

PETER'S DATA ENTRY SUITE INSTALLATION GUIDE



Click on any of these topics to jump to them:

- ◆ Easy Installation with the Web Application Updater
- ◆ First Time Installation
- ◆ Adding To The Visual Studio/Visual Web Developer Toolbox
- ◆ Installing a Service Release
- ◆ Installing Licenses
- ◆ Convert Native Controls to their DES equivalents

- ◆ Converting from Trial Version To Paid Version
- ◆ Upgrading from Professional Validation And More
- ◆ Migrating from Peter's Date Package
- ◆ Converting from ASP.NET 1.x to ASP.NET 2.0

- ◆ Deploying Your Web Applications
- ◆ Installing DES onto a New Development Computer
- ◆ Using Alternative Locations for Files in the DES Folder

- ◆ Exploring The Current Settings
- ◆ Installing Into A Partial Trust Environment
- ◆ Using IIS 7
- ◆ Troubleshooting
- ◆ Table of Contents

Table of Contents

OVERVIEW	5
Easy Installation with the Web Application Updater	5
Installation That is Not Handled by the Web Application Updater	5
WHAT'S INCLUDED	7
FIRST TIME INSTALLATION	8
FAQs - First Time Installation	10
ADDING TO THE VISUAL STUDIO/VISUAL WEB DEVELOPER TOOLBOX	12
FAQs – Adding to the Toolbox	13
INSTALLING A SERVICE RELEASE	14
INSTALLING LICENSES	15
Trial License	16
Consultant License	17
Suite – Web Server Licenses	18
Suite – Site License	20
Suite – Redistribution License	21
Peter’s Professional Validation – Web Server Licenses	22
Peter’s Professional Validation – Site License	24
Peter’s Professional Validation – Redistribution License	25
Peter’s More Validators – Web Server Licenses	26
Peter’s More Validators – Site License	28
Peter’s More Validators – Redistribution License	29
Peter’s TextBoxes – Web Server Licenses	30
Peter’s TextBoxes – Site License	32
Peter’s TextBoxes – Redistribution License	33
Peter’s Date And Time – Web Server Licenses	34
Peter’s Date And Time – Site License	36

Peter's Date And Time – Redistribution License	37
Peter's Interactive Pages – Web Server Licenses	38
Peter's Interactive Pages – Site License	40
Peter's Interactive Pages – Redistribution License	41
Peter's Input Security – Web Server Licenses	42
Peter's Input Security – Site License	44
Peter's Input Security – Redistribution License	45
Application_Start vs. Application_OnStart	45
When the Web Application does not have a Global.asax file	46
Troubleshooting Licenses	47
CONVERTING FROM TRIAL VERSION TO PAID VERSION	51
UPGRADING FROM PROFESSIONAL VALIDATION AND MORE	52
FAQs – Upgrading from Professional Validation And More	52
MIGRATING FROM PETER'S DATE PACKAGE	57
FAQs – Migrating from Peter's Date Package	72
CONVERTING FROM ASP.NET 1.X TO ASP.NET 2.0	74
CONVERT NATIVE CONTROLS TO THEIR DES EQUIVALENTS	76
Steps for Converting	76
Differences in ValidatorCalloutExtender Controls	77
DEPLOYING YOUR WEB APPLICATIONS	79
Confirming a Deployment	80
INSTALLING DES ONTO A NEW DEVELOPMENT COMPUTER	81
INSTALLING ASP.NET DESIGN MODE EXTENDER	82
FAQs – Installing ADME	82
USING ALTERNATIVE LOCATIONS FOR FILES IN THE DES FOLDER	83
INSTALLING INTO A PARTIAL TRUST ENVIRONMENT	85

USING IIS 7	87
TROUBLESHOOTING	89
What to do when you get version errors	89
Visual Studio Problems	90
Exploring The Current Settings	91
TECHNICAL SUPPORT AND OTHER ASSISTANCE	92

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 **[ProductFolder]**. Here are the actions you can take with the Web Application Updater:

- First time installation – Add its assemblies, support folder, and optionally convert Microsoft controls to their DES equivalents. It will update your Visual Studio project files if needed. See "First Time Installation".
- Install a service release – Run this each time you get a new service release. It will replace the assemblies and scripts in your web application. See "Installing a Service Release".
- Upgrade from Professional Validation And More – For **Professional Validation And More** ("VAM") users to upgrade their application's VAM controls to DES controls. It will install the assemblies and support folder if needed. It will convert the VAM controls in all of your web forms and code files to their DES equivalents. After running this, most users will be fully ready to run under DES. *Users might encounter problems will have used undocumented client or server side functions. Compiler errors should assist in resolving these.* See "Upgrading from Professional Validation And More".
- Migrate from Peter's Date Package – **Peter's Date Package** users can migrate to the DES equivalent controls. This is not required. Peter's Date Package controls can coexist with DES. You only need to migrate if you want to take advantage of new features. The **Web Application Updater** does only part of the migration. Please see "Steps for Migration" to finish the process which moves your style sheets and images from Peter's Date Package to DES.
- Convert Native Controls to their DES equivalents – If you have an existing web application with DES installed but are still using the Microsoft validation and textbox controls, you can quickly change them into their DES equivalents. See "Convert Native Controls to their DES equivalents".

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**.

Terminology

In this documentation, the term **[ProductFolder]** refers to the folder where you installed **Peter's Data Entry Suite**. For example: C:\Program Files\Peters Data Entry Suite v4.0.7.

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 Studio 2005's ASP.NET Development Server, it is usually **[My Documents]\Visual Studio 2005\Websites\MyWebApp**. For example, the web app "MyWebApp" could be in C:\documents and settings*user name*\my documents\Visual Studio 2005\Websites\MyWebApp.

What's Included

When you extracted **Peter's Data Entry Suite**, it created a folder with a number of files and folders. Here is a description of each file or folder:

