Word for WinForms

2018.04.12 更新

グレープシティ株式会社

Word for WinForms

目次

Word for WinForms の概要

ComponentOne に導入された **Word for WinForms** には、高度な機能を備えた Word ドキュメントを作成するために機能豊富な API が提供されています。**Word** コンポーネントでは、Microsoft Word Open XML 形式の拡張子（*.docx）を持つ Word ドキュメントおよびリッチテキスト形式の拡張子（*.rtf）を持つ RTF ドキュメントの作成、読み取り、書き込みを行うことができます。

Word コンポーネントは **C1.C1Word.C1WordDocument** を使用します。このクラスは、Microsoft Word ドキュメントと RTF ドキュメントの生成に必要な高度なプロパティおよびメソッドをすべて提供します。Word コンポーネントを使用して生成されたドキュメントは、ファイルシステムに保存することも、標準の Microsoft Word ドキュメント形式にエクスポートすることも簡単です。

主要な機能

Word for WinForms の主要な機能は次のとおりです。

- **充実したオブジェクトモデル**
  Word for WinForms は、機能豊富で強力だが簡単にプログラム可能なオブジェクトモデルを提供します。高度なプロパティとメソッドをすべて提供する C1WordDocument クラスだけを使用して、Microsoft Word および RTF ドキュメントの両方を作成できます。

- **高度なライブラリ**
  Word for WinForms は高度なメソッドを使用して、ピクチャ、段落、テキスト、フォント、グラフィック、曲線、および表を Word ドキュメントに追加します。

- **テキストの描画**
  Word for WinForms を使用すると、Word ドキュメント内でさまざまなフォントでテキストを描画したり、フォントプロパティを使用することができます。

- **表の追加**
  Word for WinForms の RTFTable オブジェクトを使用して、表セルにデータを追加できます。

- **ハイパーリンクとブックマーク**
  Word for WinForms は、ドキュメント内を移動するためのブックマークと、別の URL に移動するためのハイパーリンクを提供します。

- **メタファイルの描画と再生**
  C1Word クラスを使用して Word ドキュメント内でメタファイルを描画および再生します。Word for WinForms には、メタファイル画像（.wmf、.emf）を取り込んでドキュメントに追加する独自の DrawMetafile メソッドが組み込まれています。

- **DOCX/RTF ファイルの読み書き**
  Word for WinForms を使用して、DOCX および RTF ファイルを読み書きできます。機能豊富な **C1Word** API を使用して、これらのファイルを変更することもできます。

オブジェクトモデルの概要

C1Word は、機能豊富で強力だが簡単にプログラム可能なオブジェクトモデルを提供します。**C1.C1Word.C1WordDocument** クラスは、Microsoft Word および RTF ドキュメントを作成するための高度なプロパティとメソッドをすべて提供します。

C1WordDocument オブジェクト
Add, AddBookmark, AddBookmarkStart, AddBookmarkEnd, AddLink, AddParagraph, AddPicture, ColumnBreak, DrawArc, DrawRectangle, DrawString, FillPie, Load, Count, Current, Hyperlink, ShapesWord2007Compatible
RTFPageSize オブジェクト
A0, A1, A4, A10, B1, B5, HalfLetter, Legal, Letter
Strings
ResourceManager, UICulture
Strings.Errors
ClosedDocument, UnsupportedRotationAngle

クイックスタート

クイックスタートガイドでは、**Word** コンポーネントについて詳しく説明します。このセクションでは、Visual Studio で Windows フォームアプリケーションを使用して新しいプロジェクトを作成する方法を学びます。ドキュメントを作成して保存するには、C1Word の参照 (dll) をプロジェクトに追加し、ボタンを Windows フォームアプリケーションに追加する必要があります。

手順 1：アプリケーションの設定

この手順では、まず**新しいプロジェクト**を作成し、そこに Visual Studio で **Windows フォームアプリケーション**を作成します。アプリケーションを作成したら、そこに **C1Word** 参照（dll）と 1 つのボタンコントロールを追加します。

