enfocus CONNECT 2021

HTML Job Tickets

02 - 2022

enfocus

Contents

1. Copyrights **3**

2. Introduction **4**

3. HTML and CSS: the basics **5**

3.1. What is HTML? 5

3.2. What is CSS? 5

3.3. How to apply CSS to an HTML document? 5

4. Explaining a sample HTML Job Ticket and the associated CSS file **8**

4.1. About the file structure 8

4.2. About the subtitle 10

4.3. About the input fields 10

4.4. About the Submit button 11

4.5. Samples 12

4.5.1. Basic Job Ticket - sample HTML code 12

4.5.2. Basic Job Ticket with extra fields - sample HTML code 13

4.5.3. Basic Job Ticket - sample CSS file 16

5. Known limitations **19**

1. Copyrights

© 2022 Enfocus BV all rights reserved. Enfocus is an Esko company.

Certified PDF is a registered trademark of Enfocus BV.

Enfocus PitStop Pro, Enfocus PitStop Workgroup Manager, Enfocus PitStop Server, Enfocus BoardingPass, Enfocus Connect YOU, Enfocus Connect ALL, Enfocus Connect SEND, Enfocus StatusCheck, Enfocus CertifiedPDF.net, Enfocus PDF Workflow Suite, Enfocus Switch, Enfocus SwitchClient, Enfocus SwitchScripter, Enfocus TestDrive, Enfocus SwitchScriptTool, Enfocus Browser and Enfocus Appstore are product names of Enfocus BV.

Adobe, Acrobat, Distiller, InDesign, Illustrator, Photoshop, FrameMaker, PDFWriter, PageMaker, Adobe PDF Library™, the Adobe logo, the Acrobat logo and PostScript are trademarks of Adobe Systems Incorporated.

Datalogics, the Datalogics logo, PDF2IMG™ and DLE™ are trademarks of Datalogics, Inc.

Apple, Mac, macOS, Macintosh, iPad and ColorSync are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. Windows and Windows Server are registered trademarks of Microsoft Corporation.

PANTONE® Colors displayed here may not match PANTONE-identified standards. Consult current PANTONE Color Publications for accurate color. PANTONE® and other Pantone, Inc. trademarks are the property of Pantone, Inc. ©Pantone, Inc., 2006.

OPI is a trademark of Aldus Corporation.

Quark, QuarkXPress, QuarkXTensions, XTensions and the XTensions logo among others, are trademarks of Quark, Inc. and all applicable affiliated companies, Reg. U.S. Pat. & Tm. Off. and in many other countries.

This product and use of this product is under license from Markzware under U.S.Patent No. 5,963,641.

Other brand and product names may be trademarks or registered trademarks of their respective holders. All specifications, terms and descriptions of products and services are subject to change without notice or recourse.

2. Introduction

About this document

In this document, we will explain the basics of HTML and CSS and provide you with a sample HTML Job Ticket and a sample CSS file.

Why this document?

Although knowledge of HTML or CSS is not required for using the Job Ticket functionality - Connect provides you with a default Job Ticket, in a standard layout and with easily configurable fields - , you can considerably enhance the look and feel and the content of your Job Tickets, if you are familiar with HTML and CSS.

Job Tickets in Connect - where HTML and CSS come in

Enfocus Connect supports three types of Job Tickets:

- The **basic Job Ticket** defined in Connect
- The **local HTML Job Ticket**, basically an HTML file uploaded from a local system
- The **external HTML Job Ticket**, an HTML file linked to e.g. a website

All three types rely on *HTML* - also the basic Job Ticket. Although the basic Job Ticket is defined in the Connect application, it is internally transformed into HTML before it is displayed to the user.

The look and feel - colors, font, layout,... - of all three types of Job Tickets is defined using *CSS*. If you're using the basic Job Ticket with the standard layout, a default CSS file present in the background is used; if you're using the option to customize the layout of the basic Job Ticket (through the **HTML Layout** button), you are asked to upload a CSS file yourself. This document will help you in creating such a CSS file.

3. HTML and CSS: the basics

3.1. What is HTML?

HTML (**H**yper **T**ext **M**arkup **L**anguage) is the language used to create web pages.

It's based on XML and uses markup tags (such as 'table', 'list', 'paragraph', ...) to describe the structure of web pages.

