

Licensing Guide

Updated: October 12, 2008

Legal Note

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, email addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of TallComponents BV.

TallComponents BV may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from TallComponents BV, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Acrobat and PDF are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Windows NT, 2000, XP, and Server 2003, Vista, .NET Framework and Internet Information Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

© 2001-2008 TallComponents BV. All rights reserved.

Contents

Legal Note	2
Contents.....	3
License Model.....	4
Server License.....	5
ASP.NET URL License	5
ASP.NET Domain License.....	5
Web Application License	6
Site License.....	6
Client Application License.....	7
Client Component License.....	8
Help Me Select a License	9
License Key.....	11
Obtain a License Key.....	11
Install a Generated License Key	13
No Developer License	15
Upgrades.....	16
Updates.....	17
FAQ	18
Code Snippets	19

License Model

After many years of both developing and using software components we have designed a set of license types that aims to meet both our interests as a vendor and your requirements as a customer.

We offer the following license types:

- Server License
- ASP.NET URL License
- ASP.NET Domain License
- Web Application License
- Site License
- Client Application License
- Client Component License

These license types do not depend on the component that you want to license. Not all license types are available for each components, though. E.g. the Web Application license is not available for a Winforms Control library such as PDFReaderControls.NET.

After you buy a license, you obtain a license key. This key is a signature of a runtime attribute that corresponds to the type of license that you have purchased. E.g. for a Server License, the runtime attribute is the NETBIOS name of the machine. The following table shows the runtime attribute per license type.

License type	Runtime attribute
Server license	NETBIOS name of the machine.
ASP.NET URL License	The combination of the host name and the virtual folder name. E.g. 'www.tallcomponents.com/virtualfolder'.
ASP.NET Domain license	The domain of the incoming HTTP request. E.g. 'tallcomponents.com'.
Web Application license	The combination of the company and product assembly attributes of the calling assembly (= managed DLL). Works only in an ASP.NET context.
Site license	Active Directory Domain name.
Client Application license	The combination of the company and product assembly attributes of the entry assembly (= managed EXE).
Client Component license	The combination of the company and product assembly attributes of the calling assembly (= managed DLL). Works only if no entry assembly (= managed EXE) can be found.

Table 1. Runtime attribute per license type

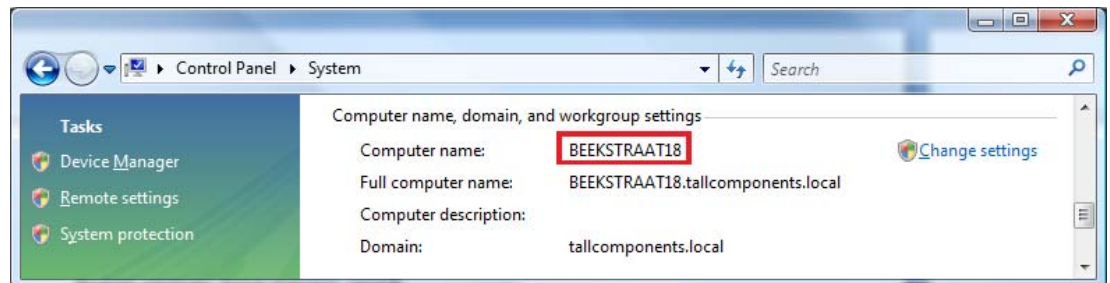
At runtime, when our software gets invoked, it makes a runtime check to see if the component is used within the terms of the license. It basically checks whether the actual runtime attribute of the license type corresponds to the license key.

The following sections discuss each license type in more detail.

Server License

Runtime attribute: NETBIOS name.

The Server license allows you to use our component on exactly one machine for any purpose. We allow you to transfer the license once to another machine. It does not matter how many different applications on that machine invoke our component or how many CPUs the machine has. It also does not matter how many (remote) clients access the application server concurrently.



