XDoc.PDF Developer Guide – Form Module

Table of Contents

Add form fields to an exist PDF file

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

//  add a radio button field with default setting
AFRadioButton field1 = new AFRadioButton("AF_RadioButton_01");
field1.PageIndex = 0;
field1.Position = new PointF(100F, 100F);
fields.Add(field1);

//  add a checkbox field with default setting
AFCheckBox field2 = new AFCheckBox("AF_CheckBox_01");
field2.PageIndex = 0;
field2.Position = new PointF(300F, 100F);
fields.Add(field2);

//  add a checkbox field with default setting
AFTextBox field3 = new AFTextBox("AF_TextBox_01");
field3.PageIndex = 0;
field3.Position = new PointF(100F, 300F);
fields.Add(field3);

//  add a list box field with default setting
AFListBox field4 = new AFListBox("AF_ListBox_01");
field4.PageIndex = 0;
field4.Position = new PointF(100F, 500F);
field4.Items = new String[4] { "Item 1", "Item 2", "Item 3", "Item 4" };
fields.Add(field4);

//  add a combo box field with default setting
AFComboBox field5 = new AFComboBox("AF_ComboBox_01");
field5.PageIndex = 0;
field5.Position = new PointF(300F, 500F);
field5.Items = new String[4] { "Item 1", "Item 2", "Item 3", "Item 4" };
fields.Add(field5);

//  add a button field with default setting
AFButton field6 = new AFButton("AF_Button_01");
field6.PageIndex = 0;
field6.Position = new PointF(100F, 700F);
fields.Add(field6);

//  add fields to the input file
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output.pdf"
```

```
Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' add a radio button field with default setting
Dim field1 As AFRadioButton = New AFRadioButton("AF_RadioButton_01")
field1.PageIndex = 0
field1.Position = New PointF(100.0F, 100.0F)
fields.Add(field1)

' add a checkbox field with default setting
Dim field2 As AFCheckBox = New AFCheckBox("AF_CheckBox_01")
field2.PageIndex = 0
field2.Position = New PointF(300.0F, 100.0F)
fields.Add(field2)

' add a checkbox field with default setting
Dim field3 As AFTextBox = New AFTextBox("AF_TextBox_01")
field3.PageIndex = 0
field3.Position = New PointF(100.0F, 300.0F)
fields.Add(field3)

' add a list box field with default setting
Dim field4 As AFListBox = New AFListBox("AF_ListBox_01")
field4.PageIndex = 0
field4.Position = New PointF(100.0F, 500.0F)
field4.Items = New String() {"Item 1", "Item 2", "Item 3", "Item 4"}
fields.Add(field4)

' add a combo box field with default setting
Dim field5 As AFComboBox = New AFComboBox("AF_ComboBox_01")
field5.PageIndex = 0
field5.Position = New PointF(300.0F, 500.0F)
field5.Items = New String() {"Item 1", "Item 2", "Item 3", "Item 4"}
fields.Add(field5)

' add a button field with default setting
Dim field6 As AFButton = New AFButton("AF_Button_01")
field6.PageIndex = 0
field6.Position = New PointF(100.0F, 700.0F)
fields.Add(field6)

' add fields to the input file
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Retrieve all form fields from a PDF file

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";

List<BaseFormField> fields = PDFFormHandler.GetFormFields(inputFilePath);
Console.WriteLine("Number of Fields: " + fields.Count);
if (fields.Count > 0)
{
   foreach (BaseFormField field in fields)
   {
      Console.WriteLine("Field");
      Console.WriteLine("  Name:     " + field.Name);
      Console.WriteLine("  Visible:  " + field.IsVisible);
      Console.WriteLine("  Page:     " + field.PageIndex);
      Console.WriteLine("  Position: " + field.Position.ToString());
      Console.WriteLine("  Size: " + field.Size.ToString());
   }
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"

