きます。
7. 左ペインから**[プロジェクト]**→**[ソリューション]**を選択し、**FlexReport.SQLite**を選択し、**[OK]**をクリックします。これで、**FlexReport.SQLite.dll** がプロジェクトの**[参照]**フォルダに追加されます。

 **メモ:** FlexReport で SQLite を使用できるようにするために、**FlexReport.SQLite プロジェクト**を参照としてアプリのプロジェクトに追加する必要があります。**FlexReport.SQLite** は、FlexReport アセンブリが SQLite をハード参照しなくても済むように、ラッパーとして機能します。

8. 「手順 1/2: レポート定義の作成」でデザイナーを使用して作成されたレポート定義ファイルをプロジェクトの**[アセット]**フォルダに追加し、その**[ビルドアクション]**プロパティを **Content** に設定します。
9. **C1NWind.mdb** データベースを**[アセット]**フォルダに追加し、その**[ビルドアクション]**プロパティを **Content** に設定します。
10. コードビューに以下の **Page_Loaded** イベントを追加して、FlexViewer コントロールでレポートをロードしてレンダリングします。

Visual Basic

```
Private Sub Page_Loaded(sender As Object, e As RoutedEventArgs)
    ' SQLite データベースファイルをアプリの Assets から LocalFolder にコピーします。
    .FLXR レポート定義では、
    ' これを ?(SpecialFolder.SystemDefault) として参照できます。
    ' Data Source=? (SpecialFolder.SystemDefault)\C1NWind.db
    ' レポートの設計時は、?(SpecialFolder.SystemDefault) は
    Environment.SpecialFolder.MyDocuments を指します。したがって、
    ' レポートデータベースファイルを MyDocuments フォルダに置くと、レポートの設計とテスト実行に便利です。
    Dim dbPath = Path.Combine(ApplicationData.Current.LocalFolder.Path,
        "C1NWind.db")
    If Not File.Exists(dbPath) Then
        File.Copy("Assets\C1NWind.db", dbPath)
    End If

    ' レポートを作成してロードします。
    Dim report As New C1FlexReport()
    Using fs As Stream = File.OpenRead("Assets/ProductsUWP.flxr")
        report.Load(fs, "ProductList")
    End Using

    ' レポートをビューアの DataSource に割り当てます。レポートは、ビューアに表示されるときに
    ' 自動的に生成されます(デフォルトでは非同期)。
    Me.flexViewer.DataSource = report
End Sub
```

o C#

```
private async void Page_Loaded(object sender, RoutedEventArgs e)
{
    // SQLite データベースファイルをアプリの Assets から LocalFolder にコピーします。.FLXR レポート定義では、
    var dbPath = Path.Combine(ApplicationData.Current.LocalFolder.Path, "C1NWind.db");
    if (!File.Exists(dbPath))
        File.Copy(@"Assets\C1NWind.db", dbPath);

    // レポートを作成してロードします。
    C1FlexReport report = new C1FlexReport();
    using (Stream fs = File.OpenRead("Assets/ProductsUWP.flxr"))
        report.Load(fs, "ProductList");

    // レポートをビューアの DataSource に割り当てます。レポートは、ビューアに表示されるときに
    // 自動的に生成されます(デフォルトでは非同期)。
}
```

```
    this.flexViewer.DocumentSource = report;  
}
```