If you are unsure what your server name is, then you can use the following code to reproduce it:

```
string getRuntimeAttribute() {
    if ( null != System.Web.HttpContext.Current ) {
        return System.Web.HttpContext.Current.Server.MachineName;
    }
    else {
        return System.Environment.MachineName;
    }
}
```

ASP.NET URL License

Runtime attribute: URL = host name plus application name.

The URL license allows you to run our software on any number of servers that serve the same URL. Not surprisingly, the runtime attribute is the URL (host name plus application name). For example, you have a web application named 'myWebApp' deployed on your domain 'www.mycompany.com'. The URL license will then check each request to see if it matches 'www.mycompany.com/myWebApp/'.

If you are unsure what your URL is, then you can reproduce this runtime attribute using the following code:

```
string getRuntimeAttribute() {
    return HttpContext.Current.Request.Url.Host +
        HttpContext.Current.Request.ApplicationPath;
}
```

ASP.NET Domain License

Runtime attribute: ASP.NET domain.

This license validates successfully if it runs inside the context of an ASP.NET request coming from a given domain such as 'tallcomponents.com'. This license is less restrictive than the URL license.

If you are unsure what your ASP.NET domain is, then you can reproduce this runtime attribute using the following code:

```

string getRuntimeAttribute() {
    if ( null != System.Web.HttpContext.Current ) {
        string host = HttpContext.Current.Request.Url.Host;
        return extractDomainFromHost( host );
    }
    else {
        throw NotSupportedException();
    }
}

static string extractDomainFromHost( string host ) {
    // actual code omitted
    // returns "tallcomponents.com" from "admin.tallcomponents.com"
    // returns "shell.co.uk" from "www.shell.co.uk"
    // etc.
}

```

Web Application License

Runtime attribute: Combination of Company and Product assembly attributes of the calling assembly (DLL).

If you are a vendor of a web application that is installed at your customer's site and you have no control over the machine name or the ASP.NET domain, then the previous license types will not work for you. To cover this scenario we offer the Web Application license. The runtime attribute is the combination of the Company and Product assembly attributes of the assembly (DLL) that calls our assembly. Consequently, the license key does not depend on any customer specific information.

The Company and Product assembly attributes are typically found in AssemblyInfo.cs or AssemblyInfo.vb that is included in the project that calls our assembly. They normally are defined like this in C#:

```

[assembly: AssemblyProduct( "StatementGenerator" )]
[assembly: AssemblyCompany( "BigBank" )]

```

Or like this in VB.NET:

```

<Assembly: AssemblyProduct( "StatementGenerator" )>
<Assembly: AssemblyCompany( "BigBank" )>

```

If you are unsure, you can reproduce this runtime attribute using the following code (null and bounds checks omitted):

```

// see Chapter Code Snippets for methods getCallingAssembly,
// getCompanyAttribute and getProductAttribute
string getRuntimeAttribute() {
    Assembly callingAssembly = getCallingAssembly();

    string company = getCompanyAttribute( callingAssembly );
    string product = getProductAttribute( callingAssembly );

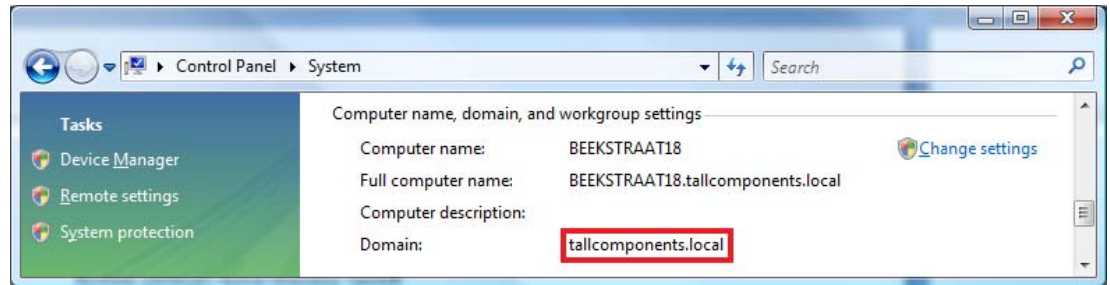
    return string.Format( "{0}_{1}", company, product );
}

```

Site License

Runtime attribute: Windows Server Domain

The Site license allows you to redistribute our component to any number of machines that are part of a given **Windows Server Domain**. This license is especially suitable for corporate / in-house applications.