- License Agreement.pdf – The License Agreement. PLEASE READ IT.
- Web Application Updater.exe – Adds various features to your web application. See “Easy Installation with the Web Application Updater”.
- Installation Guide.pdf – This file. (Peter's Input Security has additional installation requirements in its own Installation Guide.)
- General Features.pdf – Features used throughout the various modules of the product, including the PageManager control, NativeControlExtender control, AJAX, String Lookup, and ViewStateMgr.
- Validation Users Guide.pdf – The Validation controls modules. Applies if you have a license for the Suite, Peter's Professional Validation, or Peter's More Validators.
- TextBoxes Users Guide.pdf – The Peter's Textboxes module. Applies if you have a license for the Suite or Peter's TextBoxes.
- Interactive Pages Users Guide.pdf – The Peter's Interactive Pages module. Applies if you have a license for the Suite or Peter's Interactive Pages.
- Date and Time Users Guide.pdf – The Peter's Date and Time module. Applies if you have a license for the Suite or Peter's Date and Time.
- Peters Input Security – Peter's Input Security files. It is a separate installation. This folder contains its own Installation Guide and User's Guide for this module.
- Tutorial.pdf – Step-by-step examples of using the controls.
- Using Third Party Controls.pdf – Assistance with integrating various third party controls.
- What's New.pdf – For upgraders from Professional Validation And More and Peter's Date Package.
- Assemblies\ASPNET 1_x – A folder containing the assemblies for Peter's Data Entry Suite that are intended for ASP.NET 1.0 and ASP.NET 1.1 web sites. *These are installed automatically with the Web Application Updater.* PeterBlum.DES.dll is the primary assembly. PeterBlum.DES.NativeValidators.dll contains Microsoft native validators used with the controls of Peter's TextBoxes and Peter's Date and Time when you do not have a license for Peter's Professional Validation.
- Assemblies\ASPNET 2_0 – A folder containing the assemblies for Peter's Data Entry Suite that are intended for ASP.NET 2.0 web sites. *These are installed automatically with the Web Application Updater.*
- Copy To Web Application Folder – A folder containing the **DES** folder. *This is installed automatically with the Web Application Updater.* The **DES** folder contains DES's XML configuration files, **DES.config** and **custom.DES.config**. The subfolders contain the default Cascading Style Sheet and image files, and the location of license file(s) issued to you.
- Development Computer Setup – Used with Visual Studio 2002 and 2003 to install the ASP.NET Design Mode Extender (“ADME”). DES uses ADME both in design mode and at runtime. It must be installed on development computers and any web server that will access DES's assemblies. See “Installing ASP.NET Design Mode Extender”.
- StringLookup – Folder of resources for using the string lookup (localization) features of DES. See the topic “The String Lookup System” in the **General Features Guide**.
- Global Settings Editor.exe – Run this to edit the numerous global properties in the **custom.DES.config** file. See the topic “Global Settings: The Editor and custom.DES.config File” in the **General Features Guide**.
- Third Party Controls – Extensions to DES for the various third party controls. Includes Telerik RadControls, Infragistics NetAdvantage, and EasyListBox. See the **Using Third Party Controls** document.
- When Missing Global.asax – Starter files for adding **Global.asax** to your web application.

First Time Installation

Note: If you are updating an existing DES installation, see “Installing a Service Release” for quick steps.

Note: This process automatically modifies your web application using the Web Application Updater program. It 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 - First Time Installation”. If you still have concerns, email 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.
1. Exit from Visual Studio and Visual Web Developer, if opened.
 2. Run the program **Web Application Updater.exe**, found in the **[ProductFolder]**, or from the **Start** menu.
 3. Click **Next**.
 4. Select the first radio button, **Add to a web application (first time)**.
 5. If you want to use DES for your validation instead of the native Microsoft validators, or use DES TextBoxes instead of the native Microsoft TextBox, mark the checkbox **Also convert native controls to their DES equivalents** and make a selection from the **Type of Control** dropdownlist. *For details on which controls are converted, see “Convert Native Controls to their DES equivalents”.*
 6. Follow the prompts.
 7. 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 with a detailed description of the problem, the log file, and any web application files that are concerning you.

8. Install your trial or paid licenses. See “Installing Licenses”.

Problems?

See “Troubleshooting Licenses”.

9. 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”.
10. Your web application is ready to use. DES provides several additional tools that you can use in your web applications:
 - 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**.

MORE ON THE NEXT PAGE

- 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**.
- If you are using third party controls, they may require additional setup. See **Using Third Party Controls.pdf**.
- 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** guide.

FAQs - First Time Installation

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

First use the **Web Application Updater** for a first time installation. It will prompt you to select the type of assembly to install: ASP.NET 1.x, ASP.NET 2.0, or None. Select None.

Then install all of the assemblies found in the **[ProductFolder]\Assemblies\[version of ASPNET]** folder into the Global Assembly Cache.

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.

2. 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.

3. 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**.

4. 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.

5. 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.

6. 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".

7. I didn't convert Microsoft web controls to their DES equivalents. Can I do so later?

Yes. Run the **Web Application Updater** 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.

8. I have **Professional Validation And More** installed in this web application. Is it converted?

No. See "Upgrading from Professional Validation And More".

9. I have **Peter's Date Package** installed in this web application. Is it converted?

No. See "Migrating from Peter's Date Package".

10. 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”.

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

See “Deploying Your Web Applications”.

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

See “Confirming a Deployment”.

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

See “Convert Native Controls to their DES equivalents”.

14. 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.

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:

Note: **Visual Studio 2002/3 Users ONLY** *If you have not done so, please install ADME into the development computer now. See “Installing ASP.NET Design Mode Extender”.*

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. Choose **Customize Toolbox** (VS 2002), **Add/Remove Toolbox Items** (VS 2003), or **Choose Items** (VS 2005, 2008, 2010 and Visual Web Developer).
6. Select the **.Net Framework Components** tab.
7. Click **Browse** and select the **PeterBlum.DES.dll**.
 - Visual Studio 2002/3 users will find it in **[ProductFolder]\Assemblies\ASPNET 1_x**.
 - Visual Studio 2005, 2008, 2010 and Visual Web Developers users should use the assembly in their web application’s **\bin** folder. (If it is not setup there, find it in **[ProductFolder]\Assemblies\ASPNET 2_0**.)

Note: See FAQ #2 below for why the \bin folder is preferred.
8. Click **OK**. Over 60 controls are added to the “Peter’s Data Entry Suite” tab.
9. 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.

Peter’s Data Entry Suite	DES Validators	DES TextBoxes	DES Date and Time	DES Interactive Pages
Button	All validators 	CurrencyTextBox	AnniversaryTextBox	CalculationController
ImageButton	CombinedErrorMessages	DecimalTextBox	Calendar	ContextMenu
LinkButton	RequiredFieldMarker	FilteredTextBox	DateTextBox	FieldStateController
LocalizableLabel	RequiredFieldsDescription	IntegerTextBox	DurationTextBox	FSCOnCommand
NativeControlExtender	ValidationSummary	MultiSegmentDataEntry	MonthYearPicker	MultiFieldStateController
PageManager		PercentTextBox	MonthYearTextBox	MultiFSCOnCommand
		TextBox	PopupCalendar	TextCounter
			PopupMonthYearPicker	
			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 DES 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 [ProductFolder]?

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 **[ProductFolder]**, 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.

Installing a Service Release

1. Exit from Visual Studio and Visual Web Developer, if opened.
2. Run the program **Web Application Updater.exe**, found in the **[ProductFolder]** or in the **Start** menu.
3. Click **Next**.
4. Select the second radio button, **Update a web application (service release)**.
5. Follow the prompts.
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 with a detailed description of the problem, the log file, and any web application files that are concerning you.