Word コンポーネントの使用を開始するには、次の手順を実行します。

1. Visual Studio で**[新しいプロジェクト]**→**[Windows フォームアプリケーション]**を選択します。
2. アプリケーションに **C1Word** 参照（dll）を追加します。
3. **C1WordDocument** をフォームに追加します。
4. コードビューに切り替えて、次の名前空間をコードに追加します。

Visual Basic

```
Imports C1.C1Word
```

C#

```
using C1.C1Word;
```

手順 2：単純なテキストの追加

コードビューを表示したまま、**InitializeComponent()** メソッドの下に次のコードを追加することで、**C1Word** コンポーネントを使用して Word ドキュメントを作成します。

Visual Basic

```
' ドキュメントを作成します
Dim c1Word As New C1WordDocument()
c1Word.Info.Title = "c"

Dim font As New Font("Cambria", 24, FontStyle.Bold)
c1Word.AddParagraph("こんにちは。C1Wordコンポネントの単純なテキスト例。",
        font, Color.MediumPurple)

c1Word.Save("simple.docx")
Process.Start("simple.docx")
```

C#

```
// ドキュメントを作成します
C1WordDocument c1Word = new C1WordDocument();
c1Word.Info.Title = "単純なテキストサンプル";
```

```
Font font = new Font("Cambria", 24, FontStyle.Bold);
c1Word.AddParagraph("こんにちは。C1Wordコンポネントの単純なテキスト例。",
        font, Color.MediumPurple);

c1Word.Save("Simple.docx");
Process.Start("Simple.docx");
```

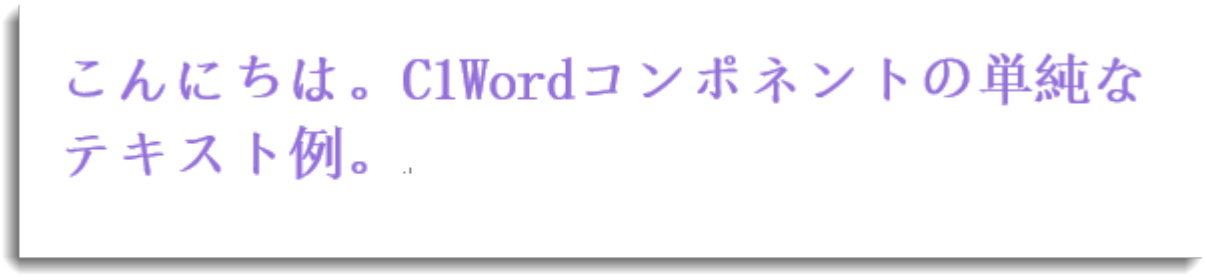
上記のコードは、「**単純なテキストサンプル**」というタイトルの Word ドキュメントを作成し、**AddParagraph** メソッドを使用してテキストを追加します。最後に、ドキュメントは **Simple.docx** という名前で保存されます。

手順 3：アプリケーションの実行

前の手順では、テキストを作成して追加し、Word ドキュメントを保存するコードを追加しました。この手順では、アプリケーションを実行して、作成したドキュメントを表示します。次の手順に従って、アプリケーションを実行します。

[F5]キーを押してアプリケーションを実行します。

ドキュメントが開かれると、次の図のように表示されます。



Word for WinForms の操作

Word コンポーネントは機能豊富な API とオブジェクトモデルを備えており、Word ドキュメントのほかに、Microsoft Word や他のエディタでサポートされている RTF ドキュメントも作成できます。**Word** コンポーネントの詳細な仕組みを次のトピックに示します。

基礎レベルの操作

Word コンポーネントを使用して、ピクチャ、グラフィック、曲線などの単純なイラストを Word ドキュメントに追加できます。 Word コンポーネントを使用すると、このようなイラストを数行のコードで追加できます。次のトピックでは、単純なテキストといくつかの基本的なイラストを追加する方法について説明します。

単純なテキストの追加

C1Word を使用して、Word ドキュメントに単純なテキストを追加できます。単純なテキストを追加するには、**AddParagraph** メソッドを使用して、そこに目的のテキストを書き込む必要があります。Word ドキュメントに表示するテキストに対して、フォントのスタイル、ファミリ、色などのプロパティを設定することもできます。Word ドキュメントへの単純なテキストの追加は、次のコードで実装されます。