If you are unsure, you can reproduce this runtime attribute using the following code:

```
using System.Security.Principal;

string getRuntimeAttribute() {
    WindowsIdentity user = WindowsIdentity.GetCurrent();
    string domain = user.Name.Split('\\')[0];
    return domain;
}
```

It basically retrieves the fully qualified name of the current user and then chops off the user name which leaves the Windows Server Domain.

Client Application License

Runtime attribute: Combination of Company and Product assembly attributes of the entry assembly (managed EXE) that hosts our component.

This is a royalty free license that allows you to redistribute our component with a single Winforms application to an unlimited number of seats. This license type requires that your application is a managed application because the runtime attribute is based on assembly attributes of the so-called entry assembly (= EXE).

The Company and Product assembly attributes are typically found in AssemblyInfo.cs that is included in the project that compiles to your EXE. They normally are defined like this in C#:

```
[assembly: AssemblyProduct("StatementGenerator")]
[assembly: AssemblyCompany("BigBank")]
```

Or like this in VB.NET:

```
<Assembly: AssemblyProduct("StatementGenerator")>
<Assembly: AssemblyCompany("BigBank")>
```

If you are unsure, you can reproduce this runtime attribute using the following code (null and bounds checks omitted):

```

using System.Reflection;

// see Chapter Code Snippets for methods
// getCompanyAttribute and getProductAttribute
string getRuntimeAttribute() {
    Assembly entryAssembly = Assembly.GetEntryAssembly();

    string company = getCompanyAttribute( entryAssembly );
    string product = getProductAttribute( entryAssembly );

    return string.Format( "{0}/{1}", company, product );
}

```

Client Component License

Runtime attribute: Combination of Company and Product assembly attributes of the calling assembly (managed DLL) that calls our component.

If your application does not have an entry assembly, e.g. because you are using our component with an unmanaged legacy application, then the Client Component license may be a good fit. It does a similar check against the calling assembly (DLL) instead of the entry assembly (EXE). The calling assembly is the inner-most managed DLL found in the call stack.

The Company and Product assembly attributes are typically found in AssemblyInfo.cs that is included in the project that compiles to your DLL. They normally are defined like this in C#:

```

[assembly: AssemblyProduct( "StatementGenerator" )]
[assembly: AssemblyCompany( "BigBank" )]

```

Or like this in VB.NET:

```

<Assembly: AssemblyProduct( "StatementGenerator" )>
<Assembly: AssemblyCompany( "BigBank" )>

```

If you are unsure, you can reproduce this runtime attribute using the following code (null and bounds checks omitted):

```

// see Chapter Code Snippets for methods getCallingAssembly,
// getCompanyAttribute and getProductAttribute
string getRuntimeAttribute() {
    Assembly callingAssembly = getCallingAssembly();

    string company = getCompanyAttribute( callingAssembly );
    string product = getProductAttribute( callingAssembly );

    return string.Format( "{0}_{1}", company, product );
}

```


Help Me Select a License

In this section we will help you select the correct license using the following straightforward questionnaire.

Question 1: Are you using our component in an ASP.NET application?

Answer: Yes, only in an ASP.NET application. *Please proceed to question 2.*

Answer: No. *Please proceed to question 13.*

Answer: Actually, I need to deploy your software to both an ASP.NET application and a non-ASP.NET application. *In that case, you need at least 2 different license types. Take this questionnaire for both deployment scenarios, respectively.*

Question 2: Will you deploy to web servers that all serve the same URL? (The URL is defined as the combination of the host name and the application name. This is reproduced by `HttpContext.Current.Request.Url.Host + HttpContext.Current.Request.ApplicationPath`.)

