



How To

Expertise at Your Fingertips

www.exsys.com

Using CSS with Corvid Servlet Runtime Templates

A system demonstrating this “How To” can be run under the “Using CSS with Corvid Servlet Runtime Templates” section of:

<http://www.exsys.com/support/howto>

The code for the sample system can be downloaded from the same page.



The Exsys Corvid Servlet Runtime uses templates to ask questions and show the results. The templates are created using HTML to define the look and feel. There are special Corvid commands embedded in the templates that tell Corvid what sections to use and in what way, but these commands are inserted as HTML comments. Corvid does not require anything beyond simple HTML, but anything that is supported in an HTML page can be used. Cascading Style Sheets (CSS) is a good example of something that is not required, but which can be used to make pages look better, and make them more functional.

CSS makes it possible to easily separate the details of the look-and-feel of a template from the content of the template. The template only needs to have the bare structure of asking the question. An associated CSS file can control appearance (fonts, colors, sizes, layout, etc). By simply changing the CSS file, the look-and-feel of a system can be completely changed. CSS also allows the appearance of the system to change depending on how it is being viewed. One design may be used for full size monitors and another for PDAs.

The advantages of CSS are:

1. It allows the '.html' of the template to be very simple. The templates contain only content placeholders for the actual question content (most of which comes from the KB), and only simple formatting to specify the type of control to use to ask questions (radio button, checkbox, etc.) and some simple arrangement information.
2. To change the look-and-feel, only the '.css' file needs to be changed.
3. When the '.css' file is changed, the look-and-feel changes consistently in all templates if they all use the same '.css' file to control the look-and-feel.
4. The size of the .html files are much smaller so they download faster. (The CSS file is only downloaded once and cached by the browser.)
5. It is possible for the user to switch between '.css' files. The user can select the look-and-feel of the page for special situations. (When using the sample templates, click on the 'High Contrast' link and the whole look-and-feel of the web page will change.)
6. A '.css' file can automatically adjust the design for specific hardware such as PDA's, printers or projectors.

Remember, that while CSS is very widely supported in modern browsers, some users may be using an old browser that does not support CSS. This is rapidly becoming increasingly rare, and for many user communities you can count on the browser supporting CSS. The percentage that may not have a current browser will depend on the intended audience for your system. It is also possible that some users may have CSS support turned off. Because of this, the templates should be designed so that they will still function without CSS – although without the desired appearance.

NOTE: These instructions assume some knowledge of CSS. They cover the issues in using CSS in Corvid templates, but this is not a CSS tutorial. For more information on CSS, see one of the many books on it.

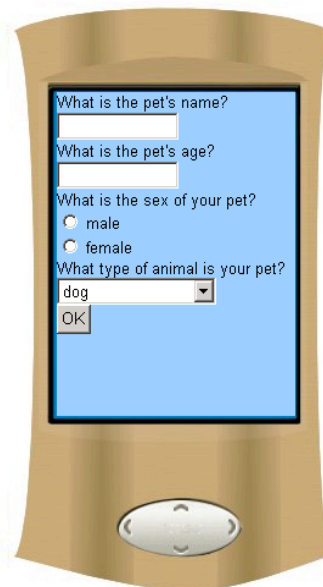
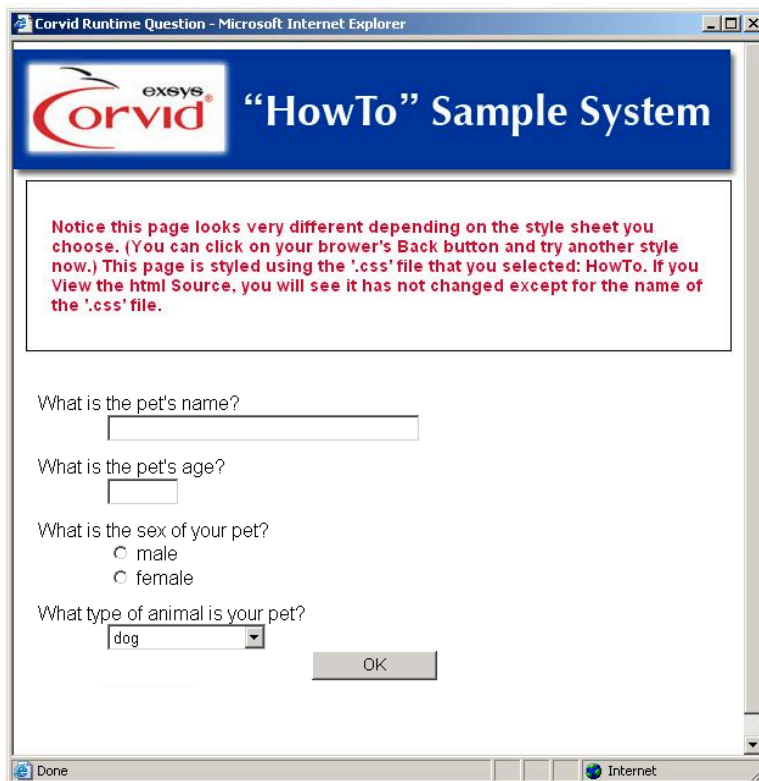
To see the effect of using CSS, run the demo system at <http://www.exsys.com/support/howto>.

After the title screen, the first question allows you to select the CSS style sheet used for the other questions in the system.

Style sheet to use:

- HowTo - This is the standard look and feel of the Corvid How To Demos
- PDA - This simulates a PDA and uses a layout appropriate to the small screen
- HighContrast - This is a style that can assist the visually impaired
- Trace - This displays the Corvid Trace (Trace is hidden in all other styles.)
- Card - This is an example of another Look-and-Feel

Each of the CSS options will make the system look quite different, but the only change to the system is the CSS style sheet used. The “How To” style looks like the other Corvid “How To” samples, while the “PDA” style emulates a PDA and uses a layout appropriate for the smaller screen.



