

# PETER'S DATA ENTRY SUITE v5 INSTALLATION GUIDE



Click on any of these topics to jump to them:

- ◆ [Prepare a web application or site \(first time installation, service releases, and upgrades\)](#) ..... [FAQs](#)
  - [Adding to the Visual Studio/Visual Web Developer Toolbox](#) ..... [FAQs](#)
  - [Upgrading from DES v4.0](#) ..... [FAQs](#)
  - [Converting from Professional Validation and More](#) ..... [FAQs](#)
  - [Converting from Peter's Date Package](#) ..... [Read This First](#) ..... [FAQs](#)
  - [Converting from ASP.NET 2.0 to ASP.NET 4.0](#)
  - [Using Alternative Locations for Files in the DES Folder](#)
  
- ◆ [Installing Licenses](#) ..... [Troubleshooting](#)
  - [Converting from Trial Version To Paid Version](#)
  
- ◆ [Convert Native Controls to their DES equivalents](#)
  
- ◆ [Add or Change Peter's Business Logic Driven UI support](#)
  
- ◆ [Deploying Your Web Applications](#)
- ◆ [Installing DES onto a New Development Computer](#)
  
- ◆ [Exploring the Current Settings](#)
- ◆ [Installing into a Partial Trust Environment](#)
- ◆ [Using IIS 7](#)
- ◆ [Troubleshooting](#)
- ◆ [Table of Contents](#)

## Overview

**Peter's Data Entry Suite** ("DES") is an extensive set of controls and tools for developing powerful data entry ASP.NET web forms. Use this document to modify your web application with its assemblies and support files, migrate from older versions, and convert Microsoft controls to their DES equivalents.

### *Easy Installation with the Web Application Updater*

DES does most of the work for you through the **Web Application Updater** program that is in the **Start** menu or the [DES installation folder]. Here are the actions you can take with the **Web Application Updater**:

- **[Prepare a web application or site](#)** – Use it for:
  - First time installation
  - Install a service release (from DES 5.0.x to 5.0.y)
  - [Upgrading from DES v4.0](#)
  - [Convert Native Controls to their DES equivalents](#)
  - [Converting from Professional Validation and More](#) (any version)
  - [Converting from Peter's Date Package](#) (any version)
  - Repair an application
  - Add validation support for a third party productSee "[Prepare a web application or site](#)".
- **Convert files not previously converted or a non-web application library** – Use this if you have an existing web application with DES installed but have some files that need conversion or upgrading. It can convert native controls to their DES equivalents, upgrade from DES 4, Professional Validation and More, and Peter's Date Package. Also use it if you have a class library where you have created code based on DES.
- **Add or change Peter's Business Logic Driven UI support to your web application or site**– Peter's Business Logic Driven UI lets you build applications separating the business logic from the user interface. It includes numerous support files that change how you develop your application. See "[Add or Change Peter's Business Logic Driven UI support](#)".

The **Web Application Updater** can be safely run multiple times on the same web application. It will only make changes that are necessary and will backup most files before changing them.

### *Installation that is not handled by the Web Application Updater*

The following are installation issues that are described in this document.

- Install the DES controls into the toolbox of Visual Studio and Visual Web Developer. See "[Adding to the Visual Studio/Visual Web Developer Toolbox](#)".
- Setup Licenses for the Suite or its modules. See "[Installing Licenses](#)".
- Support third party data entry controls that you want DES's validators to work with which were not listed in the **Web Application Updater**. See the **Using Third Party Controls** document.

Once **Peter's Data Entry Suite** has been installed, **Peter's Input Security** has a separate set of installation instructions as the process involves securing a web site, not just adding a few web controls. You are not required to use **Peter's Input Security** although it is strongly recommended if you are concerned about hackers. See the **Peter's Input Security User's Guide**.

## ***Platform Support***

This product was written for Microsoft ASP.NET version 2.0 and higher. It includes assemblies specific to ASP.NET 2.0 and ASP.NET 4.0. It is compatible with all browsers, scaling down automatically when the browser has a limitation. In some cases, that means the control turns off its client-side functionality or turns itself off entirely.

This product is designed to scale properly even when the Page's **ClientTarget** property causes the `HttpBrowserCapabilities (Request.Browser)` to falsely state the browser. In other words, you can't fool these controls with an upLevel clientTarget. This is absolutely necessary because feeding the wrong browser will generate incorrect client side scripts giving the user's scripting errors. It was also considered a requirement to hide features that didn't work on the browser to give the user the best interface. For more, see "Browser Support" in the **General Features Guide**.

## Terminology

In this documentation, the term **[DES installation folder]** refers to the folder where you installed **Peter's Data Entry Suite**. For example: C:\Program Files\Peters Data Entry Suite v5.0.0.

The term **[webapplicationroot]** refers to the folder that contains the web application on your server. For example, when using IIS, the domain <http://localhost> is usually in C:\inetpub\wwwroot. Web applications are usually in a subfolder. For example, the web app "MyWebApp" is in C:\inetpub\wwwroot\MyWebApp.

When using Visual Studios ASP.NET Development Server, it is usually **[My Documents]\Visual Studio 20##\Websites\[MyWebApp]**. For example, the web app "MyWebApp" could be in C:\documents and settings\*user name*\my documents\Visual Studio 2010\Websites\MyWebApp.

## Prepare a web application or site

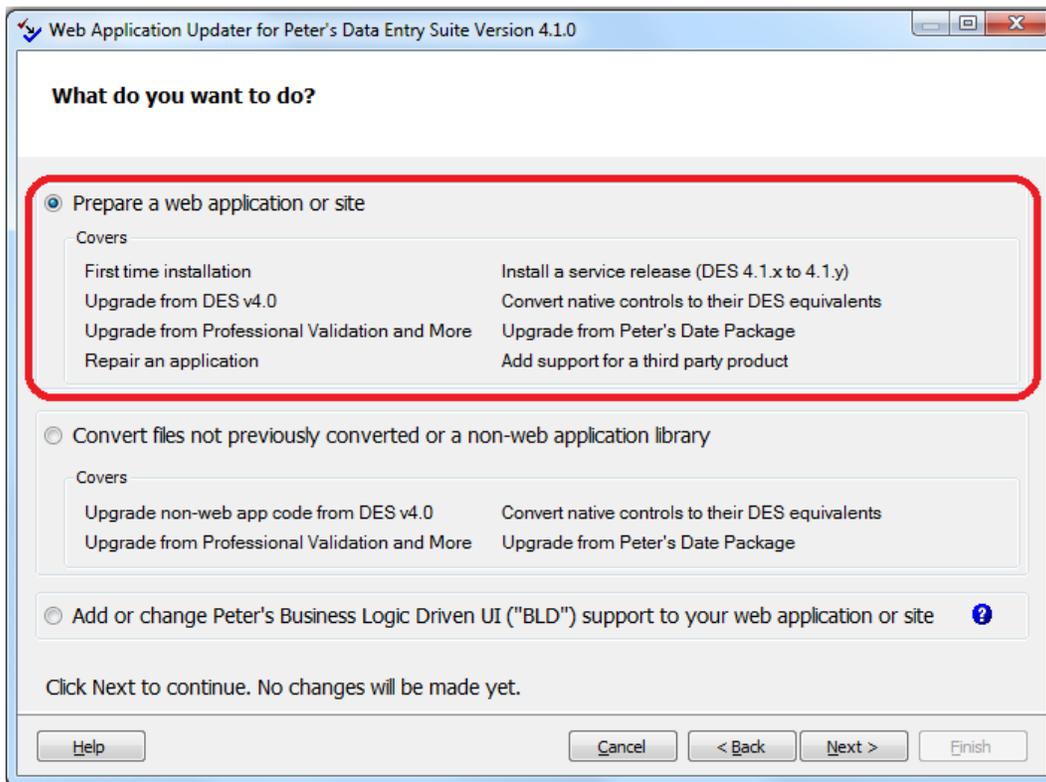
This process automatically modifies your web application using the **Web Application Updater** program. Use it for:

- First time installation
- Install a service release (from DES 5.0.x to 5.0.y)
- Upgrade from **DES v4.0**
- Convert native textboxes, validators, and buttons to their DES equivalents
- Upgrade from **Professional Validation And More** (any version)
- Upgrade from **Peter's Date Package** (any version)
- Repair an application
- Add validation support for a third party product

The **Web Application Updater** provides the following features that make it very safe to modify your web application: It has a very detailed log, creates a backup of every file it modifies, and offers a trial mode that logs the expected changes without actually making any changes. See "[FAQs - Prepare a web application or site](#)". If you still have concerns, email [support@peterblum.com](mailto:support@peterblum.com) with your questions.

### Prerequisites:

- You have created a web application, whether in Visual Studio, Visual Web Developer, or manually.
- The web application has a **web.config** file.
- The web application has a **global.asax** file.



*INSTRUCTIONS START ON THE NEXT PAGE*

## Step by step instructions

1. Exit from **Visual Studio** or **Visual Web Developer**, if opened.
2. Run the program **Web Application Updater.exe**, found in the [DES installation folder], or from the **Start** menu.
3. Click **Next**.
4. Select the first radio button, **Prepare a web application or site**.
5. Click **Next** and follow the remaining prompts.

**ALERT:** *If you want to convert from Peter's Date Package, read "[What you should know before](#)."* now

6. When finished, review the log and follow the information in the "What's Next" section.

### Problems?

The Log file provides extensive details to what was changed. Before contacting tech support, please try to work through the directions it provides. After all, you are most familiar with your web application. If problems still remain, please email [support@peterblum.com](mailto:support@peterblum.com) with a detailed description of the problem, the log file, and any web application files that are concerning you.

7. Install your trial or paid licenses. See "[Installing Licenses](#)".

### Problems?

See "[Troubleshooting Licenses](#)".

8. When using **Visual Studio** or **Visual Web Developer**, add the DES controls to the toolbox. See "[Adding to the Visual Studio/Visual Web Developer Toolbox](#)".
9. If you have upgraded from **Peter's Date Package**, see "[Converting from Peter's Date Package](#)". Your web application may require some edits to account for breaking changes and differences in style sheet files.
10. If you have upgraded from **Professional Validation and More**, see "[FAQs – Converting from Professional Validation and More](#)".

Your web application is ready to use.

### What first time users need to know

- When your page uses AJAX to update its controls, you **must** follow the steps described in the **General Features Guide** under the topic "Using These Controls With AJAX". *Remember this! It will help you avoid JavaScript errors, a lack of functionality, and the frustration of tracking down these problems later.*
- The String Lookup System is where you can define strings globally and use them for localization or standard textual templates throughout DES. For example, Validator control error messages can be more consistent when defined once in the String Lookup System. Consider setting it up now. Make decisions on where you will store your strings: resources files, a database or some other storage. That way as you define your controls, you simply add the messages into your storage. See the topic "The String Lookup System" in the **General Features Guide**.
- There are numerous global properties that expand how these controls work. You can define most of these properties in the **Global Settings Editor**. Consider running this utility now to get oriented to the global settings. As you work through the documentation, it will refer you to this utility and specific properties. See "Global Settings: The Editor and custom.DES.config File" in the **General Features Guide**.

### Need help getting started?

- **Getting Started.pdf** – The best place to get an orientation. This document is in the **[Program Files]** folder and in the **Start** menu.
- **Tutorials.pdf** – Step-by-step examples for validation and FieldStateControllers. This document is in the **[Program Files]** folder and in the **Start** menu.
- <http://samples.peterblum.com> – An interactive tour through most controls.

## FAQs - Prepare a web application or site

1. Can I install assemblies for ASP.NET 1.x?

DES 5.0 does not support ASP.NET 1.x. The previous version, DES 4.0, does. Contact tech support to get DES 4.0.

2. How can I install the assemblies into the Global Assembly Cache?

Run the **Web Application Updater**. It will prompt you to select the type of assembly to install: ASP.NET 2.0, ASP.NET 3.5, ASP.NET 4.0, or **None**. Select **None**.

Then install all of the assemblies found in the **[DES installation folder]\Assemblies\** folder into the Global Assembly Cache as follows:

### ASP.NET 4.0

**PeterBlum.DES.dll** from **[DES installation folder]\Assemblies\ASPNET 4.0**

**PeterBlum.DES.BLD.dll** from **[DES installation folder]\Assemblies\ASPNET 4.0**

**PeterBlum.DES.NativeToDES.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

**PeterBlum.DES.NativeValidators.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

### ASP.NET 3.5

**PeterBlum.DES.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

**PeterBlum.DES.BLD.dll** from **[DES installation folder]\Assemblies\ASPNET 3.5**

**PeterBlum.DES.NativeToDES.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

**PeterBlum.DES.NativeValidators.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

### ASP.NET 2.0

**PeterBlum.DES.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

**PeterBlum.DES.NativeToDES.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

**PeterBlum.DES.NativeValidators.dll** from **[DES installation folder]\Assemblies\ASPNET 2.0**

One typical problem with this kind of installation is that there can be file version conflicts with other assemblies that use the **PeterBlum.DES.dll**, including your own web application's assemblies. By using the **Web Application Updater**, you will have set up version redirection in the form of the `<assemblyBinding>` tag of the **web.config** file. This avoids the issue.

3. What has changed in my web application?

Assemblies were added to the **bin** folder. The **DES** folder was added to your web application; it contains many files for configuration, style sheets, images, and more.

The **web.config** file was modified.

If you elected to convert Microsoft controls, web forms and code behind files with those controls were changed.

When using a Visual Studio project file, it will have references to the new assemblies.

If you added support for third party controls, you may have additional assemblies and/or source code files.

Please see the log file. It provides extensive details.

4. Where is the log file?

A folder starting with "DES Conversion" followed by a date+time stamp was created side-by-side with your web application root folder. It contains **log.txt** which is the log file.

For example, if your web application folder is **C:\inetpub\wwwroot\myapp**, look for it in **C:\inetpub\wwwroot\DES Conversion yyyyymmdd hhmmss\log.txt**.

5. Can I undo the conversion?

Yes. If any changes were made, the same “DES Conversion” folder contains a backup of all changed files, including any assemblies that were replaced. You can either manually copy the desired file(s) or use the **Restore.BAT** file to let it restore all files. **Restore.BAT** is also in the “DES Conversion” folder. Just double-click to run it.

6. The log file does not tell exactly what changed inside of web forms and code files. How can I see what changed?

Use a file comparison program like [ExamDiff](#) from PrestoSoftware. The original files are located in the DES Conversion\Backup folder. Select the desired file from there and then select the file in your web application.

7. I don't like the default location of supporting files such as the images and style sheets. Can I change them?

Yes. See “[Using Alternative Locations for Files in the DES Folder](#)”.

8. I didn't convert Microsoft web controls to their DES equivalents. How do I convert it later?

See “[Convert Native Controls to their DES equivalents](#)”.

9. I have a non-web application library with code from **Peter's Date Package, Professional Validation and More**, or **DES 4.0**. How do I prepare that?

Run the **Web Application Updater** with the **Convert files not already converted or non-web application library** option.

Click **Next** and use that screen to select the files to convert.

Click **Next** and select the **Convert native controls to their DES equivalents** option on the second panel. You can still select which type of controls to convert. You can convert individual files, folders, or the complete web application.

Follow the remaining prompts.

10. I have **Professional Validation And More** installed in this web application and had it converted. Is there anything else I need to do?

Review “[FAQs – Converting from Professional Validation and More](#)”.

11. I have **Peter's Date Package** installed in this web application and had it converted. Is there anything else I need to do?

Yes. There are breaking changes. See “[Converting from Peter's Date Package](#)”.

12. Visual Studio/Visual Web Developer shows licensing errors on the DES controls when in design mode. What should I do?

First restart Visual Studio/Visual Web Developer. Often users add the licenses when these apps are opened, preventing them from being read in right away. If that does not work, make sure you have installed licenses (“[Installing Licenses](#)”) and review “[Troubleshooting Licenses](#)”.

13. How do I know what to deploy to my production server?

See “[Deploying Your Web Applications](#)”.

14. How can I confirm that key files are in place, especially after an update or deployment?

See “[Confirming a Deployment](#)”.

*CONTINUED ON THE NEXT PAGE*

15. Which controls were converted when using the **Convert Native Controls to their DES equivalents** option?

See "[Convert Native Controls to their DES equivalents](#)".

