



Form Creation Walkthrough For HTML v3.1.9079

Last Updated 06/08/09 Brandon J.

Prerequisites

- a. You will need to install the **Universal Document Driver** from <http://www.print-driver.com/>.
- b. You will need a form of html editor, whether it maybe Visual Studios, Dreamweaver or whatever it is, just as long as you are able to create html documents.
- c. You will need to acquire a decent image resizing utility. I suggest using the Paint.NET application. It's located under the tools directory.

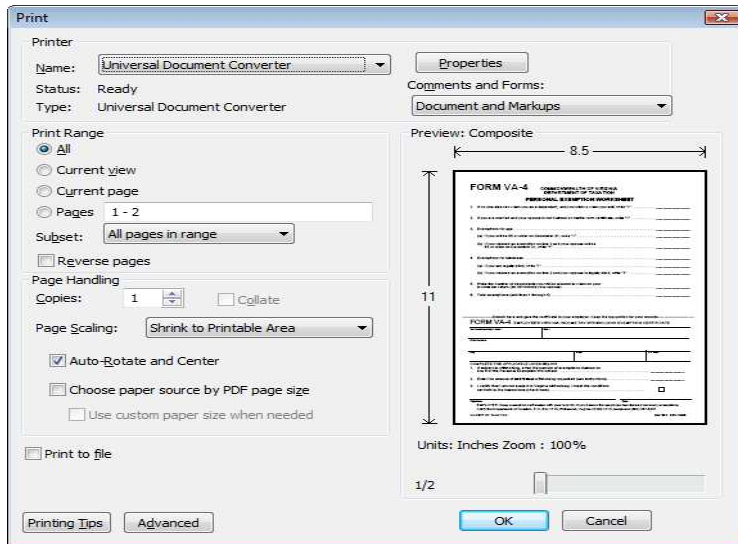
Getting Started

- 1) Open your html editor and start a new project. Name it something logical to the type of project you are working on, so that other people down the line will easily understand by simply look at it. **i.e.) Federal_W4**
- 2) Navigate to the directory where DocCenter is installed, usually **C:\Program Files\TempWorks Software\DocCenter 3** and copy the following files to your website project.
 - ✓ Calendar.png
 - ✓ Date.js
 - ✓ Datepicker.css
 - ✓ All JQuery files
 - ✓ Masterstyle.css
- 3) Acquire the PDF that you would like to imitate and save it to your project directory.
- 4) Open up the PDF and go to **print**.

Form Creation Walkthrough For HTML v3.1.9079

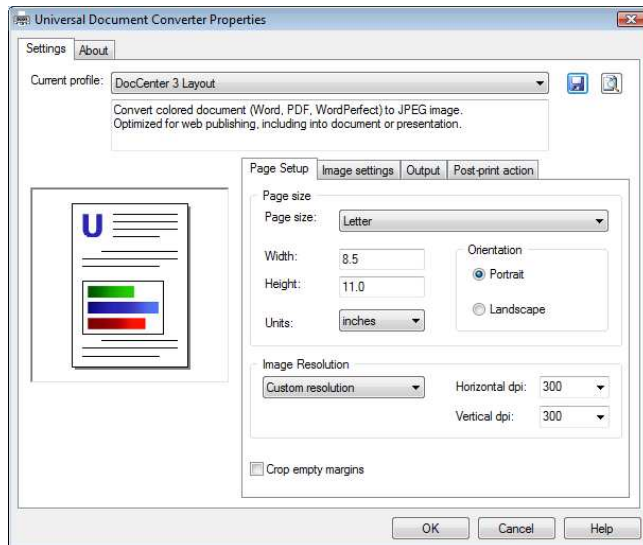
Last Updated 06/08/09 Brandon J.

5) When you see the following window select the option; **Universal Document Converter**. Once that is selected, press the **Properties** button.



6) From the options window you will see the following options available.

