

PETER'S DATA ENTRY SUITE v5 GETTING STARTED



Hello, I am Peter Blum. Thank you for using my code!

Peter's Data Entry Suite's roots started in 2002 when ASP.NET was new. At the time, it was "Peter's Date Package v1" with a DateTextBox, Calendar, and a number of supporting validators. These days, Peter's Data Entry Suite ("DES") is loaded! It has over 100 controls, several "frameworks", and a multitude of options to allow it to address the needs of all kinds of users.

I strongly encourage you to review this document. It will prepare you for what's ahead.



Click on any of these topics to jump to them:

- ◆ [What does Peter's Data Entry Suite do?](#)
- ◆ [I just want to use a few controls of your huge suite. Will DES work for me?](#)
- ◆ [How can I quickly learn these tools?](#)
- ◆ [There are thousands of pages of documentation. How can I find what I'm looking for?](#)
- ◆ [What do I need to know to use DES?](#)
- ◆ [What else do you want me to know when using DES?](#)
- ◆ [What tools are provided to simplify setting these controls up?](#)
- ◆ [Are there any great controls or tools that users often overlook?](#)
- ◆ [When should I use Peter's Business Logic Driven UI?](#)
- ◆ [Where can I get additional assistance?](#)

What does Peter's Data Entry Suite do?

Peter's Data Entry Suite ("DES") is a suite of ASP.NET controls designed around the concepts of data entry: validation, entry fields, and interactive behaviors in response to a user's action. The suite includes several modules that can be purchased stand-alone. This User's Guide covers the data entry controls.

Peter's Data Entry Suite was designed to improve upon the form validation concept build into the .Net framework. It completely replaces Microsoft's original Validator controls, as they imposed serious limitations on how a Validator can look and act. As part of the work, extensive client-side JavaScript code was written. This JavaScript code lends itself well to other client-side tasks, such as formatting the entry of a CurrencyTextBox and setting focus to a field. Since validation is always part of data entry, **Peter's Data Entry Suite** addresses numerous other requirements of a good user interface for data entry.

Here are the major aspects of **Peter's Data Entry Suite**:

- **Peter's Business Logic Driven UI** ("BLD") – BLD provides a major shift in the way you write applications, using techniques programmers have long favored: separation of concerns. Instead of creating all of your business logic inside the web forms, you define classes which manage the business logic. The classes are completely user interface independent. Web forms use a suite of smart controls like BLDFormView, BLDListView, and BLDDataField, which follow the rules in the business logic classes.

BLD uses the rest of the Peter's Data Entry Suite to implement some very smart interfaces. If you choose not to use BLD, you will still use the tools described throughout the remainder of this section.

BLD requires a minimum of ASP.NET 3.5 or higher.

See the **BLD User's Guide** and <http://learningbld.peterblum.com/>.

- **Validation** – The same idea as the concept introduced by Microsoft, with Validator controls to detect and report errors on the page. DES provides extensive enhancements over Microsoft's controls that allow your sites to have a more professional appearance with Validators and make it much easier to evaluate the data in your web form. Its rich feature set lets you set a few properties instead of writing custom code and hacks to work around the limitations of Microsoft's Validators. It includes 28 Validators and several other controls including:
 - The ValidationSummary control displays a consolidated list of all errors reported as the page is submitted.
 - The RequiredFieldMarker and RequiredFieldDescription controls standardize the user interface for indicating a field is required.
 - The CombinedErrorMessages control merges the error messages from several Validators to save screen real estate.
 - The LocalizableLabel control is an enhanced Label that supports localization. Localization is an important aspect to DES. Labels are optionally shown in the error messages.
 - A variety of button controls and ways to submit the page so that validation is run automatically.

See the **Validation User's Guide** and <http://learningdes.peterblum.com/validation/menu.aspx>.

- **TextBoxes** – DES provides a number of controls as replacements to Microsoft's TextBox control.
 - Its own enhanced TextBox control, the basis for all other textboxes, introduces numerous common client-side tricks for TextBoxes.
 - The IntegerTextBox, DecimalTextBox, CurrencyTextBox, and PercentTextBox handle data entry of these numeric formats.
 - The FilteredTextBox limits entry to the character set of your choice.
 - The MultiSegmentDataEntry control combines multiple textboxes and dropdownlists together to handle a single field that has a distinctive pattern, such as phone numbers and credit card numbers. It is a great substitute for a masked textbox with a more powerful user interface.