FlexReport の操作

FlexReport for UWP は、さまざまなシナリオで使用できますが、ほとんどのシナリオに関連する主な手順は次のとおりです。

1. **レポート定義の作成 - C1FlexReportDesigner** デスクトップアプリケーションを使用して実行できます。このデザイナーで .FLXR レポート定義ファイルを作成したら、これをアプリケーションのランタイムから使用できるようにして、**C1FlexReport** コンポーネントにロードされるようにする必要があります。または、**C1FlexReport** コンポーネントのリッチオブジェクトモデルを使用して、レポート定義をコードで完全に実行時に作成する方法もあります。
2. **レポートへのデータの提供** - レポート定義は、UWP バージョンで使用できるデータソースを念頭に作成する必要があります。実行時にレポート定義を **C1FlexReport** コンポーネントにロードしたら、データをアクセス可能にして、レポートを生成できるようにする必要があります。レポートは完全にクライアントで生成されます。使用できるデータソースの詳細については、以下を参照してください。
3. **レポートのレンダリングと出力** - 通常、レポートは次のいずれかを目的として生成されます。
 - **プレビュー**: **C1FlexReport** を **C1FlexViewer.DocumentSource** プロパティに割り当てただけで、レポートが生成されて FlexViewer に表示されます。FlexViewer コントロールの UI を使用して、レポートを対話式に出力またはエクスポートすることもできます。
 - **印刷**: レポートをレンダリングし、**C1DocumentSource.ShowPrintUIAsync** メソッドを使用して直接印刷できます。
 - **エクスポート**: レポートは、**C1FlexReport.RenderToFilter**、**C1FlexReport.RenderToFilterAsync**、または **C1FlexReport.RenderToFilterAsyncEx** メソッドを使用して、PDF、HTML などのサポートされている形式に直接レンダリングできます。

FlexReport のデータ連結

FlexReport でレポートを作成するには、レポート定義のほかに実際のデータが必要です。通常は、データベースからデータを取得しますが、他の場所から取得することもできます。以下のトピックでは、SQLite や他のソースからデータを取得する方法を確認します。

外部オブジェクトを使用したデータ連結

FlexReport では、外部オブジェクトを使用して容易にデータ連結を行うことができます。ここでは、Open Data Protocol (OData) クライアントライブラリを使用したデータ連結について説明します。OData を使用すると、Representational State Transfer (REST) リソースと同様にデータにアクセスできます。Simple.OData.Client ライブラリを使用してデータを連結する方法を以下に示します。

レポート定義を作成する

コードでレポート定義を作成し、OData クライアントライブラリを使用してデータを連結します。

1. コードビューで次の名前空間を追加します。
using Simple.OData.Client;
2. 次のコードを使用して、FlexReport のオブジェクトを作成します。

Visual Basic

```
Dim _report As New C1FlexReport()
```

C#

```
C1FlexReport _report = new C1FlexReport();
```

3. 次のコードを追加して、OData サービスにデータを要求します。

Visual Basic

```
' OData サービスにデータを要求します
Dim client = New ODataClient(ODataUri)
' すべてのカテゴリと各カテゴリの製品を選択します
Dim categories = (Await client.[For](Of Category)
().Expand(Function(x) New From { _
    x.Products _
}).FindEntriesAsync()).ToList()
Dim products = (From c In categories From p In c.Products New With { _
    Key .CategoryID = c.ID, _
    Key .CategoryName = c.Name, _
    Key .ID = p.ID, _
    Key .Name = p.Name, _
    Key .Description = p.Description, _
    Key .ReleaseDate = p.ReleaseDate, _
    Key .DiscontinuedDate = p.DiscontinuedDate, _
    Key .Rating = p.Rating, _
    Key .Price = p.Price _
}).ToList()
```

C#

```
// OData サービスにデータを要求します
var client = new ODataClient(ODataUri);
// すべてのカテゴリと各カテゴリの製品を選択します
var categories = (await client.For<Category>().Expand(x => new {
x.Products })).FindEntriesAsync().ToList();
var products = (
from c in categories
from p in c.Products
select new
{
    CategoryID = c.ID,
    CategoryName = c.Name,
    ID = p.ID,
    Name = p.Name,
    Description = p.Description,
    ReleaseDate = p.ReleaseDate,
    DiscontinuedDate = p.DiscontinuedDate,
    Rating = p.Rating,
    Price = p.Price,
}).ToList();
```

4. アプリケーションに Resources という名前の新しいフォルダを追加し、それにレポートを追加します。ここでは、Reports.flxr レポートを使用します。
5. 次のコードを使用して、Resources フォルダからレポート定義をロードします。

Visual Basic

```
' リソースからレポート定義をロードします
Dim asm As Assembly = GetType(MainPage).GetTypeInfo().Assembly
Using stream As Stream =
```

```
asm.GetManifestResourceStream("Binding.Resources.Reports.flxr")
    _report.Load(stream, "Products")
End Using
```

```
' レポートにデータセットを割り当てます
_report.DataSource.Recordset = products
```