Example:

In HTML, a paragraph is represented by the paragraph element, written as <p>.

```
<p>This is a paragraph</p>
```

3.2. What is CSS?

CSS (**C**ascading **S**tyle **S**heets) is a language that is used to define the look and feel of web pages, including colors, fonts, the layout ... It was originally developed to separate the style elements of the document from its structure.

CSS works with rules that are usually defined in a stylesheet (an external file with extension .css). If you want to change the style of the document, it is sufficient to only change the style sheet in order to have the changes on all webpages where the stylesheet is applied.

Example

If your stylesheet contains the following line, all paragraphs in the document the stylesheet applies to will be displayed in blue.

```
p {
  color:blue;
  }
```

3.3. How to apply CSS to an HTML document?

In order to apply a particular CSS stylesheet to a particular HTML document, the HTML document must refer to the stylesheet using the <link> element. As such, CSS is one of the resources of the HTML document.

Example

Suppose you have prepared an HTML document (MyWeb.html) and a CSS stylesheet (MyWebPage.css), placed in two different directories (see screenshot below).

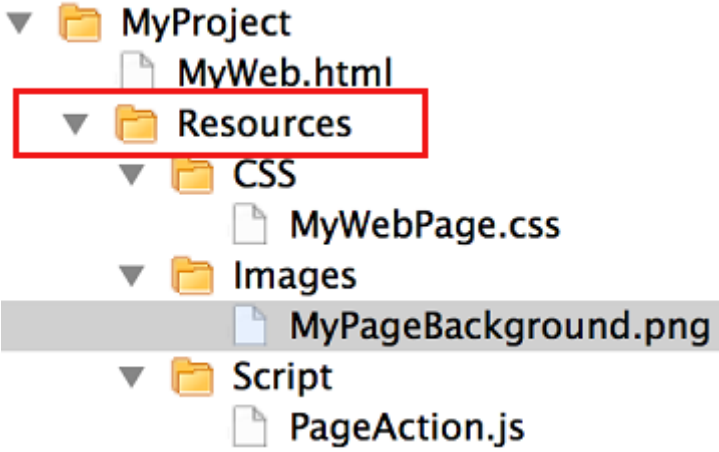
In order to correctly apply the stylesheet to the document, the HTML file (MyWeb.html) should contain the following code:

```
<head>
<link rel="stylesheet" type="text/css" href="CSS/MyWebPage.css"/>
</head>
```

Note the location of the CSS file. It is defined using a relative path.

Important remark

Beside CSS, a number of other resources can be associated with an HTML document, for example: JavaScript files, images, etc. If you are developing a Job Ticket for use with Enfocus Connect, make sure to place **all these resources under one single resource folder**! This is mandatory!

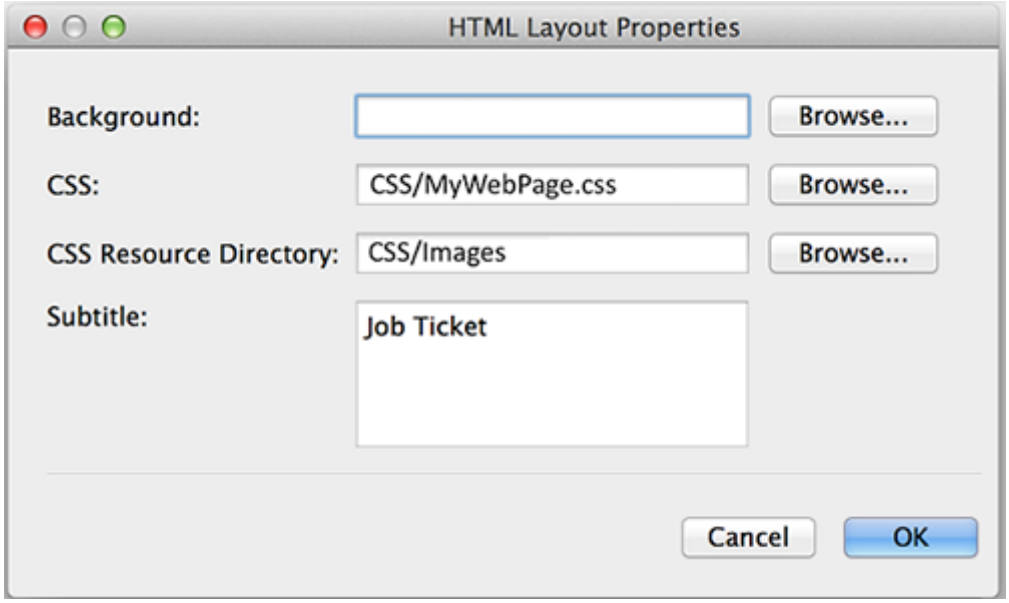


CSS resources

CSS stylesheets can also refer to resources, for example to a background image or a logo. To work well with Connect, these resources must be placed in a separate directory.

