XDoc.PDF Developer Guide – Security Module

Table of Contents

Document Open Password

Open a PDFDocument with password

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String userPassword = @"you";
//  open an encrypted document
PDFDocument doc = PDFDocument.Open(intputFilePath, userPassword);

String outputFilePath = Program.RootPath + "\\" + "Output.pdf";
//  remove the password
doc.Save(outputFilePath);
```

Remove the password for an encrypted document

Output to a new file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String userPassword = @"you";
String outputFilePath = Program.RootPath + "\\" + "Remove.pdf";
//  remove password in the input file and output to a new file
int errorCode = PDFDocument.RemovePassword(intputFilePath, outputFilePath, userPassword);
if (errorCode == 0) Console.WriteLine("Success");
else Console.WriteLine("Failed");
```

Overwrite the original file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String userPassword = @"you";
//  remove password in the file
int errorCode = PDFDocument.RemovePassword(intputFilePath, userPassword);
if (errorCode == 0) Console.WriteLine("Success");
else Console.WriteLine("Failed");
```

Remove the password for an encrypted file stream

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "Remove.pdf";

FileStream inputStream = File.Open(intputFilePath, FileMode.Open, FileAccess.Read);
FileStream outputStream = File.Open(intputFilePath, FileMode.Create, FileAccess.ReadWrite);
String userPassword = @"you";

int errorCode = PDFDocument.RemovePassword(inputStream, outputStream, userPassword);
if (errorCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Add password to a plain file

Output to a new file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_with_pw.pdf";
String userPassword = "you";
String ownerPassword = "me";
//  create password setting
PasswordSetting setting = new PasswordSetting(userPassword, ownerPassword);
//  add password to plain file
int errorCode = PDFDocument.AddPassword(intputFilePath, outputFilePath, setting);
if (errorCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Overwrite the original file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String userPassword = "you";
String ownerPassword = "me";
// create password setting
PasswordSetting setting = new PasswordSetting(userPassword, ownerPassword);
// add password to plain file
int errorCode = PDFDocument.AddPassword(intputFilePath, setting);
if (errorCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Add password to a plain file stream

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_with_pw.pdf";

FileStream inputStream = File.Open(intputFilePath, FileMode.Open, FileAccess.Read);
FileStream outputStream = File.Open(intputFilePath, FileMode.Create, FileAccess.ReadWrite);
String userPassword = "you";
String ownerPassword = "me";
//  create password setting
PasswordSetting setting = new PasswordSetting(userPassword, ownerPassword);
//  add password to a plain file stream
int errorCode = PDFDocument.AddPassword(inputStream, outputStream, setting);
if (errorCode == 0)
{
    Console.WriteLine("Success");
}
else
{
    Console.WriteLine("Failed");
}
```

Change password for an encrypted file

Output to a new file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_with_pw.pdf";
String userPassword = "you";
String newUserPassword = "fill";
String newOwnerPassword = "watch";
//  create setting for the new password
PasswordSetting setting = new PasswordSetting(newUserPassword, newOwnerPassword);
//  change password for an encrypted file
int errorCode = PDFDocument.ChangePassword(intputFilePath, outputFilePath, userPassword, setting);
if (errorCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Overwrite the original file

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String userPassword = "you";
String newUserPassword = "fill";
String newOwnerPassword = "watch";
//  create setting for the new password
PasswordSetting setting = new PasswordSetting(newUserPassword, newOwnerPassword);
//  change password for an encrypted file
int errorCode = PDFDocument.ChangePassword(intputFilePath, userPassword, setting);
if (errorCode == 0)
{
    Console.WriteLine("Success");
}
else
{
    Console.WriteLine("Failed");
}
```

Change password for an encrypted file stream

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_with_pw.pdf";

FileStream inputStream = File.Open(intputFilePath, FileMode.Open, FileAccess.Read);
FileStream outputStream = File.Open(intputFilePath, FileMode.Create, FileAccess.ReadWrite);
String userPassword = "you";
String newUserPassword = "fill";
String newOwnerPassword = "watch";
//  create setting for the new password
PasswordSetting setting = new PasswordSetting(newUserPassword, newOwnerPassword);
//  change password for an encrypted file
int errorCode = PDFDocument.ChangePassword(inputStream, outputStream, userPassword, setting);
if (errorCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Save a PDFDocument with password

```
String intputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output.pdf";

//  load a plain file
PDFDocument doc = new PDFDocument(intputFilePath);

String userPassword = "fill";
String ownerPassword = "watch";
//  create password setting
PasswordSetting setting = new PasswordSetting(userPassword, ownerPassword);

//  save PDFDocument object with password
doc.Save(outputFilePath, setting);
```

Verify if a file is encrypted or not

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

bool isEncrypted = PDFDocument.IsEncrypted(intputFilePath);
bool hasUserPassword = PDFDocument.HasUserPassword(intputFilePath);

Console.WriteLine(@"This file is encrypted: " + isEncrypted);
Console.WriteLine(@"Password is not empty: " + hasUserPassword);
```

Remarks:

If a file does not been encrypted, it should never have any password.