See the **TextBoxes User's Guide** and <http://learningdes.peterblum.com/textboxes/menu.aspx>.

- **Date And Time** – Controls for entry of date and time information.
 - DateTextBox – Date entry with a popup calendar
 - AnniversaryTextBox – Date entry without the year.
 - MonthYearTextBox – Month and year entry
 - TimeOfDayTextBox – Time of day entry
 - DurationTextBox – Time duration entry
 - Calendar – A very powerful replacement to the native ASP.NET calendar
 - MonthYearPicker – Month and year entry using the mouse
 - TimePicker – Time entry using the mouse
 - Popup versions of Calendar, MonthYearPicker, and TimePicker
 - Validators for these controls. There are versions for the DES Validation Framework and Native ASP.NET Validation Framework if you choose not to use the DES Validation Framework.

See the **Date and Time User's Guide** and <http://learningdes.peterblum.com/DateAndTime/Menu.aspx>.

- **Interactive Pages** – There are numerous ways to make your web forms more interactive and user friendly through JavaScript. These techniques will make your web forms feel more like a window in a Windows application.
 - The FieldStateController monitors clicks and changes on a field and modifies other controls. It can modify almost any attribute of a field: visibility, enabled, style sheet class, value and more. For example, use it when you want a textbox to be disabled until the user marks a checkbox. There are four versions of the FieldStateController: FieldStateController, MultiFieldStateController, FSCOnCommand, and MultiFSCOnCommand.
 - The CalculationController lets you describe a calculation that uses the textboxes on the page. It can display the result of the calculation in a label or another textbox. Validators can validate the result of the calculation. For example, a RangeValidator can make sure the total of 3 textboxes is within 0 to 100.
 - Add a ContextMenu to your web forms, supplying javascript commands to regions and controls within the page. The DropDownMenu is a variation of the ContextMenu with a button that toggles open the ContextMenu.
 - The TextCounter monitors the number of keystrokes in a textbox, displaying the count and warning the user as they near and reach the limits.
 - The Interactive Hints system shows a hint as the user moves into a field. The hint can appear in a popup (floating element) or on the page in a Label or Panel. It can replace standard ToolTips and can appear in the browser's status bar too.
 - The ChangeMonitor watches for edits in the page. Until a change is made, selected buttons are disabled. After, they are enabled.
 - Enhanced Buttons with several javascript tricks including showing a confirmation message, disabling until data is changed (see the ChangeMonitor), and disabling on submit.
 - Direct the Enter key to click a specific button. Enhances DES's TextBoxes with the **EnterSubmitsControlID** property.

See the **Interactive Pages User's Guide** and <http://learningdes.peterblum.com/interactivepages/menu.aspx>.

- **Input Security** – Hackers attack your web site through its inputs – data entry controls, query strings, hidden fields, and cookies – to access your database (called SQL Injection) and modify your pages with scripts (called Script Injection or Cross-site scripting). These attacks can be very damaging to a business, destroying data, exposing private customer information, or exposing customers to content that you would never want on your site. Every public web site should be designed with a defense system. With Peter's Input Security, you have that defense system.

Validators play an important role in blocking these attacks. However, they have their limitations. Peter's Input Security introduces specialized Validators to detect and block attacks. It also provides a "best practices" framework for protecting your site against attacks.

See the **Input Security User's Guide**.

- **General Features** – The features throughout this product are supported by these controls and tools:
 - PageManager control – Each page has numerous settings. Use this control to apply those settings in without programming.
 - Global Settings Editor – A stand-alone Windows application to customize numerous global settings used by DES.
 - NativeControlExtender control – Extends various native controls to support DES Validation, Interactive Hints, the ChangeMonitor, and more.
 - String Lookup System – A mechanism to set most string-type properties from data stored in resources or a database.
 - LocalizableLabel control – The Label control enhanced to support the String Lookup System.

See the **General Features Guide** and <http://learningdes.peterblum.com/General%20Features/Menu.aspx>.

Throughout this product, one of the most important design concepts is to allow expansion through the object-oriented concepts of subclassing and delegation. This allows you to build your web site the way you want it. The controls are built upon several discrete object classes. Even the client-side JavaScript is designed for expansion. See the [Developer's Kit](#) to learn how to program with **Peter's Data Entry Suite**.