C#

```
// リソースからレポート定義をロードします
Assembly asm = typeof(MainPage).GetTypeInfo().Assembly;
using (Stream stream =
asm.GetManifestResourceStream("Binding.Resources.Reports.flxr"))
_report.Load(stream, "Products");

// レポートにデータセットを割り当てます
_report.DataSource.Recordset = products;
```

6. レポート定義をロードしたら、次のコードを使用してレポートをビルドし、それを FlexViewer コントロールに表示します。

Visual Basic

```
Try
    ' レポートをビルドします
    prMain.IsActive = True
    Await BuildProductsReport()
    prMain.IsActive = False

    ' プレビューペインにレポートを割り当てます
    flxViewer.DocumentSource = Nothing
    flxViewer.DocumentSource = _report
Catch ex As Exception
    Dim md As New MessageDialog(String.Format("Failed to show ""
{0}"" report, error:" & vbCr & vbLf & "{1}", reportName,
ex.Message))
    Await md.ShowAsync()
End Try
```

C#

```
try
{
    // レポートをビルドします
    prMain.IsActive = true;
    await BuildProductsReport();
    prMain.IsActive = false;

    // プレビューペインにレポートを割り当てます
    flxViewer.DocumentSource = null;
    flxViewer.DocumentSource = _report;
}
catch (Exception ex)
{
    MessageDialog md = new MessageDialog(string.Format("Failed to
show \"{0}\" report, error:\r\n{1}", reportName, ex.Message));
```

```

await md.ShowAsync();
}

```

レポートは、次の図のように表示されます。

The screenshot shows a window titled "Binding" with a dropdown menu showing "006" and "000cts". The main content is a report titled "Products List".

飲料

商品名	発売日	評価	価格
果汁100% オレンジ	24-08-2016	1	¥ 200.00
果汁100% グレープ	24-08-2016	1	¥ 200.00
果汁100% レモン	24-08-2016	1	¥ 1,000.00
果汁100% ビーチ	24-08-2016	1	¥ 1,000.00
コーヒーマイルド	24-08-2016	1	¥ 1,000.00
コーヒーピター	24-08-2016	1	¥ 1,000.00
コーヒーミルク	24-08-2016	1	¥ 500.00
ピリピリ ビール	24-08-2016	1	¥ 100.00
パードワイン	24-08-2016	1	¥ 210.00
ローカロリー牛乳	24-08-2016	1	¥ 50.00

調味料

商品名	発売日	評価	価格
ホワイトソルト	24-08-2016	1	¥ 100.00
ブラックペッパー	24-08-2016	1	¥ 100.00
ビュアシュガー	24-08-2016	1	¥ 1,000.00

レポートのエクスポート

レポートをファイルにエクスポートし、それをクライアントや同僚に電子メールで配布する場合があります。これは、FlexReport コントロールを使用して行うことができます。FlexReport は、次のエクスポート形式をサポートしています。

形式	説明
RTF (*.rtf)	一般的なワードプロセッサ (Microsoft Word や WordPad) で開くことができる RTF ファイルを作成します。これは、ページ付き XML ドキュメントまたは Open XML ドキュメントとして保存できます。
Microsoft Excel (*.xlsx)	Microsoft Excel で開くことができる XLSX ファイルを作成します。
TIFF (*.tiff)、BMP、PNG、JPEG、GIF 画像	TIFF (Tag Image File Format)、BMP (Bitmap Images)、PNG (Portable Network Graphic)、JPEG、または GIF 型の画像ファイルを作成します。
PDF (*.pdf)	Adobe の Acrobat ビューアまたはブラウザプラグインを備えたコンピュータ上で表示できる PDF ファイルを作成します。
HTML (*.html)	レポートの HTML ファイルを作成します。

レポートをXLSX形式にエクスポートする

「クイックスタート」セクションで作成したレポートをXLSX形式にエクスポートするには、以下の手順に従います。

1. コードビューに次の名前空間を追加します。

Visual Basic

```
Imports C1.Xaml.Document.Export
```

C#

```
using C1.Xaml.Document.Export;
```

2. **XlsFilter**クラスを使用して、レポートを**XLSX**形式にエクスポートするために次のコードを追加します。同じクラスを使用してレポートを**XLS**形式にエクスポートすることもできます。