次のコードを使用して、**C1WordDocument** クラスのオブジェクトを作成し、**AddParagraph**メソッドを使用してテキストを追加します。

Visual Basic

```
Dim C1Word As New C1WordDocument()
```

C#

```
C1WordDocument C1Word = new C1WordDocument();
```

次のコードを追加して、作成する Word ドキュメントにテキストを追加します。

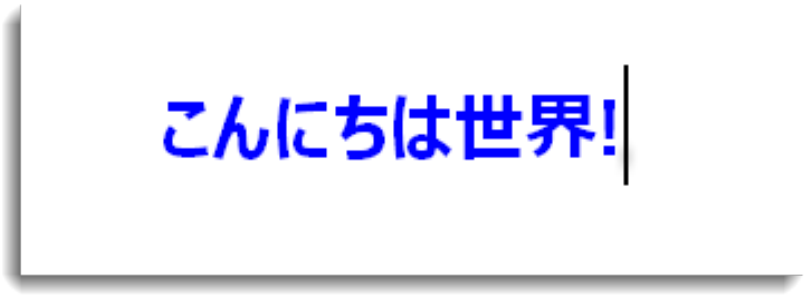
Visual Basic

```
Dim font As New Font("Tahoma", 24, FontStyle.Italic)
C1Word.AddParagraph("こんにちは世界！", font, Color.Blue)
```

C#

```
Font font = new Font("Tahoma", 24, FontStyle.Italic);
C1Word.AddParagraph("こんにちは世界！", font, Color.Blue);
```

ドキュメントは、次の図のように表示されます。



ピクチャの挿入

Word ドキュメントにテキストに加えて画像を挿入して、全体的に見栄えをよくしたい場合があります。それには、**AddPicture** メソッドを使用してピクチャを Word ドキュメントに挿入し、適宜配置を整えます。次のコードは、**AddPicture** メソッドと **RtfHorizontalAlignment** 列挙を使用して、ピクチャの水平方向の配置を設定する例を示します。

Visual Basic

```
Dim img As Image = New Bitmap(dlg.FileName)
C1Word.AddPicture(img, RtfHorizontalAlignment.Left)
```

C#

```
Image img = new Bitmap(dlg.FileName);
C1Word.AddPicture(img, RtfHorizontalAlignment.Left);
```

ドキュメントは、次の図のように表示されます。

グラフィックの描画

グラフィックを追加することで、ドキュメントの見栄えがよくなり、視覚に訴えることができます。ドキュメントに、円弧、ベジェ、楕円、直線、円、多角形、折れ線、四角形などのさまざまなタイプの図形を追加できます。直線、四角形、ベジェなどのグラフィックを追加するには、次のコードを使用します。

Visual Basic

```
' ドキュメントを作成します。
Dim c1Word As New C1WordDocument()
c1Word.Info.Title = "グラフィックスプリミティブ"

Dim sf As New StringFormat()
sf.Alignment = StringAlignment.Center
sf.LineAlignment = StringAlignment.Center
Dim rc = New RectangleF(250, 100, 150, 20)
Dim font As New Font("Arial", 14, FontStyle.Italic)
c1Word.DrawString(c1Word.Info.Title, font, Color.DeepPink, rc, sf)

c1Word.DrawLine(Pens.Green, 200, 190, 400, 190)

rc = New RectangleF(150, 150, 190, 80)
Using pen As New Pen(Brushes.Blue, 5F)
        c1Word.DrawRectangle(pen, rc)
End Using
c1Word.FillRectangle(Color.Gold, rc)
c1Word.ShapeFillOpacity(50)
c1Word.ShapeRotation(25)

rc = New RectangleF(300, 150, 80, 80)
c1Word.DrawEllipse(Pens.Red, rc)
c1Word.FillEllipse(Color.Pink, rc)
c1Word.ShapeFillOpacity(70)

Dim pts As PointF() = New PointF(3) {}
pts(0) = New PointF(200, 200)
pts(1) = New PointF(250, 300)
pts(2) = New PointF(330, 250)
pts(3) = New PointF(340, 140)
c1Word.DrawPolyline(Pens.BlueViolet, pts)

sf = New StringFormat()
sf.Alignment = StringAlignment.Center
sf.LineAlignment = StringAlignment.Far
sf.FormatFlags = sf.FormatFlags Or StringFormatFlags.DirectionVertical
rc = New RectangleF(450, 150, 25, 75)
font = New Font("Verdana", 12, FontStyle.Bold)
c1Word.DrawString("垂直", font, Color.Black, rc, sf)

pts = New PointF(3) {}
pts(0) = New PointF(372, 174)
pts(1) = New PointF(325, 174)
pts(2) = New PointF(325, 281)
pts(3) = New PointF(269, 281)
c1Word.DrawBeziers(Pens.HotPink, pts)

Dim Sdlg As New SaveFileDialog()
Sdlg.FileName = "document"
Sdlg.Filter = "RTF files (*.rtf)|*.rtf|DOCX (*.docx)|*.docx"
```

```
Sdlg.ShowDialog()

c1Word.Save(Sdlg.FileName)
MessageBox.Show("Wordドキュメントは正常に保存されました。")
```