I just want to use a few controls of your huge suite. Will DES work for me?

Yes. You can purchase licenses to modules containing the desired controls instead of licensing the entire suite. (Exception: Peter's Business Driven Logic UI requires a license for the Suite.)

Many controls work stand-alone. For controls like the `DateTextBox` and `IntegerTextBox`, just add them. I encourage you to learn about their properties to get and set their value, such as `DateValue`, `DateBindable`, and `DateNullable` on `DateTextBox`. Check out the "Using" section in the documentation for each control for the topic "Getting and Setting the value".

When it comes to validation and input security (against hackers), you are no longer dealing with single controls. Instead you are working with "frameworks". The web controls are simply a user interface to those frameworks. As you learn about validation, you realize that it is built into buttons and the Page object (`Page.IsValid`, for example). Some controls need others, like the `ValidationSummary` only appears if there are validators to report errors on.

If you want to use any of the controls in the **Peter's Professional Validation** module, you will be using the *DES Validation Framework*. When you already have a page with validators setup, you will need to switch them to the DES Validation Framework. Fortunately it's *easy*! Run the **Web Application Updater** program found in the DES Product Folder and use either of the first two options. It lets you pick a single file, a list of files or the entire application.

I strongly recommend reading the topic "Adding Validation to a WebForm" in the **Validation User's Guide**. (There is a link called "Adding" at the top of the first page in that document.)

If you want to use any of the controls in the **Peter's Input Security** module, there is much more to setup. After all, the goal is to build a solid defense against crafty hackers who enjoy looking for holes in your defenses. It may help to learn about how hackers attack a site so that you can build in these defenses as you build each page. The **Input Security User's Guide** has a detailed overview in the section called "Your Opponents: The hackers of the world".

How can I quickly learn these tools?

The primary documentation is in these PDF documents. Each control includes a topic called “Adding the [controlname]” which gives you step-by-step instructions, probably best designed for the noob.

As you want to learn what each control can do, check out the topic “Using the [controlname]”. You will learn about how to get and set values, establish formatting, add optional features, and so much more.

All of the properties are also described in these documents. However, if you use Visual Studio design mode, the Smart Tags and Properties Editor will also help you find what you need. See “[What tools are provided to simplify setting these controls up?](#)”.

I’m sure that you like online demos and walkthroughs. DES includes those resources too.

Make an appointment Walkthrough – <http://learningdes.peterblum.com/Walkthrough/MakeAnAppointment.aspx>. This is a great way to explore nearly all controls and the associated concepts. If you are learning the entire suite (excluding BLD), this is where to start.

DES Samples – <http://learningdes.peterblum.com>. This explores most of the controls, showing key properties, interactive demos and source code.

BLD Guided Tour – <http://learningbld.peterblum.com/Guided%20Tour.aspx>. If you are working with BLD, this is a guided tour that shows how to build an application.

Make an appointment Walkthrough for BLD – <http://learningbld.peterblum.com/Walkthrough/MakeAnAppointment.aspx>. This is a great way to explore nearly all controls and the associated concepts. It uses the same Make an Appointment form as in the earlier but uses BLD to construct it.

There are thousands of pages of documentation. How can I find what I'm looking for?

I love to write documentation. I am a verbose writer. (Sorry about that!) I prefer that my documentation actually answers questions instead of creating new questions. That means you have more to scan, but also more chance of learning something of value.

These PDF documents each start with a table of key features that are hyperlinked to their topics. For example:

Click on any of these topics to jump to them:		
◆ Enhanced TextBox Control	Using	Adding Properties
◆ IntegerTextBox Control	Using	Adding Properties
◆ DecimalTextBox Control	Using	Adding Properties
◆ CurrencyTextBox Control	Using	Adding Properties
◆ PercentTextBox Control	Using	Adding Properties
◆ FilteredTextBox Control	Using	Adding Properties
◆ MultiSegmentDataEntry Control	Using	Adding Properties
● Segments of the MultiSegmentDataEntry Control	Using	Adding Properties
◆ Page Level Properties (The PeterBlum.DES.Globals.Page object)		
◆ Validation with the Native Validation Framework		
◆ JavaScript Support Functions		
◆ Troubleshooting		
◆ Table Of Contents		

In fact, the documentation is heavily hyperlinked. Adobe Reader makes moving back to where you jumped easy through these toolbar commands:



Note: If your Adobe Reader lacks these buttons, they can be added to the toolbar. Right click on the toolbar and select More Tools. Locate and mark "Previous View" and "Next View" under the Page Navigation Toolbar node.