Visual Basic

```
' ユーザから対象のファイルを要求
Dim fileSavePicker As New FileSavePicker()
fileSavePicker.FileTypeChoices.Add("XLSX files", New String()
{".xlsx"})
Dim storageFile As StorageFile = Await
fileSavePicker.PickSaveFileAsync()
If storageFile Is Nothing Then
    ' ユーザーがエクスポートをキャンセルしました
    Return
End If
' XlsFilterを初期化します。
Dim filter As New XlsFilter()
filter.UseZipForMultipleFiles = True
filter.StorageFile = storageFile

' レポートをフィルタにレンダリングします。
Await report.RenderToFilterAsync(filter)

' エクスポートされたファイルを起動します。
Await Windows.System.Launcher.LaunchFileAsync(storageFile)
```

o C#

```
// ユーザからのターゲットファイルの要求
FileSavePicker fileSavePicker = new FileSavePicker();
fileSavePicker.FileTypeChoices.Add("XLSX files", new string[] { ".xlsx" });
StorageFile storageFile = await fileSavePicker.PickSaveFileAsync();
if (storageFile == null)
    // ユーザーがエクスポートをキャンセルしました
    return;

// XlsFilterを初期化します
XlsFilter filter = new XlsFilter();
filter.UseZipForMultipleFiles = true;
filter.StorageFile = storageFile;
```



```
// レポートをフィルタにレンダリングします
await report.RenderToFilterAsync(filter);

// エクスポートされたファイルを起動します
await Windows.System.Launcher.LaunchFileAsync(storageFile);
```

同様に、レポートを RTF、HTML、PDF 形式にエクスポートすることができます。

レポートの画像ファイル形式へのエクスポート

上のコードを使用してレポートを画像ファイルにエクスポートすることはできますが、複数ページから成るレポートを画像ファイルにエクスポートすると、レポートの最初のページしか一度にエクスポートされません。これは、画像形式フィルタが複数ページを1つのファイルに入れたレポートを直接サポートしていないためです。しかし、レポートの各ページに対応する複数の画像ファイルを生成して1つの ZIP ファイルに格納することは可能です。次のコードは、画像形式フィルタクラスの1つ **JpegFilter** を使用して、複数ページから成るレポートを JPEG 形式にエクスポートし、エクスポートされた画像から成る1つの ZIP ファイルを作成します。

Visual Basic

```
' ユーザから対象のファイルを要求
Dim fileSavePicker As New FileSavePicker()
fileSavePicker.FileTypeChoices.Add("ZIP files", New String() {".zip"})
Dim storageFile As StorageFile = Await
fileSavePicker.PickSaveFileAsync()
If storageFile Is Nothing Then
    ' ユーザーがエクスポートをキャンセルしました
    Return
End If

' JpegFilterを初期化します。
Dim filter As New JpegFilter()
filter.UseZipForMultipleFiles = True
filter.StorageFile = storageFile

' レポートをフィルタにレンダリングします。
Await report.RenderToFilterAsync(filter)

' エクスポートされたファイルを起動します。
Await Windows.System.Launcher.LaunchFileAsync(storageFile)
```

C#

```
// ユーザから対象のファイルを要求
FileSavePicker fileSavePicker = new FileSavePicker();
fileSavePicker.FileTypeChoices.Add("ZIP files", new string[] {
    ".zip"
});
StorageFile storageFile = await fileSavePicker.PickSaveFileAsync();
if (storageFile == null)
// ユーザーがエクスポートをキャンセルしました
    return;

// JpegFilterを初期化します。
JpegFilter filter = new JpegFilter();
filter.UseZipForMultipleFiles = true;
```

FlexReport for UWP

```
filter.StorageFile = storageFile;

// レポートをフィルタにレンダリングします。
await report.RenderToFilterAsync(filter);

// エクスポートされたファイルを起動します。
await Windows.System.Launcher.LaunchFileAsync(storageFile);
```

印刷

FlexReport では、**C1DocumentSource** クラスの **ShowPrintUIAsync** メソッドを使用してレポートを印刷できます。次のコードは、レポートを印刷するための **ShowPrintUIAsync** メソッドを実装します。この例では、「[クイックスタート](#)」で作成したサンプルを使用します。