C#

```
// ドキュメントを作成します。
C1WordDocument c1Word = new C1WordDocument();
c1Word.Info.Title = "グラフィックスプリミティブ";

StringFormat sf = new StringFormat();
sf.Alignment = StringAlignment.Center;
sf.LineAlignment = StringAlignment.Center;
var rc = new RectangleF(250, 100, 150, 20);
Font font = new Font("Arial", 14, FontStyle.Italic);
c1Word.DrawString(c1Word.Info.Title, font, Color.DeepPink, rc, sf);

c1Word.DrawLine(Pens.Green, 200, 190, 400, 190);

rc = new RectangleF(150, 150, 190, 80);
using(Pen pen = new Pen(Brushes.Blue, 5.0 f)) {
  c1Word.DrawRectangle(pen, rc);
}
c1Word.FillRectangle(Color.Gold, rc);
c1Word.ShapeFillOpacity(50);
c1Word.ShapeRotation(25);

rc = new RectangleF(300, 150, 80, 80);
c1Word.DrawEllipse(Pens.Red, rc);
c1Word.FillEllipse(Color.Pink, rc);
c1Word.ShapeFillOpacity(70);

PointF[] pts = new PointF[4];
pts[0] = new PointF(200, 200);
pts[1] = new PointF(250, 300);
pts[2] = new PointF(330, 250);
pts[3] = new PointF(340, 140);
c1Word.DrawPolyline(Pens.BlueViolet, pts);

sf = new StringFormat();
sf.Alignment = StringAlignment.Center;
sf.LineAlignment = StringAlignment.Far;
sf.FormatFlags |= StringFormatFlags.DirectionVertical;
rc = new RectangleF(450, 150, 25, 75);
font = new Font("Verdana", 12, FontStyle.Bold);
c1Word.DrawString("垂直", font, Color.Black, rc, sf);

pts = new PointF[4];
pts[0] = new PointF(372, 174);
pts[1] = new PointF(325, 174);
pts[2] = new PointF(325, 281);
pts[3] = new PointF(269, 281);
```