7. If you are using the Visual Studio toolbox, you may experience problems adding a control because Visual Studio 2005, 2008, 2010 and Visual Web Developer will continue to look for the assembly used in the original installation. It is recommended that you always add the **PeterBlum.DES.dll** assembly from the **\bin** folder of your web application into the toolbox to limit the impact.

If you have not done this previously, delete your current entries in the toolbox and add the new **PeterBlum.DES.dll** from your **\bin** folder. See "Adding To The Visual Studio/Visual Web Developer Toolbox".

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:

Trial License Consultant License

Suite (Complete Product)

Web Server Site Redistribution

Peter’s Professional Validation Module

Web Server Site Redistribution

Peter’s More Validators Module

Web Server Site Redistribution

Peter’s TextBoxes Module

Web Server Site Redistribution

Peter’s Date and Time Module

Web Server Site Redistribution

Peter’s Interactive Pages Module

Web Server Site Redistribution

Peter’s Input Security Module

Web Server Site Redistribution

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 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.Page.Suite_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.ProfessionalValidation_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.MoreValidators_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.TextBoxes_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.DateAndTime_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.InteractivePages_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 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.Page.InputSecurity_LicenseKey**.

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

For example, the serial number is 999-111222333:

```
PeterBlum.DES.Globals.Page.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 **[ProductFolder]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	\Licenses
DES PV	\Licenses\Professional Validation
DES MV	\Licenses\More Validators
DES TB	\Licenses\Text Boxes
DES DT	\Licenses\Date and Time
DES IP	\Licenses\Interactive Pages
DES IS	\Licenses\Input Security

2. Assign the Licensing Key to the **PeterBlum.DES.Globals.modulename_LicenseKey** or **PeterBlum.DES.Globals.Page.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.Page.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.Page.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 to order the product.

Converting from Trial Version To Paid Version

1. Update the License Files and License Key. See “Installing Licenses”.
2. Let the **Web Application Updater** determine if the product you downloaded with the paid version is newer. Use “Installing a Service Release”.

Upgrading from Professional Validation And More

Use these instructions to install an upgrade from any version of **Professional Validation And More**.

1. Exit from Visual Studio and Visual Web Developer, if opened.
3. Run the program **Web Application Updater.exe**, found in the **[ProductFolder]** or in the **Start** menu.
4. Click **Next**.
5. Select the third radio button, **Upgrade from Professional Validation And More**.
6. Follow the prompts.
7. 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 with a detailed description of the problem, the log file, and any web application files that are concerning you.

PLANNING AHEAD: When your page uses AJAX to update its controls, you **must** follow the steps described in each User's 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.*

FAQs – Upgrading 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 **[VAM product 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). 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;
}

.DESVALCheckboxWithError
{
    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;
}

.DESVALCheckboxWithError
{
    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.

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

This is a partial list.

- Changed “PeterBlum.VAM” to “PeterBlum.DES”. PeterBlum.DES is the new namespace and assembly name.
- Changed some references of “PeterBlum.VAM.Security” to “PeterBlum.DES”.
- Changed these class names starting with VAM to start with DES: VAMPage, VAMTypeConverter, and VAMException.
- 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.
- Commented out old LicenseKeys
- Inserted the line <%= PeterBlum.DES.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 your 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 **[VAM Product 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

Migrating 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 migrating

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 some 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. Here are the actions that you will take when the **Web Application Updater** is finished.

- If you modified the Peter's Date Package style sheet files or created your own classes, you will have to apply them to the new style sheet files in DES.
- 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 have similar functionality in DES, but 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

1. Exit from Visual Studio and Visual Web Developer, if opened.
2. Run the program **Web Application Updater.exe**, found in the **[ProductFolder]** or in the **Start** menu.
3. Click **Next**.
4. Select the third radio button, **Migrate from Peter's Date Package**.
5. Follow the prompts.
6. 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

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 with a detailed description of the problem, the log file, and any web application files that are concerning you.

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.

As a result, the best approach is to utilize the new style sheet files and merge your edited style sheet classes in.

What follows are charts to take you through the differences, control by control. The charts assume that you will modify the existing styles of the style sheet files and change the value of the associated property from any style name you were using to the style defined in the file.

DateTextBox, AnniversaryTextBox, MonthYearTextBox, TimeOfDayTextBox, DurationTextBox.

Textbox and help button only. Popups are handled elsewhere

PDP file*: PetersDatePackage\Appearance\DateTextBox.css and TimeTextBox.css

DES file: DES\Appearance\Date And Time\DateAndTimeTextBoxes.css

PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	<u>additions or edits</u> deletions
DTBToggleCalendar DES_CalPopup	xPopupCalendar.CssClass PopupCalendar.CssClass	Moved to Calendar.css No change
DTBToggleCalendarPressed DES_CalPopupPressed	xPopupCalendar. xPressedCssClass n/a	Moved to Calendar.css Merges with DES_CalPopup when used color: <u>darkblue</u> ;
DTBHelpButton and TMTB_HelpButton DES_DATHelpButton	xHelpButtonCSSClass HelpButtonCssClass	No change
DTBHelpButtonPressed and TMTB_HelpButtonPressed DES_DATHelpButtonPressed	xHelpButtonPressedCSSClass n/a	No change Merges with DES_DATHelpButton when used

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

Calendar		
<i>Popups are handled elsewhere</i>		
PDP file*: PetersDatePackage\Appearance\Calendar.css DES file: DES\Appearance\Date And Time\Calendar.css		
PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	additions or edits deletions
CSCCalendar DES_CalControl	CssClass CssClass	<i>Moved to Calendar.css</i> <i>No changes for v2 users</i> <i>For v1.x users:</i> <code>border-right: #E0E0E0 ridge 4px;</code> <code>border-top: #E0E0E0 ridge 4px;</code> <code>font-size: 8pt;</code> <code>filter: progid:...</code> <code>border-left: #E0E0E0 ridge 4px;</code> <code>border-bottom: #E0E0E0 ridge 4px;</code> <code>font-family: Arial;</code> <code>background-color: white;</code>
CSC_CalendarNoFilter n/a	CssClass CssClass	<i>OBSOLETE. Assign CssClass to DES_CalControl.</i>