16. Are there any compatibility issues with ASP.NET that need additional setup?

If you are using Partial Trust security, see "[Installing into a Partial Trust Environment](#)".

If you are using IIS 7 and your **web.config** file does not have a `<system.webServer>` section, see "[Using IIS 7](#)".

If you are using FormsAuthentication, your **web.config** file now should have a `<location path="DES">` tag. This covers all files in the DES folder and its child folders, including images and style sheets. If you move any of the files or point to files outside of this folder, be sure that those files are accessible on pages not protected by FormsAuthentication.

17. I am using third party controls. Is there any additional setup required?

Most are fully setup if you marked them within the **Web Application Updater** wizard. See **Using Third Party Controls.pdf**. It covers all supported third party controls. It describes any special setup at the time you add a control.

Keep in mind that DES supports validation through menu and toolbar controls.

You may be using some third party data entry controls that were not listed while running the **Web Application Updater**. To make them work with DES, see the **Using Third Party Controls.pdf** guide.

## Adding to the Visual Studio/Visual Web Developer Toolbox

If you are using Visual Studio, add the controls from the **PeterBlum.DES.dll** assembly into the Visual Studio toolbox. It is recommended that you create a new tab in the toolbox called “Peter’s Data Entry Suite”. If you are going to use the controls of Peter’s TextBoxes or Peter’s Date And Time with the native Microsoft validators, add the controls from **PeterBlum.DES.NativeValidators.dll** into the Visual Studio toolbox. Give it a different tab, named “Native Validators for DES Controls”.

***IMPORTANT:** This will add all of the controls defined in PeterBlum.DES.dll. That is the entire suite. You will only be able to use controls for which you have licenses. Visual Studio’s toolbox has tools to organize the controls, including moving them to other tabs and deleting them. Please use the Visual Studio documentation for directions.*

The steps follow:

1. Open the **Toolbox** view. (**View; Toolbox**)
2. If you are attempting to replace the controls from an earlier service release, delete the existing tabs. Right click on each tab for this product and choose **Delete Tab**.
3. Right click in the Toolbox view and choose **Add Tab**.
4. Name the tab “Peter’s Data Entry Suite”.
5. Right click in the Toolbox view inside the new tab area. Select **Choose Items** and use the **.Net Framework Components** tab.
6. Click **Browse** and select the **PeterBlum.DES.dll** from the web application’s **\bin** folder. (Recommended.)  
*Note: See FAQ #2 below for why the \bin folder is preferred.*
7. Click **OK**. Over 60 controls are added to the “Peter’s Data Entry Suite” tab. If you are using the PeterBlum.DES.dll for ASP.NET 4.0, it will include controls for Peter’s Business Logic Driven UI.
8. If you are going to use the controls of Peter’s TextBoxes or Peter’s Date And Time with the native Microsoft validators, add the controls from **PeterBlum.DES.NativeValidators.dll** into the Visual Studio toolbox. Give it a different tab, named “Native Validators for DES Controls”.

*FAQS ARE ON THE NEXT PAGE*

## FAQs – Adding to the Toolbox

1. There are so many controls. How can I better organize my toolbox?

Visual Studio’s toolbox has tools to organize the controls, including moving them to other tabs and deleting them. Please use the Visual Studio documentation for directions. Here is a suggested approach by module. (The Peter’s Business Logic Driven UI module is in a separate assembly, PeterBlum.DES.BLL.dll which can be installed into its own tab.)

| Peter’s Data Entry Suite | DES Validators   | DES TextBoxes         |
|--------------------------|--|-----------------------|
| Button                   | All validators  | CurrencyTextBox       |
| ImageButton              | CombinedErrorMessages  | DecimalTextBox        |
| LinkButton               | RequiredFieldMarker  | FilteredTextBox       |
| LocalizableLabel         | RequiredFieldsDescription  | IntegerTextBox        |
| NativeControlExtender    | ValidationSummary  | MultiSegmentDataEntry |
| PageManager              |  | PercentTextBox        |
| HtmlList                 |  | TextBox               |

| DES Date and Time    | DES Interactive Pages     |
|----------------------|---------------------------|
| AnniversaryTextBox   | CalculationController     |
| Calendar             | ContextMenu               |
| DateTextBox          | DropDownMenu              |
| DurationTextBox      | FieldStateController      |
| MonthYearPicker      | FSCOnCommand              |
| MonthYearTextBox     | MultiFieldStateController |
| PopupCalendar        | MultiFSCOnCommand         |
| PopupMonthYearPicker | TextCounter               |
| PopupTimePicker      |                           |
| QuickDateMenu        |                           |
| SpecialDates         |                           |
| TimeOfDayTextBox     |                           |
| TimePicker           |                           |

*Note: While the items in the Peter’s Data Entry Suite group may be better located in multiple groups (DES Validators and Peter’s Interactive Pages in particular), it appears that Visual Studio only permits one copy of a control per assembly.*

2. Why add the DES product assemblies from the \bin folder instead of the [DES installation folder]?

Each time you drop a control from its toolbox, Visual Studio 2005, 2008, 2010 and Visual Web Developer attempts to access the assembly in the path you give it here. The **\bin** folder provides a more stable location for the **PeterBlum.DES.dll** than the **[DES installation folder]**, which may change with a service release or product reinstallation. In addition, Visual Studio 2008 appears to be more version sensitive, preferring the assembly in the toolbox to have the same version as the assembly in the **\bin** folder.

## Using Alternative Locations for Files in the DES Folder

By default, each web application has its own folder and subfolders for DES's licenses, appearance folder, scripts and config files. It is the **[webapplicationroot]\DES** folder. You can choose alternative names and locations for any of these folders. To do this, you must update your **web.config** file with entries in the <appSettings> section.

*Note: The .Net framework uses the tilde (~) character to indicate the Application root path. It should be used when you have an IIS application defined under the domain. For example, http://localhost is your domain and you have app1/ under it, use "~/folderofapp1/file.aspx" and .Net will use the virtual path "/app1/folderofapp1/file.aspx".*

The VirtualPath keys below all support the tilde notation so long as it's the first character of the virtual path string.

There are six keys that you can setup here:

- **DES\_LicenseVirtualPath** – The value is the virtual path to the Licenses folder, which contains your license files. When not specified, it defaults to **[webapplicationroot]\DES\Licenses**. ASP.NET 2.0+ users must use the tilde (~) notation, as described in the note below. This string cannot start with a lead slash.
- **DES\_LicenseFilePath** – This is an alternative to **DES\_LicenseVirtualPath**. It provides a real file path (C:\folder\folder2) to the Licenses folder. *Remember that your web server Windows NT account must have network rights to read from the folder specified here.*
- **DES\_AppearanceVirtualPath** – This is a virtual path to the Appearance folder that contains the default image and style sheet files. When not specified, it defaults to **[webapplicationroot]\DES\Appearance**.

This setting only affects properties that start with "{APPEARANCE}", a token that uses this key. For example, the DateTextBox uses the URL "{APPEARANCE}/Date and Time/Calendar.jpg" in its **DateTextBox.PopupCalendar.ToggleImageUrl** property.

It also affects the default location of style sheet files. You can override the default location and filename for each style sheet as well. See "Using Style Sheets" in the **General Features Guide**.

**ALERT:** If you specify a path starting with http:// or https://, you must also modify settings associated with the style sheets. By default, style sheets are read from disk to be merged and compressed. Without a file path (not a internet URL), DES cannot read it. You need to use one of the following options:

**ALERT:** When your site uses Forms Authentication, be sure that the folder specified here always allow access without requiring authentication. Generally you will add a <location path="url"> tag to the web.config file. See "[When Forms Authentication Is Used, Add This](#)".

### Abandon Merge and Compression Feature

You can turn off the compression and merging feature so that individual style sheet files are added to the page. That way, the browser will apply the http:// or https:// based URL. Add this line to the <appSettings> section of the **web.config** file:

```
<add key="DES_StyleSheetCompression" value="SeparateFiles" />
```

See the "Compressing and Merging Files: The GetFiles.aspx File" section of the **General Features Guide** for details.

### Specify file-based URLs to each style sheet file

You can provide alternative paths to individual style sheet files. Those paths must refer to a real file, either using the virtual path syntax of "~/folder/" or the file:// header to the file. You will add lines to the <appSettings> section of the **web.config** file that identify the location of each style sheet file.

See the "Customizing the URLs to each Style Sheet File" section of the **General Features Guide**.

Examples:

```
<add key="DES_StyleSheetCalendar" value="~/DES/Date And Time/Calendar.css" />
<add key="DES_StyleSheetCalendar"
value="file://mycomputer/myapp/DES/Date And Time/Calendar.css" />
```

- **DES\_ConfigFilePath** – The file path to the **DES.config** and **custom.DES.config** files, which configure DES both in design mode and at runtime. **Peter's Input Security** will use this to find its **Security Config Files** folder, which

should be in the folder specified here. Remember that your web server Windows NT account must have network rights to read from the folder specified here.

- `DES_ConfigVirtualPath` – This is an alternative to `DES_ConfigFilePath`. The value is a virtual path to the folder that contains the **DES.config** and **custom.DES.config** files. ASP.NET 2.0+ users must use the tilde (~) notation, as described in the note below. This string cannot start with a lead slash.

**ALERT:** When your site uses Forms Authentication, be sure that the folder specified here always allow access without requiring authentication. Generally you will add a `<location path="url">` tag to the web.config file. See [“When Forms Authentication Is Used, Add This”](#).

- `DES_GetFilesVirtualPath` – The full URL to the **GetFiles.aspx** file. The **GetFiles.aspx** file is located in the **[webapplicationroot]\DES** folder by default. It provides merged and compressed versions of scripts and style sheet files to a page for ASP.NET 1.x users and for users who elect to use it instead of the HttpHandler built into DES. In these cases, it is an essential element to DES’s output.

ASP.NET 2 users switch from using the HttpHandler to the **GetFiles.aspx** file by including this key, usually with the value of `“~/DES/GetFiles.aspx”`. Do NOT use this if you want the HttpHandler to deliver files.

If DES is not loading style sheets and scripts, it may mean the default URL is being ignored by your web server. You can relocate the **GetFiles.aspx** file to another location, such as the web application root. Always specify the filename too, such as `“~/GetFiles.aspx”`. ASP.NET 2.0+ users must use the tilde (~) notation, as described in the note below. This string cannot start with a lead slash.

**ALERT:** When your site uses Forms Authentication, be sure that the folder specified here always allow access without requiring authentication. Generally you will add a `<location path="url">` tag to the web.config file. See [“When Forms Authentication Is Used, Add This”](#).

### Example

```
<configuration>
  <appSettings >
    <add key="DES_LicenseVirtualPath" value="~/Private/Licenses/" />
    <add key="DES_AppearanceVirtualPath" value="~/MyImages/" />
    <add key="DES_ConfigFilePath" value="C:\inetpub\wwwroot\ConfigFiles\" />
  </appSettings>
</configuration>
```

### When Forms Authentication Is Used, Add This

```
<configuration>
  <location path="MyImages">
    <system.web>
      <authorization>
        <allow users="*" />
      </authorization>
    </system.web>
  </location>
  <location path="ConfigFiles">
    <system.web>
      <authorization>
        <allow users="*" />
      </authorization>
    </system.web>
  </location>
</configuration>
```

## Installing Licenses

When you downloaded the product, you should have received one or more License Files. If you have none, it will work in a trial mode. See “[If you have NOT received a Trial License file](#)”. You also received a serial number if you purchased a license. (The Trial Version doesn’t issue a serial number.)

There are two actions to get license files to work:

- Place the License Files into the **[webapplicationroot]\DES\Licenses** folder or its subfolder on each web server.
- Setup the License Key global variable in `Application_Start()` or `Page_Load()` methods. *Trial version users do not take this action.*

After installation and deployment to another server, you can review the current licenses configuration. See “[Exploring the Current Settings](#)”.

If you have problems, see “[Troubleshooting Licenses](#)”.

**Click on your license type for instructions:**

|  |                                    |                                |
|--|------------------------------------|--------------------------------|
| <a href="#">Trial License</a>                        | <a href="#">Consultant License</a> |                                |
| <b><u>Suite (Complete Product)</u></b>               |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s Professional Validation Module</u></b> |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s More Validators Module</u></b>         |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s TextBoxes Module</u></b>               |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s Date and Time Module</u></b>           |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s Interactive Pages Module</u></b>       |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |
| <b><u>Peter’s Input Security Module</u></b>          |                                    |                                |
| <a href="#">Web Server</a>                           | <a href="#">Site</a>               | <a href="#">Redistribution</a> |

## ***Trial License***

### **If you have received a Trial License file**

The Trial License file gives you a 30 day time limit to evaluate the entire **Peter's Data Entry Suite**. You can use any of the controls described throughout this package.

1. Place the license file **DES Trial expires on yyyy-mm-dd.lic** into the **[webapplicationroot]\DES\Licenses** folder. Recommended only for non-production servers (development, testing, and staging). You can put it on production (live) servers so long as you accept the limitations.
2. That's it.

Runtime Limitations: Allows only the first 15 unique IP addresses to access DES controls per application startup; expires 30 days after you requested it. If either limit is hit, an exception is thrown.

### **If you have NOT received a Trial License file**

DES will run without any license file. You can use any of the controls described throughout this package.

Each page will display text like this ( **DES-License file has not been setup** ) below most of your HTML. If it is in the way, click on it to hide it. (It will return the next time the page is refreshed.) If you do not want the small graphic, you can get a time limited Trial Version license file at <http://www.peterblum.com/des/trial.aspx>. It will demand a valid email address as the file is emailed to you.

Runtime Limitations: Allows only the first 2 unique IP addresses to access DES controls per application startup. If the limit is hit, an exception is thrown.

*Reasons you did not get a license file:*

- The email with the file is in your junk mail folder
- You entered an incorrect email address. This is common.
- When using a general email service like yahoo, gmail, or msn, your email address has been discontinued.

## ***Consultant License***

The Consultant License is designed for non-production server use.

1. Place the license file **DES Consultant.lic** into the **[webapplicationroot]\DES\Licenses** folder.
2. That's it.

Runtime Limitations: Allows only the first 15 unique IP addresses to access DES controls per application startup. If the limit is hit, an exception is thrown.

## Suite – Web Server Licenses

Web Server Licenses come with two types of files: **DES Suite Development.lic** and **DES Suite SingleServer####.lic**. You must have one **DES Suite SingleServer####.lic** for each production server. **DES Suite Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES Suite SingleServer####.lic** files. You will only have one **DES Suite Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES Suite Development.lic** and all **DES Suite SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses** folder.
2. Gather the product serial numbers from the emails containing the **DES Suite SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application’s **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES Suite SingleServer####.lic** files to find a match. If there was no matching computer name, it uses **DES Suite Development.lic**. **DES Suite Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.Suite\_LicenseKey**.

```
PeterBlum.DES.Globals.Suite_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.Suite_LicenseKey = "999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Suite – Site License

The Site License provides a single license file with this name: **DES Suite Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES Suite Site####.lic** into the **[webapplicationroot]\DES\Licenses** folder.
2. Gather the product serial number from the email containing the **DES Suite Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the Application\_Start() method. If your **Global.asax** file contains the method Application\_OnStart(), see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to Application\_Start() that assigns your serial number to **PeterBlum.DES.Globals.Suite\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.Suite_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Suite_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the Application\_Start() method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Suite – Redistribution License

The Redistribution License provides a single license file with this name: **DES Suite R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES Suite R####.lic** into the **[webapplicationroot]\DES\Licenses** folder.
2. Gather the product serial number from the email containing the **DES Suite R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.Suite\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.Suite_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.Suite_LicenseKey = "999-111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Professional Validation – Web Server Licenses

Web Server Licenses come with two types of files: **DES PV Development.lic** and **DES PV SingleServer####.lic**. You must have one **DES PV SingleServer####.lic** for each production server. **DES PV Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES PV SingleServer####.lic** files. You will only have one **DES PV Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES PV Development.lic** and all **DES PV SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\Professional Validation** folder.
2. Gather the module serial numbers from the emails containing the **DES PV SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES PV SingleServer####.lic** files to find a match. If there was no matching computer name, it uses **DES PV Development.lic**. **DES PV Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.ProfessionalValidation\_LicenseKey**.