In the example below, the Images folder is the resource folder for the CSS file "MyWebPage.css".

The previous explained how to apply a CSS stylesheet to an HTML document. However, to use your own CSS file with the Connect basic Job Ticket, you should only enter the path to your CSS stylesheet and CSS Resource directory in the HTML Layout properties dialog (see screenshot).



4. Explaining a sample HTML Job Ticket and the associated CSS file

About the samples

The Connect user has no control over the HTML of the Basic Job Ticket. Connects provides a user interface for the definition of the fields and creates the Job Ticket in the background during the creation of the Connector.

However, in order to understand how your CSS - the part that you can control - will affect the Job Ticket, we provide you with two samples of the internal HTML used for the basic Job Ticket.

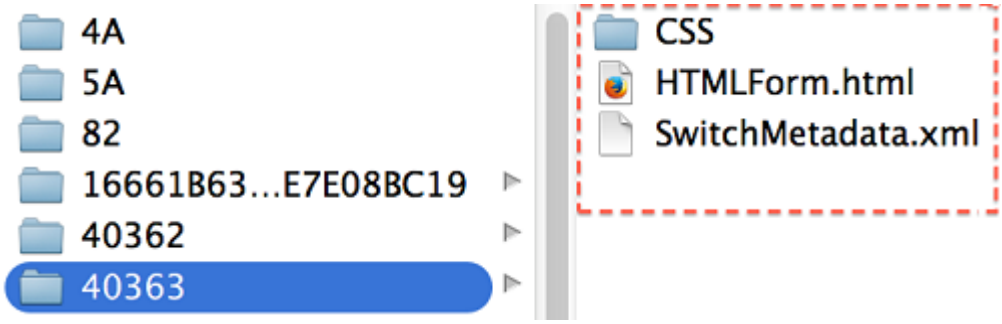
The CSS sample file provided here is the one used by Connect if you do not upload your own CSS. It's meant as an example to help you build your own CSS.

The sample files are enclosed in this document, at the end of this chapter.

Location of the sample files on your computer

The **HTML sample file** is not meant to be viewed by Connect users. However, there is a way to capture it:

1. In Connect, open the Job Ticket tab.
2. Turn on **Enable Job Tickets** and choose a **basic** Job Ticket.
3. Click the **Preview** button.
4. Leave the Preview open, while searching for a recently created folder in the **temp** folder. There you should find the CSS folder, HTMLForm.HTML and SwitchMetadata.xml.



The **CSS sample file** can be found on your local machine:

- Mac: /Library/Application Support/Enfocus/Connect ALL/<version>/Connectors/HTMLMetadataResources/CSS
- Windows: C:\ProgramData\Enfocus\Connect ALL\<version>\Connectors\HTMLMetadataResources\CSS

4.1. About the file structure

A HTML consists of two main parts:

- HEAD (enclosed in <head> tags) : contains the scripts that control the elements in the body part and a link to the CSS file.

- BODY (enclosed in <body> tags) : contains the form elements that are displayed when submitting a job file to the Connector.

```
1  <html xmlns:EnfocusConnectNS="http://www.enfocus.com/2013/EnfocusConnectMetadataPreset">
2  <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4      <script type="text/javascript">
5      </script>
6      <link rel="stylesheet" type="text/css" href="CSS/EnfocusSwitchMetadata.css">
7      <title>Preview of Job Ticket: 'Basic JT'</title>
8  </head>                                                                    Head
9  <body style="background: url(Images/BackgroundImage) no-repeat fixed center">
10     <form action="http://" name="EnfocusConnectMetadata" enctype="application/x-www-form-urlencoded">
11         <fieldset>
12             <p class="introduction">Job Ticket </p>
13             <p class="note">
14                 Fields marked with an asterisk (<abbr title="Required Field">*</abbr>) are required
15             </p>
16             <div id="MF_1">
17                 <label for="MF_1_text" title="">
18                     Name:<abbr title="Required Field">*</abbr>
19                 </label>
20                 <input value="" name="MF_1" id="MF_1_text" required="" type="text">
21             </div>
22
23             <input type="submit" value="Submit" class="controls">
24         </fieldset>
25     </form>
26 </body>                                                                    Body
27 </html>
28
```

The code in the CSS file describes how the body and form part look like.