Document Permission Setting

Add password with access permission setting

```
String inputFilePath = Program.RootPath + "\\" + "3.pdf";
String outputFilePath = Program.RootPath + "\\" + "3_pw_a.pdf";

//  create a password setting object with user password is "Hello World"
PasswordSetting passwordSetting = new PasswordSetting("Hello World");
//  set encryption level to AES-128
passwordSetting.Level = EncryptionLevel.AES_128bit;

//  print is allowed
passwordSetting.IsPrint = true;
//  print with high-resolution
passwordSetting.IsHighReso = true;
//  change document is allowed
passwordSetting.IsModify = true;
//  annotation is allowed
passwordSetting.IsAnnot = true;
//  fill form is allowed
passwordSetting.IsFillForm = true;
//  content extraction is allowed
passwordSetting.IsExtract = true;
//  copy is allowed
passwordSetting.IsCopy = true;
//  assemble the document is allowed
passwordSetting.IsAssemble = true;

//  add password to the file
PDFDocument.AddPassword(inputFilePath, outputFilePath, passwordSetting);
```

```
String inputFilePath = Program.RootPath + "\\" + "3.pdf";
String outputFilePath = Program.RootPath + "\\" + "3_pw_b.pdf";

//  create a password setting object with user password is "Hello World"
PasswordSetting passwordSetting = new PasswordSetting("Hello World");
//  set encryption level to AES-128
passwordSetting.Level = EncryptionLevel.AES_128bit;

//  print is allowed
passwordSetting.IsPrint = true;
//  print with low-resolution
passwordSetting.IsHighReso = false;
//  change document is not allowed
passwordSetting.IsModify = false;
//  annotation is not allowed
passwordSetting.IsAnnot = false;
//  fill form is not allowed
passwordSetting.IsFillForm = false;
//  content extraction is allowed
passwordSetting.IsExtract = true;
//  copy is allowed
passwordSetting.IsCopy = true;
//  assemble the document is allowed
passwordSetting.IsAssemble = true;

//  add password to the file
PDFDocument.AddPassword(inputFilePath, outputFilePath, passwordSetting);
```

Digital Signature

Search unsigned signature field in the document

```
String inputFilePath = @"...";
List<SignatureField> vals = PDFDigitalSignatureHandler.SearchUnsignedFields(inputFilePath);
if (vals == null || vals.Count == 0)
{
    Console.WriteLine("No Signature Field");
}
else
{
    Console.WriteLine("Signature Fields");

    for (int i = 0; i < vals.Count; i++)
    {
        Console.WriteLine(i + ": " + vals[i].ID + " (Pg " + vals[i].PageIndex + ")");
    }
}
```

Prepare a certification which used to sign a document

```
StoreName storeName = StoreName.My;
StoreLocation storeLocation = StoreLocation.CurrentUser;
String subjectName = "";

List<String> allSubjects = PDFDigitalSignatureHandler.GetAllSubjectNames(storeName,
storeLocation, true);
foreach (String str in allSubjects)
{
    Console.WriteLine("Make certification by name: " + str);
    X509Cert cert = PDFDigitalSignatureHandler.MakeCert(storeName, storeLocation, str);

    //  ...
}
```

Add an empty signature field

Add an empty signature field in page 0 with the field boundary [10, 10, 50, 150].

```
String inputFilePath = @"...";
String outputFilePath = @"...";

PDFSignature signature = new PDFSignature(new RectangleF(10F, 10F, 250F, 150F));
if (PDFDigitalSignatureHandler.CreateSignatureField(inputFilePath, outputFilePath, signature,
0) == 0)
{
    Console.WriteLine("[SUCCESS]");
}
else
{
    Console.WriteLine("[FAILED]");
}
```

Sign a document

```
String inputFilePath = @"...";
String outputFilePath = @"...";
String imagePath = @"...";
String certSubjectName = @"...";
String fieldID = @"...";

X509Cert cert = PDFDigitalSignatureHandler.MakeCert(StoreName.My, StoreLocation.CurrentUser,
certSubjectName);
//  set cert properties
cert.Location = "CHINA SHANGHAI";
cert.Reason = "Reason";
cert.APMode = APMode.Text;      //  APMode.Text | APMode.Image
cert.Image = new Bitmap(imagePath);

if (PDFDigitalSignatureHandler.Sign(inputFilePath, outputFilePath, cert, fieldID) == 0)
{
    Console.WriteLine("[SUCCESS]");
}
else
{
    Console.WriteLine("[FAILED]");
}
```