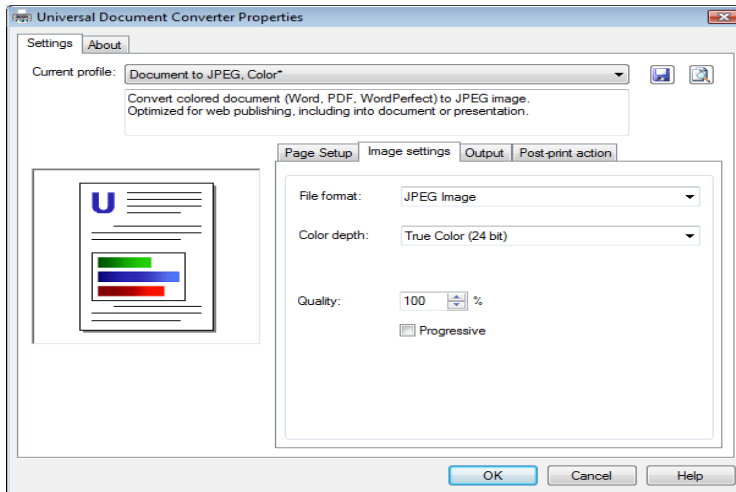
- a. First you'll need to adjust the size of the image when it actually gets saved. You can do that by simply setting the document **Height** and **Width** to equal **8.5" W x 11" H**. When setting your size, make sure you set the unit dropdown to equal **inches**. Also make sure the **Portrait** orientation option is selected and not the **Landscape** orientation.



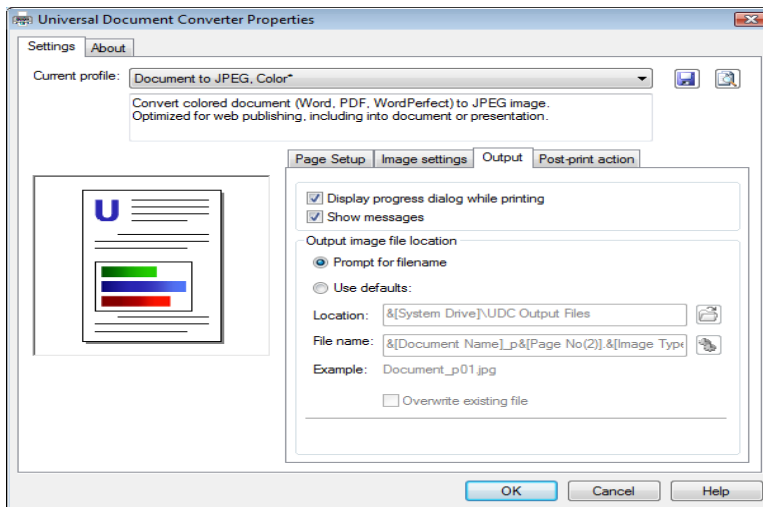
Form Creation Walkthrough For HTML v3.1.9079

Last Updated 06/08/09 Brandon J.

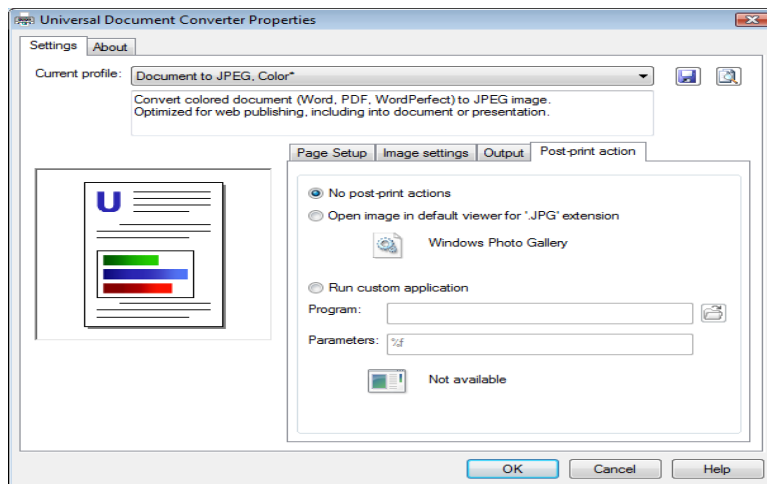
- b. The next tab in the properties window is **image settings**. Just make sure the settings match the image below.