- The entire Job Ticket (= the complete "html") gets a gray background:

```
html {
    background: #eee;
     }
```

- The body part ("body") gets the same background. The text in the body part will use the Helvetica Neue font family (or, if not available: Helvetica, Arial, sans-serif) and have a fixed size.

```
body {
    background: #eee;
    color: #333;
    font: 75%/1.3 "Helvetica Neue", Helvetica, Arial, sans-serif;
    padding: 20px;
      }
```

- The form part ("form") of the body is embedded in a box (actually all HTML elements can be considered boxes). The margins, border and padding of this box can be changed through CSS. Padding is the white space between the element content and the element border.

```
form {
    padding: 20px;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
     }
```

4.2. About the subtitle

There is no specific <subtitle> HTML element used; instead, the subtitle is enclosed in paragraph tags:

```
<p class="introduction">Job Ticket</p>
```

This paragraph is assigned a class ("introduction"), which determines how it looks. The class is defined as follows in the CSS file:

```
form p.introduction
    {
    font-size: 1.5em;
    font-weight: bold;
    line-height: 1.6;
    margin: 0 0 2em;
    color: rgba(255,255,255,1);
    /* background: linear-gradient(to right, rgba(154,169,16,1), rgba(154,169,16,0));
 Esko green */
    /* background: linear-gradient(to right, rgba(249,153,69,1), rgba(249,153,69,0));
 Connect orange */
    background: linear-gradient(to right, rgba(155,155,155,1), rgba(155,155,155,0));
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
    padding: 6px 0 6px 10px;
    }
```

The first line (*form p.introduction*) is called the CSS selector: it indicates where the CSS code should be applied; in this example, it should be applied to a paragraph element (*p*) inside a form (*form*), which is assigned the class "introduction" (*.introduction*) - the dot refers to a class.

Note: If you're using the basic Job Ticket in Connect, you can change the subtitle via the Connect interface (**HTML Layout** button).

4.3. About the input fields

The code below defines an input field in the form:

```
<div id="MF_1">
     <label for="MF_1_text" title="">
          Name:<abbr title="Required field">*</abbr>
     </label>
     <input value="" name="MF_1" id="MF_1_text" required="" type="text">
</div>
```

Each input field is put inside a (a division or a section in an HTML document);

- the <label> element is used to display which information you need - in this example "Name"
- the <input> element is used to create an input field where the user can enter the requested information.
- The id (id="MF_1_text") makes clear the label and the input field belong together.

The input is decorated through *two sets of CSS selectors*:

- The first CSS selector places a shadow effect around the input area:

```
fieldset div input,
fieldset div textarea,
fieldset div select {
    line-height: 1;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
}
```

- The second CSS selector displays a highlighted boundary around the input when the input has keyboard focus.

```
/* Custom focus selection */
input:focus,
textarea:focus,
select:focus {
    border: 2px solid #26a0ac;
}
```

Note: ":focus" is a pseudo-class in CSS; a class that is used to define a special state of an element. Other pseudo-classes are ":hover", ":checked", ...

Example of a possible change to the CSS

Suppose you want to use an input field with a round corner. This can easily be done by using the *border-radius* property:

```
fieldset div input{
    line-height: 1;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
    border-radius: 5px;
}

fieldset div textarea,
fieldset div select {
    line-height: 1;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
    border-radius: 5px;
}
```

Same change if the input field has focus:

```
/* Custom focus selection */
input:focus,
textarea:focus,
select:focus {
    border: 2px solid #F99945;
    border-radius: 5px;
}
```

4.4. About the Submit button

The HTML code below defines the Submit button:

```
<input type="Submit" value="Submit" class="controls">
```

The 'value' refers to the actual string displayed to the user; this can be anything.