Answer: Yes. *Please proceed to question 3.*

Answer: No. *Please proceed to question 5.*

Question 3: Will you deploy to one or two machines and not more?

Answer: Yes. *Please proceed to question 4.*

Answer: No, I will deploy to at least three servers or plan to do so in the foreseeable future. *You need an ASP.NET URL License.*

Question 4: Do you know the NETBIOS names of your web servers at deployments time?

Answer: Yes. *Although the ASP.NET URL License will work for you, the Server License is the most economic choice. You will need one per machine.*

Answer: No. *You need an ASP.NET URL License.*

Question 5: Will you deploy to web servers that all serve HTTP requests to a common ASP.NET domain (such as 'yahoo.com' or 'shell.co.uk')?

Answer: Yes. *Please proceed to question 6.*

Answer: No. *Please proceed to question 8.*

Question 6: Will you deploy to five or less machines?

Answer: Yes. *Please proceed to question 7.*

Answer: No, I will deploy to at least six servers or plan to do so in the foreseeable future. *You need an ASP.NET Domain License.*

Question 7: Do you know the NETBIOS names of your web servers at deployments time?

Answer: Yes. *Although the ASP.NET Domain License will work for you, the Server License is the most economic choice. You will need one per machine.*

Answer: No. *You need an ASP.NET Domain License.*

Question 8: Will you deploy to web servers that are all members of the same Windows Server Domain that is known at deployment time?

Answer: Yes. *Please proceed to question 9.*

Answer: No. *Please proceed to question 11.*

Question 9: Will you deploy to five or less machines and not more?

Answer: Yes. *Please proceed to question 10.*

Answer: No, I will deploy to atleast six servers or plan to do so in the foreseeable future. *You need a Site License.*

Question 10: Do you know the NETBIOS names of your web servers at deployment time?

Answer: Yes. *Although the Site License will work for you, the Server License is the most economic choice. You will need one per machine.*

Answer: No. *You need a Site License.*

Question 11: Will you deploy to nine or less machines?

Answer: Yes. *Please proceed to question 12.*

Answer: No, I will deploy to atleast ten servers or plan to do so in the foreseeable future. *You need a Web Application License.*

Question 12: Do you know the NETBIOS names of your web servers at deployment time?

Answer: Yes. *Although the Web Application License will work for you, the Server License is the most economic choice. You will need one per machine.*

Answer: No. *You need a Web Application License.*

Question 13: Will you redistribute our software with a non-server application? In other words, will you deploy our component with a client or desktop application?

Answer: Yes. *Please proceed to question to 14.*

Answer: No. Although my application is not an ASP.NET application, it is a server application in the sense that potentially multiple remote users or remote machines invoke its functionality over the network. *Please proceed to question 15.*

Question 14: Will our software always run inside the same managed EXE?

Answer: Yes. *You need a Client Application license per application with which you plan to redistribute our software.*

Answer: No, my application is a legacy application. The hosting EXE is not managed. *You need a Client Component license.*

Question 15: Will you deploy to server machines that are all members of the same Windows Server Domain that is known at deployment time?