CSCWeekRowsTable DES_CalWeekRowsTable	xWeekRowCSSClass WeekRowCssClass	<i>No changes</i>
CSCDay DES_CalDay	xDayCSSClass DayCssClass	<i>Get the border colors shown below from the Calendar.xDayCellBorderColor property.</i> <code>color: Black;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code> <i>/* only specify color of borders. Lines are added programmatically */</i> <code>border-right-color: #808080;</code> <code>border-top-color: #808080;</code> <code>border-left-color: #808080;</code> <code>border-bottom-color: #808080;</code>
CSCSelectedDay DES_CalSelected	xSelectedCSSClass SelectedCssClass	<i>The attributes here are merged with those in DES_CalDay at runtime.</i> <code>background-color: gold !important;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code>
CSCSelectedToday n/a	xSelectedTodayCSSClass n/a	<i>OBSOLETE. DES_CalToday merges with DES_CalDay and DES_CalSelected at runtime</i>
CSCToday DES_CalToday	xTodayCSSClass TodayCssClass	<i>The attributes here are merged with those in DES_CalDay at runtime.</i> <code>font-weight: bold;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code>
CSCSpecialDay DES_CalSpecialDay	xSpecialCSSClass SpecialCssClass	<i>The attributes here are merged with those in DES_CalDay at runtime.</i> <code>background-color: #ffffad;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code>

CSCOtherMonthDays DES_CalOtherMonthDay	xOtherMonthDayCSSClass OtherMonthDayCssClass	Get the border colors shown below from the Calendar.xOtherMonthDayCellBorderColor property. color: gray; cursor: pointer; cursor: hand; /* only specify color of borders. Lines are added programmatically */ border-right-color: #C0C0C0; border-top-color: #C0C0C0; border-left-color: #C0C0C0; border-bottom-color: #C0C0C0;
CSCOutsideRangeDay DES_CalOutsideRangeDay	xOutsideRangeCSSClass OutsideRangeCssClass	Get the border colors shown below from the Calendar.xDayCellBorderColor property. color: grey; cursor: default; /* only specify color of borders. Lines are added programmatically */ border-right-color: #808080; border-top-color: #808080; border-left-color: #808080; border-bottom-color: #808080;
CSCWeekNumberColumn DES_CalWeekNumbers	xWeekNumberCssClass WeekNumberCssClass	No change for v2 users For v1.x users: border-bottom: blue 1px solid; background-color: lightblue;
CSCDayHeader DES_CalDayHeader	xDayHeaderCSSClass DayHeaderCssClass	No change for v2 users For v1.x users, REMOVE CSCDayHeader TR and use this for DES_CalDayHeader: border-bottom: blue 1px solid; font-size: 7pt; background-color: lightblue;
CSCDayHeaderCell DES_CalDayHeaderCell	xDayHeaderCellCSSClass DayHeaderCellCssClass	No change for v2 users For v1.x users: font-size: 7pt;
CSCHeaderTable DES_CalHeader	xHeaderCSSClass HeaderCssClass	No change for v2 users For v1.x users: margin-top: 2px; margin-bottom: 2px; border-bottom: blue 1px solid; font-family: Arial; font-size: 9pt;
CSCHeaderRows DES_CalHeaderRows	xHeaderRowsCSSClass xHeaderRowsCssClass	No change
CSCIncrementalButtons DES_CalMonthButton	xIncrementalButtonCSSClass IncrementalButtonCssClass	No change
CSCIncrementalButtonsPressed DES_CalMonthButtonPressed	xIncrementButtonPressed CSSClass n/a	Merges with DES_CalMonthButton when used color: darkblue; cursor: pointer; cursor: hand;
CSCJumpButtons DES_CalYearButton	xJumpButtonCSSClass JumpButtonCssClass	No change
CSCJumpButtonsPressed DES_CalYearButtonPressed	xJumpButtonPressedCSSClass n/a	Merges with DES_CalMonthButton when used color: darkblue; cursor: pointer; cursor: hand;
CSCCloseButton DES_CalCloseButton	xCloseButtonCSSClass CloseButtonCssClass	No change

CSCCloseButtonPressed DES_CalCloseButtonPressed	xCloseButtonPressedCSSClass CloseButtonPressedCssClass	No change Merges with DES_CalCloseButton when used
CSCHelpButton DES_CalHelpButton	xHelpButtonCSSClass HelpButtonCssClass	No change
CSCHelpButtonPressed DES_CalHelpButtonPressed	xHelpButtonPressedCSSClass HelpButtonPressedCssClass	No change Merges with DES_CalHelpButton when used
CSCFooterTable DES_CalCommands	xFooterCSSClass CommandAreaCssClass	No change
CSCFooterButton DES_CalCommandButton	xFooterButtonCSSClass CommandButtonCssClass	No change
CSCFooterButtonPressed DES_CalCommandButton Pressed	xFooterButtonPressed CSSClass n/a	Merges with DES_CalCommandButton when used border-right: #E0E0E0 thin inset; border-top: #E0E0E0 thin inset; border-left: #E0E0E0 thin inset; border-bottom: #E0E0E0 thin inset; background-color: lightblue; color: black; font-size: 8pt; font-family: Arial; cursor: pointer; cursor: hand;
CSCFooterCurrentDate DES_CalCurrentDateView	xCurrentDateCSSClass CurrentDateCssClass	No change
CSCMultiMonthContainer DES_CalMultiMonthContainer	xMultiMonthContainerCSSClass MultiMonthContainerCssClass	No change
CSCMultiMonthHeader DES_CalMultiMonthHeader	xMultiMonthHeaderCSSClass MultiMonthHeaderCssClass	No change

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

MonthYearPicker and PopupMonthYearPicker

Textbox and help button only. Popups are handled elsewhere

PDP file*: PetersDatePackage\Appearance\MonthYearPicker.css

DES file: DES\Appearance\Date And Time\MonthYearPicker.css

PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	additions or edits deletions
MYPMonthYearPicker DES_MYPControl	CssClass CssClass	border-right: gray 1px solid; border-top: gray 1px solid; border-left: gray 1px solid; border-bottom: gray 1px solid; font-family: Arial; font-size: 8pt; background-color: white; cursor:default;
MYPHeader DES_MYPHHeader	xHeaderCSSClass HeaderCssClass	background-color:transparent; padding-left: 2px; padding-right: 2px; width: 100%;
MYPOutOfRangeCell DES_MYPOutOfRangeCell	xOutOfRangeCellsCSSClass OutOfRangeCellsCssClass	No change
MYPMonthTable DES_MYPMonthTable	xMonthTableCSSClass MonthTableCssClass	No change
MYPYearTable DES_MYPYearTable	xYearTableCSSClass YearTableCssClass	No change
MYPMonthCell DES_MYPMonthCell	xMonthCellsCSSClass MonthCellsCssClass	No change
MYPSelectedMonthCell DES_MYPSelectedMonthCell	xSelectedMonthCellsCSSClass SelectedMonthCellsCssClass	No change
MYPYearCell DES_MYPYearCell	xYearCellsCSSClass YearCellsCssClass	No change
MYPMoveYearsButtons DES_MYPYearButton	xMoveYearsButtonsCSSClass MoveYearsButtonsCssClass	color: mediumblue; cursor: pointer; cursor: hand;
MYPMoveYearsButtons Pressed DES_MYPYearButtonPressed	xMoveYearsButtonsPressed CSSClass n/a	Merges with DES_MYPYearButton when used color: darkblue;
MYPTogglePicker DES_MYPPopup	PopupMonthYearPicker. CssClass PopupMonthYearPicker. CssClass	No change
MYPTogglePickerPressed DES_MYPPopupPressed	PopupMonthYearPicker. xPressedCSSClass n/a	Merges with DES_MYPPopup when used color: darkblue;
FYPYearTitle DES_FYPYearTitle	xYearTitleCSSClass YearTitleCssClass	No change

FYPFooterButton DES_FYPFooterButton	xFooterButtonCSSClass FooterButtonCssClass	border-right: #E0E0E0 thin outset; border-top: #E0E0E0 thin outset; border-left: #E0E0E0 thin outset; border-bottom: #E0E0E0 thin outset; background-color: lightblue; font-family: Arial; font-size: 8pt; color: black; cursor: pointer; cursor: hand; margin-left: 20px; margin-right: 20px; margin-top: 4px; margin-bottom: 4px;
FYPFooterButtonPressed DES_FYPFooterButton Pressed	xFooterButtonPressed CSSClass n/a	Merges with DES_FYPFooterButton when used border-right: #E0E0E0 thin inset; border-top: #E0E0E0 thin inset; border-left: #E0E0E0 thin inset; border-bottom: #E0E0E0 thin inset; background-color: lightblue; font-family: Arial; font-size: 8pt; color: black; cursor: pointer; cursor: hand;

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

MultiSelectionCalendar

Popups are handled elsewhere

PDP file*: PetersDatePackage\Appearance\MultiSelectionCalendar.css

DES file: DES\Appearance\Date And Time\ MultiSelectionCalendar.css

PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	<u>additions or edits</u> deletions
MSCMessageCenter DES_MSCMessageCenter	xMessageCenterCssClass MessageCenterCssClass	No change