```
PeterBlum.DES.Globals.ProfessionalValidation_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.ProfessionalValidation_LicenseKey = "999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Professional Validation – Site License

The Site License provides a single license file with this name: **DES PV Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES PV Site####.lic** into the **[webapplicationroot]\DES\Licenses\Professional Validation** folder.
2. Gather the module serial number from the email containing the **DES PV Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the Application\_Start() method. If your **Global.asax** file contains the method Application\_OnStart(), see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to Application\_Start() that assigns your serial number to **PeterBlum.DES.Globals.ProfessionalValidation\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.ProfessionalValidation_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.ProfessionalValidation_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the Application\_Start() method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Professional Validation – Redistribution License

The Redistribution License provides a single license file with this name: **DES PV R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES PV R####.lic** into the **[webapplicationroot]\DES\Licenses\Professional Validation** folder.
2. Gather the module serial number from the email containing the **DES PV R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.ProfessionalValidation\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.ProfessionalValidation_LicenseKey =  
"serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.ProfessionalValidation_LicenseKey =  
"999-111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's More Validators – Web Server Licenses

Web Server Licenses come with two types of files: **DES MV Development.lic** and **DES MV SingleServer####.lic**. You must have one **DES MV SingleServer####.lic** for each production server. **DES MV Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES MV SingleServer####.lic** files. You will only have one **DES MV Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES MV Development.lic** and all **DES MV SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\More Validators** folder.
2. Gather the module serial numbers from the emails containing the **DES MV SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES MV SingleServer####.lic** files to find a match. If there was no matching computer name, it uses

**DES MV Development.lic**. **DES MV Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.MoreValidators\_LicenseKey**.

```
PeterBlum.DES.Globals.MoreValidators_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.MoreValidators_LicenseKey = "999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's More Validators – Site License

The Site License provides a single license file with this name: **DES MV Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES MV Site####.lic** into the `[webapplicationroot]\DES\Licenses\More Validators` folder.
2. Gather the module serial number from the email containing the **DES MV Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to `Application_Start()` that assigns your serial number to **PeterBlum.DES.Globals.MoreValidators\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.MoreValidators_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.MoreValidators_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's More Validators – Redistribution License

The Redistribution License provides a single license file with this name: **DES MV R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES MV R####.lic** into the **[webapplicationroot]\DES\Licenses\More Validators** folder.
2. Gather the module serial number from the email containing the **DES MV R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.MoreValidators\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.MoreValidators_LicenseKey =  
"serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.MoreValidators_LicenseKey = "999-  
111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's TextBoxes – Web Server Licenses

Web Server Licenses come with two types of files: **DES TB Development.lic** and **DES TB SingleServer####.lic**. You must have one **DES TB SingleServer####.lic** for each production server. **DES TB Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES TB SingleServer####.lic** files. You will only have one **DES TB Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES TB Development.lic** and all **DES TB SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\TextBoxes** folder.
2. Gather the module serial numbers from the emails containing the **DES TB SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-
1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES TB SingleServer####.lic** files to find a match. If there was no matching computer name, it uses

**DES TB Development.lic**. **DES TB Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.TextBoxes\_LicenseKey**.

```
PeterBlum.DES.Globals.TextBoxes_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.TextBoxes_LicenseKey =
"999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's TextBoxes – Site License

The Site License provides a single license file with this name: **DES TB Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES TB Site####.lic** into the **[webapplicationroot]\DES\Licenses\TextBoxes** folder.
2. Gather the module serial number from the email containing the **DES TB Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the Application\_Start() method. If your **Global.asax** file contains the method Application\_OnStart(), see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to Application\_Start() that assigns your serial number to **PeterBlum.DES.Globals.TextBoxes\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.TextBoxes_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.TextBoxes_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the Application\_Start() method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's TextBoxes – Redistribution License

The Redistribution License provides a single license file with this name: **DES TB R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES TB R####.lic** into the **[webapplicationroot]\DES\Licenses\TextBoxes** folder.
2. Gather the module serial number from the email containing the **DES TB R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.TextBoxes\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.TextBoxes_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.TextBoxes_LicenseKey = "999-111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Date And Time – Web Server Licenses

Web Server Licenses come with two types of files: **DES DT Development.lic** and **DES DT SingleServer####.lic**. You must have one **DES DT SingleServer####.lic** for each production server. **DES DT Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES DT SingleServer####.lic** files. You will only have one **DES DT Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES DT Development.lic** and all **DES DT SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\Date and Time** folder.
2. Gather the module serial numbers from the emails containing the **DES DT SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES DT SingleServer####.lic** files to find a match. If there was no matching computer name, it uses

**DES DT Development.lic**. **DES DT Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.DateAndTime\_LicenseKey**.

```
PeterBlum.DES.Globals.DateAndTime_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.DateAndTime_LicenseKey = "999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Date And Time – Site License

The Site License provides a single license file with this name: **DES DT Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES DT Site####.lic** into the **[webapplicationroot]\DES\Licenses\Date and Time** folder.
2. Gather the module serial number from the email containing the **DES DT Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the Application\_Start() method. If your **Global.asax** file contains the method Application\_OnStart(), see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to Application\_Start() that assigns your serial number to **PeterBlum.DES.Globals.DateAndTime\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.DateAndTime_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.DateAndTime_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the Application\_Start() method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Date And Time – Redistribution License

The Redistribution License provides a single license file with this name: **DES DT R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES DT R####.lic** into the **[webapplicationroot]\DES\Licenses\Date and Time** folder.
2. Gather the module serial number from the email containing the **DES DT R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.DateAndTime\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.DateAndTime_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.DateAndTime_LicenseKey = "999-111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Interactive Pages – Web Server Licenses

Web Server Licenses come with two types of files: **DES IP Development.lic** and **DES IP SingleServer####.lic**. You must have one **DES IP SingleServer####.lic** for each production server. **DES IP Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES IP SingleServer####.lic** files. You will only have one **DES IP Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES IP Development.lic** and all **DES IP SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\Interactive Pages** folder.
2. Gather the module serial numbers from the emails containing the **DES IP SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES IP SingleServer####.lic** files to find a match. If there was no matching computer name, it uses **DES IP Development.lic**. **DES IP Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.InteractivePages\_LicenseKey**.

```
PeterBlum.DES.Globals.InteractivePages_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.InteractivePages_LicenseKey = "999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Interactive Pages – Site License

The Site License provides a single license file with this name: **DES IP Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES IP Site####.lic** into the `[webapplicationroot]\DES\Licenses\Interactive Pages` folder.
2. Gather the module serial number from the email containing the **DES IP Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to `Application_Start()` that assigns your serial number to **PeterBlum.DES.Globals.InteractivePages\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.InteractivePages_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.InteractivePages_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Interactive Pages – Redistribution License

The Redistribution License provides a single license file with this name: **DES IP R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES IP R####.lic** into the **[webapplicationroot]\DES\Licenses\Interactive Pages** folder.
2. Gather the module serial number from the email containing the **DES IP R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.InteractivePages\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.InteractivePages_LicenseKey =  
"serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.InteractivePages_LicenseKey = "999-  
111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Input Security – Web Server Licenses

Web Server Licenses come with two types of files: **DES IS Development.lic** and **DES IS SingleServer####.lic**. You must have one **DES IS SingleServer####.lic** for each production server. **DES IS Development.lic** is used on unlimited non-production servers.

If you have purchased more than one Web Server License, you will have several **DES IS SingleServer####.lic** files. You will only have one **DES IS Development.lic**.

*Note: Use these steps whether you are working on a non-production (development, testing, staging) or production server. The steps are designed to set up the licensing once, on the non-production server. Then as you deploy to other servers, it will continue to work.*

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license files **DES IS Development.lic** and all **DES IS SingleServer####.lic** into the **[webapplicationroot]\DES\Licenses\Input Security** folder.
2. Gather the module serial numbers from the emails containing the **DES IS SingleServer####.lic** files. You should have one for each Web Server License purchased.
3. Gather the computer names of each production server.

On Windows NT and XP, it is available on the System control panel under the “Computer Name” tab. If you are using a hosted server, email [support@PeterBlum.com](mailto:support@PeterBlum.com) to request an easy way to get the computer name for the Web Server License. Please provide your serial number. (While you wait, you can use a temporary name so that you can move forward. Your license will work on non-production servers.)

*HINT: The most common error users make is to get the wrong computer name. Often they use a domain name or change the computer. Both will not match to the computer name.*

*Note: If you do not know the name of the production server at this time, use a fake name. Perhaps one to remind you to fill it in later like “AssignToProductionServerName”. Remember to update this prior to deploying to production.*

4. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) If you do not have a *Global.asax* file, see “[When the Web Application does not have a Global.asax file](#)”.
5. Locate the `Application_Start()` method. If your **Global.asax** file contains the method `Application_OnStart()`, see “[Application\\_Start vs. Application\\_OnStart](#)”. If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

6. Determine your **License Key**. It will be used in `Application_Start()` in the next step.

- For one Web Server License, it is this format:

```
serialnumber | computername
```

A pipe character separates the two.

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
999-111222333 | MarsServer
```

- For two to four Web Server Licenses, it is this format:

```
serialnumber | computername ; serialnumber2 | computername2 ; serialnumber3 | comput
ername3 ; serialnumber4 | computername4
```

The delimiter after the serial number is a pipe (“|”). The delimiter after the computer name is a semi-colon (“;”)

You can put serial numbers and computer names in any order so long as each serial number and computer name is unique.

For example, with 3 Web Server licenses:

```
999-1100000001|MarsServer;999-1100000002|JupiterServer; 999-
1100000003|SaturnServer
```

- For five Web Server Licenses, you are granted a Site License. Your License Key is one serial number (no computer name either). You can pick any serial number of the 5.

DES selects which of the license files will be active based on the License Key. It locates a serial number by matching the computer name to the names you’ve entered. With that serial number, it looks through your

**DES IS SingleServer####.lic** files to find a match. If there was no matching computer name, it uses

**DES IS Development.lic**. **DES IS Development.lic** allows only the first 15 computers to access DES controls as it’s designed for non-production computers.

7. Add a line to `Application_Start()` that assigns your License Key to **PeterBlum.DES.Globals.InputSecurity\_LicenseKey**.

```
PeterBlum.DES.Globals.InputSecurity_LicenseKey = "License Key"
```

For example, the serial number is 999-111222333 and computer name is “MarsServer”:

```
PeterBlum.DES.Globals.InputSecurity_LicenseKey =
"999-111222333|MarsServer"
```

8. Restart your web application so it reruns the `Application_Start()` method.

Please see [“Troubleshooting Licenses”](#) if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Input Security – Site License

The Site License provides a single license file with this name: **DES IS Site####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES IS Site####.lic** into the **[webapplicationroot]\DES\Licenses\Input Security** folder.
2. Gather the module serial number from the email containing the **DES IS Site####.lic** file.
3. Open your web application's **Global.asax** file to its code view. (In Visual Studio 2002 and 2003, this is the code behind file: **Global.asax.cs** or **Global.asax.vb**) *If you do not have a Global.asax file, see "[When the Web Application does not have a Global.asax file](#)".*
4. Locate the Application\_Start() method. If your **Global.asax** file contains the method Application\_OnStart(), see "[Application\\_Start vs. Application\\_OnStart](#)". If it is missing, add it like this:

[C#]

```
protected void Application_Start(Object sender, EventArgs e)
{
}
```

[VB]

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
End Sub
```

5. Add a line to Application\_Start() that assigns your serial number to **PeterBlum.DES.Globals.InputSecurity\_LicenseKey**. This is called the **License Key**.

```
PeterBlum.DES.Globals.InputSecurity_LicenseKey = "serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.InputSecurity_LicenseKey = "999-111222333"
```

6. Restart your web application so it reruns the Application\_Start() method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## Peter's Input Security – Redistribution License

The Redistribution License provides a single license file with this name: **DES IS R####.lic**. It can be used on unlimited production and non-production servers as described in the License Agreement.

*Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see [this troubleshooting topic](#) first.*

1. Place the license file **DES IS R####.lic** into the **[webapplicationroot]\DES\Licenses\Input Security** folder.
2. Gather the module serial number from the email containing the **DES IS R####.lic** file.
3. For each web form or user control that uses DES, you must add the **License Key** to the `Page_Load()` method. Add a line to `Page_Load()` that assigns your serial number to **PeterBlum.DES.Globals.WebFormDirector.InputSecurity\_LicenseKey**.

```
PeterBlum.DES.Globals.WebFormDirector.InputSecurity_LicenseKey =  
"serialnumber"
```

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.WebFormDirector.InputSecurity_LicenseKey = "999-  
111222333"
```

*Note: When adding DES to a page for the first time, the steps to add a DES control remind you to do this.*

4. Restart your web application so it reruns the `Application_Start()` method.

Please see "[Troubleshooting Licenses](#)" if you receive licensing exception errors when you access a page that uses DES controls.

## ***Application\_Start vs. Application\_OnStart***

The .net framework defines two method names that can initialize your application within the **Global.asax** file: `Application_Start()` and `Application_OnStart()`. Your web application may be using either but cannot use both.

This documentation will always refer to `Application_Start()`. You should substitute `Application_OnStart` if that is the method name you have defined in **Global.asax**.

## ***When the Web Application does not have a Global.asax file***

When setting up a Web Server or Site license, you are instructed to use the `Application_Start()` method in the **Global.asax** file. This topic helps you when you do not have the **Global.asax** file in your web application.

You have several choices.

### **Using Visual Studio (any version)**

You can add it by following these steps:

- Open Solution Explorer
- Right click on the web application root node and select **Add New Item**.
- Select the icon “Global Application Class”.

### **Using files supplied with DES**

DES supplies versions of the **Global.asax** file for C# and VB.net applications. Locate the right one in the **[DES installation folder]\When Missing Global.asax** folder for the language you use. Add it to the web application root folder.

### **Setting up licensing on each page instead of Global.asax**

This technique is an excellent workaround when you cannot use the **Global.asax** file solution. Add the `LicenseKey` line of code as the first line of the `Page_Load()` method. It is the same line of code you would add to **Global.asax**.

Make sure you add it to every page that contains DES controls.

## Troubleshooting Licenses

If you run into errors when the page is rendered, the error message should direct you to the problem. See “[Licensing Error Messages](#)” for more.

**HELPFUL HINT:** You can review the current licenses configuration to assist you. See “[Exploring the Current Settings](#)”. This feature displays exactly the license files it has detected and is using. It also tells you what licenses are required by the controls and features used on this page.

There are two steps to setting up the licenses correctly:

1. Add the license files to the **[webapplicationroot]\DES\Licenses** folder or subfolder on each computer.

Common errors in this step include:

- Omitting a license file. If you are missing a license file, visit <http://www.peterblum.com/LicensedDownloads/RetrieveLicense.aspx> to get a new copy.
- Placing the file in the wrong folder.

| <i>Files that start with</i> | <i>Belong in this folder below [webapplicationroot]\DES</i> |
|------------------------------|---|
| DES Suite                    | <b>\Licenses</b>  |
| DES PV                       | <b>\Licenses\Professional Validation</b>                    |
| DES MV                       | <b>\Licenses\More Validators</b>                            |
| DES TB                       | <b>\Licenses\Text Boxes</b>                                 |
| DES DT                       | <b>\Licenses\Date and Time</b>                              |
| DES IP                       | <b>\Licenses\Interactive Pages</b>                          |
| DES IS                       | <b>\Licenses\Input Security</b>                             |

2. Assign the Licensing Key to the **PeterBlum.DES.Globals.modulename\_LicenseKey** or **PeterBlum.DES.WebFormDirector.modulename\_LicenseKey** property.

Common errors in this step include:

- With a Web Server license, using the computer name of a non-production server in the License Key. It must be the name of the production server.
- With a Web Server license, getting the wrong text for the computer name in the License Key. For example, the domain name. The software checks the computer name by checking the property `System.Environment.ComputerName`. You may want to test this property yourself to get the string.
- `Application_Start()` is not being called when your application starts. Usually this is because the **Global.asax** file has not been deployed. ASP.NET depends on this file to be present before it runs any methods in the Global class (which includes `Application_Start`.) If this fails to solve the problem, tech support recommends this solution:

Copy the License Key code from `Application_Start()` into the `Page_Load()` of a page that is using DES controls. Compile and test that page. If it works, `Application_Start()` is broken. Repeat copying the License Keys into all page’s `Page_Load()` methods that use DES controls.

ASP.NET 2.0 and higher users sometimes have this problem because the code behind file (**Global.asax.vb** or **Global.asax.cs**) that was created in ASP.NET 1.x was moved into the **App\_Code** folder during the upgrade to 2.0. Move that file into the **[web application folder]**.

- Leaving the `modulename_LicenseKey` property blank.
- On a Trial or Consultant License, assigning a value to the `modulename_LicenseKey` property. It should be blank.
- Forgetting to compile your web application (specifically the **Global.asax** file) so that changes are reflected in the web application.

- Not restarting the web application.
- With a Redistribution license, forgetting to assign the License Key to **PeterBlum.DES.Globals.WebFormDirector.modulename\_LicenseKey** property on each `Page_Load()` method that uses these controls or forgetting to compile.
- You added the `Application_Start()` method to your **Global.asax** file when it already includes the `Application_OnStart()` method. ASP.NET only runs `Application_OnStart()` when both are defined. So move the contents of `Application_Start()` into `Application_OnStart()`.

See also "[Licensing Error Messages](#)".

## Licensing Error Messages

Here are the error messages you may encounter and how to resolve them. If a problem is related to a licensing file, identify the server that generated the error first. Errors may be specific to a server. Don't forget that after you change the contents of the Licenses folder, you must wait 1 minute (after an error message) for the cache to clear or restart the web application.

**HELPFUL HINT:** You can review the current licenses configuration to assist you. See [“Exploring the Current Settings”](#). This feature displays exactly the license files it has detected and is using. It also tells you what licenses are required by the controls and features used on this page.

### **Could not match the License Key [license key] to the [module name] license files in [filepath].**

The value of `PeterBlum.DES.Globals.modulename_LicenseKey` or `PeterBlum.DES.Globals.WebFormDirector.modulename_LicenseKey` does not match to the license files found in the file path.

See the two steps of [“Troubleshooting Licenses”](#), above.

### **Your non-production server license [path] has reached its limit of 15 unique IP Addresses accessing it.**

The `PeterBlum.DES.Globals.modulename_LicenseKey` property matched to a Trial or Development license file. Both of these files impose a limit of 15 unique browser IP Addresses as they are intended for internal development and product evaluation. The first 15 IP addresses are cached. Your options are:

- If you are using a Web Server license, the License Key string contains the wrong computer name. In `Global.asax`, locate `Application_Start()` and review the License Keys. Correct any that is not using a production server's computer name. Recompile and deploy.
- If you are running the Trial version, do not run it on a server that has published your site.

When a page uses a Trial or Development license, it will add this text near the `</form>` tag:

```
<!-- DESLIC: Suite=TRIAL; [modulename]=DEVELOPMENT -->
```

When determining if you fixed the problem, view the HTML source of the page to see this tag. If it is missing or the module with the error is not shown, the license has been corrected for use on production servers.

### **Access to the path "filepath" is denied.**

While the license file may be completely accessible to your IIS Web Server, when DES must read from the file, it must go through the Windows operating system to open the file. Windows has its own security restrictions on files and folders. Make sure the web application has Windows operating system rights to read files from the given filepath.

### **There are no valid [module name] license files found in [path].**

Cannot find any license files in the correct license folder. See step 1 of [“Troubleshooting Licenses”](#).

### **The License Key is blank for [module name]. It must be setup.**

The `modulename_LicenseKey` property has not been setup or the `Application_Start` method may not be called when the application starts up.

See step 2 of [“Troubleshooting Licenses”](#).

### **The file path [filepath] could not be found**

You are missing the `DES\Licenses` folder within your web application's folder or the folder specified by the `web.config` file in the `DES_LicenseFilePath` or `DES_LicenseVirtualPath` entries of `<appSettings>`. See [“Using Alternative Locations for Files in the DES Folder”](#).

### **The serial number string contains duplicate codes. It must contain unique codes.**

When you have multiple serial numbers in the `modulename_LicenseKey` property, if there are any duplicates, this error occurs.

***The serial number [serial number] matched to a Web Server license. A Web Server license requires the registration code to be followed by '|' and the computer name that runs the web server for this license. Please add it.***

The serial number is associated with a Single Server license. You have not provided the computer name in the *modulename\_LicenseKey* property. Please see step 2 of "[Troubleshooting Licenses](#)".

***The Trial Version License expired on [date]***

Your Trial version license has expired. Please visit [www.PeterBlum.com](http://www.PeterBlum.com) to order the product.

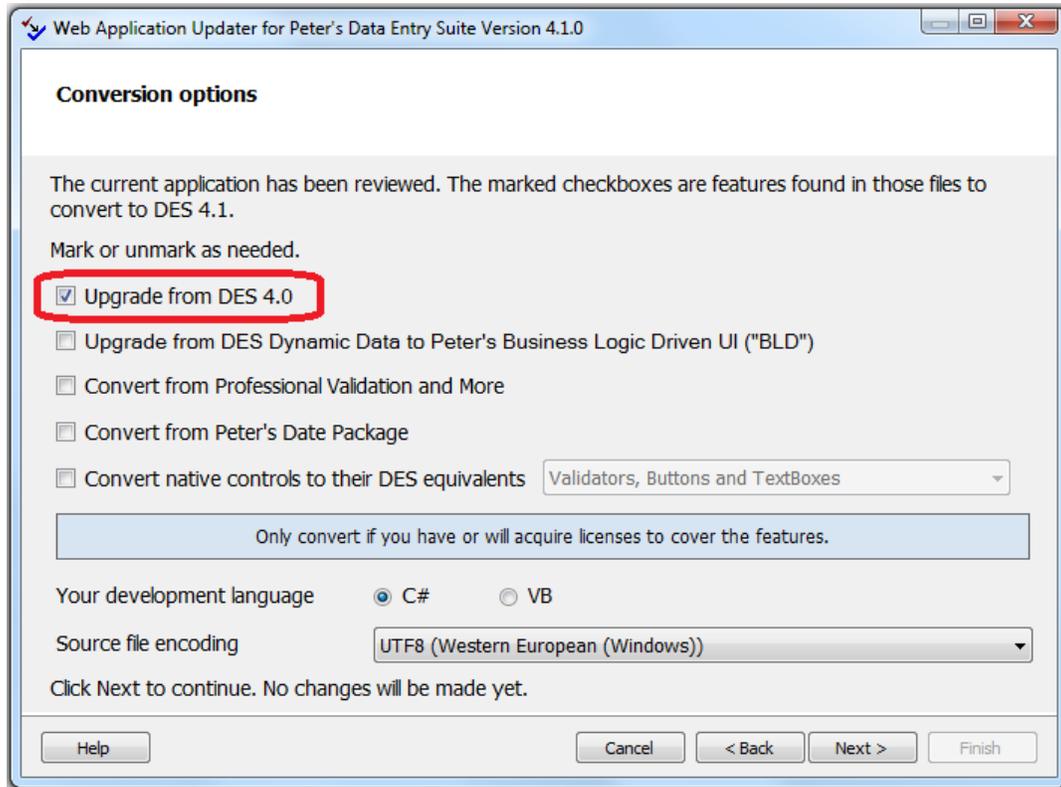
## Converting from Trial Version To Paid Version

1. Update the License Files and License Key. See "[Installing Licenses](#)".
2. Run the **Web Application Updater** with the **Prepare a web application or site** option if the product you downloaded with the paid version is newer. Use "[Prepare a web application or site](#)".

## Upgrading from DES v4.0

Follow the instructions of “[Prepare a web application or site](#)”. The **Web Application Updater** should detect the presence of DES 4.0 features and mark the **Upgrade from DES 4.0** option on the **Conversion Options** panel.

If you also were using DES Dynamic Data, be sure the **Upgrade from DES Dynamic Data** checkbox is marked too.



## FAQs – Upgrading from DES v4.0

1. I created new classes based on DES. How are they handled?

If you developed your own classes based on DES or VAM classes, they may need to be reworked. You will know if you get compiler errors.

Contact technical support with your existing class code for assistance.

2. What are the breaking changes?

- On the MultiSegmentDataEntry control, the **AutoPostBackValidates** property changed its type from boolean to `PeterBlum.DES.AutoPostBackValidates`. This enumerated type has three values. When correcting your code, consider these values:
  - No - AutoPostBack does not validate. Postback always occurs. *If your code used false, change it to No.*
  - Control - AutoPostBack runs all validators associated with the Control To Extend. *If your code used true and you want the same behavior, change it to Control.*
  - ValidationGroup - AutoPostBack runs all validators associated with the validation group. The validation group name is specified in the **ValidationGroup** property.
- If you modify the items of a ContextMenu built into DES controls like DateTextBox and Calendar, the methodology differs. Changes to the **Items** collection must occur within the new **ContextMenuUpdated** event on the ContextMenu object. Currently your code is setup in `Page_Load()`. Create the new ContextMenuUpdated event and relocate your code. The ContextMenu object is passed to you through the `e` parameter as `e.ContextMenu`. See “Modifying a ContextMenu (or Help Button menu)” in the **Date and Time User’s Guide**.

All other changes are either converted for you by the Web Application Updater or continue to be supported, but marked as Obsolete (which will give you compiler warnings).

3. Are there any changes to my style sheet files?

Yes. The standard style sheet files installed by DES 4.0 have been reviewed and modified automatically. All modifications have inline comments. While the Calendar control has a new appearance, it will not be used unless you follow the directions of #6 below.

4. Do I need new license files?

No. DES 5.0 uses the same license files and serial numbers as DES 4.0.

5. What changes were made to my web forms, user controls and code files?

- The PeterBlum.DES namespace changed to PeterBlum.DES.Web.WebControls. Some of the types in DES are still in PeterBlum.DES. Others are now in PeterBlum.DES.Web.
- A number of properties changed, especially within the Peter’s Date and Time module where groups of related controls were moved from the control level to a child property. Yet, all old property names still work. You will see compiler warnings describing them as obsolete. You can use those warnings as directions to using their new names, but that is not needed.
- Numerous typecasts will now be using an interface instead of a class. For example, to typecast a validator, it uses PeterBlum.DES.IBaseAnyValidator instead of PeterBlum.DES.Web.WebControls.BaseAnyValidator.
- The CustomValidator’s ServerCondition property has made a first class “event” instead of a delegate. This changes the syntax of how its setup in your forms and codes.
- These classes in the Peter’s Input Security module were renamed: PeterBlum.DES.Security.Globals to PeterBlum.DES.Web.SecurityGlobals; PeterBlum.DES.Security.ConfigFile to PeterBlum.DES.Web.SecurityConfigFile.
- Peter’s Business Logic Driven UI users will notice new “using” or “Imports” definitions. There are separate aliases for the controls and DataAnnotation attributes of Peter’s Business Logic Driven UI.

6. How do I use the new appearance of the Calendar control?

DES 5 provides a new appearance for the Calendar control and the toggle buttons for the DateTextBox and TimeOfDayTextBox. You can see this appearance here.

After conversion, you will be using the appearance that came with DES 4. You can easily switch to the new style sheet files and images by removing (or commenting out) this line in the <appSettings> section of the **web.config** file.

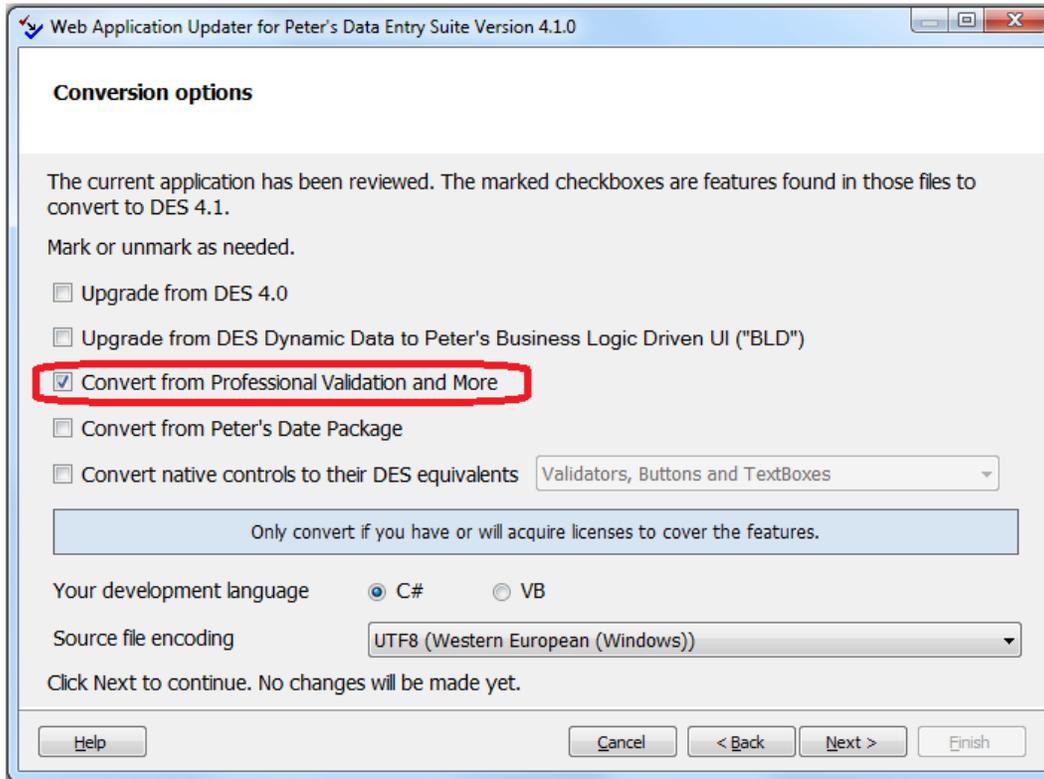
```
<add key="DES_DES4Appearance" value="" />
```

If you only want to switch the image files while keeping the style sheet files from Peter’s Date Package, use this guide.

| Image   | Property to change                              | File name within Appearance/Date and Time |
|---|---|---|
|  | DateTextBox.PopupCalendar.ToggleImageUrl        | ToggleCalendar20x20.GIF                   |
|  | DateTextBox.PopupCalendar.ToggleImageUrl        | ToggleCalendar15x15.GIF                   |
|  | TimeOfDayTextBox.PopupTimePicker.ToggleImageUrl | ToggleClock15x15.GIF                      |
|  | Calendar.PreviousMonthCommand.ImageUrl          | LeftCmdSolidBlue.GIF                      |
|  | Calendar.NextMonthCommand.ImageUrl              | RightCmdSolidBlue.GIF                     |
|  | Calendar.PreviousYearCommand.ImageUrl           | LeftCmd2Overlay1Blue.GIF                  |
|  | Calendar.NextYearCommand.ImageUrl               | RightCmd2Overlay1Blue.GIF                 |

## Converting from Professional Validation and More

Follow the instructions of “[Prepare a web application or site](#)”. The **Web Application Updater** should detect the presence of VAM features and mark the **Convert from Professional Validation and More** option on the **Conversion Options** panel.



## FAQs – Converting from Professional Validation and More

1. I wrote client-side scripts that used functions in VAM. Will they still work?

Most will. During conversion, the **Web Application Updater** has added this key to your <appSettings> section of the **web.config** file:

```
<add key="DES_VAMCompatibleScriptFile" value="" />
```

This key tells DES to load a special script file called **VAMCompatible.js** each time any control found in VAM is used. **VAMCompatible.js** contains the function names that were defined in VAM. It maps them to the new function names.

This has its limits. It only handles functions that were documented in the various User’s Guides, plus a few that Tech Support felt might have been used. So test run your pages. DES provides the source script file **VAMCompatible.js** in the **[DES installation folder]\Upgrading** folder. Use it if you are getting JavaScript errors to see if a function you are using is still supported. If not, please allow Tech Support assist you ([support@peterblum.com](mailto:support@peterblum.com)). Provide tech support with details of any script error, including the actual VAM-compatible script code and the error message.

2. What happens to my style sheets?

VAM provided a style sheet file called **VAMStyleSheets.css** that is located by default in the **[web application]\VAMAppearance** folder.

DES provides replacements to that file: **[web application]\DES\Appearance\Validation\Validation.css** and **[web application]\DES\Appearance\TextBoxes\TextBoxes.css**.

The style sheet class names have changed, such as **VAMFieldWithError** becoming **DESVALFieldWithError**. The **Web Application Updater** will find all of the class names declared in VAM and renamed them. So your pages will already be compatible with the new files *except for any class names that you created*.

In addition, any style sheet classes that you had edited have not been transferred.

Here are two ways to migrate new and edited VAM style sheet classes to DES.

**Use your old VAMStyleSheets.css file**

DES can run in a VAM compatibility mode. When you use this feature, your original file is used. In addition, when DES sees any of the new class names from the **Validation.css** file, it renames it to the old name. As a result, all style sheet classes declared in the original **VAMStyleSheets.css** file, whether edited or new, are used.

*ALERT: DES will not use the Validation.css or TextBoxes.css file in this case. In future work, do not modify those files. Instead, modify the VAMStyleSheet.css.*

- a) Determine the location of the **VAMStyleSheet.css** file. You may want to relocate it from the **VAMAppearance** folder to the **[web application]\DES\Appearance\** folder.
- b) Add these two lines to the <appSettings> section of the **web.config** file:

```
<add key="DES_StyleSheetVAMCompatible" value="[URL to VAMStyleSheet.css]" />
<add key="DES_EnableConvertToVAMClassNames" value="" />
```

DES\_StyleSheetVAMCompatible tells DES to use the **VAMStyleSheet.css** file instead of its **Validation.css** and **TextBoxes.css** files. DES\_EnableConvertToVAMClassNames instructs DES to rename the style sheet class names to their original VAM names at runtime.

The URL can start with “~” to indicate the web application folder, or “{APPEARANCE}” to use the location of the Appearance folder.

If you use **[web application]\DES\Appearance\VAMStyleSheet.css**, use this:

```
<add key="DES_StyleSheetVAMCompatible" value="{APPEARANCE}/VAMStyleSheet.css" />
```

If you use **[web application]\VAM\Appearance\VAMStyleSheet.css**, use this:

```
<add key="DES_StyleSheetVAMCompatible"
value="~/VAM/Appearance/VAMStyleSheet.css" />
```

- c) Since DES introduces a few new style sheet class names, add these to your VAMStyleSheet.css file.