If these commands are missing, go to the **View; Toolbars** menu and select **Page Navigation**.

Also take advantage of the **Find** command (type Ctrl+F).

What do I need to know to use DES?

AJAX

When your page uses AJAX, DES requires additional setup to let its controls know to generate scripts and style sheets differently. Instructions are in the “Using these controls with AJAX” section of the **General Features Guide**. *Failure to follow these directions can result in incorrect behavior and JavaScript errors.*

In most cases, you just need to set the **AJAXFramework** property on the PageManager control.

```
<des:PageManager runat="server" AJAXFramework="MicrosoftAJAX" />
```

Yet you should review the steps offered to handle the other cases. Users often overlook step 5, used when controls are added for the first time by the callback process.

```
<des:PageManager runat="server" AJAXFramework="MicrosoftAJAX"
  PreLoadForAJAX-Validators="True" PreLoadForAJAX-TextBoxes="True" />
```

Always protect against bad data

Being an expert in validation and input security, take my advice: *illegal or unwanted data will get passed to your application.* You must defend against it, especially if it's from hackers. Yet respectable can send illegal data to the server if their browsers don't work with the JavaScript based features of these controls, or have turned off JavaScript.

When you add a textbox, assume that none of its JavaScript features are working, and add validator controls that will run on the server side to catch the bad data. Each TextBox control has a topic called “Data Entry Validation” within its “Using” section in the various User's Guides. It explains how to validate both in the DES and native ASP.NET Validation Frameworks.

If you are concerned about illegal data from hackers, they will attack through cookies, querystring parameters, hidden fields and other types of controls. You should use the techniques described in the **Input Security User's Guide**.

Make sure server side validation is setup

After adding validators and testing them, users move on time-and-time again without making sure that server side validation is doing its job. Yet it is so easy to setup! In a button's Click event handler method, test **PeterBlum.DES.Globals.WebFormDirector.IsValid** is true before using the data from the page.

[C#]

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (PeterBlum.DES.Globals.WebFormDirector.IsValid)
    {
        // code that saves or uses the data of the web form
    }
}
```

[VB]

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles Button1.Click

    If PeterBlum.DES.Globals.WebFormDirector.IsValid Then
        ' code that saves or uses the data of the web form
    End If
End Sub
```

In any other event handler where data is saved, first call

PeterBlum.DES.Globals.WebFormDirector.Validate("validation group name") then test

PeterBlum.DES.Globals.WebFormDirector.IsValid is true.

What else do you want me to know when using DES?

Style sheets

DES uses style sheets to establish the appearance of its output. There are numerous style sheet files, based on the type of control. This way it only loads the style sheets needed for the controls added to the page.

- Style sheets are found in the **[web application]\DES\Appearance\control category** folders.
- They are heavily commented. The User's Guides also describe them associated with the properties that are assigned to them.
- You can create alternative style sheet classes and assign their names to properties on the controls.
- DES automatically adds the style sheets to your page. You never need to set up <link> tags. When it does this, it strips out the comments and whitespace to deliver a very compact file to your browser. It also uses GZIP or DEFLATE compression in the transmission of the file.

Learn more with the "Using Style Sheets" section of the **General Features Guide**.

Localization and String Lookup

Dates, times, numbers, and currencies are culture sensitive. DES respects the rules of the [System.Globalization.CultureInfo](#) object which is setup on each ASP.NET page. If you don't take any other action, DES will give you formatting to the server you are running on. One common issue is that when a user deploys to another server, its default localization is different. Make sure your site or pages are using the culture you want, not the one on your development server.

See the "Establishing Localization for the Web Form" section in the **General Features Guide**.

Strings are also culture sensitive as sites may need to be multilingual. ASP.NET has a technology to support this [through resources](#). DES goes one step further with its **String Lookup System**. It is more flexible and transparent, allowing the use of resources, database and other storage mechanisms.

See the "String Lookup System" section of the **General Features Guide**.

