

PDF Security, part 1 – create or open password protected PDF documents

Written by Apitron Documentation Team

Introduction

PDF allows you to protect the contents of your documents by setting a password. There are two types of passwords that can be set - an *owner password* and a *user password*.

Whether additional operations will be allowed on a decrypted document depends on which password (if any) was supplied when the document was opened, and on any access restrictions that were specified when the document was created:

- Opening the document with the correct *owner password* should allow full access to the document. This unlimited access includes the ability to change the document's passwords and access permissions.
- Opening the document with the correct *user password*, or opening a document with the default password, should allow additional operations to be performed according to the user access permissions specified for the document.

The complete documentation can be found in section *7.6.4 Public-Key Security Handlers* in PDF specification.

User access permissions control the printing, content extraction and editing, and other possible modifications of the source document. For the complete list of user access permissions see the section *7.6.4.2 Public-Key Encryption Dictionary*.

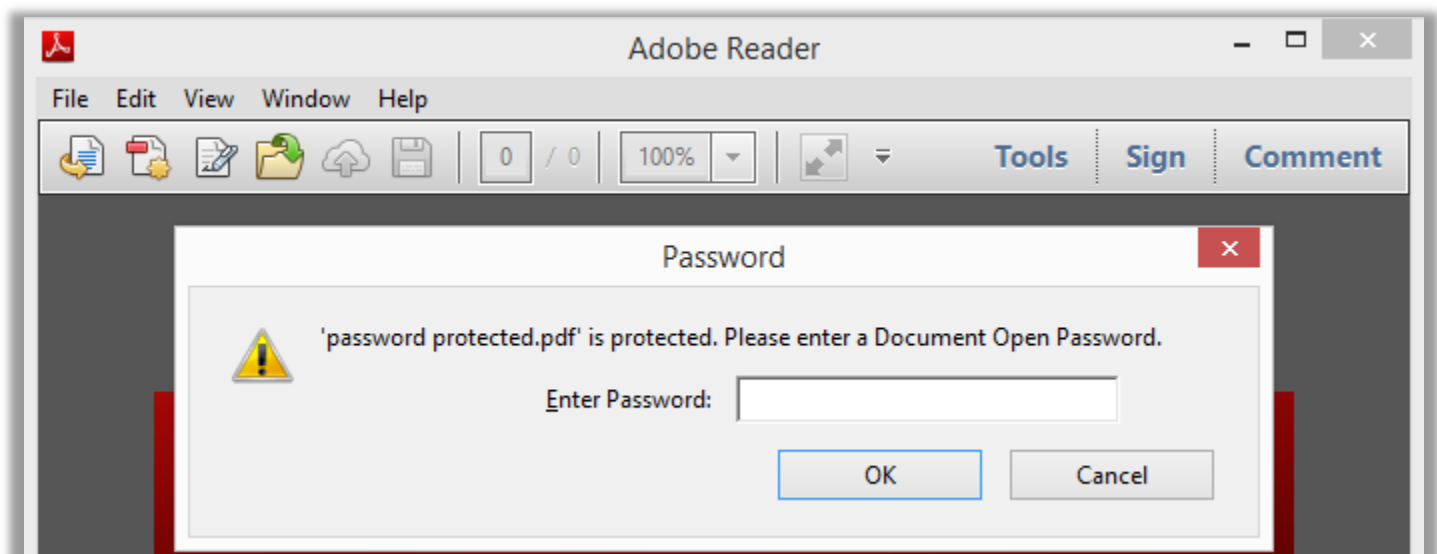
Setting PDF document password

The code below opens existing PDF document and sets both passwords for it. It also sets the optional encryption algorithm and permissions entry.

```
public void SetDocumentPassword()
{
    // open existing document and create output stream
    using (Stream inputStream = File.OpenRead("hello world.pdf"),
        outputStream = File.Create("password protected.pdf"))
    {
        using (FixedDocument doc = new FixedDocument(inputStream))
        {
            // create security settings for the document
            doc.SecuritySettings = new SecuritySettings();
            // set document passwords
            doc.SecuritySettings.OwnerPassword = "owner";
            doc.SecuritySettings.UserPassword = "user";
            // set document's encryption level and permissions
            doc.SecuritySettings.EncryptionLevel = EncryptionLevel.AES_128bit;
            doc.SecuritySettings.Permissions = Permissions.AllowAllPermissions;

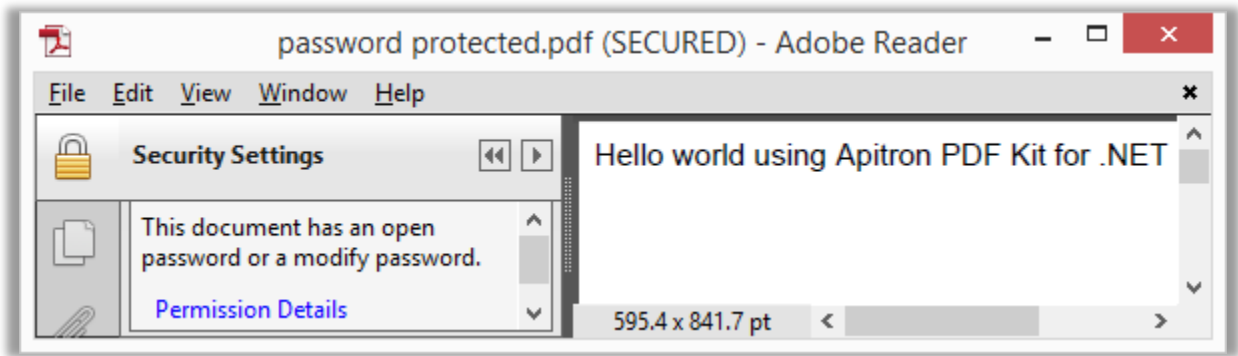
            // save document
            doc.Save(outputStream);
        }
    }
}
```