Visual Basic

```
' 印刷のUIを表示します
report.ShowPrintUIAsync()
```

- C#

```
// 印刷のUIを表示します
report.ShowPrintUIAsync();
```

ShowPrintUIAsync メソッドを使用するほかに、FlexViewer の UI を使用してレポートを印刷することもできます。FlexViewer コントロールでは、[印刷]アイコンを使用して、UI から直接レポートを印刷できるオプションが提供されています。アイコンをクリックすると、プリンタの選択、印刷する方向、サイズ、ページ数の選択など、標準のプリンタ設定が表示されます。

FlexReportDesigner について

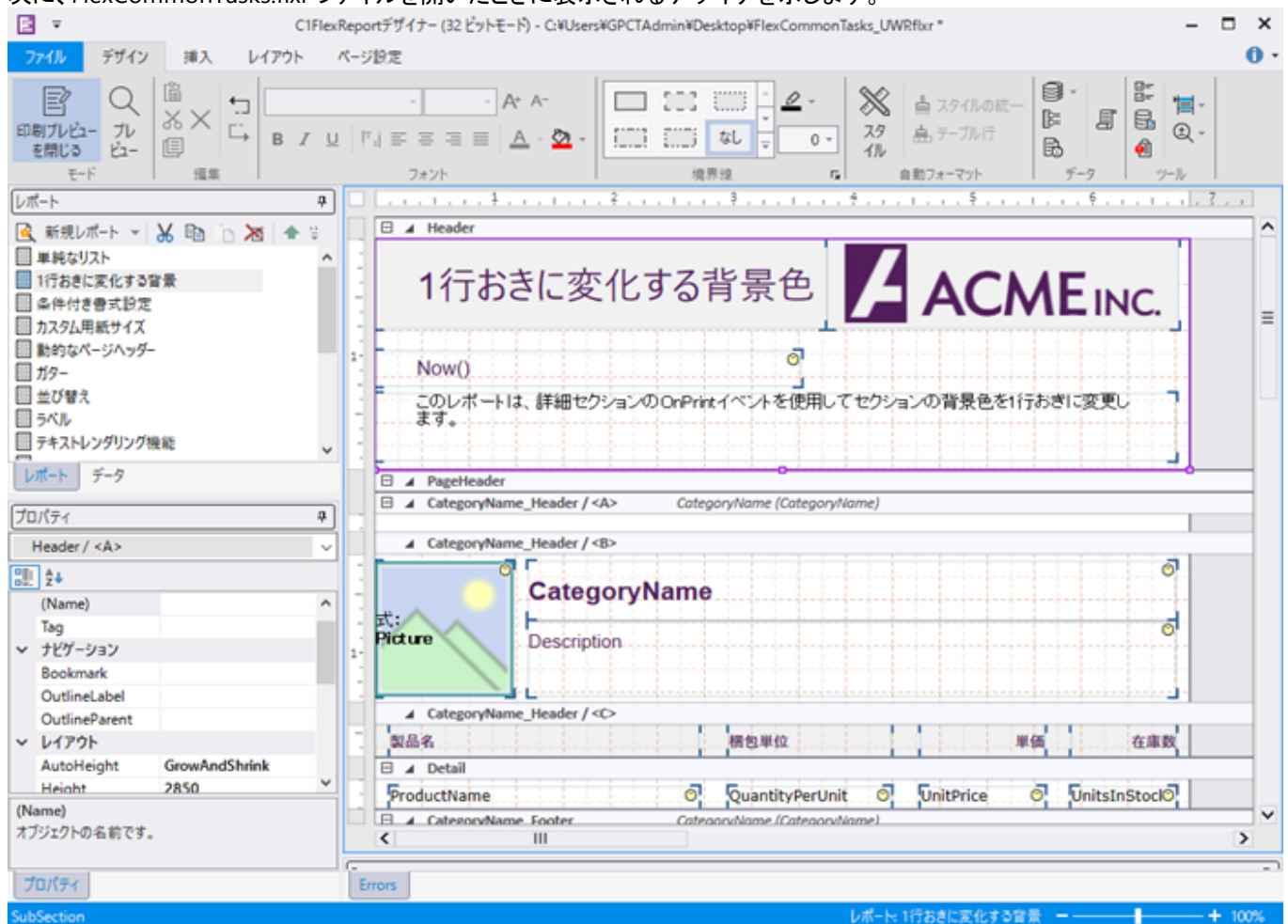
FlexReportDesigner アプリケーションは、**C1FlexReport** レポート定義ファイルを作成および編集するためのツールです。このデザイナーでは、.flxr 拡張子が付いたファイルの作成、編集、ロード、および保存を行うことができます。Microsoft Access ファイル(.mdb)や Crystal Reports (.rpt) から C1Report (.xml) やレポート定義をインポートすることもできます。