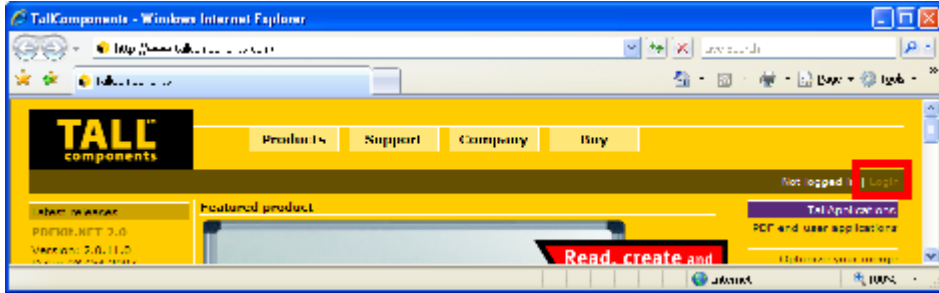
Answer: Yes. *You need a Site License.*

Answer: No. *Please contact sales@tallcomponents.com.*

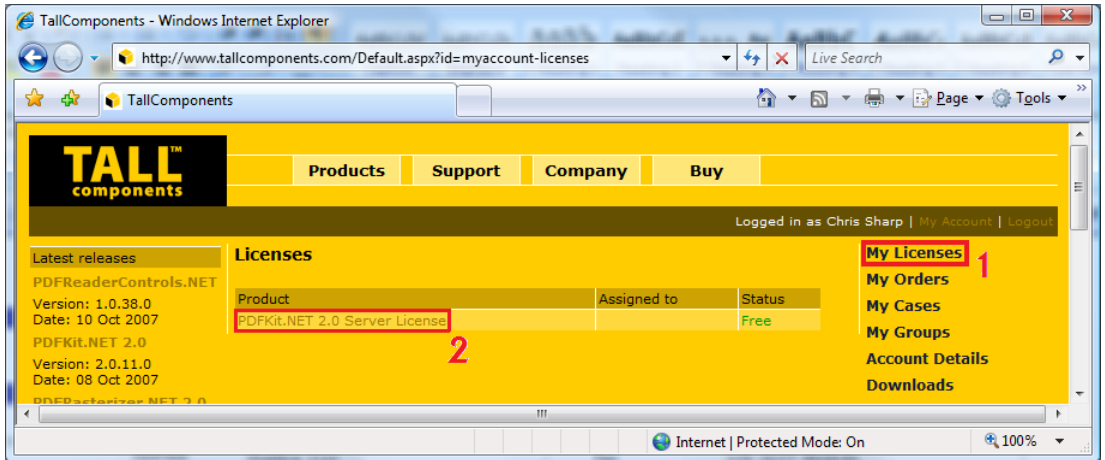
License Key

Obtain a License Key

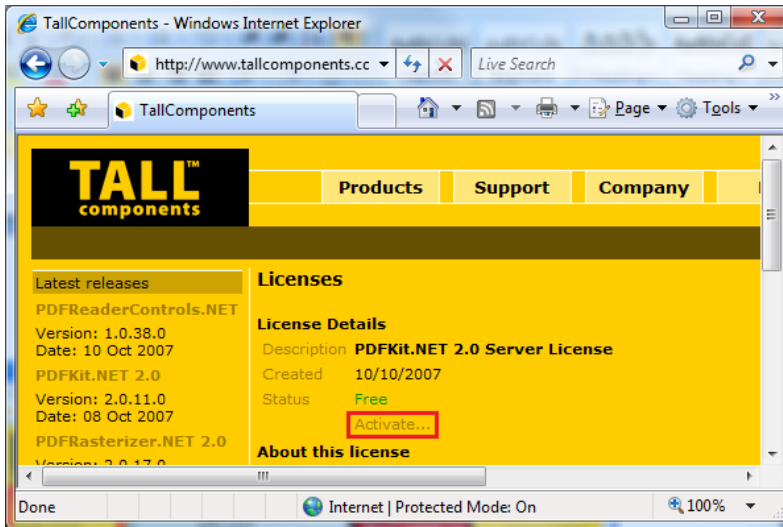
You can generate your license key yourself from your online TallComponents account. After you purchase a license, you receive a confirmation that includes your username and password. In the following, it is assumed that you already have a TallComponents account and know your username and password.