```
.DESVALListWithError
{
    color: red;
}

.DEVSVALCheckboxWithError
{
    color: red;
}
```

- d) For VAM version 1 and 2 users, also these to your VAMStyleSheet.css file.

```
.VAMValSummaryAltRows
{
    color: red;
    background-color: lightblue;
}
.VAMMultiSegContainer
{
    border-left: lightgrey thin inset;
    border-top: lightgrey thin inset;
    border-right: lightgrey thin inset;
    border-bottom: lightgrey thin inset;
    padding-left: 2px;
    padding-top: 2px;
    padding-right: 2px;
    padding-bottom: 2px;
}
```

```
.VAMMultiSegTextBox
{
  border-left: whitesmoke thin solid;
  border-top: whitesmoke thin solid;
  border-right: whitesmoke thin solid;
  border-bottom: whitesmoke thin solid;
}
```

### Edit the Validation.css and TextBoxes.css files

DES has split the contents of **VAMStyleSheet.css** between two files, **Validation.css** and **TextBoxes.css**, so that only style sheets needed by a particular module are sent to the browser. Aside from renaming the style sheet class names and adding a few new classes, little else has changed.

As a result, it's very easy to just copy the contents of your **VAMStyleSheet.css** file into the other two files. Here's how.

- a) Open all three .css files in a text editor.
- b) Copy all styles except **VAMMultiSegContainer** and **VAMMultiSegTextBox** into the **Validation.css** file. You can delete their DES equivalents.
- c) Since DES introduces a few new style sheet class names, add these to the **Validation.css** file.

```
.DESVALListWithError
{
  color: red;
}

.DEVSVALCheckboxWithError
{
  color: red;
}
```

- d) Copy **VAMMultiSegContainer** and **VAMMultiSegTextBox** into the **TextBoxes.css** file. Delete their DES equivalents.
- e) Add this line to the <appSettings> section of the **web.config** file:

```
<add key="DES_EnableConvertToVAMClassNames" value="" />
```

`DES_EnableConvertToVAMClassNames` instructs DES to rename the style sheet class names to their original VAM names at runtime.

*CONTINUED ON THE NEXT PAGE*

3. What are the key changes applied to my web forms and code behind files?

This is a partial list.

- Changed the “PeterBlum.VAM” namespace to “PeterBlum.DES”, “PeterBlum.DES.Web”, and “PeterBlum.DES.Web.WebControls”.
- Changed some references of “PeterBlum.VAM.Security” to “PeterBlum.DES.Web” and “PeterBlum.DES.Web.WebControls”.
- Changed these class names: VAMPPage to WebFormDirector, VAMTypeConverter to DESTypeConverter, and VAMException to DESEException.
- Changed web form tag prefixes from <vam: to <des: and <m2v: to <despval:
- Changed these client-side global variable names to their DES equivalents: gVAMCanEval, gVAMSubmitEvent, gVAMActions, and gVAM\_PageIsValid.
- Inserted the line <%= PeterBlum.DES.Web.WebControls.StyleSheetManager.GetLinkTags() %> into the <head> of each page to provide style sheets.

4. I have fully converted from Professional Validation and More to DES’s controls. How can I clean up my web application?

- First confirm that you have fully converted. Just search for the text “VAM” throughout your web application. You may find using clauses (C#), Imports clauses (VB), and <% @ Register %> tags that all can be removed if the contents of the page are clean. Recompile after removing those entries to see if you were successful.
- Use the **Global Settings Editor** to edit your **custom.des.config** file. Look for any URLs that point to the **VAM** folder. Edit those URLs to locate the file in a different folder. If you are using the original files from VAM, the same files will be found in the **DES\Appearance** folder under a subfolder, such as Validation or TextBoxes.
- Remove Visual Studio project references or entries in the **\bin** folder any assembly that starts with **PeterBlum.VAM**. Do not remove **PeterBlum.ADME.dll** unless you are using ASP.NET 2+.
- Compile to confirm there are no errors.
- Remove the **VAM** folder unless you continue to use its images or style sheets. SUGGESTION: rename it and test you web application for any visual problems introduced by this step.
- In **web.config**, go to the <appSettings> section and remove any key that begins with “VAM\_”.

5. I use the String Lookup System to retrieve strings from resources. Do I have to make any changes?

Only if you plan to use the String Lookup System with the controls introduced in this version. The **Web Application Updater** program has added this line to your <appSettings> section of the **web.config** file which makes DES’s String Lookup System use the resource file names from VAM:

```
<add key="DES_VAMNamesInStringLookup" value="" />
```

DES has changed the default names of resource files from VAMxyz.resx to DESxyz.resx. When the above key is in place, it will continue to use the old names: VAMMessages.resx, VAMLabels.resx, and VAMMisc.resx.

There are several new resource files. Some, like DESHints.resx, are a new location for resources that were previously in the VAMMisc.resx file. You can put your resources either in the new location or the VAMMisc.resx file. DES will first look in the new location for a LookupID. If it’s not found there, it searches VAMMisc.resx.

DES provides empty resource files to help you quickly introduce support for the new controls.

- Add the following files from **[DES installation folder]\StringLookup** to your web application in the same folder that now contains VAMMessages.resx:
  - DESConfirmMessages.resx
  - DESContextMenu.resx
  - DESDateTime.resx
  - DESHints.resx
  - DESPopupViews.resx
  - DESTextCounter.resx
  - DESValidationMisc.resx
  -

6. I use the String Lookup System to retrieve strings from my database. Do I have to make any changes?

No. The **Web Application Updater** program has added this line to your <appSettings> section of the **web.config** file which makes DES's String Lookup System use the stored procedure name and String Group Names from VAM:

```
<add key="DES_VAMNamesInStringLookup" value="" />
```

DES has changed one String Group Name, "MESSAGE", to "ERRORMSG". When the above key is in place, both versions of the String Group Name work; you can continue to use "MESSAGE".

DES has also introduced new String Groups Names as shown here. See the **General Features Guide** for details.

| DES       | VAM     |
|-----------|---------|
| ERRORMSG  | MESSAGE |
| LABEL     | LABEL   |
| VALMISC   |         |
| MENU      |         |
| DATETIME  |         |
| HINT      |         |
| CONFIRM   |         |
| POPUPVIEW |         |
| COUNTER   |         |
| MISC      | MISC    |

## Converting from Peter's Date Package

Use these instructions to migrate the **Peter's Date Package** controls to their **Peter's Data Entry Suite** ("DES") equivalents.

### *What you should know before converting*

DES's Date and Time controls and Peter's Date Package controls can work together in the same web application, so long as you don't try to use them together on the same web form. As a result, carefully consider converting to DES. A good reason to convert is when you want to take advantage of new functionality. See the **Whats New.pdf** document for details on the new functionality.

In addition, several features found in Peter's Date Package are only supported when you have other modules than just the Peter's Date and Time module. They are:

- These properties on textboxes have been replaced by similar functionality in the Peter's Professional Validation module: **xInvalidDateMsg**, **xOutOfRangeDateMsg**, **xInvalidTimeMsg**, **xOutOfRangeTimeMsg**, **xErrorAlertOnChangeB**, **xFocusOnErrorB**, **xErrorForeColor**, **xErrorBackColor**.
- The stand-alone ContextMenu control is now part of **Peter's Interactive Pages**. In addition, the Help button requires **Peter's Interactive Pages** because it utilizes the ContextMenu.

If you chose to migrate, you can convert individual pages or the entire application.

The **Web Application Updater** does most of the work: it modifies enough of your web forms to be able to compile and run them. It still needs your help and that is handled in this section. It also converts style sheet files. Here are the actions that you will take when the **Web Application Updater** is finished.

- Your style sheet files have been restructured to reflect how DES 5 draws the Calendar control (including how it works in the DateTextBox). Be prepared to tweak the look.
- If you modified the Peter's Date Package image files, you will have to move them into **DES's Appearance\Date And Time** folder.
- If you created new image files, you can move them and update the associated properties.
- Some properties are obsolete. Most obsolete properties have similar functionality in DES. Some still work while others require some changes to your web forms. The **Web Application Updater** log file provides you with a list of each obsolete property that was found and directions for addressing them.

### *Steps for Migration*

Follow the instructions of "[Prepare a web application or site](#)". The **Web Application Updater** should detect the presence of PDP features and mark the **Convert from Peter's Date Package** option on the **Conversion Options** panel.

There will be additional panels specific to conversion of Peter's Date Package. They are self documenting.

When finished, review the log and follow the information in the What's Next section. In addition, follow the directions in the next few sections:

- ◆ [Migrating your Style Sheet Classes](#)
- ◆ [Migrating your Image files](#)
- ◆ [Resolving Obsolete Properties](#)
- ◆ [Other Changes: TimeOfDayTextBox.xOnChangeFunctionName Property](#)

## Migrating your Style Sheet Classes

There are numerous differences between the Peter’s Date Package and DES style sheet files. Here are some examples:

- Most properties that specified colors have been replaced by style sheet classes.
- Buttons now offer mouse over effects which require new style sheet classes. The style sheet name must have a specific syntax: use the name of the style sheet class for the normal appearance + “MouseOver”.
- Properties that specified the style sheet class used when the mouse is pressed are no longer supported. Instead, the style sheet name must have a specific syntax: use the name of the style sheet class for the normal appearance + “Pressed”.
- Borders on the Calendar’s day cells get their color from in style sheets. Previously they were handled programmatically. You **must** introduce the border colors (ex: border-left-color: gray) to the style sheets to have borders. Code still generates the presence of a line by defining the border line style and width.
- Mouse overs on the Calendar day cells, the MonthYearPicker month and year cells, and TimePicker time cells are implementing in style sheets.
- Previously the Calendar created the appearance of Day cells by using unique style sheet classes for various situations: normal, selected, today, and specialdate. DES still has separate style sheet classes, but they are designed to be merged. DES will always get the normal style then add the selected or today style as needed. So the style definitions for selected, today, and specialdate have fewer attributes.
- Several colors that were specified by name are now specified by their RGB color to accommodate browsers that lacked support for the color name.

## How your PDP Style Sheet Files are converted

When you use the Migrate from Peters Date Package option of the Web Application Updater, it will attempt to convert your Peter’s Date Package style sheet files.

If the Web Application Updater finds your style sheet files in the **[web app]/PetersDatePackage/Appearance** folder, it will convert them. If they are not there, it helps to copy them into that folder before conversion. If you forget, you can copy them after and rerun the conversion.

The following files are transferred into the **[web app]/DES/Appearance/Date And Time** folder.

| Source                     | Destination  |
|----------------------------|--|
| Calendar.css               | Calendar DES4.css  |
| DateTextBoxes.css          | DateAndTimeTextBoxes.css   |
| MonthYearPicker.css        | MonthYearPicker DES4.css   |
| MultiSelectionCalendar.css | MultiSelectionCalendar.css   |
| SpecialDates.css           | SpecialDates DES4.css  |
| TimePicker.css             | TimePicker DES4.css  |
| ContentMenu.css            | <i>Not copied. Use new file in Appearance/Interactive Pages/Menu.css</i> |
| TimeTextBoxes.css          | <i>Not copied. Similar properties now found in DateAndTimeTextBoxes.</i> |

It then opens each of these files to:

- Rename class names
- Add, change, or remove styles within those classes

In both cases, the file will be commented to indicate the changes.

**Migrating your Image files**

Peter’s Date Package provides several standard graphic files, found in the **[web application]\PetersDatePackage\**

**Appearance** folder. They are the Calendar toggle () , month changing arrows () , multiple date selection arrows () , spinner arrows () , TimePicker toggle () , QuickDateMenu toggle () , Help button () , and Close button () .

Peter’s Data Entry Suite has equivalent graphics, using the same file names and identical images. They are in the **[web application]\DES\Appearance\Date And Time** folder and **[web application]\DES\Appearance\Shared** folder (for spinner arrows, Help button and Close button.)

- If you have not edited any graphics and are using the default filepaths, you do not need to do anything. The migrated controls will use the graphics from their new locations.
- If you have edited the existing graphic files, you need to copy them to their new locations. In addition, DES has introduced mouseover versions of the graphic files. For example, for **Calendar.jpg**, there is now a file called **CalendarMouseOver.jpg**. Since you do not have a mouse over graphic file at this point, you either have to edit the one supplied by DES or delete it to avoid it being used.
- If you created new graphic files, you need to review the URLs you specified in properties. They can remain in their current location or be relocated. As you review a URL, if it contains the token “{APPEARANCE}”, the token no longer points to the original location. You should replace the token with URL. Typically, you replace “{APPEARANCE}” with “~/PetersDatePackage/Appearance/”.

Here are the properties that have associated images throughout the product.

| Date and Time TextBoxes   |  |   |
|---|--|---|
| PDP Property Name   | DES Property Name  | Defaults  |
| xHelpButtonImageUrl   | HelpCommandAndButton.ButtonImageUrl  |  HelpCmd.gif, HelpCmdPressed.gif<br>Add HelpCmdMouseOver.gif     |
| xPopupCalendar.xToggleImageUrl<br>xPopupMonthYearPicker.<br>xToggleImageUrl | PopupCalendar.ToggleImageUrl<br>PopupMonthYearPicker.ToggleImageUrl          |  Calendar.jpg, CalendarPressed.jpg<br>Add CalendarMouseOver.jpg |
| xPopupTimePicker.ToggleImageUrl   | PopupTimePicker.ToggleImageUrl   |  Clock.gif, ClockPressed.gif<br>Add ClockMouseOver.gif          |
| xPrevCommandImageUrl  | PeterBlum.DES.Globals.WebFormDirector.<br>SpinnerManager.PrevCommandImageUrl |  DnArrow1.gif, DnArrow1Pressed.gif<br>Add DnArrow1MouseOver.gif |
| xNextCommandImageUrl  | PeterBlum.DES.Globals.WebFormDirector.<br>SpinnerManager.NextCommandImageUrl |  UpArrow1.gif, UpArrow1Pressed.gif<br>Add UpArrow1MouseOver.gif |

| Calendar and MultiSelectionCalendar             |   |   |
|---|---|---|
| PDP Property Name                               | DES Property Name                             | Defaults  |
| xHelpButtonImageUrl                             | HelpCommand.ButtonImageUrl                    |  HelpCmd.gif, HelpCmdPressed.gif.<br>Add HelpCmdMouseOver.gif  |
| xCloseButtonImageUrl                            | CloseCommand.ButtonImageUrl                   | <input checked="" type="checkbox"/> CloseCmd.gif, CloseCmdPressed.gif<br>Add CloseCmdMouseOver.gif  |
| xPrevMonthButtonImageUrl                        | PreviousMonthCommand.ButtonImageUrl           |  LeftCmd_3DBlue.gif,<br>LeftCmd_3DBluePressed.gif<br>Add LeftCmd_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with LeftCmd_                         |
| xNextMonthButtonImageUrl                        | NextMonthCommand.ButtonImageUrl               |  RightCmd_3DBlue.gif,<br>RightCmd_3DBluePressed.gif<br>Add RightCmd_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with RightCmd_                     |
| xJumpBackButtonImageUrl                         | JumpBackCommand.ButtonImageUrl                |  LeftCmd2_3DBlue.gif,<br>LeftCmd2_3DBluePressed.gif<br>Add LeftCmd2_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with LeftCmd2_                     |
| xJumpForwardButtonImageUrl                      | JumpForwardCommand.ButtonImageUrl             |  RightCmd2_3DBlue.gif,<br>RightCmd2_3DBluePressed.gif<br>Add RightCmd2_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with RightCmd2_                 |
| xTodayButtonImageUrl                            | TodayCommand.ButtonImageUrl                   | No defaults   |
| xSpecialButtonImageUrl                          | Special Command.ButtonImageUrl                | No defaults   |
| xClearButtonImageUrl                            | Clear Command.ButtonImageUrl                  | No defaults   |
| xShowSelectionButtonImageUrl                    | ShowSelection Command.ButtonImageUrl          | No defaults   |
| xApplyButtonImageUrl                            | Apply Command.ButtonImageUrl                  | No defaults   |
| xExpandedDateFormatter.<br>xCloseButtonImageUrl | ExpandedDateFormatter.<br>CloseButtonImageUrl | <input checked="" type="checkbox"/> CloseCmd.gif, CloseCmdPressed.gif<br>Add CloseCmdMouseOver.gif  |
| xSelectWeeksImageUrl                            | SelectWeeksImageUrl                           |  ToggleWeekSolidGray.gif,<br>ToggleWeekSolidGray Pressed.gif<br>Add ToggleWeekSolidGrayMouseOver.gif<br>There are several versions of this image, all starting with ToggleWeek     |
| xSelectMonthImageUrl                            | SelectMonthImageUrl                           |  ToggleMonthSolidGray.gif,<br>ToggleMonthSolidGray Pressed.gif<br>Add ToggleMonthSolidGrayMouseOver.gif<br>There are several versions of this image, all starting with ToggleMonth |
| xSelectDOWImageUrl                              | SelectDOWImageUrl                             |  ToggleDOWSolidGray.gif,<br>ToggleDOWSolidGray Pressed.gif<br>Add ToggleDOWSolidGrayMouseOver.gif<br>There are several versions of this image, all starting with ToggleDOW         |

| Other controls  |                          |                         |  |
|-----------------|--------------------------|-------------------------|--|
| Control name    | PDP Property Name        | DES Property Name       | Defaults   |
| MonthYearPicker | xCloseButtonImageUrl     | CloseButtonImageUrl     |  CloseCmd.gif, CloseCmdPressed.gif<br>Add CloseCmdMouseOver.gif  |
| MonthYearPicker | xPrevYearsButtonImageUrl | PrevYearsButtonImageUrl |  LeftCmd_3DBlue.gif,<br>LeftCmd_3DBluePressed.gif<br>Add LeftCmd_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with LeftCmd_     |
| MonthYearPicker | xNextYearsButtonImageUrl | NextYearsButtonImageUrl |  RightCmd_3DBlue.gif,<br>RightCmd_3DBluePressed.gif<br>Add RightCmd_3DBlueMouseOver.gif<br>There are several versions of this image, all starting with RightCmd_ |
| TimePicker      | xCloseButtonImageUrl     | CloseButtonImageUrl     |  CloseCmd.gif, CloseCmdPressed.gif<br>Add CloseCmdMouseOver.gif  |
| PopupTimePicker | xToggleImageUrl          | ToggleImageUrl          |  Clock.gif, ClockPressed.gif<br>Add ClockMouseOver.gif   |
| QuickDateMenu   | xToggleImageUrl          | ToggleImageUrl          |  Range.gif, RangePressed.gif<br>Add RangeMouseOver.gif   |

## Resolving Obsolete Properties

Peter's Date Package used a different naming convention for its property names than DES. It started each property with a lowercase x and finished boolean properties with an uppercase B. Several properties were renamed in other ways, for clarity and consistency. The **Web Application Updater** has changed many of these names, although not all. For example, **xPopupCalendar** became **PopupCalendar** and **xDate** became **DateValue**.

For each that is not changed, you should expect to see a compiler warning that the property is obsolete (except for properties in ASP.NET Markup). This warning will give you further instructions to resolve the problem. Use this section as a guide to resolving those names. In many cases, you can ignore the warning because the program will still work.

Here are the warning messages.

*New property name: [name]. This name still works for SET but not for GET.*

The old property name can remain in the code when it's shown inside of the ASP.NET Markup. For example:

```
<des:Calendar xWeekendBackground="green" />
```

When used programmatically, the old code can work if you are assigning a value to it.

```
Calendar1.xWeekendBackground = Color.Green
```

When used programmatically, you must change the code if you are getting the value from it:

```
CallMyFunction(Calendar1.WeekendBackground)
```

*New property name: [name]. This name still works.*

No changes are required. All existing code should work. Rename it only if the warning message bothers you.

*New property name: [name]. This name still works except in ASP.NET Markup.*

When used programmatically, no changes are required. All existing code should work. Rename it only if the warning message bothers you.

When used in the ASP.NET Markup, you must change the code to the new property name.

*This property is ignored. Replaced by [description].*

You must change the code and ASP.NET Markup. While the description may help, the log file output by the Web Application Updater will contain more directions for each of these obsolete properties.

## Other Changes: TimeOfDayTextBox.xOnChangeFunctionName Property

The TimeOfDayTextBox and DurationTextBox allow you to setup a JavaScript function that is identified by the **xOnChangeFunctionName** property. While the feature remains (using the new property name **OnChangeFunctionName**), the parameters of your JavaScript function have changed and require editing.

In Peter's Date Package, there were 4 parameters. *The pError parameter was added in later releases and is optional. It may be missing in your code.*

```
function MyOnChangeFnc(pTBId, pDTBId, pTime, pError)
```

In DES, there are 3 parameters, with the pDTBId parameter removed.

```
function MyOnChangeFnc(pTBId, pTime, pError)
```

To fix this, follow these steps:

1. Remove the *pDTBId* parameter.
2. If you need the value of that parameter, which is the ID to the DateTextBox connected to the TimeOfDayTextBox, modify your script to get it like this:

```
var pDTBId = DES_TMTBGetDTBId(pTBId);
```

## FAQs – Converting from Peter’s Date Package

1. How do I use the new appearance of the Calendar control?

DES 5 provides a new appearance for the Calendar control and the toggle buttons for the DateTextBox and TimeOfDayTextBox. You can see this appearance here.

After conversion, you will be using the appearance that came with Peter’s Date Package. You can easily switch to the new style sheet files and images by removing (or commenting out) this line in the <appSettings> section of the **web.config** file.

```
<add key="DES_DES4Appearance" value="" />
```

If you only want to switch the image files while keeping the style sheet files from Peter’s Date Package, use this guide.

| Image   | Property to change                              | File name within Appearance/Date and Time |
|---|---|---|
|  | DateTextBox.PopupCalendar.ToggleImageUrl        | ToggleCalendar20x20.GIF                   |
|  | DateTextBox.PopupCalendar.ToggleImageUrl        | ToggleCalendar15x15.GIF                   |
|  | TimeOfDayTextBox.PopupTimePicker.ToggleImageUrl | ToggleClock15x15.GIF                      |
|  | Calendar.PreviousMonthCommand.ImageUrl          | LeftCmdSolidBlue.GIF                      |
|  | Calendar.NextMonthCommand.ImageUrl              | RightCmdSolidBlue.GIF                     |
|  | Calendar.PreviousYearCommand.ImageUrl           | LeftCmd2Overlay1Blue.GIF                  |
|  | Calendar.NextYearCommand.ImageUrl               | RightCmd2Overlay1Blue.GIF                 |

2. Why are there two <% =PeterBlum.DES.Web.WebControls.StyleSheetManager.GetLinkTags() %> entries?

Peter’s Date Package had some users add the tag

```
<% =PeterBlum.PetersDatePackage.CommonFunctions.GetStyleSheetLinkTags(Page) %>
```

to the <head> tag. The **Web Application Updater** has replaced this with the call used by DES. If this page was previously converted, either from VAM or native validators, the **Web Application Updater** also added the `GetLinkTags()` code.

You can remove one of them, if you like. One may have a parameter, *Page*. That one is the recommended line to remove, but it’s safe if you remove the other. *The GetLinkTags() line is not required for ASP.NET 2+ users so long as the <head> tag has a runat="server" attribute and does not contain any <% %> tags. In that case, DES will call GetLinkTags for you as part of generating the page.*

You do not have to remove the duplicate though. Only the first call to it is actually used. So aside from making your code look cleaner, the extra `GetLinkTags()` call is harmless.

3. I’ve learned the property names for Peter’s Date Package. How much do I need to relearn?

Most properties are very similar, where their lead x character and trailing uppercase B character are removed. For example, **xDateTextBoxControlID** becomes **DateTextBoxControlID** and **xShowWeekNumbersB** becomes **ShowWeekNumbers**.

However, any property that creates the appearance of a button in the Calendar, MonthYearPicker, or TimePicker has been relocated. Each button now has a single property named for the button, like “NextYearCommand”, with its child properties handling all of the formatting. For example, “NextYearButtonImageUrl” becomes **NextYearCommand.ImageUrl**.

4. I have fully converted from Peter’s Date Package to DES’s Date and Time controls. How can I clean up my web application?

- First confirm that you have fully converted. Just search for the text “PetersDatePackage” throughout your web application. You may find uses clauses (C#), Imports clauses (VB), and <% @ Register %> tags that all

can be removed if the contents of the page are clean. Recompile after removing those entries to see if you were successful.

- Remove Visual Studio project references or entries in the `\bin` folder for these assemblies:

**PetersDatePackage.dll**

**PeterBlum.PDPtoVAM.dll**

- Compile to confirm there are no errors.
- Remove the **PetersDatePackage** folder unless you continue to use its images or style sheets. SUGGESTION: rename it and test you web application for any visual problems introduced by this step.
- In **web.config**, go to the `<appSettings>` section and remove any key that begins with “PDP\_”.

5. I wrote client-side scripts that used functions in Peter’s Date Package. Will they still work?

Most will. During conversion, the **Web Application Updater** has added this key to your `<appSettings>` section of the **web.config** file:

```
<add key="DES_PDPCCompatibleScriptFile" value="" />
```

This key tells DES to load a special script file called **PDPCCompatible.js** each time any control found in Peter’s Date Package is used. **PDPCCompatible.js** contains the function names that were defined in VAM. It maps them to the new function names.

This has its limits. It only handles functions that were documented in the Peter’s Date Package User’s Guide, plus a few that Tech Support felt might have been used. So test run your pages. DES provides the source script file **PDPCCompatible.js** in the **[DES installation folder]\Upgrading** folder. Use it if you are getting JavaScript errors to see if a function you are using is still supported. If not, please allow Tech Support assist you ([support@peterblum.com](mailto:support@peterblum.com)). Provide tech support with details of any script error, including the actual PDP-compatible script code and the error message.

## Converting from ASP.NET 2.0 to ASP.NET 4.0

If you upgrade to the ASP.NET 4.0 framework, consider using the **PeterBlum.DES.dll** that is compiled for ASP.NET 4.0. It is not a requirement. The older assembly still works, but the new one takes advantage of new features in ASP.NET 4.0 including ASP.NET Dynamic Data.

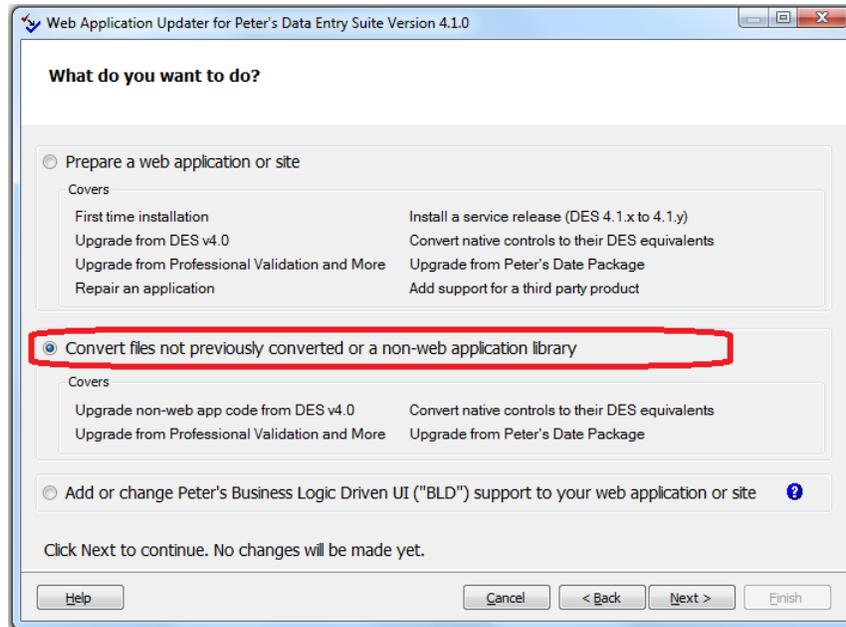
To convert, simply update your project reference to use the **PeterBlum.DES.dll** assembly found in **[DES installation folder]\Assemblies\ASPNET 4.0**.

If your **\bin** folder or project contains **PeterBlum.DES.BLD.dll**, replaced it when the assembly found in **[DES installation folder]\Assemblies\ASPNET 4.0**.

## Convert Native Controls to their DES equivalents

You can quickly convert the native controls in your web application to their DES equivalents with the **Web Application Updater** program.

1. Run the **Web Application Updater** with the **Convert files not already converted** option.
2. Click **Next** and use that screen to select the files to convert.
3. Click **Next** and select the **Convert native controls to their DES equivalents** option on the second panel. You can still select which type of controls to convert.
4. Follow the remaining prompts.



This feature converts the following controls:

- System.Web.UI.WebControls.TextBox (<asp:TextBox>) to PeterBlum.DES.TextBox
- System.Web.UI.WebControls.Button (<asp:Button>) to PeterBlum.DES.Button
- System.Web.UI.WebControls.LinkButton (<asp:LinkButton>) to PeterBlum.DES.Button
- System.Web.UI.WebControls.ImageButton (<asp:Button>) to PeterBlum.DES.ImageButton
- System.Web.UI.WebControls.RequiredFieldValidator (<asp:RequiredFieldValidator>) to PeterBlum.DES.NativeToDES.RequiredFieldValidator
- System.Web.UI.WebControls.CompareValidator (<asp:RequiredFieldValidator>) to PeterBlum.DES.NativeToDES.CompareValidator
- System.Web.UI.WebControls.RangeValidator (<asp:RequiredFieldValidator>) to PeterBlum.DES.NativeToDES.RangeValidator
- System.Web.UI.WebControls.RegularExpressionValidator (<asp:RequiredFieldValidator>) to PeterBlum.DES.NativeToDES.RegularExpressionValidator
- System.Web.UI.WebControls.CustomValidator (<asp:RequiredFieldValidator>) to PeterBlum.DES.NativeToDES.CustomValidator
- AjaxControlToolkit.ValidatorCalloutExtender (<cc1:ValidatorCalloutExtender>) to PeterBlum.DES.NativeToDES.ValidatorCalloutExtender. See [“Differences in ValidatorCalloutExtender Controls”](#).

Use the **Type of Control** drop down list (at the bottom) to determine the controls that are converted. Then follow the prompts.

## Differences in ValidatorCalloutExtender Controls

The DES ValidatorCalloutExtender does *not* mimic the features of the AJAX Control Toolkit ValidatorCalloutExtender. Instead, it changes the ErrorFormatter of the target Validator control to use DES's PopupErrorFormatter. DES includes style sheets and images that allow it to create a PopupView that is similar in appearance to the AJAX Control Toolkit. It also offers several properties found on the PopupErrorFormatter class so you can take advantage of those features without directly editing the ErrorFormatter property on the Validator.

## Matching Appearance

DES includes the images from the ValidatorCalloutExtender. The close button and large alert image are both automatically applied. DES will ignore any setting you made to the **CloseImageUrl** and **WarningIconImageUrl** properties. If you want to change its defaults, open the **Global Settings Editor** and edit each of these ErrorMessage PopupViews: Callout-small, Callout-medium, and Callout-large.

Within the PopupView, substitute the **CloseButtonImageUrl** property for the **ValidatorCalloutExtender.CloseImageUrl**. Substitute the **BodyImageUrl** property for the **ValidatorCalloutExtender.WarningIconImageUrl**.

Images for Close buttons are in the **[WebApplicationRoot]\DES\Appearance\Shared** folder:

- **CloseCmd.gif** (☒) is the standard DES closebutton.
- **CloseCmd2.gif** (✖) is the ValidatorCalloutExtender close button, augmented with images for mouseover and pressed effects.

Images for the Warning icons are in the **[WebApplicationRoot]\DES\Appearance\Validation** folder:

- **BigValErrorIcon.GIF** () is the standard large image for DES.
- **BigValErrorIcon2.GIF** () is the ValidatorCalloutExtender large image (called **alert-large.GIF** in the toolkit).
- **ValErrorIcon.GIF** () is the standard small image for DES.
- **BigValErrorIcon2.GIF** () is the ValidatorCalloutExtender small image (called **alert-small.GIF** in the toolkit).

The remaining elements that control appearance are defined in style sheet definitions in the file **[WebApplicationRoot]\DES\Appearance\Validation\PopupErrorMessages.css**. Look for the class definitions under the heading "ValidatorCalloutExtender Scheme".

The appearance is not identical to the AJAX Control Toolkit ValidatorCalloutExtender. It has a larger area above and below the image and error message text. The area at the top is for a titlebar. You can add a title and the user can drag when the mouse is in that area. It's size defaults to 6 px, which is the height of the close button. If you intend to use a title, consider a larger size. Set it in the DES\_PEFHeaderCallout class by editing the `font-size` and `height` styles.