デザイナーを実行するには、**C1FlexReportDesigner.exe (64 ビットプラットフォームの場合)** ファイルまたは **C1FlexReportDesigner32.4.exe (32 ビットプラットフォームの場合)** をダブルクリックします。このファイルは、.NET 4.0 の場合、デフォルトで次のパスにあります。

- C:\Program Files (x86)\ComponentOne\Apps\v4.0 (64 ビットプラットフォームの場合)
- C:\Program Files\ComponentOne\Apps\v4.0 (32 ビットプラットフォームの場合)

この場所は、デフォルトのインストールパスを反映します。インストールパスを変更した場合は、異なる場所になります。

次に、FlexCommonTasks.flxr ファイルを開いたときに表示されるデザイナーを示します。



デザイナーウィンドウのコンポーネントの詳細については、「WinForms 用の FlexReportDesigner アプリケーション」を参照してください。

FlexViewer for UWP

FlexViewer は、FlexReportおよびPDFドキュメントの表示に使用できるプレビューコントロールです。**FlexViewer** コントロールは、対話式でユーザーフレンドリな最新のユーザーインターフェース (UI) を備えています。**FlexViewer** コントロールでは、ページナビゲーションオプションを使用してレポートページ間を移動したり、[ページ] テキストボックスにページ番号を入力して特定のページにジャンプすることができます。

Balance Sheet		ACME INC.	
Total Assets	\$6,500,800	Total Liabilities	\$6,500,800
Current Assets		Current Liabilities	
Cash in Bank	\$45,000	Accounts Payable	\$2,585,600
Inventory	\$45,000	Taxes Payable	\$56,263
Prepaid Expenses	\$600	Notes Payable (due within 12 months)	\$216
Other	\$10,000	Current Portion Long-term Debt	\$3,800
Total	\$100,600	Other Current Liabilities (specify)	\$3,000
Fixed Assets		Total	
Machinery & Equipment	\$56,200		\$2,648,879
Furniture & Fixtures	\$32,400	Long-Term Liabilities	
Leasehold Improvements	\$6,300	Bank Loans Payable (greater than 12	\$200
Real Estate / Buildings	\$6,250,000	Less: Short-term Portion	\$560
Other	\$7,000	Notes Payable to Stockholders	\$6,203
Total	\$6,351,900	Other long-term debt (specify)	\$450
		Total	\$7,413

FlexViewer コントロール内からレポートを複数の形式に簡単にエクスポートできます。**FlexViewer** コントロールから[ページ設定]アイコンをクリックして、レポートのページ設定を変更することもできます。さらに、**FlexViewer** コントロールでは、標準的な印刷設定を提供する[印刷]アイコンを使用して、レポートを印刷することもできます。

主な機能

FlexViewer の主要な機能は次のとおりです。

- **最新の使いやすい UI**

FlexViewer は、対話式の使いやすい UI を持ち、FlexReport、SSRS、PDFドキュメントなどのドキュメントタイプをプレビューできます。FlexViewer は、UWP アプリのデザイン標準に従っており、Windows 10 PC、Windows Phone などのあらゆるデバイスによくマッチするデザインになっています。

- **ページナビゲーション**

ページナビゲーションは、FlexViewer コントロールの上部で行うことができます。レポートページ間を移動したり、ページ番号テキストボックスにページ番号を入力して特定のページ番号にジャンプすることができます。

- **ビューモード**





FlexViewer は、レポート/ドキュメントの各種ビューをサポートしています。




- 連続 - ページを連続で表示します。
- 実際のサイズ - 実際のサイズでページを表示します。
- ページ幅 - ページをプレビューウィンドウの幅に合わせます。