The look of the Submit button is defined by the following lines in the CSS file:

```
input[type=submit] {
    cursor: pointer;
    float: right;
    background: url(images/Submit.png) no-repeat 0 0;
    width: 115px;
    height: 45px;
    margin: 0;
    overflow: hidden;
    padding: 32px 0 0 116px;
    border: none;
}

input[type=submit]:active {
    background: url(images/Submit_down.png) no-repeat 0 0;
}
```

Note: The CSS selector is written in a slightly different way, i.e. through an attributes type (input[type=submit]); this notation says that the style should be applied to <input> of the type Submit.

If you have a look at the code, you can see that there are two different styles defined for the button. The first one is the style for the normal state of the button, the second one is how the button will look when it is clicked upon. To define this last state, the pseudo class *:active* is used.

In this part of the CSS, two background images are specified. They are assumed to be placed in a directory that is defined relative to the CSS file.

4.5. Samples

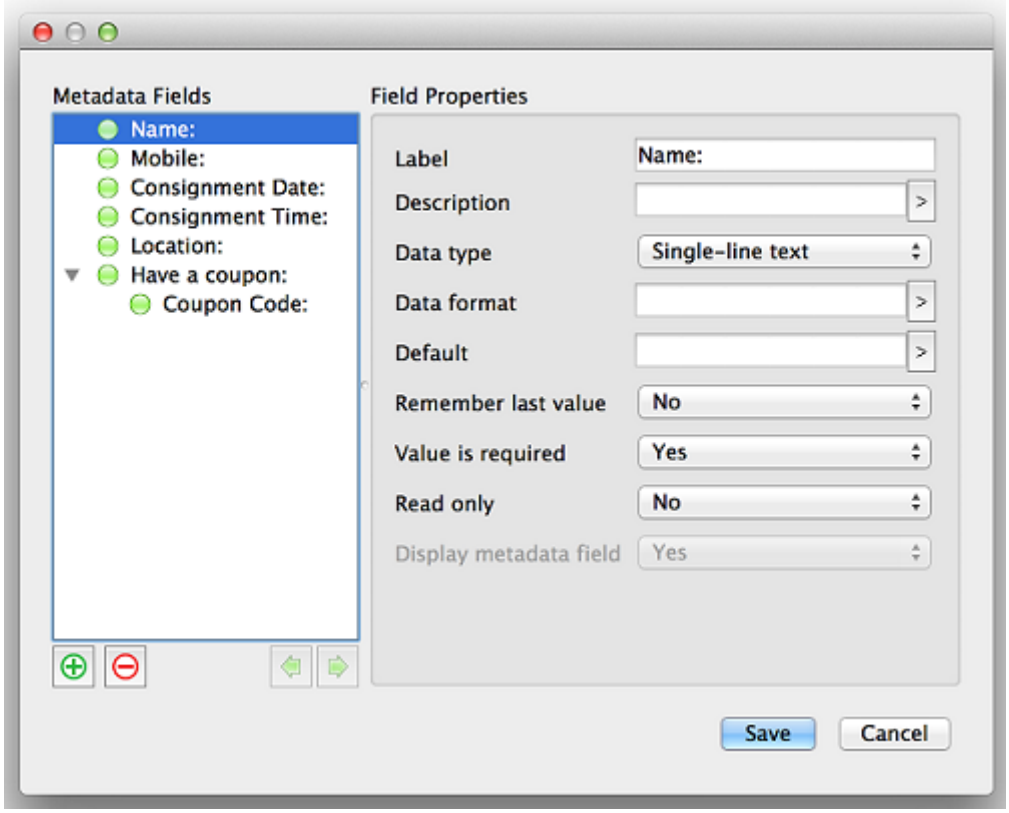
4.5.1. Basic Job Ticket - sample HTML code