Dim fields As List(Of BaseFormField) = PDFFormHandler.GetFormFields(inputFilePath)
Console.WriteLine("Number of Fields: " + fields.Count)
If (fields.Count > 0) Then
   For Each field As BaseFormField In fields
      Console.WriteLine("Field")
      Console.WriteLine("  Name:     " + field.Name)
      Console.WriteLine("  Visible:  " + field.IsVisible)
      Console.WriteLine("  Page:     " + field.PageIndex)
      Console.WriteLine("  Position: " + field.Position.ToString())
      Console.WriteLine("  Size:     " + field.Size.ToString())
   Next
End If
```

Select a field in a page by position

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";

//  select a field at position [110, 310] in page 1 (page index 0)
int pageIndex = 0;
PointF pos = new PointF(110, 310);
//  get the form field object
BaseFormField field = PDFFormHandler.GetFormField(inputFilePath, pageIndex, pos);
if (field != null)
{
   Console.WriteLine("Field " + field.Name + " in page " + field.PageIndex + " at " +
field.Position.ToString());
}
else
{
   Console.WriteLine("Field " + field.Name + " does not exist");
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"

' select a field at position [110, 310] in page 1 (page index 0)
Dim pageIndex As Integer = 0
Dim pos As PointF = New PointF(110, 310)
' get the form field object
Dim field As BaseFormField = PDFFormHandler.GetFormField(inputFilePath, pageIndex, pos)
If Not IsNothing(field) Then
   Console.WriteLine("Field " + field.Name + " in page " + field.PageIndex + " at " +
field.Position.ToString())
Else
   Console.WriteLine("Field " + field.Name + " does not exist")
End If
```

Select a field in a document by the name

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";

//  select a field with name "AF_RadioButton_01"
String fieldName = @"AF_RadioButton_01";
//  get the form field object
BaseFormField field = PDFFormHandler.SelectFormField(inputFilePath, fieldName);
if (field != null)
{
   Console.WriteLine("Field " + field.Name + " in page " + field.PageIndex + " at " +
field.Position.ToString());
}
else
{
   Console.WriteLine("Field " + field.Name + " does not exist");
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"

' select a field with name "AF_RadioButton_01"
Dim fieldName As String = "AF_RadioButton_01"
' get the form field object
Dim field As BaseFormField = PDFFormHandler.SelectFormField(inputFilePath, fieldName)
If Not IsNothing(field) Then
   Console.WriteLine("Field " + field.Name + " in page " + field.PageIndex + " at " +
field.Position.ToString())
Else
   Console.WriteLine("Field " + field.Name + " does not exist")
End If
```

Delete a form field in the page

By position:

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";
String outputFilePath = Program.RootPath + "\\" + "output.pdf";

//  delete a field at position [110, 310] in page 1 (page index 0)
int pageIndex = 0;
PointF pos = new PointF(110, 310);
//  remove the field and output the new document
int errCode = PDFFormHandler.RemoveFormField(inputFilePath, pageIndex, pos, outputFilePath);
if (errCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "output.pdf"

' delete a field at position [110, 310] in page 1 (page index 0)
Dim pageIndex As Integer = 0
Dim pos As PointF = New PointF(110, 310)
' remove the field and output the new document
Dim errCode As Integer = PDFFormHandler.RemoveFormField(inputFilePath, pageIndex, pos,
outputFilePath)
If errCode = 0 Then
   Console.WriteLine("Success")
Else
   Console.WriteLine("Failed")
End If
```

By field name:

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";
String outputFilePath = Program.RootPath + "\\" + "output.pdf";

//  remove a field by name and output the new document
String fieldName = "AF_RadioButton_01";
int errCode = PDFFormHandler.RemoveFormField(inputFilePath, fieldName, outputFilePath);
if (errCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "output.pdf"