MSCMessageCenterTitle DES_MSCMessageCenterTitle	xMessageCenterTitleCssClass MessageCenterTitleCssClass	No change
MSCMessageCenterBody DES_MSCMessageCenterBody	xMessageCenterBodyCssClass MessageCenterBodyCssClass	No change
MSCMessageCenterNextClick DES_MSCMessageCenter NextClick	MessageCenterNextClick CssClass xMessageCenterNextClick CssClass	No change

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

SpecialDates		
PDP file*: PetersDatePackage\Appearance\SpecialDates.css		
DES file: DES\Appearance\Date And Time\ SpecialDates.css		
PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	additions or edits deletions
SD_Selectable1 DES_SDSelectable1	Various in SpecialDates and each item added to SpecialDates.Dates.	background-color: mistyrose; cursor: pointer; cursor: hand; /* only specify color of borders. Lines are added programmatically */ border-right-color: #808080; border-top-color: #808080; border-left-color: #808080; border-bottom-color: #808080;
SD_SelectableOM1 DES_SDSelectableOM1	Various in SpecialDates and each item added to SpecialDates.Dates.	background-color: powderblue; cursor: pointer; cursor: hand; /* only specify color of borders. Lines are added programmatically */ border-right-color: #C0C0C0; border-top-color: #C0C0C0; border-left-color: #C0C0C0; border-bottom-color: #C0C0C0;
SD_Selectable2 DES_SDSelectable2	Various in SpecialDates and each item added to SpecialDates.Dates.	background-color: palegoldenrod; cursor: pointer; cursor: hand; /* only specify color of borders. Lines are added programmatically */ border-right-color: #808080; border-top-color: #808080; border-left-color: #808080; border-bottom-color: #808080;
SD_SelectableOM2 DES_SDSelectableOM2	Various in SpecialDates and each item added to SpecialDates.Dates.	background-color: gainsboro; cursor: pointer; cursor: hand; /* only specify color of borders. Lines are added programmatically */ border-right-color: #C0C0C0; border-top-color: #C0C0C0; border-left-color: #C0C0C0; border-bottom-color: #C0C0C0;
SD_Unselectable1 DES_SDUnselectable1	Various in SpecialDates and each item added to SpecialDates.Dates.	color: gray; background-color: mistyrose; /* only specify color of borders. Lines are added programmatically */ border-right-color: #808080; border-top-color: #808080; border-left-color: #808080; border-bottom-color: #808080;
SD_UnselectableOM1 DES_SDUnselectableOM1	Various in SpecialDates and each item added to SpecialDates.Dates.	color: gray; background-color: powderblue; /* only specify color of borders. Lines are added programmatically */ border-right-color: #C0C0C0; border-top-color: #C0C0C0; border-left-color: #C0C0C0; border-bottom-color: #C0C0C0;

SD_Unselectable2 DES_SDUnselectable2	Various in SpecialDates and each item added to SpecialDates.Dates.	<code>color: gray;</code> <code>background-color: palegoldenrod;</code> <code>/* only specify color of borders. Lines are added programmatically */</code> <code>border-right-color: #808080;</code> <code>border-top-color: #808080;</code> <code>border-left-color: #808080;</code> <code>border-bottom-color: #808080;</code>
SD_UnselectableOM2 DES_SDUnselectableOM2	Various in SpecialDates and each item added to SpecialDates.Dates.	<code>color: gray;</code> <code>background-color: gainsboro;</code> <code>/* only specify color of borders. Lines are added programmatically */</code> <code>border-right-color: #C0C0C0;</code> <code>border-top-color: #C0C0C0;</code> <code>border-left-color: #C0C0C0;</code> <code>border-bottom-color: #C0C0C0;</code>
SD_TimeRow DES_SDTimeRow		<i>No change</i>
SD_TimeRowAlt DES_SDTimeRowAlt		<i>No change</i>
EDFMain DES_EDFMain	xExpandedDateFormatter .xCssClass ExpandedDateFormatter .CssClass	<code>font-size: 8pt;</code> <code>font-family: Arial;</code> <code>color: black;</code> <code>background-color: aliceblue;</code> <code>border-right: black 1px solid;</code> <code>border-top: black 1px solid;</code> <code>border-left: black 1px solid;</code> <code>border-bottom: black 1px solid;</code> <code>padding-right: 5px;</code> <code>padding-left: 5px;</code> <code>padding-bottom: 5px;</code> <code>padding-top: 5px;</code> <code>max-height: 400px;</code> <code>overflow:auto;</code>
EDFTitle DES_EDFTitle	xTitleCSSClass TitleCssClass	<i>No change</i>

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

TimePicker and PopupTimePicker		
PDP file*: PetersDatePackage\Appearance\TimePicker.css		
DES file: DES\Appearance\Date And Time\TimePicker.css		
PDP style sheet class name	PDP PropertyName	Style Attribute Changes
DES style sheet class name	DES PropertyName	additions_or_edits deletions
TPTimePicker DES_TPControl	CssClass CssClass	<i>No change</i>
TPTimeSpan DES_TPTimeCell	xTimeSpanCSSClass TimeValueCssClass	<code>cursor: pointer;</code> <code>cursor: hand;</code> <code>white-space : nowrap; /* recommended */</code>
TPSelectedTimeSpan DES_TPSelected	xSelectedTimeSpanCssClass SelectedTimeValueCssClass	<code>background-color: gold;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code> <code>white-space : nowrap; /* recommended */</code>
TPFiller DES_TPFillerCell	xTimeSpanCSSClass TimeValueCssClass	<code>color: gray;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code> <code>white-space : nowrap; /* recommended */</code>
TPTitle DES_TPTitle	xTitleCSSClass TitleCssClass	<code>background-color: gold;</code> <code>border-top: black 1px solid;</code> <code>border-left: black 1px solid;</code> <code>white-space : nowrap; /* recommended */</code>
TPHeader DES_TPHeader	xHeaderTitleCSSClass HeaderTitleCssClass	<code>background-color:transparent;</code> <code>padding-left: 2px;</code> <code>padding-right: 2px;</code>
TPFooterButton TPCommandButton	xFooterButtonCSSClass FooterButtonCssClass	<i>No change</i>
TPFooterButtonPressed DES_TPCommandButton Pressed	xFooterButtonPressed CSSClass n/a	<i>Merges with DES_TPCommandButton when used</i> <code>border-right: #E0E0E0 thin inset;</code> <code>border-top: #E0E0E0 thin inset;</code> <code>border-left: #E0E0E0 thin inset;</code> <code>border-bottom: #E0E0E0 thin inset;</code> <code>background-color: lightblue;</code> <code>font-family: Arial;</code> <code>font-size: 8pt;</code> <code>color: black;</code> <code>cursor: pointer;</code> <code>cursor: hand;</code>
TPTogglePicker DES_TPPopup	PopupTimePicker.CssClass PopupTimePicker.CssClass	<i>No change</i>
TPTogglePickerPressed DES_TPPopupPressed	PopupTimePicker. xPressedCSSClass n/a	<i>Merges with DES_TPPopup when used</i> <code>color: darkblue;</code>

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

ContextMenu

PDP file*: PetersDatePackage\Appearance\ContextMenu.css

DES file: DES\Appearance\Interactive Pages\Menu.css

Since the PDP format is so different, it is better just to start with the new file.

Apply the desired font, background color and borders to DESMenu.

Several ContextMenu properties that had colors are replaced by style sheets. This chart will help you transfer the color from the property to the style sheet class defined in Menu.css

PDP Property name	DES Style Sheet Class name
xSeparatorColor	DESMenuSeparator
All xHint colors	DESMenuHint
xSelectedForeColor and xSelectedBackColor	DESMenuMouseOver

* - PDP v2 users use these files. PDP v1.x users have everything in the file StyleSheet1.css.

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 () .

Professional Validation And More 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	HelpButtonImageUrl	 HelpCmd.gif, HelpCmdPressed.gif Add HelpCmdMouseOver.gif
xPopupCalendar.xToggleImageUrl xPopupMonthYearPicker. xToggleImageUrl	PopupCalendar.ToggleImageUrl PopupMonthYearPicker.ToggleImageUrl	 Calendar.jpg, CalendarPressed.jpg Add CalendarMouseOver.jpg
xPopupTimePicker.ToggleImageUrl	PopupTimePicker.ToggleImageUrl	 Clock.gif, ClockPressed.gif Add ClockMouseOver.gif
xPrevCommandImageUrl	PeterBlum.DES.Globals.Page. SpinnerManager.PrevCommandImageUrl	 DnArrow1.gif, DnArrow1Pressed.gif Add DnArrow1MouseOver.gif