- ページ全体 - プレビューウィンドウにページ全体を合わせます。
 - 時計回りに回転 - ビューを時計回りに回転します。
 - 反時計回りに回転 - ビューを反時計回りに回転します。
 - 1 ページ - レポートページを単一ページビューに 1 ページずつ表示します。
 - 見開きページ - レポートページを並べて表示します。
 - 2 ページ - 2 ページビューを表示します。
 - 2 ページ - ページを 4x4 モードで表示します。
 - 8 ページ - ページを 8x8 モードで表示します。
- **パラメータの使用/リセット**
FlexViewer コントロールは、レポートに表示する必要があるデータパラメータを入力することができ、対話性が向上しています。String、Boolean、Date、Integer、Float 型のパラメータをサポートします。
 - **サムネイルと階層の表示**
FlexViewer は、レポートページのサムネイルビューを表示できます。レポートに見出しマップが含まれる場合は、FlexViewer にアウトラインパネルが表示され、ジャンプする先を選択できます。
 - **ページ設定**
FlexViewer コントロールを使用すると、レポートを印刷する前に、要件に応じてページ設定を変更できます。左パネルの[ページ設定]アイコンをクリックして、以下のオプションを設定できます。
 - 向き - 縦/横
 - サイズ
 - マージン - 上マージン、下マージン、左マージン、右マージン
 - **印刷**
FlexViewer では、標準的な印刷設定を提供する[印刷]アイコンを使用して、レポートを印刷することもできます。
 - **エクスポート**
FlexViewer を使用すると、HTML、PDF、RTF、GIF、JPEG、PNG、BMP、TIFF、Open XML Exce、Open XML Word などの形式にレポートやドキュメントをエクスポートできます。エクスポート後に、エクスポートしたドキュメントを自動的に開くように選択することもできます。
 - **RightToLeft**
UI から直接アクセスできる上記の機能のほかに、FlowDirection プロパティを RightToLeft または LeftToRight に設定することで、[FlexViewer ツール]パネルの方向を設定できます。

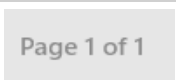
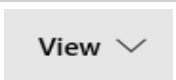

ツールバー

ツールバーは **FlexViewer** コントロールの左側に表示されます。次のコマンドボタンがあります。

コマンドボタン	コマンドボタン名	説明
	ツール	FlexViewer コントロールで使用できるツールのリストを表示します
	サムネイル	表示されたレポートで使用できるすべてのページのサムネイルが表示されます
	アウトライン	レポートページの概要を表示します
	パラメータ	レポートページのパラメータセットを表示します

	エクスポート	ユーザーがさまざまな形式にレポートをエクスポートできます
	ページ設定	レポートページの方向、サイズ、マージンを設定できます
	印刷	レポートを印刷できます

FlexViewer コントロールの上部に他のコマンドもあります。

コマンドボタン	コマンドボタン名	説明
	ページ	レポートの現在のページ番号と合計のページ数を表示します
	ビュー	さまざまなビューでレポートページを表示できます
	検索	レポート内のテキストを検索できます

レポートビューの回転

FlexViewer では、要件に合わせて柔軟に、さまざまな角度にレポートビューを回転させることができます。レポートビューをさまざまな角度に回転させるには、C1FlexViewer クラスの **RotateView** プロパティを設定します。**RotateView** プロパティは、ビューの回転角度を記述する **FlexViewerRotateView** 列挙に含まれる次の値を受け取ります。

- **NoRotation:** ビューに回転が適用されません。
- **Rotation180:** ビューを 180 度回転します。
- **Rotation90Clockwise:** ビューを時計回りに 90 度回転します。
- **Rotation90CounterClockwise:** ビューを反時計回りに 90 度回転します。

実行時のレポートビューの回転

実行時に FlexViewer コントロールの右上にある【ビュー】ドロップダウンリストからビューの回転を選択して、レポートビューを回転させることができます。

プログラムによるレポートビューの回転

レポートビューを回転させるには、**FlexViewerRotateView** 列挙を使用します。次のコードは、**FlexViewerRotateView** 列挙の使用例を示します。

Visual Basic

```
flxViewer.RotateView =  
C1.Xaml.FlexViewer.FlexViewerRotateView.Rotation90Clockwise
```

- **C#**

```
flxViewer.RotateView = C1.Xaml.FlexViewer.FlexViewerRotateView.Rotation90Clockwise;
```

Page 1 of 4 View ▾ 🔍

Tuesday, September 13, 2016

This report uses the **OnPrint** event of the detail section to alternate the **BackColor** property of the section.