' remove a field by name and output the new document
Dim fieldName As String = "AF_RadioButton_01"
Dim errCode As Integer = PDFFormHandler.RemoveFormField(inputFilePath, fieldName, outputFilePath)
If errCode = 0 Then
   Console.WriteLine("Success")
Else
   Console.WriteLine("Failed")
End If
```

Add a Radio Button Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output0.pdf";

List<BaseFormField> fields = new List<BaseFormField>();
//  create a Radio Button Field object
AFRadioButton tbField = new AFRadioButton("AF_RadioButton_01");
//  in first page (page index 0)
tbField.PageIndex = 0;
//  position of top left corner of the field: [100 pixels, 250 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(100, 250);
//  size of the field: width 400 pixels, height 100 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(400, 100);
//  set field visible
tbField.IsVisible = true;
//  set the label of the Radio Button field
tbField.Text = "Radio Button";
tbField.SetTextFont(PSType1Font.Helvetica_Oblique, 12);
tbField.TextColor = System.Drawing.Color.Black;
//  set the alignment of the radio button and label text in the field region
tbField.HorizontalAlignment = HorizontalAlignment.Left;
tbField.VerticalAlignment = VerticalAlignment.Center;

fields.Add(tbField);

//  add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output0.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a Radio Button Field object
Dim tbField1 As AFRadioButton = New AFRadioButton("AF_RadioButton_01")
' in first page (page index 0)
tbField1.PageIndex = 0
' position of top left corner of the field: [100 pixels, 250 pixels] (in 96 dpi)
tbField1.Position = New System.Drawing.PointF(100, 250)
' size of the field: width 400 pixels, height 100 pixels (in 96 dpi)
tbField1.Size = New System.Drawing.SizeF(400, 100)
' set field visible
tbField1.IsVisible = True
' set the label of the Radio Button field
tbField1.Text = "Radio Button"
tbField1.SetTextFont(PSType1Font.Helvetica_Oblique, 12)
tbField1.TextColor = System.Drawing.Color.Black
' set the alignment of the radio button and label text in the field region
tbField1.HorizontalAlignment = HorizontalAlignment.Left
```

```
tbField1.VerticalAlignment = VerticalAlignment.Center

fields.Add(tbField1)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Add a Check Box Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output1.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

//  create a Check Box Field object
AFCheckBox tbField = new AFCheckBox("AF_CheckBox_01");
//  in first page (page index 0)
tbField.PageIndex = 0;
//  position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(150, 300);
//  size of the field: width 300 pixels, height 80 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(300, 80);
//  set field visible
tbField.IsVisible = true;
//  set the label of the Check Box field
tbField.Text = "Check Box";
tbField.SetTextFont(PSType1Font.Helvetica, 16);
tbField.TextColor = System.Drawing.Color.Black;
//  set the alignment of the checkbox and label text in the field region
tbField.HorizontalAlignment = HorizontalAlignment.Center;
tbField.VerticalAlignment = VerticalAlignment.Center;

//  set the initial state of the CheckBox field to ON
tbField.IsChecked = true;

fields.Add(tbField);

//  add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output1.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a Check Box Field object
Dim tbField As AFCheckBox = New AFCheckBox("AF_CheckBox_01")
' in first page (page index 0)
tbField.PageIndex = 0
' position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = New System.Drawing.PointF(150, 300)
' size of the field: width 300 pixels, height 80 pixels (in 96 dpi)
tbField.Size = New System.Drawing.SizeF(300, 80)
' set field visible
tbField.IsVisible = True
' set the label of the Check Box field
tbField.Text = "Check Box"
tbField.SetTextFont(PSType1Font.Helvetica, 16)
```

```
tbField.TextColor = System.Drawing.Color.Black
' set the alignment of the checkbox and label text in the field region
tbField.HorizontalAlignment = HorizontalAlignment.Center
tbField.VerticalAlignment = VerticalAlignment.Center

' set the initial state of the CheckBox field to ON
tbField.IsChecked = True

fields.Add(tbField)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Add a Text Box Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output2.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

//  create a Text Box Field object
AFTextBox tbField = new AFTextBox("AF_TextBox_01");
//  in first page (page index 0)
tbField.PageIndex = 0;
//  position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(150, 300);
//  size of the field: width 400 pixels, height 300 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(400, 300);
//  set field visible
tbField.IsVisible = true;
//  background of the text box: Light Gray
tbField.BackgroundColor = System.Drawing.Color.LightGray;
//  initial content and font and font size of the content in the text box
tbField.Text = "";
tbField.SetTextFont(PSType1Font.Helvetica, 12);
tbField.TextColor = System.Drawing.Color.Black;
//  multi-line flag: true
tbField.IsMultiLine = true;
//  readonly flag: false
tbField.IsReadOnly = false;
//  set border of the field: 3 pixels in width, lightblue
tbField.BorderWidth = 3;
tbField.BorderColor = System.Drawing.Color.LightBlue;