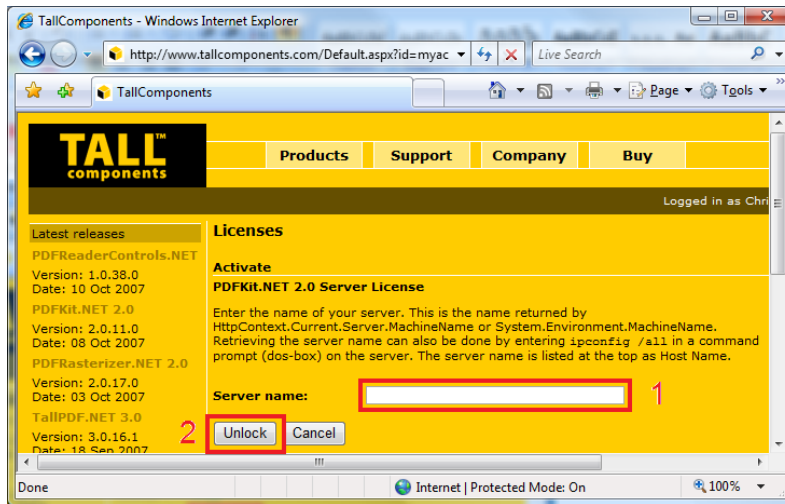
Step 1: login



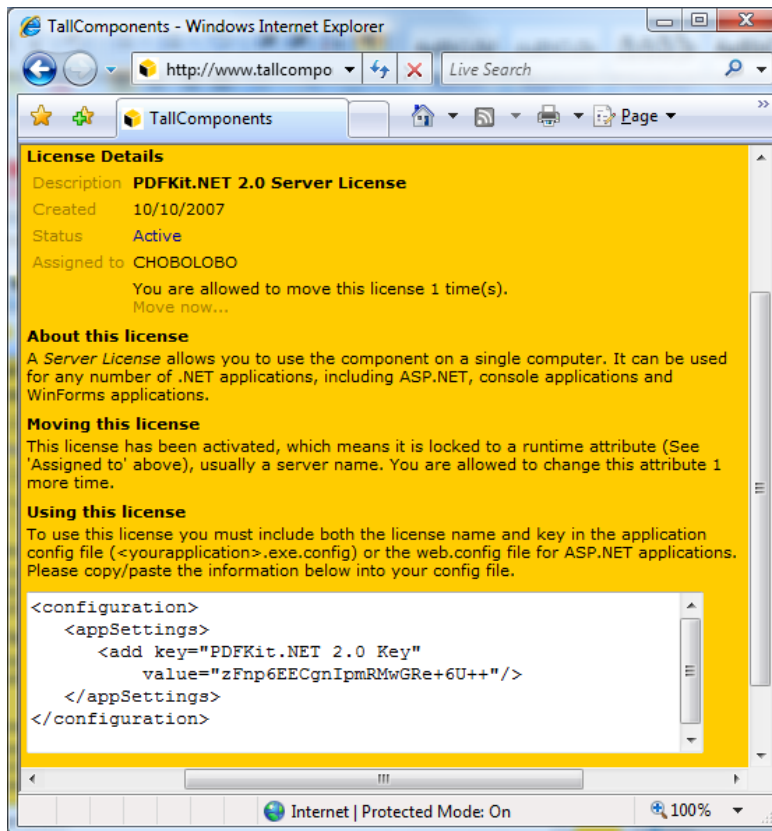
Step 2: Go to My Account -> My Licenses and select the license that you want to activate



Step 3: Click the [Activate...] link



Step 4: Enter runtime attribute and click the Unlock button



Step 5: The license key is generated as a section that you can add to your web.config or app.config file

Note that the last step presents you with a license key that is preformatted as a configuration file section. The next section discusses how to install this license key.

Install a Generated License Key

After you have generated a license key, you must associate it with your application or environment so that our software finds it at runtime. We offer different ways of doing this. You can pick the one that suits you best.

The previous section described how to generate a license key. The result was a configuration file section that looks something like this:

```
<configuration>
  <appSettings>
    <add key="PDFKit.NET 2.0 Key"
      value="zFnp6EECgnIpmRMwGRe+6U++" />
  </appSettings>
</configuration>
```

Two elements are of importance:

1. The *key name*: PDFKit.NET 2.0 Key
2. The *key value*: zFnp6EECgnIpmRMwGRe+6U++

Method 1: web.config

Applies to license types: Server License; ASP.NET Domain License; Web Application License.