- c. The next tab in the properties window is **Output**. The only thing important here, is that you should set the **Prompt for filename** option so that it doesn't save it automatically somewhere outside of your soon to be project directory.



- d. The last tab in the collection of properties is the **Post-print action**, which simply asks what you want to do after it's done saving your images. Just set the **No post-print actions** option. After everything is set, click the **OK** button to save your changes.



7) Now that you are back to the print window, go ahead and click **OK** to continue with your printing process. When the save window pops up, make sure you save any forms that have the potential to be used later in the shared directory.

NOTE: If the PDF is multiple pages then make sure your naming them accordingly. **i.e.) FederalW4_Page1, etc.**

8) Open each picture up in Paint.NET and resize the image down to as close as you can to **1005px** wide. After you make your adjustments, save the image.

9) Now that you have your images in your same folder as your project and you have your project open and ready to go, let's begin our form. If you haven't done so already, go ahead and create a new blank html page; name it whatever you want.

10) If it isn't already, open it so that you see the html code. Once opened, we will begin to start coding the form. Refer the following example as they relate to your page.

- a. The top part of your page should contain at the very least the following statements. The **title** tag is simple the title of the page and isn't really relevant as DocCenter uses its own method of gathering the title from the dcfx_FormConfig table. The **link** tags represent the connection for the external CSS documents that hold all the styling for the calendar control. This is a must in order to make sure the calendar is formatted and aligned as it should. The next are the java script external document references. Those can be distinguished by the **script** tags as you see below. Once these are defined in the page you plan on using the java script you can use any function from any one of those linked scripts.

Figure 1.

```
<title>Federal W-4</title>
<link rel="stylesheet" type="text/css" href="masterstyle.css" />
<link rel="stylesheet" type="text/css" href="datePicker.css" />
<script src="jquery.js" type="text/javascript"></script>
<script src="date.js" type="text/javascript"></script>
<script src="jquery.datePicker.js" type="text/javascript"></script>
<script src="jquery.validate.js" type="text/javascript"></script>
```

Form Creation Walkthrough For HTML v3.1.9079

Last Updated 06/08/09 Brandon J.

- b. On most of all the forms that will be done the calendar control will need to be used. How this is approached is simply by clicking on the date field and the calendar will pop up in place of it. How is this done? Simple; we use the **script** tag again to write some of our own logic. Notice the first function in that resides under the script tag below (Figure 2). This is the main function that gets fired from the body tags **onload** event of the page. When the InitCalendar function is fired, it will begin to execute one line after another, first being, setting today's date to the date field on the page. After which time it will then initialize an **onclick** event for the date field because we need to be able to have the form open the calendar when we click the date field. Once the system initializes the Date field **onclick** event, the system knows that when you click that Date field, it will then fire the **OpenCalendar** function. The **OpenCalendar** function does two main things; it will first display the calendar, and second move the calendar to what the dpSetOffset property is set to.

Figure 2.

```

<script type="text/javascript">
    function InitCalendar() {
        $("#Date").val(new Date().asString());
        document.getElementById("Date").onclick = function() {
            OpenCalendar("#Date");
            var aCalLink = document.getElementById("aCal");
            if (null != aCalLink)
                RemoveCalendar(aCalLink);
        };
        document.getElementById("Date").onmouseover = function() {
            document.getElementById("Date").title = "Please click to change the date";
        };
    }
    function OpenCalendar(element) {
        $("input" + element).datePicker({ clickInput: true }); //Open Calendar when clicking on
        $(element).dpSetOffset(20, 0); //Offsets the calendar to 20 pixel
    }
    function RemoveCalendar(element) {
        element.parentNode.removeChild(element);
    }
</script>

```

the Date field.
down from the date text field.