fields.Add(tbField);

//  add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output2.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a Text Box Field object
Dim tbField As AFTextBox = New AFTextBox("AF_TextBox_01")
' in first page (page index 0)
tbField.PageIndex = 0
' position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = New System.Drawing.PointF(150, 300)
' size of the field: width 400 pixels, height 300 pixels (in 96 dpi)
tbField.Size = New System.Drawing.SizeF(400, 300)
' set field visible
tbField.IsVisible = True
```

```
' background of the text box: Light Gray
tbField.BackgroundColor = System.Drawing.Color.LightGray
' initial content and font and font size of the content in the text box
tbField.Text = ""
tbField.SetTextFont(PSType1Font.Helvetica, 12)
tbField.TextColor = System.Drawing.Color.Black
' multi-line flag: true
tbField.IsMultiLine = True
' readonly flag: false
tbField.IsReadOnly = False
' set border of the field: 3 pixels in width, lightblue
tbField.BorderWidth = 3
tbField.BorderColor = System.Drawing.Color.LightBlue

fields.Add(tbField)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Add a List Box Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output3.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

// create a List Box Field object
AFListBox tbField = new AFListBox("AF_ListBox_01");
// in first page (page index 0)
tbField.PageIndex = 0;
// position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(150, 300);
// size of the field: width 150 pixels, height 60 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(150, 60);
// set field visible
tbField.IsVisible = true;
// set option items in the list box
tbField.Items = new String[] { "Item 1", "Item 2", "Item 3", "Item 4" };
// set properties of the item text
tbField.SetTextFont(PSType1Font.Helvetica, 12);
tbField.TextColor = System.Drawing.Color.Black;
// set border of the field: 2 pixels in width, black
tbField.BorderWidth = 2;
tbField.BorderColor = System.Drawing.Color.Black;
// multi-selection flag: false
tbField.IsMultiSelect = false;

fields.Add(tbField);

// add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output3.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a List Box Field object
Dim tbField As AFListBox = New AFListBox("AF_ListBox_01")
' in first page (page index 0)
tbField.PageIndex = 0
' position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = New System.Drawing.PointF(150, 300)
' size of the field: width 150 pixels, height 60 pixels (in 96 dpi)
tbField.Size = New System.Drawing.SizeF(150, 60)
' set field visible
tbField.IsVisible = True
' set option items in the list box
tbField.Items = New String() {"Item 1", "Item 2", "Item 3", "Item 4"}
' set properties of the item text
```

```
tbField.SetTextFont(PSType1Font.Helvetica, 12)
tbField.TextColor = System.Drawing.Color.Black
' set border of the field: 2 pixels in width, black
tbField.BorderWidth = 2
tbField.BorderColor = System.Drawing.Color.Black
' multi-selection flag: false
tbField.IsMultiSelect = False

fields.Add(tbField)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Add a Combo Box Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output4.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

//  create a Combo Box Field object
AFComboBox tbField = new AFComboBox("AF_ComboBox_01");
//  in first page (page index 0)
tbField.PageIndex = 0;
//  position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(150, 300);
//  size of the field: width 150 pixels, height 30 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(150, 30);
//  set field visible
tbField.IsVisible = true;
//  set field background color: lightgray
tbField.BackgroundColor = System.Drawing.Color.LightGray;
//  set option items in the combo box
tbField.Items = new String[] { "Item 1", "Item 2", "Item 3", "Item 4" };
//  set properties of the item text
tbField.SetTextFont(PSType1Font.Helvetica, 12);
tbField.TextColor = System.Drawing.Color.Black;
//  set border of the field: 2 pixels in width, black
tbField.BorderWidth = 2;
tbField.BorderColor = System.Drawing.Color.Black;

//  set inital selected index: 2 (the 3rd item)
tbField.SelectedIndex = 2;

fields.Add(tbField);

//  add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output4.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a Combo Box Field object
Dim tbField As AFComboBox = New AFComboBox("AF_ComboBox_01")
' in first page (page index 0)
tbField.PageIndex = 0
' position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = New System.Drawing.PointF(150, 300)
' size of the field: width 150 pixels, height 30 pixels (in 96 dpi)
tbField.Size = New System.Drawing.SizeF(150, 30)
' set field visible
```

```
tbField.IsVisible = True
' set field background color: lightgray
tbField.BackgroundColor = System.Drawing.Color.LightGray
' set option items in the combo box
tbField.Items = New String() {"Item 1", "Item 2", "Item 3", "Item 4"}
' set properties of the item text
tbField.SetTextFont(PSType1Font.Helvetica, 12)
tbField.TextColor = System.Drawing.Color.Black
' set border of the field: 2 pixels in width, black
tbField.BorderWidth = 2
tbField.BorderColor = System.Drawing.Color.Black

' set inital selected index: 2 (the 3rd item)
tbField.SelectedIndex = 2

fields.Add(tbField)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Add a Button Field

C#