When you open the generated document the password request windows appears:



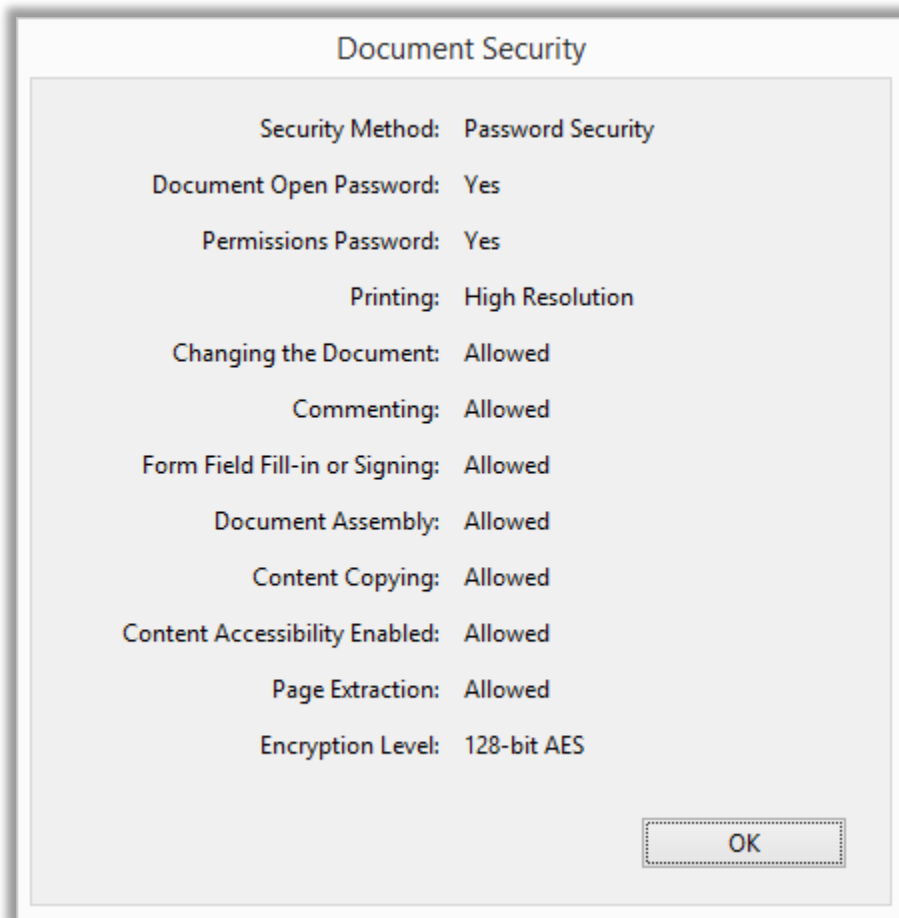
Pic. 1 Opening PDF document - password request window

After entering the correct password, you'll be able to see document's content and examine its security properties. See the image below:



Pic. 2 Opened password protected PDF document

Clicking on *Permission Details* opens the details window showing all properties set:



Pic. 3 Document security properties

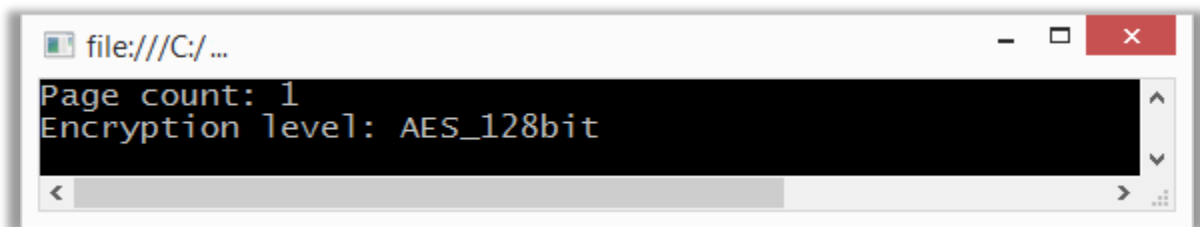
Opening password protected PDF document

The sample below opens password protected document created using the code from previous example, and prints the number of pages in this document and encryption level used.

```
public void OpenPasswordProtectedDocument()
{
    // open existing pdf document
    using (Stream inputStream = File.OpenRead("password protected.pdf"))
    {
        // provide the owner password to read the document
        FixedDocument doc = new FixedDocument(inputStream, "owner");

        Console.WriteLine("Page count: {0}", doc.Pages.Count);
        Console.WriteLine("Encryption level: {0}", doc.SecuritySettings.EncryptionLevel);
    }
}
```

Resulting output is shown below:



Pic. 4 Reading password protected pdf document

Conclusion

In this post we demonstrated how to create or open password protected PDF documents using [Apitron PDF Kit](#) library. This library can be used by applications targeting many platforms like Android and iOS (via Xamarin), Windows, Windows Phone, Windows Store and others. It's also can be used to create server, cloud and web solutions, starting from small business to enterprise level. Read our [free book](#) showing the component in action, and see for yourself. Contact us if you have any questions, we are always ready to help.