Troubleshooting

I put much of my tech support solutions into my documentation. If you don't find the answer directly in the docs for a control, please jump to the end of the User's Guides. They all have a "Troubleshooting" section.

The two that are most extensive are in the **General Features Guide** and **Installation Guide**.

The one topic I direct users to the most is called "Handling JavaScript Errors" in the **General Features Guide**. It even has a hyperlink from the first page of the document, near the bottom. If you ever get a JavaScript error, please start there.

What tools are provided to simplify setting these controls up?

Visual Studio and Visual Web Developer provide a terrific design mode system, and I have included many extensions to it.

Smart Tags

I really love the Smart Tag feature of design mode. If you don't know that it is, point to a control and look for this image: . Click on it to bring up a worksheet with the ability to edit properties and invoke commands.

If you have used a Smart Tag elsewhere, you may find I do things a little differently. Microsoft recommends Smart Tags for a short list of commands and one-or-two property settings. I think it's an awesome way to prepare a control when you first add it. So my Smart Tags feature some the best properties for getting a control setup.

Enhancements to the Properties Editors

The Properties Editor view has new editors on many of the properties in DES controls. The documentation shown on each property is probably as comprehensive and helpful as you've seen anywhere.

One challenge you have is that my controls have many properties. Finding your way through them may be a challenge. That's why I recommend the "Expanded Properties Editor". This command brings up a dialog box with the typical PropertiesGrid, plus a special button called Best Order (). When clicked, properties are ordered by how I think they should be grouped, not alphabetically. The most important are nearer to the top of categories. In addition, you can hide lesser used properties so you can make your way to the most important properties.

The Expanded Properties Editor command can be found on the Smart Tag, the control's context menu, and at the bottom of the main Properties Editor view.

For more, see the "Expanded Properties Editor" section in the **General Features Guide**.

Intellisense

If you prefer to type instead of drag and drop, Intellisense will deliver documentation on properties.

Global Settings Editor

A well designed site uses a consistent user interface. DES lets you define numerous site-level rules with the **Global Settings Editor**, a program in the **DES product folder**. Each web application has a file called **custom.des.config** located in the **[web application]DES** folder where your site level rules are stored.

For more, see the "Global Settings Editor and custom.des.config file" section in the **General Features Guide**.

Web Application Updater

When adding a third party assembly to your app, "getting started" really means adding the product files and modifying any existing files as needed. This work is handled by the **Web Application Updater** program in the **DES product folder**. It has a wizard-driven interface, backs up files, and creates a detailed log of what actions it took.

See the **Installation Guide** for details.

Are there any great controls or tools that users often overlook?

At least once a month, a user emails me saying how they just discovered the FieldStateController and how great it is. This control provides an interactive client-side interface by monitoring user actions on fields and changing the attributes and styles of other controls on the page. For example, when the user clicks a checkbox, show a previously hidden <div>. The FieldStateControllers often eliminate the effort to build the browser-independent JavaScript required to monitor events, look at field data, and change attributes and styles.

For more, see the **Interactive Pages User's Guide** or <http://learningdes.peterblum.com/interactivepages/menu.aspx>.

While you are there, read the summaries of the other controls and tools of the **Peter's Interactive Pages** module. Programmers are so busy adding textboxes, validators and buttons to their pages, that they don't realize how many great things exist if they only opened the **Interactive Page's User's Guide**. For example, the ChangeMonitor changes buttons from disabled to enabled as edits occur. Interactive Hints show help balloons on your controls. And the CalculationController performs math and displays results.

When should I use Peter's Business Logic Driven UI?

Many programmers prefer to separate their business logic code from the user interface. They write "Entity" and "Data Access Object" classes describing the business logic. That code has no user interface, allowing it to be tested easily and used in a variety of applications.

This "Separation of Concerns" is achieved by using Peter's Business Driven UI ("BLD"). Your user interface fully respects the business logic code, adding the right validators and data entry controls. BLD contains 30 controls to build a user interface, plus classes to help you build your business logic.

Explore more here: <http://www.peterblum.com/bld/home.aspx>.

Where can I get additional assistance?

Please visit <http://www.peterblum.com/des/support.aspx> for all of your options.

PeterBlum.com is my full time job (since 2002). Usually you won't have to wait a day just to hear from me.