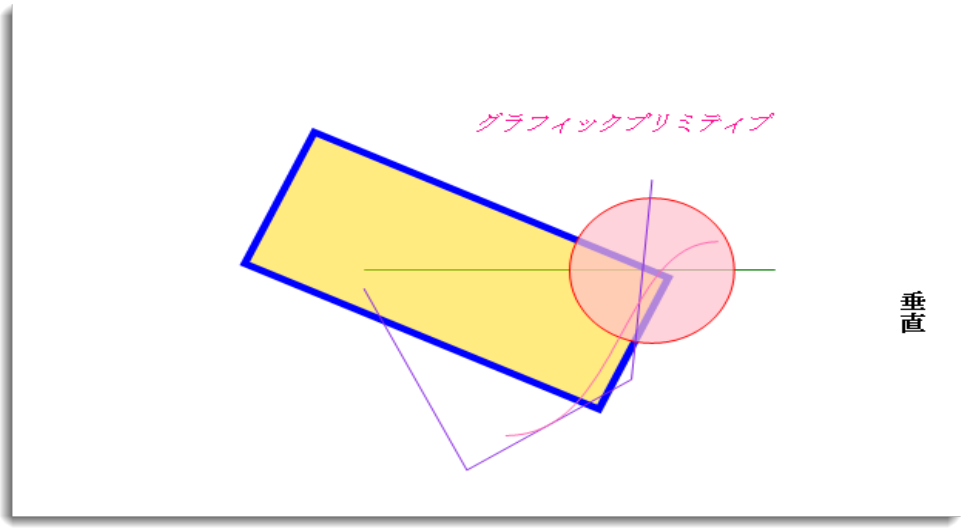
```
c1Word.DrawBeziers(Pens.HotPink, pts);

SaveFileDialog Sdlg = new SaveFileDialog();
Sdlg.FileName = "document";
Sdlg.Filter = "RTF files (*.rtf)|*.rtf|DOCX (*.docx)|*.docx";
Sdlg.ShowDialog();

c1Word.Save(Sdlg.FileName);
MessageBox.Show("Wordドキュメントが正常に保存されました。");
```

上記のコードは、**DrawLine**、**DrawRectangle**、**DrawEllipse**、**DrawPolyline**、および **DrawBeziers** メソッドを使用して、直線、四角形、ベジェ、楕円などのさまざまなタイプのグラフィックを描画します。

ドキュメントは、次の図のように表示されます。



曲線の追加

Word コンポーネントを使用して、Word ドキュメントに曲線を追加できます。**DrawArc** メソッドを使用すると、ドキュメントに曲線を追加できます。**DrawArc** メソッドは、次のコードで実装されます。

Visual Basic

```
rc = New RectangleF(120, 100, 150, 80)
C1Word.DrawArc(Pens.Red, rc, 0, 90)
```

C#

```
rc = new RectangleF(120, 100, 150, 80);
C1Word.DrawArc(Pens.Red, rc, 0, 90);
```

上記のコードで Word ドキュメントに赤い曲線が追加されます。

ドキュメントは、次の図のように表示されます。

上級レベルの操作

通常、Word ドキュメントにはテキストが入りますが、これに画像、イラスト、表、メタファイルなどを追加することで、見栄えをよくし、かつわかりやすくすることができます。次に示すトピックでは、**Word** コンポーネントを使用して、これらの高度な機能を Word ドキュメントに追加する方法を説明します。

表の挿入

表は、Word ドキュメントでデータをいくつかの行と列に整えて表示するために使用されます。Word ドキュメントで、データをいくつかの行と列に整えて表示するために表を使用することはごく一般的なことです。**Word** コンポーネントを使用すると Word ドキュメントに表を追加できます。それには **RtfTable** クラスを使用して表を作成し、**RtfParagraph** クラスを使用して表にコンテンツを挿入します。

次のコードは、Word ドキュメントに表を追加します。

Visual Basic

```
Dim rows As Integer = 4
Dim cols As Integer = 2
Dim table As New RtfTable(rows, cols)
C1Word.Add(table)
For row As Integer = 0 To rows - 1
        For col As Integer = 0 To cols - 1
                Dim paragraph As New RtfParagraph()
                paragraph.Content.Add(New RtfString(String.Format("table
cell {0}:{1}.", row, col)))
                table.Rows(row).Cells(col).Content.Add(paragraph)
        Next
Next
```

C#

```
int rows = 4;
int cols = 2;
RtfTable table = new RtfTable(rows, cols);
```

```
C1Word.Add(table);
for (int row = 0; row < rows; row++)
{
        for (int col = 0; col < cols; col++)
        {
        RtfParagraph paragraph = new RtfParagraph();
        paragraph.Content.Add(new RtfString(string.Format("table cell
{0}:{1}.", row, col)));
        table.Rows[row].Cells[col].Content.Add(paragraph);
        }
}
```