```
String inputFilePath = Program.RootPath + "\\" + "empty.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output5.pdf";

List<BaseFormField> fields = new List<BaseFormField>();

//  create a Push Button Field object
AFButton tbField = new AFButton("AF_Button_01");
//  in first page (page index 0)
tbField.PageIndex = 0;
//  position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = new System.Drawing.PointF(150, 300);
//  size of the field: width 120 pixels, height 30 pixels (in 96 dpi)
tbField.Size = new System.Drawing.SizeF(120, 30);
//  set field visible
tbField.IsVisible = true;
//  set field background color: gray
tbField.BackgroundColor = System.Drawing.Color.Gray;
//  set label in the push button
tbField.Text = "CLICK";
tbField.SetTextFont(PSType1Font.Helvetica, 16);
tbField.TextColor = System.Drawing.Color.Black;
//  set border of the field: 1 pixel in width, drakgray
tbField.BorderColor = System.Drawing.Color.DarkGray;
tbField.BorderWidth = 1;

fields.Add(tbField);

//  add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "empty.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "Output5.pdf"

Dim fields As List(Of BaseFormField) = New List(Of BaseFormField)

' create a Push Button Field object
Dim tbField As AFButton = New AFButton("AF_Button_01")
' in first page (page index 0)
tbField.PageIndex = 0
' position of top left corner of the field: [150 pixels, 300 pixels] (in 96 dpi)
tbField.Position = New System.Drawing.PointF(150, 300)
' size of the field: width 120 pixels, height 30 pixels (in 96 dpi)
tbField.Size = New System.Drawing.SizeF(120, 30)
' set field visible
tbField.IsVisible = True
' set field background color: gray
tbField.BackgroundColor = System.Drawing.Color.Gray
' set label in the push button
tbField.Text = "CLICK"
```

```
tbField.SetTextFont(PSType1Font.Helvetica, 16)
tbField.TextColor = System.Drawing.Color.Black
' set border of the field: 1 pixel in width, drakgray
tbField.BorderColor = System.Drawing.Color.DarkGray
tbField.BorderWidth = 1

fields.Add(tbField)