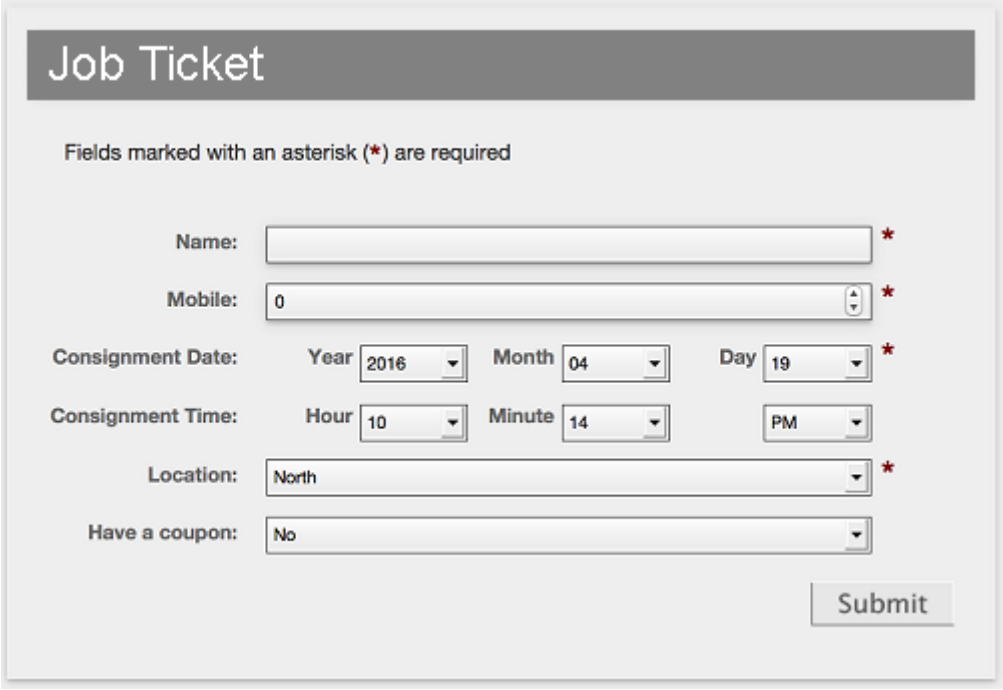
```
<html xmlns:EnfocusConnectNS="http://www.enfocus.com/2013/
EnfocusConnectMetadataPreset">

<head>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
      <script type="text/javascript">
      </script>
      <link rel="stylesheet" type="text/css" href="CSS/EnfocusSwitchMetadata.css">
      <title>Preview of Job Ticket: 'Basic JT'</title>
</head>

<body style="background: url(Images/BackgroundImage) no-repeat fixed center">

      <form action="http://" name="EnfocusConnectMetadata" enctype="application/x-www-
form-urlencoded">
            <fieldset>
                  <p class="introduction">Job Ticket'</p>
```

```
                    <p class="note">Fields marked with an asterisk (abbr
 title="Required Field">*</abbr>) are required</p>
                    </p>
                    <div id="MF_1">
                          <label for="MF_1_text" title="">
                               Name:<abbr title="Required field">*</abbr>
                          </label>
                          <input value="" name="MF_1" id="MF_1_text" required=""
 type="text">
                    </div>

                    <input type="Submit" value="Submit" class="controls">
            </fieldset>
        </form>
</body>

</html>
```