複雑なテキストの追加

テキスト、画像、表、およびグラフィックをドキュメントに追加すると、ドキュメントがさらにインタラクティブになります。**Word** コンポーネントを使用すると、タイトル、画像、表、およびグラフィックを Word ドキュメントに簡単に追加できます。ここまでは、ドキュメントに単純なテキストを追加してきました。しかし、実際のドキュメントは、テキスト、画像、ピクチャ、グラフィックなどの集合で、それらが実際の複雑なテキストを構成しています。次のコードは、Word ドキュメントにタイトル、画像、表、およびグラフィックを追加する、オールインワンのコードです。

Visual Basic

```
' タイトルを追加します
C1Word.AddParagraph(C1Word.Info.Title, New Font("Tahoma", 24,
FontStyle.Italic), Color.BlueViolet)

' 表を追加します
C1Word.AddParagraph("picture:", New Font("Courier New", 9,
FontStyle.Regular), Color.Black)
Dim img As New Bitmap(GetManifestResource("picture.jpg"))
C1Word.AddPicture(img, RtfHorizontalAlignment.Center)

' 表を追加します
C1Word.LineBreak()
Dim rows As Integer = 7
Dim cols As Integer = 2
Dim table As New RtfTable(rows, cols)
C1Word.Add(table)
For row As Integer = 0 To rows - 1
        For col As Integer = 0 To cols - 1
                Dim paragraph As New RtfParagraph()
                paragraph.Content.Add(New RtfString(String.Format("table
cell {0}:{1}.", row, col)))
                table.Rows(row).Cells(col).Content.Add(paragraph)
        Next
Next

' グラフィックを追加します
C1Word.LineBreak()
C1Word.DrawLine(Pens.Green, 200, 90, 400, 90)

Dim rc = New RectangleF(150, 170, 90, 40)
```

```
Using pen As New Pen(Brushes.Blue, 5F)
        C1Word.DrawRectangle(pen, rc)
End Using
C1Word.FillRectangle(Color.Gold, rc)
C1Word.ShapeFillOpacity(50)
C1Word.ShapeRotation(25)

rc = New RectangleF(300, 120, 80, 80)
C1Word.DrawEllipse(Pens.Red, rc)
C1Word.FillEllipse(Color.Pink, rc)
C1Word.ShapeFillOpacity(70)
```

C#

```
// タイトルを追加します
C1Word.AddParagraph(C1Word.Info.Title, new Font("Tahoma", 24,
FontStyle.Italic), Color.BlueViolet);

// 画像を追加します
C1Word.AddParagraph("picture:", new Font("Courier New", 9,
FontStyle.Regular), Color.Black);
Bitmap img = new Bitmap(GetManifestResource("picture.jpg"));
C1Word.AddPicture(img, RtfHorizontalAlignment.Center);

// 表を追加します
C1Word.LineBreak();
int rows = 7;
int cols = 2;
RtfTable table = new RtfTable(rows, cols);
C1Word.Add(table);
for (int row = 0; row < rows; row++)
{
    for (int col = 0; col < cols; col++)
    {
        RtfParagraph paragraph = new RtfParagraph();
        paragraph.Content.Add(new RtfString(string.Format("table cell
{0}:{1}.", row, col)));
        table.Rows[row].Cells[col].Content.Add(paragraph);
    }
}

// グラフィックを追加します
C1Word.LineBreak();
C1Word.DrawLine(Pens.Green, 200, 90, 400, 90);

var rc = new RectangleF(150, 170, 90, 40);
using (Pen pen = new Pen(Brushes.Blue, 5.0f))
{
    C1Word.DrawRectangle(pen, rc);
}
C1Word.FillRectangle(Color.Gold, rc);
C1Word.ShapeFillOpacity(50);
```

```
C1Word.ShapeRotation(25);

rc = new RectangleF(300, 120, 80, 80);
C1Word.DrawEllipse(Pens.Red, rc);
C1Word.FillEllipse(Color.Pink, rc);
C1Word.ShapeFillOpacity(70);
```

フォントの追加

適切なフォントを使用するとドキュメントが引き立ちます。さまざまなフォントスタイルを使用してドキュメントを作成し、それぞれのテキストを異なるフォントスタイルで明確に表示できます。Word ドキュメントでさまざまなフォントを使用するには、次のコードを使用します。