- c. Now for looks, the following code is simply the styling side for the controls on this page. It does two things; first, every control on the form that is using the **class** name **.TextField** will be formatted in the way the class is defined as. The TextField class is meant to show minimal formatting for the end user but still present the control well enough. During design time there shouldn't be any control that uses the **.CleanField** class as it is used for when an employee is done filling out their forms. During the post procedure, the system will automatically switch every control from bordered representation to non bordered representation, TextField to CleanField.

Figure 3.

```
<style type="text/css">
  /*Standard format when viewing*/
  .TextField
  {
    border-style:solid;
    border-color:#000000;
    border-width:1px;
    border-style:dotted;
    background-color:White;
  }
  /*Used for when the form gets created to a PDF and saved to employees file. Removes border from PDF
  fields*/
  .CleanField
  {
    border-style:solid;
    border-color:#000000;
    border-width:0px;
    border-style:dotted;
    background-color:White;
  }
</style>
```

- d. Now that we have all the contents of the heading in place and explained, let's move on to the body of the form. There are a few things that are needed to be explained and are necessary in order for DocCenter to be used properly. First things first, everything that is done has to be done in between the **body** tags of the page. The **form** tag is signifying that everything inside of the form will be able to be binded to the form action for validation. The **id** and the **name** properties of the form need to be **DocCenterForm**. The **div** control by the name of **errors** is used to display any required messages that the user still needs to fill out. The validation is done from **Figure 5**. The next **div** tag is a container tag, it holds all the content because as you can see the property, **background-image** shows the image that we converted prior up at the top of the page and we want all the controls to be placed ontop of that image. Also make sure that the dimensions of the container div is set to the dimensions of the images we previously converted. The next **div** tag by the id/name of **valid**, is a tag that will determine whether the person is able to go forth in DocCenter when there are still errors on the form. The value of this should always be **false** unless you would like otherwise.

Figure 4.

```
<body onload="javascript:InitCalendar();">
  <form id="DocCenterForm" name="DocCenterForm" action="" method="post">
    <div id="errors" style="position:relative; left:50px;"></div>
    <input type="hidden" name="valid" id="valid" value="false" />
```



Form Creation Walkthrough For HTML v3.1.9079

Last Updated 06/08/09 Brandon J.

```
<div style="position:relative; background-image:url(FederalW4_Page1.jpg); background-repeat:no-repeat; width:1005px; height:1305px">
  <input id="FirstName" name="FirstName" class="TextField" style="position:absolute; top:930px; width:280px; height:auto; left: 89px;" type="text"/>
  <input id="LastName" name="LastName" class="TextField" style="position:absolute; top:929px; width:347px; height:auto; left: 379px;" type="text"/>
  <input id="Date" name="Date" class="TextField" style="position:absolute; top:1184px; width:108px; height:auto; left: 788px;" type="text"/>
  <input id="Qst" name="Qst" class="TextField" style="position:absolute; left:926px; top:1008px;" type="checkbox"/>
  
</div>
<div style="position:relative; background-image:url(FederalW4_Page2.jpg); background-repeat:no-repeat; width:1005px; height:1305px"></div>
</form>
</body>
```

- e. For more advanced validation, you will need to create validation rules and messages. Please see the below **Figure 5** for reference.

Figure 5.

```
$(document).ready(function() { //setup the validator with messages, rules, and error placement.
  var validator = $("#DocCenterForm").validate({
    messages: {
      FirstName: "<br>Please enter your first name.",
      LastName: "<br>Please enter your last name."
    },
    rules: {
      FirstName: { required: true },
      LastName: { required: true }
    },
    errorPlacement: function(error, element) {
      error.appendTo("#errors");
    }
  }); //validate the form using .valid() to validate the whole form, not just one input element.
  $("#DocCenterForm").valid();
  $("input").blur(function() {
    $(document.getElementById("Qst")).val((document.getElementById("Qst")).checked); //Set the
    value of the checkbox to itself. If you don't do this by default it uses "On" regardless of value
    $(document.getElementById("valid")).val($("#DocCenterForm").valid());
  });
});
```