```
.DES_PEFHeaderCallout
{
    font-size: 6px; /* if you use the title feature, increase this size */
    height: 6px;
/* add this if you allow dragging and want to emphasize that fact
    cursor: move;
*/
}
```

The extra area below is to provide balance to the area on top. You can modify it in the DES\_PEFOverallCallout class by editing the `padding-bottom` style.

```
.DES_PEFOverallCallout
{
    border-right: black 1px solid;
    border-top: black 1px solid;
    border-left: black 1px solid;
    border-bottom: black 1px solid;
    font-family: Verdana;
```

```
font-size: 10px;  
color: Black;  
background-color: #fffacd; /* lemonchiffon */  
padding-bottom:10px; /* creates the gap at the bottom */  
}
```

## Properties for PopupErrorFormatters

The following properties are found on the PopupErrorFormatter class. Since DES's ValidatorCalloutExtender creates a PopupErrorFormatter, these properties let you setup the PopupErrorFormatter. Detailed documentation on these properties is found in the **Validation User's Guide**.

- ErrorMessageHelp – Used by the PopupView.HelpBehavior property to do many things, such as adding help text, a title, a hyperlink, and running a script.
- PopupViewName – If you do not like the default PopupView selected by this control, define your own in the Global Settings Editor and assign its name here. When this is unassigned, it uses one of these predefined PopupViews, as determined by the Width property: Callout-small (Width <= 200px), Callout-medium (Width <=400px), Callout-large (Width > 400px).
- PopupOnFocus – Determines if focus in the data entry control displays the PopupView.
- PopupOnFocusDelay – How many milliseconds before the PopupView appears on focus.
- PopupOnMouseOver – Determines if mouseover the error message image displays the PopupView.
- PopupOnMouseOverDelay – How many milliseconds before the PopupView appears on mouseover.

## Add or Change Peter’s Business Logic Driven UI support

Peter’s Business Logic Driven UI (“BLD”) provides a different approach to application development. If you are starting an application, it deserves strong consideration.

ASP.NET Dynamic Data, from which Peter’s Business Logic Driven UI is based, adds the **DynamicData** folder to your web application, hosting Field Templates, Page Templates, and support files. BLD adds the **BLD Templates** folder to hold its own versions of these files. Learn more in the “Adding Peter’s Business Logic Driven UI” section of the **BLD User’s Guide**.

Use the **Web Application Updater** program to install these files. Before you do, you must make a decision on the technology you will use for your Entity table objects and the engine that performs CRUD operations on those objects. There are two models to consider: BLD DataAccessObjects and traditional ASP.NET Dynamic Data. Both are supported by Peter’s Business Logic Driven UI.

### BLD DataAccessObjects

Peter’s Business Logic Driven UI provides “BLD DataAccessObjects”, which are classes for developing Entity table objects and Data Access Objects (which handle the CRUD operations on the Entity table objects).

Application development enforces strong separation of user interface code from business logic. You will be writing business logic code in separate classes, usually well in advance of building the actual web forms that consume them. Here are its key features:

- Supports Entity Framework v4, LINQ to SQL, ADO.NET and even your own custom solutions. You can still use the Entity table objects defined with Entity Framework and LINQ to SQL.
- Provides a Data Access Object framework that knows how to perform CRUD on the Entity table objects you develop. You will subclass the Data Access Objects for each table to provide specific CRUD methods, such as a query that has a specific selection criteria.
- EntityDAODataSource is a DataSource control that goes on your page. It places a heavy emphasis on separation of concerns. It cannot be used to write queries or other CRUD operations. Instead, EntityDAODataSource calls upon your Data Access Object classes to do all CRUD work.

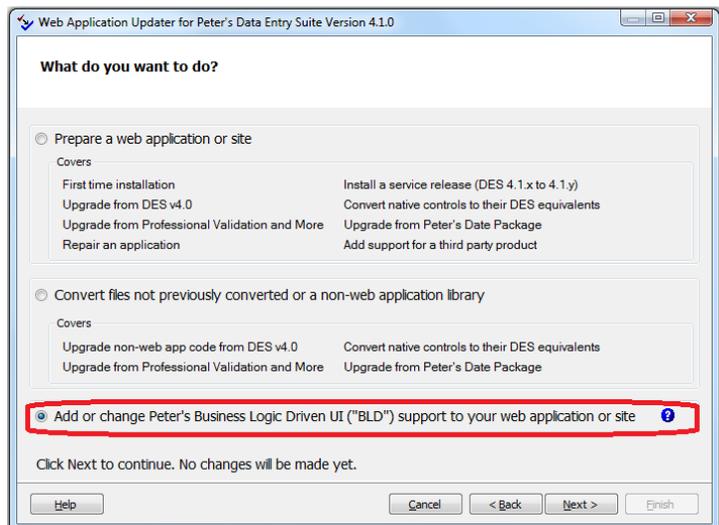
### Traditional ASP.NET Dynamic Data

ASP.NET Dynamic Data lacks the Data Access Object concept. As a result, all CRUD activity is defined within the DataSource control. So it is built in the UI layer, not in a separate business layer.

You can use either Entity Framework or LINQ to SQL for your Entity table objects. The EntityDataSource and LinqDataSource controls handle the CRUD activity on the web form.

## Installing Peter’s Business Logic Driven UI

1. Run the **Web Application Updater**.
2. Select the **Add or change Peter’s Business Logic Driven UI** option.
3. Follow the remaining prompts.



## Deploying Your Web Applications

Peter’s Data Entry Suite (“DES”) installs many more files than just one assembly into your web application. The following is a checklist of the files that you must install when you deploy to another computer.

| Filename  | Location<br>Comments   |
|---|--|
| ✓ PeterBlum.DES.dll   | [webapplicationroot]\bin or Global Assembly Cache  |
| ✓ PeterBlum.DES.NativeValidators.dll  | [webapplicationroot]\bin or Global Assembly Cache<br>Optional. If you are using DES textboxes with the native ASP.NET validation system.   |
| ✓ PeterBlum.DES.NativeToDES.dll   | [webapplicationroot]\bin or Global Assembly Cache<br>Optional. If you converted from <b>Professional Validation And More</b> or <b>Peter’s Date Package</b> .  |
| ✓ PeterBlum.DES.PDPToDES.dll  | [webapplicationroot]\bin or Global Assembly Cache<br>Optional. If you are using <b>Peter’s Date Package</b> with DES.  |
| ✓ PeterBlum.DES.BLD.dll or<br>PeterBlum.DES.BLD3_5.dll                            | [webapplicationroot]\bin or Global Assembly Cache<br>Optional. If you are using <b>Peter’s Business Logic Driven UI</b> .  |
| ✓ PeterBlum.DES.DataAnnotations.dll<br>or<br>PeterBlum.DES.DataAnnotations3_5.dll | [webapplicationroot]\bin or Global Assembly Cache<br>Optional. If you are using <b>Peter’s Business Logic Driven UI</b> .  |
| ✓ DES folder  | Root directory of your web application<br><br>Copy the entire folder. It contains the Appearance folder with images and style sheets, License folder with your existing licenses, and several XML configuration files.<br><br>You may have reconfigured DES to keep this or its child folders in different locations. The <appSettings> section of <b>web.config</b> will identify the new locations.  |
| ✓ Licenses  | Your licenses should already be established in the [webapplicationroot]\DES\Licenses folder. Your development efforts should have already tested them.<br><br>If you have a Web Server license, the software will now switch from using the <b>DES Development.lic</b> file to the <b>DES SingleServer###.lic</b> files. Often users find they have not entered the correct License Key with an error in the serial number or computer name. So review the License Key in the Application_Start() method of <b>Global.asax</b> . See “ <a href="#">Installing Licenses</a> ”. If there are licensing errors, please see “ <a href="#">Troubleshooting Licenses</a> ”.<br><br><i>Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see <a href="#">this troubleshooting topic first</a>.</i> |

THIS TABLE IS CONTINUED ON THE NEXT PAGE

| Filename                                  | Location<br>Comments   |
|---|--|
| ✓ Global.asax                             | <b>[webapplicationroot]</b><br><br>This file contains the <code>Application_Start()</code> method that you modified for Web Server and Site Licenses. <code>Application_Start()</code> does not run unless the <b>Global.asax</b> file is present.   |
| ✓ web.config                              | <b>[webapplicationroot]</b><br><br>The <b>web.config</b> file contains several DES-specific entries, including <code>&lt;appSettings&gt;</code> entries, <code>&lt;assemblyBinding&gt;</code> tags, and <code>&lt;page&gt;</code> tags.<br><br>Any key for DES declared in the <code>&lt;appSettings&gt;</code> section should be reviewed. If file paths are declared in the <code>DES_ConfigFilePath</code> or <code>DES_LicenseFilePath</code> keys, they may no longer match the file structure of the destination server. |
| ✓ Third party assemblies that support DES | <b>[webapplicationroot]\bin</b><br><br>If you use other products that implement DES classes, be sure they are migrated as well. For Telerik, Infragistics, and EasyListBox, their code is compiled into your web application assembly. So make sure that assembly is deployed.   |
| ✓ BLD Templates                           | Root directory of your web application<br><br>Copy the entire folder. It contains Template files used by Peter’s Business Logic Driven UI. It is not present when you are not using BLD.   |

If you are installing onto a hosted computer, it may be setup with more security restrictions than your development system. Please see [“Installing into a Partial Trust Environment”](#).

If you are having problems, please see the [“Troubleshooting”](#) section.

### Confirming a Deployment

You can review most of the settings of your web application by using a special querystring parameter. See [“Exploring the Current Settings”](#).

## Installing DES onto a New Development Computer

1. Add controls to the Visual Studio toolbox. See “[Adding to the Visual Studio/Visual Web Developer Toolbox](#)”.
2. If this computer has its own web servers for development, add DES to each web application. See “[Prepare a web application or site](#)”.

## Installing into a Partial Trust Environment

*Note: PeterBlum.com tech support cannot offer assistance in working with a partial trust environment beyond the technical details given here.*

The .net framework provides extensive security to prevent rogue assemblies from using resources that should be off limits. In .net v1.1 and higher, the web server can be set to impose security protections upon the web applications. This is called establishing “partial trust”. Extensive details on security in an ASP.NET environment can be found in this online book: “[Improving Web Application Security: Threats and Countermeasures](#)” by J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan of Microsoft Corporation. In particular, see Chapter 9, “[Using Code Access Security with ASP.NET](#)”.

There are two parts to security: settings within the assembly and permissions within the ASP.NET environment. This is how to use Peter’s Data Entry Suite within a partial trust environment.

### Assemblies

In a partial trust environment, assemblies must have the `AllowPartiallyTrustedCallersAttribute` established. In addition, they must be “strong named” (compatible with the Global Assembly Cache by having a publickeytoken.)

**PeterBlum.DES.dll**, **PeterBlum.DES.NativeValidators.dll**, **PeterBlum.DES.NativeToDES.dll** and **PeterBlum.PDPtoDES.dll** support `AllowPartiallyTrustedCallersAttribute` and strong names.

### Permissions within the ASP.NET environment

Your machine.config file identifies another config file that lists all of the permissions an ASP.NET application has. In this config file, you must have the following permissions for Peter’s Data Entry Suite to work.

#### FileIOPermission

File access is only used to read the license files. The software never creates, deletes or writes to files.

Here is the definition for the minimum for FileIOPermission:

```
<IPermission
  class="FileIOPermission"
  version="1"
  Read="$AppDir$"
  PathDiscovery="$AppDir$"
/>
```

This definition assumes that the licenses are located in the **[web application]** folder or one of its subfolders. If you have it in the **/aspnet\_client** folder and that folder is outside of the **[web application]** folder, relocate it into the web application.

#### SecurityPermission

It must have the Flags Execute and ControlThread. **ControlThread is not defined by default in trust level=Low.**

ControlThread allows you to assign a CultureInfo object to the current thread, which is important when dealing with dates and times.

Here is the definition for the minimum for SecurityPermission:

```
<IPermission
  class="SecurityPermission"
  version="1"
  Flags="Execution, ControlThread"
/>
```

#### ReflectionPermission

This is optionally used. When the **custom.DES.config** file uses `<property>` tags in any of these sections, `<DataTypes>`, `<ErrorFormatters>`, or `<Conditions>`, ReflectionPermission is required. If you cannot get this permission, remove the associated `<property>` tags from **custom.DES.config**.

### EventLogPermission

When you are using Visual Input Security and logging to the event log, .net demands the EventLogPermission. **This permission is NOT defined in any Trust level other than FULL. You must explicitly add it to your permissions.** If you are using a hosted site's computer, the event log should be considered off limits. So use file logging instead.

Here is the definition for the minimum for LogEventPermission:

```
<IPermission
  class="EventLogPermission"
  version="1"
/>
```

## Using IIS 7

By default, applications created under IIS 7 use the Application Pool called “DefaultAppPool”. This pool uses the “Integrated Pipeline” feature. When setup that way, third party HttpHandlers must appear in the `<system.webServer>` section of the **web.config** file. DES has such an HttpHandler called DESGetFiles.

If the Web Application Updater detects the `<system.webServer>` section, it will automatically create the HttpHandler, like this:

```
<system.webServer>
  <handlers>
    <add name="name" path="DESGetFiles.aspx" verb="GET"
        type="PeterBlum.DES.GetFilesHandler, PeterBlum.DES,
            Version=5.0.0.5000, Culture=neutral,
            PublicKeyToken=cb5182303c90db58" />
  </handlers>
  <validation validateIntegratedModeConfiguration="false" />
</system.webServer>
```

*Note: If you need to set `validateIntegratedModeConfiguration` to true, see “Using `validateIntegratedModeConfiguration=True`” below.*

You need to take no further action. The Web Application Updater will even update the version number as service releases are installed.

If you do not have the `<system.webServer>` section, you *will* get JavaScript errors because DESGetFiles feeds the script files to the browser. Here are four ways to overcome this issue:

### Copy the `<system.webServer>` tag

Copy the tag shown above into your web.config file. Put it just before the `</configuration>` tag. (Never put it at the top.)

Change the `Version=` property to reflect the current version as shown on the **PeterBlum.DES.dll** file.

### Run the **AppCmd.exe** utility

IIS provides the **AppCmd.exe** utility to handle many tasks, one of which is to create the `<system.webServer>` section. It will also create the `<add>` tag in the `<handlers>` section based on tags already located in the `<httpHandlers>` section of the **web.config** file.

This command line will create that section for you:

```
%windir%\system32\inetsrv\APPCMD.EXE migrate config <Application Path>
```

Where `<Application Path>` is the virtual path of the application containing the site name, such as “Default Web Site/app1”.

See <http://www.iis.net/default.aspx?i=928&subtabid=25&tabid=2&p=2> for details.

### Switch to “Classic” mode

Change the Application Pool assigned to this web application to “Classic”.

1. In the IIS Manager, select your web application node.
2. Click the **Basic Settings** command under the Actions section (to the right).
3. In the Edit Application dialog box, click **Select**.
4. In the **Application Pool** dropdownlist, choose “Classic NET AppPool”.
5. Click **OK**.
6. Click **OK**.

**ALERT:** IIS 7 may not use the HttpHandler if the web.config file defines it in both `<system.webServer>` and `<httpHandlers>` sections. Make sure you have it only in one of those two places.

**Disable DES's HttpHandler**

Add this line to the <appSettings> section of the **web.config** file:

```
<add key="DES_GetFilesVirtualPath" value="~/DES/GetFiles.aspx"/>
```

**Using validateIntegratedModeConfiguration=True**

When your **web.config** file must have this line:

```
<validation validateIntegratedModeConfiguration="true" />
```

you will need to turn off DES's own validation for the presence of the HttpHandler in the pre-IIS7 section "system.web/HttpHandlers" because you are not allowed to have that section.

Add this line to the <appSettings> section of the **web.config** file:

```
<add key="DES_DisableHttpHandlerCheck" value="" />
```

## Troubleshooting

Here are some issues that you may run into. Remember that technical support is available from [support@PeterBlum.com](mailto:support@PeterBlum.com). We encourage you to use this knowledge base first.

This guide contains problems specific to installation. Please see the “Troubleshooting” section of the **General Features Guide** for an extensive list of other topics including “Handling JavaScript Errors” and “Common Error Messages”.

Click on any of these topics to jump to them:

- ◆ [FAQs - Prepare a web application or site](#)
- ◆ [FAQs – Adding to the Toolbox](#)
- ◆ [Troubleshooting Licenses](#)
- ◆ [FAQs – Converting from Professional Validation and More](#)
- ◆ [FAQs – Converting from Peter's Date Package](#)
- ◆ [Confirming a Deployment](#)
- ◆ [Using IIS 7](#)
- ◆ [Installing into a Partial Trust Environment](#)
- ◆ [What to do when you get version errors](#)
- ◆ [Visual Studio Problems](#)
- ◆ [Exploring the Current Settings](#)

### ***What to do when you get version errors***

- If the assembly is any from DES (they all start with “PeterBlum.DES”), run the **Web Application Updater** utility with the option [Prepare a web application or site](#).
- If you can compile the assembly, recompile it with the same **PeterBlum.DES.dll** that is in use in your web application.
- For any other assembly, open your **web.config** file and locate an `<assemblyBinding>` tag. If any are found, they map a specific assembly in the `<assemblyIdentity>` tag to a desired version number for that assembly.
- Correct or add its `<assemblyBinding>` tag.
- Make sure the `<configuration>` tag of the file does *not* have the `xmlns=` attribute.

## Visual Studio Problems

### Visual Studio's Publish Web Site command does not copy the license files

Visual Studio users who use the **Publish Web Site** command will find that it ignores the .lic files of your web application. You can use the <buildProviders> section of **web.config** to resolve this:

```
<configuration>
  <system.web>
    <compilation>
      <buildProviders>
        <remove extension=".lic"/>
        <add extension=".lic"
          type="System.Web.Compilation.ForceCopyBuildProvider"/>
      </buildProviders>
    </compilation>
  </system.web>
</configuration>
```

*Note: The web.config file is case sensitive.*

*Warning: This solution works when using the standard way to create a web site without a project file. It does not work with the [Web Application Project](#) model.*

In Solution Explorer, view the Properties of the license file. Confirm that **Copy To Output Directory** is set to Copy Always and **Build Action** is Content.

If this does not work, change the file extension of the license files from the **.lic** to **.licx**.

### The References section of the Visual Studio Properties Pages lists duplicate DES assemblies

When you view the References section of the Visual Studio Properties Pages on your web application, you may see up to 3 copies of the DES assemblies. One will refer to the GAC, which DES is actually not using. While confusing, there is nothing wrong with your web application assembly. It is the way Visual Studio handles this group of tags in the **web.config** file:

```
<compilation debug="false" strict="false" explicit="true">
  <assemblies>
    <add assembly="PeterBlum.DES, Version=5.0.#.5000, Culture=neutral,
      PublicKeyToken=cb5182303c90db58" />
    <add assembly="PeterBlum.DES.NativeToDES, Version=5.0.#.5000,
      Culture=neutral, PublicKeyToken=cb5182303c90db58" />
    <add assembly="PeterBlum.DES.NativeValidators, Version=5.0.#.5000,
      Culture=neutral, PublicKeyToken=cb5182303c90db58" />
  </assemblies>
</compilation>
```

Do not attempt to fix it unless you are comfortable removing the above tags.

## Exploring the Current Settings

DES has a special feature that lets you review its various components on the server. Go to any page using DES controls and add a special querystring parameter to the URL. It will return a page detailing a certain aspect of DES's setup. This is very helpful after site deployment and for technical support.

This feature exposes the following information:

- **Globals** – Settings from the **Global Settings Editor** and more. Lists the values of the **PeterBlum.DES.Globals** object, many of which are loaded from the **custom.des.config** file and **web.config** file.
- **Page-Level** – Property values on the **PeterBlum.DES.Globals.WebFormDirector** object. If you are using the **PageManager** control, its settings will be reflected here too.
- **Validation** – Identifies all validators, buttons, **ValidationSummary** and **CombinedErrorMessage** controls on the page. Provides key settings to help you understand how they are working. A great way to determine why a button isn't firing validators.
- **AJAX** – How DES is setup to work with AJAX on this page. It identifies the framework and several key properties. It lists preloaded features. If the **AJAXManager.AllInAJAXUpdate** property is `false`, it also lists every DES control on the page, with its **InAJAXUpdate** property. Use it to determine if a control should be changed to **InAJAXUpdate=true**. It helps debug AJAX related problems and helps you optimize the performance of the page.
- **Overall configuration** – Identifies the file paths and URLs for the configuration files, Appearance Folder, Licenses folder, and the **GetFiles.aspx** form. Identifies the DES assemblies, by version and location. Lists all keys defined in the `<appSettings>` section of the **web.config** file.
- **Licenses** – Identifies the license files found, the License Keys, and which licenses are in use. For users of Web Server licenses, it is particularly valuable after initial deployment to production to be sure your production license files are in use instead of the limited non-production licenses.
- **Style Sheets** – Since style sheets can be merged and compressed, it helps to know if your styles are being delivered to the page. This lists the settings used by style sheets, including the URLs and file paths used to retrieve them. It also outputs the same text from your files that is sent to the browser.

## Using This Feature

When the page is run from `http://localhost` (or the IP address requesting the page is `127.0.0.1`), just add the parameter `DESDebug=` to the URL of the page. It will give you the DES Debug menu.

*Note: This feature can be disabled by using the `DES_DebugAllowedIPs` key in `<appSettings>` without declaring the IP address `127.0.0.1`.*

### Example

```
http://localhost/myfolder/myform.aspx?DESDebug=
```

When run from any other location, some additional setup is required to enable the `DESDebug` parameter or a custom parameter of your choice. See "Exploring the Current Settings" in the **General Features Guide**.

## Technical Support and Other Assistance

PeterBlum.com offers free technical support. This is just one of the ways to solve problems. This section provides all of your options and explains how technical support is setup.

### Installation and User's Guides

These guides are large because they are loaded with content. In many cases, the answers are in them. Most guides include Troubleshooting sections at the end. This information will often save you time.

### Developer's Kit

The Developer's Kit is a free download that provides documentation and sample code for building your own classes with this framework. It includes:

- Developer's Guide - Overviews of each class with examples, step-by-step guides, and other tools to develop new classes.
- MSDN-style help file - Browse through this help file to learn about all classes and their members.
- Sample code in C# and VB.

You can download it from <http://www.peterblum.com/DES/DevelopersKit.aspx>.

### PeterBlum.Com Forums

Use the forums <http://www.peterblum.com/forums.aspx> to discuss issues and ideas with other users.

### Getting Product Updates

As minor versions are released (5.0.1 to 5.0.2 is a minor version release), you can get them for free. Go to <http://www.PeterBlum.com/DES/Home.aspx>. It will identify the current version at the top of the page. You can read about all changes in the release by clicking "Release History". Click "Get This Update" to get the update. You will need the serial number and email address used to register for the license.

As upgrades are offered (v5.0 to v5.1 or 6), PeterBlum.com will determine if there is an upgrade fee at the time. You will be notified of upgrades and how to retrieve them through email.

PeterBlum.com often adds new functionality into minor version releases.

### Technical Support

You can contact Technical Support at this email address: [Support@PeterBlum.com](mailto:Support@PeterBlum.com). I (Peter Blum) make every effort to respond quickly with useful information and in a pleasant manner. As the only person at PeterBlum.com, it is easy to imagine that customer support questions will take up all of my time and prevent me from delivering to you updates and cool new features. As a result, I request the following of you:

- Please review the Users or Installation Guides, including their [Troubleshooting](#) section, first.
- Please try to include as much information about your web form or the problem as possible. I need to fully understand what you are seeing and how you have set things up.
- If you have written code that interacts with my controls or classes, please be sure you have run it through a debugger to determine that it is working in your code or the exact point of failure and error it reports.
- If you are subclassing from my controls, I provide the DES [Developer's Kit](#) that includes the [Developers Guide.pdf](#), [Classes And Types help file](#), and sample files. *I can only offer limited assistance as you subclass as this kind of support can be very time consuming.* I am interested in any feedback about my documentation's shortcomings so I can continue to improve it.
- I cannot offer general ASP.NET mentoring. If your problem is due to your lack of knowledge in ASP.NET, I will give you some initial help and then ask you to find assistance from the many tools available to the .Net community. They include:

- [www.asp.net](http://www.asp.net) forums and tutorials
- Google searches. (I virtually live in Google as I try to figure things out with ASP.NET.) <http://www.Google.com>. Don't forget to search the "Groups" section of Google!
- For DHTML, Microsoft provides an excellent guide at <http://msdn2.microsoft.com/en-us/library/ms533050.aspx>.
- For DOM, start with the DHTML guide. Topics that are also in DOM are noted under the heading "Standards Information"
- For JavaScript, I recommend <https://developer.mozilla.org/en/JavaScript/Reference>.
- <http://aspnet.4guysfromrolla.com/>, <http://www.aspalliance.com/>
- Books

As customers identify issues and shortcomings with the software and its documentation, I will consider updating these areas.

# Table of Contents

|   |           |
|---|-----------|
| <b>OVERVIEW</b>   | <b>2</b>  |
| Easy Installation with the Web Application Updater              | 2         |
| Installation That is Not Handled by the Web Application Updater | 2         |
| Platform Support  | 3         |
| <b>PREPARE A WEB APPLICATION OR SITE</b>                        | <b>5</b>  |
| Step by step instructions                                       | 6         |
| FAQs - Prepare a web application or site                        | 7         |
| <b>ADDING TO THE VISUAL STUDIO/VISUAL WEB DEVELOPER TOOLBOX</b> | <b>10</b> |
| FAQs – Adding to the Toolbox                                    | 11        |
| <b>USING ALTERNATIVE LOCATIONS FOR FILES IN THE DES FOLDER</b>  | <b>12</b> |
| <b>INSTALLING LICENSES</b>                                      | <b>14</b> |
| Trial License   | 15        |
| Consultant License  | 16        |
| Suite – Web Server Licenses                                     | 17        |
| Suite – Site License  | 19        |
| Suite – Redistribution License                                  | 20        |
| Peter’s Professional Validation – Web Server Licenses           | 21        |
| Peter’s Professional Validation – Site License                  | 23        |
| Peter’s Professional Validation – Redistribution License        | 24        |
| Peter’s More Validators – Web Server Licenses                   | 25        |
| Peter’s More Validators – Site License                          | 27        |
| Peter’s More Validators – Redistribution License                | 28        |
| Peter’s TextBoxes – Web Server Licenses                         | 29        |
| Peter’s TextBoxes – Site License                                | 31        |
| Peter’s TextBoxes – Redistribution License                      | 32        |
| Peter’s Date And Time – Web Server Licenses                     | 33        |

|   |           |
|---|-----------|
| Peter's Date And Time – Site License                          | 35        |
| Peter's Date And Time – Redistribution License                | 36        |
| Peter's Interactive Pages – Web Server Licenses               | 37        |
| Peter's Interactive Pages – Site License                      | 39        |
| Peter's Interactive Pages – Redistribution License            | 40        |
| Peter's Input Security – Web Server Licenses                  | 41        |
| Peter's Input Security – Site License                         | 43        |
| Peter's Input Security – Redistribution License               | 44        |
| Application_Start vs. Application_OnStart                     | 45        |
| When the Web Application does not have a Global.asax file     | 46        |
| Troubleshooting Licenses                                      | 47        |
| <b>CONVERTING FROM TRIAL VERSION TO PAID VERSION</b>          | <b>51</b> |
| <b>UPGRADING FROM DES V4.0</b>                                | <b>52</b> |
| FAQs – Upgrading from DES v4.0                                | 52        |
| <b>CONVERTING FROM PROFESSIONAL VALIDATION AND MORE</b>       | <b>54</b> |
| FAQs – Converting from Professional Validation and More       | 54        |
| <b>CONVERTING FROM PETER'S DATE PACKAGE</b>                   | <b>59</b> |
| What you should know before converting                        | 59        |
| Steps for Migration   | 59        |
| Migrating your Style Sheet Classes                            | 60        |
| FAQs – Converting from Peter's Date Package                   | 65        |
| <b>CONVERTING FROM ASP.NET 2.0 TO ASP.NET 4.0</b>             | <b>67</b> |
| <b>CONVERT NATIVE CONTROLS TO THEIR DES EQUIVALENTS</b>       | <b>68</b> |
| Differences in ValidatorCalloutExtender Controls              | 69        |
| <b>ADD OR CHANGE PETER'S BUSINESS LOGIC DRIVEN UI SUPPORT</b> | <b>71</b> |
| Installing Peter's Business Logic Driven UI                   | 71        |
| <b>DEPLOYING YOUR WEB APPLICATIONS</b>                        | <b>72</b> |

|   |           |
|---|-----------|
| <b>Confirming a Deployment</b>                        | <b>73</b> |
| <b>INSTALLING DES ONTO A NEW DEVELOPMENT COMPUTER</b> | <b>74</b> |
| <b>INSTALLING INTO A PARTIAL TRUST ENVIRONMENT</b>    | <b>75</b> |
| <b>USING IIS 7</b>                                    | <b>77</b> |
| <b>TROUBLESHOOTING</b>                                | <b>79</b> |
| What to do when you get version errors                | 79        |
| Visual Studio Problems                                | 80        |
| Exploring the Current Settings                        | 81        |
| <b>TECHNICAL SUPPORT AND OTHER ASSISTANCE</b>         | <b>82</b> |