' add fields
PDFFormHandler.AddFormFields(inputFilePath, fields, outputFilePath)
```

Get filled data in fields

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF_Filled.pdf";

List<BaseFormField> fields = PDFFormHandler.GetFormFields(inputFilePath);
Console.WriteLine("Number of Fields: " + fields.Count);
if (fields.Count > 0)
{
  foreach (BaseFormField field in fields)
  {
    Console.WriteLine("Field");
    Console.WriteLine("  Name:     " + field.Name);

    if (field is AFCheckBox)
    {
      Console.WriteLine("  Type:     " + "CheckBox");
      Console.WriteLine("  IsChecked: " + ((AFCheckBox)field).IsChecked);
    }
    else if (field is AFRadioButton)
    {
      Console.WriteLine("  Type:     " + "RadioButton");
      Console.WriteLine("  IsChecked: " + ((AFRadioButton)field).IsChecked);
    }
    else if (field is AFTextBox)
    {
      Console.WriteLine("  Type:     " + "TextBox");
      Console.WriteLine("  Content:   " + ((AFTextBox)field).Text);
    }
    else if (field is AFListBox)
    {
      Console.WriteLine("  Type:           " + "ListBox");
      Console.WriteLine("  Selected Item Index: " + ((AFListBox)field).SelectedIndexes[0]);
    }
    else if (field is AFComboBox)
    {
      Console.WriteLine("  Type:           " + "ComboBox");
      Console.WriteLine("  Selected Item Index: " + ((AFComboBox)field).SelectedIndex);
    }
  }
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF_Filled.pdf"

Dim fields As List(Of BaseFormField) = PDFFormHandler.GetFormFields(inputFilePath)
Console.WriteLine("Number of Fields: " + fields.Count)
If fields.Count > 0 Then
    For Each field As BaseFormField In fields
        Console.WriteLine("Field")
        Console.WriteLine("  Name:       " + field.Name)

        If TypeOf field Is AFCheckBox Then
```

```
            Console.WriteLine("  Type:      " + "CheckBox")
            Dim obj As AFCheckBox = field
            Console.WriteLine("  IsChecked: " + obj.IsChecked)
        ElseIf TypeOf field Is AFRadioButton Then
            Console.WriteLine("  Type:      " + "RadioButton")
            Dim obj As AFRadioButton = field
            Console.WriteLine("  IsChecked: " + obj.IsChecked)
        ElseIf TypeOf field Is AFTextBox Then
            Console.WriteLine("  Type:      " + "TextBox")
            Dim obj As AFTextBox = field
            Console.WriteLine("  Content:   " + obj.Text)
        ElseIf TypeOf field Is AFListBox Then
            Console.WriteLine("  Type:                " + "ListBox")
            Dim obj As AFListBox = field
            Console.WriteLine("  Selected Item Index: " + obj.SelectedIndexes(0))
        ElseIf TypeOf field Is AFComboBox Then
            Console.WriteLine("  Type:                " + "ComboBox")
            Dim obj As AFComboBox = field
            Console.WriteLine("  Selected Item Index: " + obj.SelectedIndex)
        End If

    Next

End If
```

Fill a field in the page

By position:

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";
String outputFilePath = Program.RootPath + "\\" + "output.pdf";

List<BaseFormField> fields = PDFFormHandler.GetFormFields(inputFilePath);

int cnt = 0;
foreach (BaseFormField field in fields)
{
   float x = field.Position.X + field.Size.Width / 2F;
   float y = field.Position.Y + field.Size.Height / 2F;

   if (field is AFCheckBox)
   {  //  fill a CheckBox field, set state to ON
      AFCheckBoxInput input = new AFCheckBoxInput(true);
      PDFFormHandler.FillFormField(inputFilePath, 0, new PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf");
   }
   else if (field is AFRadioButton)
   {  //  fill a RadioButton field, set state to ON
      AFRadioButtonInput input = new AFRadioButtonInput(true);
      PDFFormHandler.FillFormField(inputFilePath, 0, new PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf");
   }
   else if (field is AFTextBox)
   {  //  fill a TextBox field, change content to "Hello World"
      AFTextBoxInput input = new AFTextBoxInput("Hello World");
      PDFFormHandler.FillFormField(inputFilePath, 0, new PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf");
   }
   else if (field is AFListBox)
   {  //  fill a ListBox field, selete the 3rd item (with index value 2)
      AFListBoxInput input = new AFListBoxInput(new int[1] { 2 });
      PDFFormHandler.FillFormField(inputFilePath, 0, new PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf");
   }
   else if (field is AFComboBox)
   {  //  fill a BomboBox field, selete the 3rd item (with index value 2)
      AFComboBoxInput input = new AFComboBoxInput(2);
      PDFFormHandler.FillFormField(inputFilePath, 0, new PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf");
   }

   cnt++;
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"
```

```
Dim outputFilePath As String = Program.RootPath + "\\" + "output.pdf"