4.5.2. Basic Job Ticket with extra fields - sample HTML code

Below is an example of the code generated by Connect. It's a basic Job Ticket with extra fields added through the Connect interface (see screenshot). The second screenshot shows how the preview would look in Connect.






```
<html xmlns:EnfocusConnectNS="http://www.enfocus.com/2013/
EnfocusConnectMetadataPreset">
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <script type="text/javascript">
  ...
    </script>
    <link rel="stylesheet" type="text/css" href="CSS/EnfocusSwitchMetadata.css">
 <title>Preview of Job Ticket: 'Basic JT'</title>
</head>
<body style="background: url(Images/BackgroundImage) no-repeat fixed center"
 onload="OnLoad()">
 <form onsubmit="OnFormSubmit(event)" action="http://" method="post" autocomplete="on"
 name="EnfocusConnectMetadata" enctype="application/x-www-form-urlencoded">
  <fieldset>
   <p class="introduction">Job Ticket</p>
   <p class="note">Fields marked with an asterisk (<abbr title="Required Field">*</
abbr>) are required</p>
   <div id="MF_1">
    <label for="MF_1_text" title="">Name:<abbr title="Required Field">*</abbr></label>
    <input value="" name="MF_1" id="MF_1_text" required=""
 onchange="EnableChildNodes([[this,'','','']])" type="text">
   </div>
   <div id="MF_2">
    <label for="MF_2_number" title="">Mobile:<abbr title="Required Field">*</abbr></
label>
    <input value="0" name="MF_2" id="MF_2_number" required=""
 onchange="EnableChildNodes([[this,'','','']])" type="number">
   </div>
   <div id="MF_3">
    <label for="MF_3_date" title="">Consignment Date:<abbr title="Required Field">*</
abbr></label>
    <fieldset class="datetime">
     <input name="MF_3" type="hidden" id="MF_3_date" required=""
 onchange="EnableChildNodes([[this,'','','']])" value="2016-04-19" updateonload="true"
 onload="OnDateFieldLoading( 'MF_3' )">
     <div>
      <label for="MF_3_year">Year</label>
      <select id="MF_3_year" name="MF_3_year" onchange="DateChanged( 'MF_3' )">
       <option value="2011">2011</option>
       <option value="2012">2012</option>
       ...
       <option value="2016" selected>2016</option>
       ...
```

```
      </select>
     </div>
     <div>
      <label for="MF_3_month">Month</label>
      <select id="MF_3_month" name="MF_3_month" onchange="DateChanged( 'MF_3' )">
       <option value="1">01</option>
       ...
      </select>
     </div>
     <div>
      <label for="MF_3_day">Day</label>
      <select id="MF_3_day" name="MF_3_day" onchange="DateChanged( 'MF_3' )">
       ...
      </select>
     </div>
    </fieldset>
   </div>
   <div id="MF_4">
    <label for="MF_4_time" title="">Consignment Time:</label>
    <fieldset class="datetime">
     <input name="MF_4" type="hidden" id="MF_4_time" value="" updateonload="true"
 onload="OnTimeFieldLoading( 'MF_4' )">
     <div>
      <label for="MF_4_hour">Hour</label>
      <select id="MF_4_hour" name="MF_4_hour" onchange="TimeChanged( 'MF_4' )">
       ...
      </select>
     </div>
     <div>
      <label for="MF_4_minute">Minute</label>
      <select id="MF_4_minute" name="MF_4_minute" onchange="TimeChanged( 'MF_4' )">
       ...
      </select>
     </div>
     <div>
      <label for="MF_4_ampm"></label>
      <select id="MF_4_ampm" name="MF_4_ampm" onchange="TimeChanged( 'MF_4' )">
       <option value="AM">AM</option>
       <option value="PM" selected>PM</option>
      </select>
     </div>
    </fieldset>
   </div>
   <div id="MF_5">
    <label for="MF_5_select" title="">Location:<abbr title="Required Field">*</abbr></
label>
    <select name="MF_5" id="MF_5_select" required=""
 onchange="EnableChildNodes([[this,'','','']])">
     <option value="North" selected>North</option>
     <option value="South">South</option>
     <option value="East">East</option>
     <option value="West">West</option>
    </select>
   </div>
   <div id="MF_6">
    <label for="MF_6_select" title="">Have a coupon:</label>
    <select name="MF_6" id="MF_6_select"
 onchange="EnableChildNodes( [[this,'MF_7','Equals','Yes']] )">
     <option value="Yes">Yes</option>
     <option value="No" selected>No</option>
    </select>
   </div>
   <div id="MF_7" style="display: none;">
    <label for="MF_7_number" title="">Coupon Code:</label>
    <input value="0" name="MF_7" id="MF_7_number" type="number">
   </div>
   <input type="submit" value="Submit" class="controls">
  </fieldset>
 </form>
</body>
```

```
</html>
```