xNextCommandImageUrl	PeterBlum.DES.Globals.Page. SpinnerManager.NextCommandImageUrl	 UpArrow1.gif, UpArrow1Pressed.gif Add UpArrow1MouseOver.gif

Calendar and MultiSelectionCalendar		
PDP Property Name	DES Property Name	Defaults
xHelpButtonImageUrl	HelpButtonImageUrl	 HelpCmd.gif, HelpCmdPressed.gif. Add HelpCmdMouseOver.gif
xCloseButtonImageUrl	CloseButtonImageUrl	<input checked="" type="checkbox"/> CloseCmd.gif, CloseCmdPressed.gif Add CloseCmdMouseOver.gif
xPrevMonthButtonImageUrl	PrevMonthButtonImageUrl	 LeftCmd_3DBlue.gif, LeftCmd_3DBluePressed.gif Add LeftCmd_3DBlueMouseOver.gif There are several versions of this image, all starting with LeftCmd_
xNextMonthButtonImageUrl	NextMonthButtonImageUrl	 RightCmd_3DBlue.gif, RightCmd_3DBluePressed.gif Add RightCmd_3DBlueMouseOver.gif There are several versions of this image, all starting with RightCmd_
xJumpBackButtonImageUrl	JumpBackButtonImageUrl	 LeftCmd2_3DBlue.gif, LeftCmd2_3DBluePressed.gif Add LeftCmd2_3DBlueMouseOver.gif There are several versions of this image, all starting with LeftCmd2_
xJumpForwardButtonImageUrl	JumpForwardButtonImageUrl	 RightCmd2_3DBlue.gif, RightCmd2_3DBluePressed.gif Add RightCmd2_3DBlueMouseOver.gif There are several versions of this image, all starting with RightCmd2_
xTodayButtonImageUrl	TodayButtonImageUrl	No defaults
xSpecialButtonImageUrl	SpecialButtonImageUrl	No defaults
xClearButtonImageUrl	ClearButtonImageUrl	No defaults
xShowSelectionButtonImageUrl	ShowSelectionButtonImageUrl	No defaults
xApplyButtonImageUrl	ApplyButtonImageUrl	No defaults
xExpandedDateFormatter. xCloseButtonImageUrl	ExpandedDateFormatter. CloseButtonImageUrl	<input checked="" type="checkbox"/> CloseCmd.gif, CloseCmdPressed.gif Add CloseCmdMouseOver.gif
xSelectWeeksImageUrl	SelectWeeksImageUrl	 ToggleWeekSolidGray.gif, ToggleWeekSolidGray Pressed.gif Add ToggleWeekSolidGrayMouseOver.gif There are several versions of this image, all starting with ToggleWeek
xSelectMonthImageUrl	SelectMonthImageUrl	 ToggleMonthSolidGray.gif, ToggleMonthSolidGray Pressed.gif Add ToggleMonthSolidGrayMouseOver.gif There are several versions of this image, all starting with ToggleMonth
xSelectDOWImageUrl	SelectDOWImageUrl	 ToggleDOWSolidGray.gif, ToggleDOWSolidGray Pressed.gif Add ToggleDOWSolidGrayMouseOver.gif 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 Add CloseCmdMouseOver.gif
MonthYearPicker	xPrevYearsButtonImageUrl	PrevYearsButtonImageUrl	 LeftCmd_3DBlue.gif, LeftCmd_3DBluePressed.gif Add LeftCmd_3DBlueMouseOver.gif There are several versions of this image, all starting with LeftCmd_
MonthYearPicker	xNextYearsButtonImageUrl	NextYearsButtonImageUrl	 RightCmd_3DBlue.gif, RightCmd_3DBluePressed.gif Add RightCmd_3DBlueMouseOver.gif There are several versions of this image, all starting with RightCmd_
TimePicker	xCloseButtonImageUrl	CloseButtonImageUrl	 CloseCmd.gif, CloseCmdPressed.gif Add CloseCmdMouseOver.gif
PopupTimePicker	xToggleImageUrl	ToggleImageUrl	 Clock.gif, ClockPressed.gif Add ClockMouseOver.gif
QuickDateMenu	xToggleImageUrl	ToggleImageUrl	 Range.gif, RangePressed.gif 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 Declarative Syntax). 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 Declarative Syntax. 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 Declarative Syntax.

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 Declarative Syntax, 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 Declarative Syntax. 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 – Migrating from Peter’s Date Package

1. Why are there two `<% =PeterBlum.DES.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.

2. 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**.

Well over 500 property names were converted, yet about less than 30 were renamed dramatically.

Here is a list of a most classes, properties and methods that changed more dramatically. *Most properties still exist and can be used, although they are not shown in the Properties Editor or Intellisense.*

Old	New
CS_Calendar (class)	Calendar
CS_Menu (class)	ContextMenu
xDate	DateValue
xTimeSpan	TimeValue
xTimeSpanNullable	TimeNullable
xTotalSeconds	TimeAsSeconds
xTimeDouble	TimeAsHours
xlsValidDateB and xlsValidTimeB	IsValid
xShowPopupB	ShowPopupCalendar or ShowPopupMonthYearPicker
xAutoToolTipB	AutoHint (feature is expanded to support hints)
xToolTipInStatusB	HintInStatusBar
xShowCommandButtons	ShowSpinners
RegisterForAJAX()	InAJAXUpdate (method converted to a property)
RegisteredForAJAX	PreLoadedForAJAX
OnDateChanged (event)	SelectedDateChanged
OnMonthViewChanged (event)	MonthViewChanged
xControlToUpdateID	ShowSelectedDateControlID
xControlToUpdateDateFormat	ShowSelectedDateControlDateFormat
xFooterButtonInset	CommandButtonInset (the “Footer” is new region of the calendar. The buttons are in the “commands area”)
xFooterRowHeight	CommandButtonRowHeight
xFooterVerticalAlign	CommandButtonVerticalAlign

THIS TABLE IS CONTINUED ON THE NEXT PAGE

Old	New
TimePickerTimeSpan (class)	TimePickerTimeValue
xTimeSpanCssClass	TimeValueCssClass
xSelectedTimeSpanCssClass	SelectedTimeValueCssClass
Mode (enumerated type on Menus)	AnnualDateMode

3. 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_”.

4. 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 **[VAM product 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). Provide tech support with details of any script error, including the actual PDP-compatible script code and the error message.

Converting from ASP.NET 1.x to ASP.NET 2.0

Use this section when you are converting your web site from ASP.NET 1.x to ASP.NET 2.0 or higher. It will replace the ASP.NET 1.x version of Peter's Data Entry Suite ("DES") with the ASP.NET 2.0 version, which has an extended feature set.

Note: ASP.NET 2.0 users are not required to use the ASP.NET 2.0 version of DES. The ASP.NET 1.x version is compatible. However, it lacks features built around ASP.NET 2.0. If your web application is complete and you plan no further development, consider leaving your site using the ASP.NET 1.x version of DES.

1. Convert your site fully to ASP.NET 2 while keeping the existing DES in place. Confirm that it works.
2. If you are using resource files (resx) to supply strings to DES's String Lookup System, Visual Studio handles them differently. Your site conversion should relocate your resx files into the **App_GlobalResources** folder. You have to reconfigure the String Lookup System to locate the resources here.
 - a. Open the **Global Settings Editor** with the **custom.DES.config** file in **[webapplicationroot]\DES**.
 - b. Locate the String Lookup System topic.
 - c. Set the value of the ResAssemblyName property to "app_GlobalResources".
 - d. Save the file.
3. Remove the ASP.NET Design Mode Extender ("ADME"). It is no longer used.

Note: It is harmless to keep it. If you have a mixture of ASP.NET 1.x and 2.0 web applications. Keep it.

- a. Exit from Visual Studio, if open.
 - b. Use the **Add/Remove Programs** control panel to remove ASP.NET Design Mode Extender.
 - c. Since ADME is installed on each development computer, repeat the process on other development servers.
 - d. If you have setup a Deployment project or batch files that copy your web application to production, do not have it deploy **PeterBlum.ADME.dll**. *Note: If the file is moved to production, it will be ignored by DES and is therefore harmless.*
4. Remove the existing DES assemblies from the **\bin** folder. When using Visual Studio, always do this by removing the project references to the assembly. All assemblies start with "PeterBlum.DES".
5. Follow the instructions of "First Time Installation". The **Web Application Updater** will ask which assemblies to use. Be sure to select ASP.NET 2.0
6. If you use **Peter's Input Security** to send emails of errors that were logged, follow these steps:
 - a. Open the **Global.asax** file and comment out the line that looks like this in the `SetupInputSecurity()` method:


```
System.Web.Mail.SmtpMail.SmtpServer = "mail.mydomain.com"
```
 - b. ASP.NET 2.0 uses a new class for email, `System.Net.Mail.SmtpClient`. The `System.Net.Mail.SmtpClient` class is configured either in **machine.config** or **web.config** using this basic syntax:

```
<configuration>
  <system.net>
    <mailSettings>
      <smtp>
        <network host="email server" />
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

Set the host attribute to the name of the server, such as "mail.myserver.com".

There are other attributes available to the `<network>` tag, such as username, password, and port. Please refer to the .net documentation if you need to use any of them.

Note: PeterBlum.com cannot provide technical support on getting ASP.NET to communicate with the STMP Server. If you have problems, try to use the System.Net.Mail.SmtpClient class yourself and debug your email configuration. At its most basic, you create a web form and add this code into Page_Load():

[C#]

```
MailMessage vMailMessage = new MailMessage(From, To, Subject, Body);
SmtpClient vSmtpClient = new SmtpClient();
vSmtpClient.UseDefaultCredentials = true;
vSmtpClient.Send(vMailMessage);
```

[VB]

```
Dim vMailMessage As MailMessage = New MailMessage(From, To, Subject, Body)
Dim vSmtpClient As SmtpClient = New SmtpClient()
vSmtpClient.UseDefaultCredentials = True
vSmtpClient.Send(vMailMessage)
```

7. Recompile your web application and test it. If using Visual Studio, use Rebuild Solution.

PLANNING AHEAD: When your page uses AJAX to update its controls, you **must** follow the steps described in each User's 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.*

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. Run it and select **Convert Native Controls to their DES equivalents**. You can convert a page, group of pages, or the entire web application.