You can install the license key in the web.config file of your ASP.NET application/web service by adding the generated XML snippet *as is* to your web.config file. Note that the web.config file should be located in the folder that is configured as your virtual folder.

It is possible to install multiple license keys in a single web.config file by post-fixing the key names so that they are unique. This allows you to use a single identical web.config file on multiple servers.

Example:

```
<configuration>
  <appSettings>
    <add key="PDFKit.NET 2.0 Key A"
      value="zFnp6EECgnIpmRMwGRe+6U++" />
    <add key="PDFKit.NET 2.0 Key B"
      value="sf583m8u3845m98v4hg+6f++" />
  </appSettings>
</configuration>
```

Method 2: app.config

Applies to license types: Server License; Client Application License

If you use our component in combination with a Winforms or Console application, then you can add the license key to the app.config file. You do this by adding the generated XML snippet *as is* to the configuration file.

Note that this file is called app.config because this is the name of the file as included in your Visual Studio .NET project. When you compile your application, Visual Studio .NET automatically renames it to <yourapp>.exe.config and copies it to the same folder as where your EXE is copied. Note that this should also be the case at deployment time.

Method 3: TallComponents.Xxx.dll.config

Applies to license types: All

This file must be in the exact same folder as our assembly. Say that you want to install a PDFKit.NET 2.0 license key, then you must create a file called TallComponents.Pdf.Kit.dll.config and copy this file to the same folder where TallComponents.Pdf.Kit.dll is located. The contents of the configuration is exactly the XML snippet that is generated on our web site after you activated your license.

Method 4: TallComponents.Licensing.LicenseCollection.Add(name, value);

Applies to license types: All

Instead of installing the license in a separate configuration file, you can also install the license key by making a method call at runtime.

Let's say that this is the generated license key section:

```
<configuration>
  <appSettings>
    <add key="PDFKit.NET 2.0 Key"
      value="zFnp6EECgnIpmRMwGRE+6U++" />
  </appSettings>
</configuration>
```

Then you can install this license key programmatically by making the following method call:

```
TallComponents.Licensing.LicenseCollection.Add(
  "PDFKit.NET 2.0 Key",
  "zFnp6EECgnIpmRMwGRE+6U++" );
```

Note: You should make a call to this method before invoking our software.

Method 5: Custom Assembly Attribute

Applies to license types: All

You can also install a license key by adding a custom assembly attribute to your assembly. For the Client Application license, you must apply this attribute to the *entry assembly*. For the Client Component license and the Web Application license, you must apply this attribute to the *calling assembly*.

Let's say that this is the generated license key section:

```
<configuration>
  <appSettings>
    <add key="PDFKit.NET 2.0 Key"
      value="zFnp6EECgnIpmRMwGRE+6U++" />
  </appSettings>
</configuration>
```

Then you can install this license key by adding the following assembly attribute:

```
[assembly: TallComponents.Licensing.PDFKitLicenseAttribute(
  "PDFKit.NET 2.0 Key",
  "zFnp6EECgnIpmRMwGRE+6U++" )]
```

Note that the type of the attribute is component specific. Here are the attribute classes per component:

Component	License attribute
PDFKit.NET	TallComponents.Licensing.PDFKitLicenseAttribute
TallPDF.NET	TallComponents.Licensing.TallPDFLicenseAttribute
PDFRasterizer.NET	TallComponents.Licensing.PDFRasterizerLicenseAttribute
PDFReaderControls.NET	TallComponents.Licensing.PDFReaderControlsLicenseAttribute

Table 2. License attribute class per component

No Developer License

We do not sell developer licenses. You are free to use our components in evaluation mode for development. In evaluation mode, you compile and run against exactly the same assembly as in production. The component is fully functional and does not expire. The only difference is that an evaluation notice is printed across the generated PDF or raster output.