Dim fields As List(Of BaseFormField) = PDFFormHandler.GetFormFields(inputFilePath)

Dim cnt As Integer = 0
For Each field As BaseFormField In fields

   Dim x As Single = field.Position.X + field.Size.Width / 2.0F
   Dim y As Single = field.Position.Y + field.Size.Height / 2.0F

   Console.WriteLine("Field Location: x = " + x + ", y = " + y)

   If TypeOf field Is AFCheckBox Then
      ' fill a CheckBox field, set state to ON
      Dim input As AFCheckBoxInput = New AFCheckBoxInput(True)
      PDFFormHandler.FillFormField(inputFilePath, 0, New PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf")
   ElseIf TypeOf field Is AFRadioButton Then
      ' fill a RadioButton field, set state to ON
      Dim input As AFRadioButtonInput = New AFRadioButtonInput(True)
      PDFFormHandler.FillFormField(inputFilePath, 0, New PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf")
   ElseIf TypeOf field Is AFTextBox Then
      ' fill a TextBox field, change content to "Hello World"
      Dim input As AFTextBoxInput = New AFTextBoxInput("Hello World")
      PDFFormHandler.FillFormField(inputFilePath, 0, New PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf")
   ElseIf TypeOf field Is AFListBox Then
      ' fill a ListBox field, selete the 3rd item (with index value 2)
      Dim input As AFListBoxInput = New AFListBoxInput(New Integer() {2})
      PDFFormHandler.FillFormField(inputFilePath, 0, New PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf")
   ElseIf TypeOf field Is AFComboBox Then
      ' fill a BomboBox field, selete the 3rd item (with index value 2)
      Dim input As AFComboBoxInput = New AFComboBoxInput(2)
      PDFFormHandler.FillFormField(inputFilePath, 0, New PointF(x, y), input, outputFilePath +
cnt.ToString() + ".pdf")
   End If

   cnt += 1
Next
```

By Name:

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";
String outputFilePath = Program.RootPath + "\\" + "output.pdf";

List<BaseFormField> fields = PDFFormHandler.GetFormFields(inputFilePath);

int cnt = 0;
foreach (BaseFormField field in fields)
{
   if (field is AFCheckBox)
   {  //  fill a CheckBox field, set state to ON
      AFCheckBoxInput input = new AFCheckBoxInput(true);
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf");
   }
   else if (field is AFRadioButton)
   {  //  fill a RadioButton field, set state to ON
      AFRadioButtonInput input = new AFRadioButtonInput(true);
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf");
   }
   else if (field is AFTextBox)
   {  //  fill a TextBox field, change content to "Hello World"
      AFTextBoxInput input = new AFTextBoxInput("Hello World!");
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf");
   }
   else if (field is AFListBox)
   {  //  fill a ListBox field, selete the 3rd item (with index value 2)
      AFListBoxInput input = new AFListBoxInput(new int[1] { 2 });
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf");
   }
   else if (field is AFComboBox)
   {  //  fill a BomboBox field, selete the 3rd item (with index value 2)
      AFComboBoxInput input = new AFComboBoxInput(2);
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf");
   }

   cnt++;
}
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "output.pdf"

Dim fields As List(Of BaseFormField) = PDFFormHandler.GetFormFields(inputFilePath)
```

```
Dim cnt As Integer = 0
For Each field As BaseFormField In fields

   If TypeOf field Is AFCheckBox Then
      ' fill a CheckBox field, set state to ON
      Dim input As AFCheckBoxInput = New AFCheckBoxInput(True)
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf")
   ElseIf TypeOf field Is AFRadioButton Then
      ' fill a RadioButton field, set state to ON
      Dim input As AFRadioButtonInput = New AFRadioButtonInput(True)
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf")
   ElseIf TypeOf field Is AFTextBox Then
      ' fill a TextBox field, change content to "Hello World"
      Dim input As AFTextBoxInput = New AFTextBoxInput("Hello World!")
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf")
   ElseIf TypeOf field Is AFListBox Then
      ' fill a ListBox field, selete the 3rd item (with index value 2)
      Dim input As AFListBoxInput = New AFListBoxInput(New Integer() {2})
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf")
   ElseIf TypeOf field Is AFComboBox Then
      ' fill a BomboBox field, selete the 3rd item (with index value 2)
      Dim input As AFComboBoxInput = New AFComboBoxInput(2)
      PDFFormHandler.FillFormField(inputFilePath, field.Name, input, outputFilePath + cnt.ToString() +
".pdf")
   End If

   cnt += 1
Next
```