Visual Basic

```
' テキストを多くのフォントで描画します
Dim font As New Font("Tahoma", 9)
Dim ifc As New InstalledFontCollection()
For Each ff As FontFamily In ifc.Families
        ' フォントを作成します
        Dim sample As Font = Nothing
        For Each fs As FontStyle In [Enum].GetValues(GetType(FontStyle))
                If ff.IsStyleAvailable(fs) Then
                        sample = New Font(ff.Name, 9, fs)
                        Exit For
                End If
        Next
        If sample Is Nothing Then
                Continue For
        End If

        ' フォントを表示します
        C1Word.AddParagraph(ff.Name, font, Color.Black)
        C1Word.AddParagraph("The quick brown fox jumped over the lazy
dog. 1234567890!", sample, Color.Black)
        sample.Dispose()
Next
```

C#

```
// テキストを多くのフォントで描画します
Font font = new Font("Tahoma", 9);
InstalledFontCollection ifc = new InstalledFontCollection();
foreach (FontFamily ff in ifc.Families)
{
        // フォントを作成します
        Font sample = null;
        foreach (FontStyle fs in Enum.GetValues(typeof(FontStyle)))
        {
                if (ff.IsStyleAvailable(fs))
```

```
                {
                        sample = new Font(ff.Name, 9, fs);
                        break;
                }
        }
        if (sample == null) continue;

        // フォントを表示します
        C1Word.AddParagraph(ff.Name, font, Color.Black);
        C1Word.AddParagraph("The quick brown fox jumped over the lazy
dog. 1234567890!", sample, Color.Black);
        sample.Dispose();
}
```

メタファイルの再生

メタファイルは、さまざまなデータを格納できるファイルです。.emf 画像や .wmf 画像を Word ドキュメントに追加できます。次のコードは、**DrawMetafile** メソッドを使用してメタファイル画像をドキュメント内に描画する方法を示します。

Visual Basic

```
Dim img As Image = Metafile.FromFile(dlg.FileName)
C1Word.DrawMetafile(DirectCast(img, Metafile))
```

C#

```
Image img = Metafile.FromFile(dlg.FileName);
C1Word.DrawMetafile((Metafile)img);
```

RTF ファイルの読み取りと書き込み

Word コンポーネントでは、リッチテキスト形式の拡張子(*.rtf)を持つ RTF ドキュメントの読み取りと書き込みを行うことができます。**Load** メソッドを使用してドキュメントを読み取り、**Save** メソッドを使用してドキュメントを書き込みます。次のコードのように、**Load** メソッドを使用してドキュメントを読み取り、**Save** メソッドを使用してドキュメントを書き込むことができます。

Visual Basic

```
' Word/RTF ドキュメントをロードします
Dim C1Word As New C1WordDocument()
C1Word.Load(dlg.FileName)

' RTF ドキュメントを保存します
C1Word.Save("sample.rtf")
```

C#

```
// Word/RTF ドキュメントをロードします
C1WordDocument C1Word = new C1WordDocument();
```

```
C1Word.Load(dlg.FileName);

// RTF ドキュメントを保存します
C1Word.Save("sample.rtf");
```

Docx ファイルの読み取りと書き込み

Word コンポーネントでは、Microsoft Word Open XML 形式の拡張子（*.docx）を持つ Word ドキュメントの読み取りと書き込みを行うことができます。次のコードのように、**Load** メソッドを使用してドキュメントを読み取り、**Save** メソッドを使用してドキュメントを書き込むことができます。

Visual Basic

```
' Word/RTF ドキュメントをロードしますDim C1Word As New C1WordDocument()
C1Word.Load(dlg.FileName)

' Word(docx)ドキュメントを保存します
C1Word.Save("sample.docx")
```

C#

```
// Word/RTF ドキュメントをロードします
C1WordDocument C1Word = new C1WordDocument();
C1Word.Load(dlg.FileName);

// Word(docx)ドキュメントを保存します
C1Word.Save("sample.docx");
```

リファレンス

以下のトピックには、Word for WinForms の API リファレンスが含まれます。