Alternating Background

Beverages

Soft drinks, coffees, teas, beers, and ales

Product Name	Quantity Per Unit	Unit Price	In Stock
Chai	10 boxes x 20 bags	\$18.00	39
Chang	24 - 12 oz bottles	\$19.00	17
Sauquatch Ale	24 - 12 oz bottles	\$14.00	111
Steeleye Stout	24 - 12 oz bottles	\$18.00	20
Cole de Bray	12 - 75 cl bottles	\$263.50	17
Chartreuse verte	750 cc per bottle	\$18.00	69
Ipon Coffee	16 - 500 g tins	\$46.00	17
Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52
Outback Lager	24 - 355 ml bottles	\$15.00	15
Rhinrau Koeselbier	24 - 0.5 l bottles	\$7.75	125
Lakkaiköni	500 ml	\$18.00	57
Guaraná Fantástica	12 - 355 ml cans	\$4.50	20

Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

Product Name	Quantity Per Unit	Unit Price	In Stock
Aniseed Syrup	12 - 550 ml bottles	\$10.00	13
Chef Athon's Cajun Seasoning	48 - 6 oz jars	\$22.00	53
Grandma's Boysenberry Spread	12 - 8 oz jars	\$25.00	120
Northwoods Cranberry Sauce	12 - 12 oz jars	\$40.00	6
Garden of Eatin'	24 - 250 ml bottles	\$15.50	39
Guia Malacca	20 - 2 kg bags	\$19.45	27
Sicop d'erable	24 - 500 ml bottles	\$28.50	113

Page 1 of 4

FlexReport と FlexViewer の連結

レポートをレンダリングするには、最初にレポートをロードする必要があります。レポート定義の作成、データソースの定義、レポート定義のロードが終わったら、レポートをプレビューコントロールの **C1FlexViewer** コントロールでレンダリングできます。

C1FlexViewer コントロールでレポートをプレビューするには、次のコードを使用します。

Visual Basic

Try

・ リソースストリームからロードします

```
Using stream As Stream =
```

```
asm.GetManifestResourceStream("BindingApp.Resources.TelephoneBillReport.flxr")
```

```
rpt.Load(stream, "TelephoneBillReport")
```

```
End Using
```

```
Catch ex As Exception
```

```
Dim md As New MessageDialog(String.Format("レポートの読み込みに失敗しました。", rpt.ReportName, ex.Message), "Error")
```

```
Await md.ShowAsync()
```

```
Return
```

```
End Try
```

```
Flxviewer.DocumentSource = rpt
```

C#

FlexReport for UWP

```
try
{
    // リソースストリームからロードします
    using (Stream stream =
asm.GetManifestResourceStream("BindingApp.Resources.TelephoneBillReport.flxr"))

    rpt.Load(stream, "TelephoneBillReport");
}
catch (Exception ex)
{
    MessageDialog md = new MessageDialog(string.Format("レポートの読み込みに失敗しまし
た。", rpt.ReportName, ex.Message), "Error");
    await md.ShowAsync();
    return;
}

Flxviewer.DocumentSource = rpt;
```

Page 1 of 1 View

ACME INC.

Tim Richard
120 Hanover Sq., Victoria Lane, Near Xamine Store
Pittsburg, Pennsylvania

Page: 1 of 1
Date: Jun 22, 2016
Bill Cycle Date: 12/27/2010 - 01/26/2011
Account: 125345871322147014

Monthly Statement

Bill-At-A-Glance	
Previous Balance	\$228.33
Payment - Thank You	\$189.00
Adjustments	\$39.00
Balance	\$0.00
New Charges	\$332.07
Total Amount Due	\$332.00
Amount Due in Full by	10/2/2011

Payments & Adjustments			
S.No.	Description		
1.	Payment posted	5/15/2016	\$189.00
2.	Service Charges		\$332.07
3.	Adjustments		\$39.00
Total Payments & Adjustments			\$332.00

Wireless

Group 1 Usage Summary -
FamilyTalk Nation 850 w/Rollover Minutes - \$9.99