Fill fields in a document object

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_AF.pdf";
String outputFilePath = Program.RootPath + "\\" + "output.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

List<BaseFormField> fields = PDFFormHandler.GetFormFields(doc);

int cnt = 0;
foreach (BaseFormField field in fields)
{
   if (field is AFCheckBox)
   {  //  fill a CheckBox field, set state to ON
      AFCheckBoxInput input = new AFCheckBoxInput(true);
      PDFFormHandler.FillFormField(doc, field.Name, input);
   }
   else if (field is AFRadioButton)
   {  //  fill a RadioButton field, set state to ON
      AFRadioButtonInput input = new AFRadioButtonInput(true);
      PDFFormHandler.FillFormField(doc, field.Name, input);
   }
   else if (field is AFTextBox)
   {  //  fill a TextBox field, change content to "Hello World"
      AFTextBoxInput input = new AFTextBoxInput("Hello World!");
      PDFFormHandler.FillFormField(doc, field.Name, input);
   }
   else if (field is AFListBox)
   {  //  fill a ListBox field, selete the 3rd item (with index value 2)
      AFListBoxInput input = new AFListBoxInput(new int[1] { 2 });
      PDFFormHandler.FillFormField(doc, field.Name, input);
   }
   else if (field is AFComboBox)
   {  //  fill a BomboBox field, selete the 3rd item (with index value 2)
      AFComboBoxInput input = new AFComboBoxInput(2);
      PDFFormHandler.FillFormField(doc, field.Name, input);
   }

   cnt++;
}

doc.Save(outputFilePath);
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1_AF.pdf"
Dim outputFilePath As String = Program.RootPath + "\\" + "output.pdf"

Dim doc As PDFDocument = New PDFDocument(inputFilePath)

Dim fields As List(Of BaseFormField) = PDFFormHandler.GetFormFields(doc)

Dim cnt As Integer = 0
```

```
For Each field As BaseFormField In fields

    If TypeOf field Is AFCheckBox Then
        ' fill a CheckBox field, set state to ON
        Dim input As AFCheckBoxInput = New AFCheckBoxInput(True)
        PDFFormHandler.FillFormField(doc, field.Name, input)
    ElseIf TypeOf field Is AFRadioButton Then
        ' fill a RadioButton field, set state to ON
        Dim input As AFRadioButtonInput = New AFRadioButtonInput(True)
        PDFFormHandler.FillFormField(doc, field.Name, input)
    ElseIf TypeOf field Is AFTextBox Then
        ' fill a TextBox field, change content to "Hello World"
        Dim input As AFTextBoxInput = New AFTextBoxInput("Hello World!")
        PDFFormHandler.FillFormField(doc, field.Name, input)
    ElseIf TypeOf field Is AFListBox Then
        ' fill a ListBox field, selete the 3rd item (with index value 2)
        Dim input As AFListBoxInput = New AFListBoxInput(New Integer() {2})
        PDFFormHandler.FillFormField(doc, field.Name, input)
    ElseIf TypeOf field Is AFComboBox Then
        ' fill a BomboBox field, selete the 3rd item (with index value 2)
        Dim input As AFComboBoxInput = New AFComboBoxInput(2)
        PDFFormHandler.FillFormField(doc, field.Name, input)
    End If

    cnt += 1
Next

doc.Save(outputFilePath)
```