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.

Steps for Converting

1. Exit from Visual Studio and Visual Web Developer, if opened.
2. Run the program **Web Application Updater.exe**, found in the **[ProductFolder]** or in the **Start** menu.
3. Click **Next**.
4. Select the third radio button, **Convert Native Controls to their DES equivalents**.
5. Choose which types of controls to convert with the **Type of Control** drop down list.
Recommendation: Always convert buttons and validators on any page that you will be introducing DES's validators.
6. 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.

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 Comments
✓ PeterBlum.DES.dll	[webapplicationroot]\bin or Global Assembly Cache
✓ PeterBlum.DES.NativeValidators.dll	[webapplicationroot]\bin or Global Assembly Cache Optional. If you are using DES textboxes with the native ASP.NET validation system.
✓ PeterBlum.DES.NativeToDES.dll	[webapplicationroot]\bin or Global Assembly Cache Optional. If you converted from Professional Validation And More or Peter’s Date Package .
✓ PeterBlum.DES.PDPToDES.dll	[webapplicationroot]\bin or Global Assembly Cache Optional. If you are using Peter’s Date Package with DES.
✓ PeterBlum.ADME.dll	[webapplicationroot]\bin or Global Assembly Cache ASP.NET 1.x Users ONLY
✓ DES folder	Root directory of your web application 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. You may have reconfigured DES to keep this or its child folders in different locations. The <appSettings> section of web.config 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. If you have a Web Server license, the software will now switch from using the DES Development.lic file to the DES SingleServer###.lic 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 Global.aspx . See “Installing Licenses”. If there are licensing errors, please see “Troubleshooting Licenses”. <i>Note: Visual Studio 2005/8 users who intend to use the Publish Web Site command should see this troubleshooting topic first.</i>

THIS TABLE IS CONTINUED ON THE NEXT PAGE

Filename	Location	Comments
✓ Global.asax	[webapplicationroot]	This file contains the Application_Start() method that you modified for Web Server and Site Licenses. Application_Start() does not run unless the Global.asax file is present.
✓ web.config	[webapplicationroot]	The web.config file contains several DES-specific entries, including <appSettings> entries, <assemblyBinding> tags, and <page> tags. Any key for DES declared in the <appSettings> section should be reviewed. If file paths are declared in the DES_ConfigFilePath or DES_LicenseFilePath keys, they may no longer match the file structure of the destination server.
✓ Third party assemblies that support DES	[webapplicationroot]\bin	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.

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. **Visual Studio 2002/3 Users ONLY** Install ASP.NET Design Mode Extender. See “Installing ASP.NET Design Mode Extender”.
2. Add controls to the Visual Studio toolbox. See “Adding To The Visual Studio/Visual Web Developer Toolbox”.
3. If this computer has its own web servers for development, add DES to each web application. See “First Time Installation”.

Installing ASP.NET Design Mode Extender

Visual Studio 2002/3 Users ONLY Visual Studio 2005, 2008, 2010, and Visual Web Developer users do not use ASP.NET Design Mode Extender.

ASP.NET Design Mode Extender (“ADME”) provides **Peter’s Data Entry Suite** (“DES”) with access to its XML configuration files that are used in design mode and at runtime. Therefore, it must be installed onto any computer that will access DES’s own assemblies. This includes both development computers and web servers.

Note: Why must ADME be installed on so many computers? ADME extends your design mode of Visual Studio 2002 and 2003. So it must be installed onto your development computers. ADME provides a runtime library used by Professional Validation And More’s assemblies. So it must be installed onto the web server where PeterBlum.VAM.dll is installed.

Note: Other products may install ADME onto your computers. If you already have ADME installed, you do not need to install it again.

1. The **Web Application Updater** automatically installs ADME onto the local computer if it’s needed. You can stop if you have no other computers to update.
2. Go to a development computer.
3. Exit from Visual Studio if open. *ADME adds tools to Visual Studio.*
4. Install ADME onto the first development computer.
5. Run **Use on Visual Studio 2002 and 2003 computers.msi** within this path:
[ProductFolder]\Development Computer Setup
6. Follow the prompts.
7. The first time you go to a Web Application project in Visual Studio, ADME will prompt you to register that web application. This configures ADME. If you later need to adjust the configuration, choose the **Tools; ASP.NET Design Mode Extender** menu command in Visual Studio.
8. Go to another development computer and repeat from step 3.

FAQs – Installing ADME

1. What was changed when ADME was installed?
 - **[ProductFolder]\ADME**. It includes the **PeterBlum.ADME.dll** assembly, the **ADME Settings Editor**, and files used by Visual Studio in design mode.
 - Visual Studio includes new menu and toolbar commands. Both run the **ADME Settings Editor** application. In addition, as you switch between web application projects, the **Web Application Monitor Add-In** will prompt you to setup any web applications that you have yet to register with ADME.

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 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=4.0.7.5000, 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. 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 - First Time Installation](#)
- ◆ [FAQs – Adding to the Toolbox](#)
- ◆ [Troubleshooting Licenses](#)
- ◆ [FAQs – Upgrading from Professional Validation And More](#)
- ◆ [FAQs – Migrating from Peter's Date Package](#)
- ◆ [Confirming a Deployment](#)
- ◆ [FAQs – Installing ADME](#)
- ◆ [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 **Update a Web Application (service release)**. See “Installing a Service Release”.
- 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 2005/8's Publish Web Site command does not copy the license files

Visual Studio 2005/8 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 2005/8 Properties Pages lists duplicate DES assemblies

When you view the References section of the Visual Studio 2005/8 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=4.0.#.5000, Culture=neutral,
    PublicKeyToken=cb5182303c90db58" />
    <add assembly="PeterBlum.DES.NativeToDES, Version=4.0.#.5000,
    Culture=neutral, PublicKeyToken=cb5182303c90db58" />
    <add assembly="PeterBlum.DES.NativeValidators, Version=4.#.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.Page 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 Message Board

Use the message board at <http://groups.yahoo.com/groups/peterblum> to discuss issues and ideas with other users.

Getting Product Updates

As minor versions are released (4.0.1 to 4.0.2 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 (v4.0 to v4.1), 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. 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:

- Books
- www.asp.net forums and tutorials
- Microsoft's usenet newsgroups such as [microsoft.public.dotnet.framework.aspnet](http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&group=microsoft.public.dotnet). See <http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&group=microsoft.public.dotnet>
- 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!
- <http://aspnet.4guysfromrolla.com/>, <http://www.dotnetjunkies.com>, <http://www.aspalliance.com/>

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