Upgrades

An upgrade is defined as an increment of the major version. For example, you upgrade from 2.0 to 3.0. An Upgrade costs 60% of the list price (40% discount). If you upgrade within one month after we release an upgrade, you can upgrade at 50% of the list price (50% early-bird discount). If you purchase a license and we release a major upgrade within eight months of this purchase, you upgrade for free.

Four-months cutoff period

Four months after we release an upgrade, we stop releasing updates and bug fixes of the previous major version. From an engineering point of view, it is impossible to fix bugs in multiple source code branches. If we would have to do this, we would never be able to implement a major redesign. Major redesigns are required now and then to be able to fix bugs and add features fast.

Updates

An update is defined as any version change other than the major version. For example, you update from 2.0.5.0 to 2.0.6.0 or from 2.0.30.0 to 2.1.0.0 Normally, an update includes bug fixes and minor improvements/features. All our licenses include updates. So updates are free and do not require a paid subscription. They can be downloaded from our web site. Our home page displays all latest versions in the top left corner.



Figure 1. The TallComponents home page shows the latest versions at a glance

FAQ

Q: Some time ago, we purchased a server license. We have just renamed our server (or moved our software to another server) and now the evaluation message appears! What should we do?

A: Per server license we allow you to regenerate your license key once. You can do this yourself as follows:

1. login to your TallComponents account
2. go to My Account -> My Licenses
3. select the license you want to move
4. click the 'Move now' link
5. enter the new server name

Q: I have installed the license key per your instructions but the evaluation message still appears! What am I doing wrong?

A1: If you are using TallPDF.NET or PDFKit.NET, then please send a message with subject 'License not working' to support@tallcomponents.com and include the following:

- attach PDF (no screenshot!) that has the evaluation notice
- The username of the account that holds the license
- If applicable, the runtime attribute of the license that isn't working

Q: One of our team members has left the company. Our licenses are under his account. We want to move the licenses to another account. How can we do that?

A: Please send a message with subject 'Please move licenses' to support@tallcomponents.com and include the following information:

- current account
- new account (create this one first if it does not exist)

Make sure that your request comes from the same domain as the e-mail address of the current account. To check, we may call the general company number and ask for you.

Q: After evaluating your product I have purchased a license. Where do I download the licensed software?

A: There is no separate *licensed version* of our software. In production you will be using exactly the same assembly as for evaluation. The only difference is in the absence or presence of a license key. This determines whether our assembly runs in licensed or in evaluation mode.

Code Snippets

```
static Assembly getCallingAssembly() {
    Assembly componentAssembly = Assembly.GetCallingAssembly();
    StackTrace stackTrace = new StackTrace();
    for ( int i=0; i<stackTrace.FrameCount; i++ )
    {
        StackFrame stackFrame = stackTrace.GetFrame(i);
        if ( stackFrame.GetMethod().DeclaringType.Assembly.FullName !=
            componentAssembly.FullName )
        {
            return stackFrame.GetMethod().DeclaringType.Assembly;
        }
    }
    return null;
}

static string getCompanyAttribute( Assembly assembly ) {
    object[] attributes = assembly.GetCustomAttributes(
        typeof( AssemblyCompanyAttribute ), true );
    if ( attributes.Length > 0 ) {
        AssemblyCompanyAttribute attribute =
            attributes[0] as AssemblyCompanyAttribute;
        if ( null != attribute ) return attribute.Company;
    }
    return null;
}

static string getProductAttribute( Assembly assembly ) {
    object[] attributes = assembly.GetCustomAttributes(
        typeof( AssemblyProductAttribute ), true );
    if ( attributes.Length > 0 ) {
        AssemblyProductAttribute attribute =
            attributes[0] as AssemblyProductAttribute;
        if ( null != attribute ) return attribute.Product;
    }
    return null;
}
```