4.5.3. Basic Job Ticket - sample CSS file

```
/*
 Eric Meyer's Reset CSS: Our default setting.
 http://meyerweb.com/eric/tools/css/reset/
 v2.0 | 20110126
 License: none (public domain)
 */
html, body, blockquote, pre, abbr, acronym, address, code,
em, img, dl, dt, dd, ol, ul, li,
fieldset, form, label,  legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    vertical-align: baseline;
}

ul {
    list-style: none;
}

table {
    border-collapse: separate;
    border-spacing: 0;
}

caption, th, td {
    text-align: left;
    font-weight: normal;
}

table, td, th {
    vertical-align: middle;
}

blockquote:before, blockquote:after, q:before, q:after {
    content: "";
}

blockquote, q {
    quotes: "" "";
}

a img { border : none }

/* Custom setting start */
/* ------------------------------------------------------- */
html {
    background: #eee;
}

body {
    background: #eee;
    color: #333;
    font: 75%/1.3 "Helvetica Neue", Helvetica, Arial, sans-serif;
    padding: 20px;
}

form {
    padding: 20px;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
}

input.ValidationError:focus,
textarea.ValidationError:focus,
select.ValidationError:focus {
    border: 2px solid #ff0000;
}
```

```
/* Custom top paragraph */
form p.introduction {
    font-size: 1.5em;
    font-weight: bold;
    line-height: 1.6;
    margin: 0 0 2em;
    color: rgba(255,255,255,1);
    /* background: linear-gradient(to right, rgba(154,169,16,1), rgba(154,169,16,0));
 Esko green */
    /* background: linear-gradient(to right, rgba(249,153,69,1), rgba(249,153,69,0));
 Connect orange */
    background: linear-gradient(to right, rgba(155,155,155,1), rgba(155,155,155,0));
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
    padding: 6px 0 6px 10px;
}

form p.note {
    font-size: 1.25em;
    margin: -1.7em 0 1.25em;
    padding: 20px;
}

/* Side by side labels */
fieldset div {
    margin: 0 0 10px 160px;
}

fieldset div.controls {
    margin: 25px 0 0;
    padding: 0;
}

fieldset div label {
    color: #666;
    float: left;
    font-size: 1.16em;
    font-weight: bold;
    line-height: 1.1;
    margin: 5px 20px 0 -160px;
    width: 140px;
    text-align: right;
}

fieldset abbr {
    color: #7b0101;
    font-size: 20px;
    font-weight: bold;
    line-height: 1;
    padding: 0 1px;
    vertical-align: middle;
 }

 fieldset div label abbr {
     display: block;
     font-size: 23px;
     padding: 0;
     position: relative;
     width: 16px;
     left: 580px;
     top: -0.9em;
}

fieldset div input,
fieldset div textarea,
fieldset div select {
    line-height: 1;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.16), 0px 2px 10px 0px rgba(0, 0, 0,
 0.12) !important;
}


/* Custom input, textarea and submit button */
input,
textarea,
select {
```

```
    border: 1px solid #666666;
    padding: 5px;
    width: 420px;
    background: linear-gradient(to bottom, rgba(255,255,255,1) 0%,rgba(243,243,243,1)
 50%,rgba(237,237,237,1) 86%,rgba(255,255,255,1) 100%);
    min-height: 26px;
}

textarea {
    height: 140px;
}

input[type=submit] {
    cursor: pointer;
    float: right;
    background: url(images/Submit.png) no-repeat 0 0;
    width: 115px;
    height: 45px;
    margin: 0;
    overflow: hidden;
    padding: 32px 0 0 116px;
    border: none;
}

input[type=submit]:active {
    background: url(images/Submit_down.png) no-repeat 0 0;
}

/* Custom focus selection */
input:focus,
textarea:focus,
select:focus {
    border: 2px solid #26a0ac;
}

input[type=submit]:focus {
    border: none;
}

/* Date */
fieldset fieldset.datetime {
    color: #666;
    margin: 0;
    padding: 0;
}

fieldset fieldset.datetime div {
    background: none;
    float: left;
    margin: 0;
    padding-left: 0;
}

fieldset.datetime div label {
    width: 60px;
    margin: 5px 0 0 0;
    padding: 0;
}

fieldset.datetime div select {
    width: 75px;
    margin: 5px 0 0 5px;
    padding: 0;
}

/* EOF */
```

5. Known limitations

When creating or uploading an HTML form, make sure not to use the following elements as these are **not yet supported**:

- File upload - do not ask your customers to upload files through a Job Ticket.
- Video
- JPEG 2000 images
- Dates outside the range of 2011 – 2025 - make sure your customers do not have to enter an earlier or a later date.
- Two submit points on the same HTML page. This is not allowed because clicking a Submit button triggers file submission.
- Empty form actions (especially important for local HTML). The form with empty action will not be submitted and no form data is captured for Job Ticket.

```
WRONG:

<form action=""...>
  ...
</form>
```

```
RIGHT:

At least, add the protocal part of the URL:

<form action="http://"...>
...
</form>
```

Note: When using External HTML, or CSS for a Basic Job Ticket, we advise testing them on an external browser before using them in Connect.