The “High Contrast” style makes the screen easier to read for people with visual impairment. “Trace” displays the trace of the run. “Card” is a different design. There are limitless ways the screen can be designed using CSS.

In this demo, the CSS style used is selected by the system user. In a fielded system, there would usually be only a single style for the browser, with the design selected by the system developer. Other styles needed for special situations such as running on a PDA or printing, would be selected automatically.

Moving the Demo System Files to a Server

The demo runs using the Corvid Servlet Runtime, so you will need a servlet server. As with many of the other HowTo's, this HowTo will give the instructions for Tomcat.

- Download the CSS demo files from the link above, and unzip the file. It has two folders.
- Put the folder in 'KB' into Tomcat's 'webapps' folder in a subfolder named 'HowTo'.
- Put the folder in 'base' into Tomcat's 'webapps/ROOT/HowTo' folder.
- Open the .CVD file in Corvid editor.
- In the Properties window under the Servlet tab, change the host and port to match your server.

If your server is 'localhost' on port 8080, it is already configured and you can skip ahead to running the demo. Otherwise, change 'localhost:8080' to your server's host name and port in the CORVID_SERVLET and CORVID_LINK_BASE and also the Specific URL under the Test Run tab.

Start Tomcat if it is not already running and click on the Run button in Corvid (the blue triangle). Remember, localhost is just a synonym for the same IP as the local computer. This means the computer running the browser must also be the server. If the system is installed correctly, it will run the same as the sample system link above.

How it Works

CSS can be a complex subject. For its use in Corvid templates just remember that essentially anything you can do with CSS can be done in the template. The Corvid template does not require CSS, but it also does not prevent any use of CSS.

Basically a CSS file defines named “classes”. The various classes can describe fonts, sizes, positioning, color, layout, etc. These class names can then be applied to sections of the template. This makes the template content very simple, since the CSS style sheet provides all the detailed “look-and-feel” information.

For example in the simple question template that could be used to ask a Numeric variable:

```
<html>
<head>
  <link rel="stylesheet" href="CorvidKB.css">
</head>
<body>
  <div class=questions>
  <form method="post" action="CORVID_SERVLET">
    <div class=question>
      <span class="prompt"> VARIABLE_PROMPT </span>
      <span class="editbox">
        <input type="text" name="[VARIABLE_NAME]">
      </span>
    </div>
    <input type="submit" value="OK">
```

```
</form>
</div>

<p class=trace>
CORVID_TRACE
</p>
</body>
</html>
```

The VARIABLE_PROMPT is surrounded by ` ... ` tags. The desired appearance (markup) can be specified using CSS. If you want the prompt text in red and bold, put this "selector" in the '.css' file.

```
.prompt {
    color: red;
    font-weight: bold;
}
```

The period (.) in front of "prompt" means this selector will be applied to tags with a "class" of "prompt". It tells the browser to display the text between those tags in red and bold. The key to CSS is to use the "class" attribute to identify the tags that mark up text of a specific meaning. Notice the `` tag includes a class name. This is just a name by which you can identify the `` tags that markup the prompt text.

The `<div>` tag is used to mark large sections of the web page (divisions), in this case, the question section. Different sections of the web page will have `<div>` tags with different class names. For example, the template above has these divisions:

- "questions" This marks the entire section that includes the question or, if more than one question, **all** the questions.
- "question" This marks an individual question. It includes the prompt text and the controls (radio buttons, edit box, drop down list) used to ask the question.
- "trace" This is the section that contains the Trace text. Run with Trace on only for testing. Use CSS to hide it from the beta user but if a beta tester encounters a problem, you can have the beta tester pull down View and select Source and send you the Trace.

The `<div class=question>` tag does nothing by default until you specify its markup in the '.css' file. (Actually, it displays that section on a new line.)

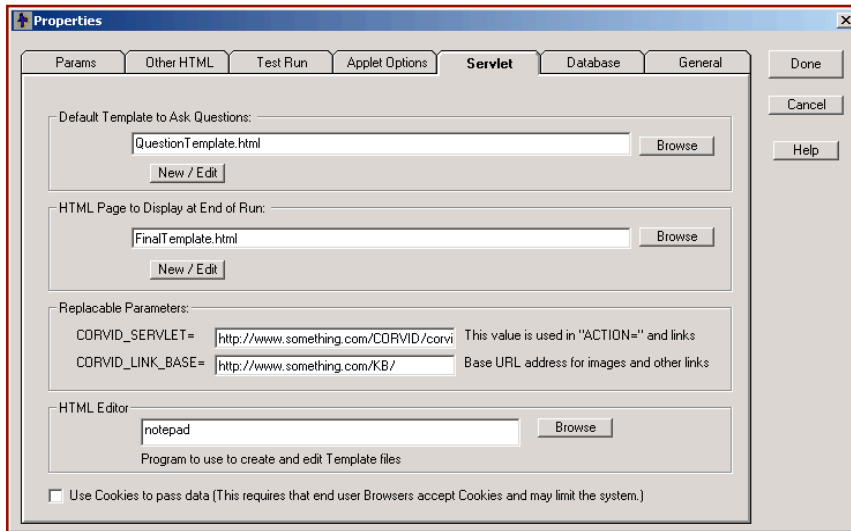
The sample 'CorvidKB.css' has comments to help you determine which CSS selector specifies the look-and-feel of the section or tag you want to customize. To change the background image, replace 'background.jpg' with your desired background file. You can comment out lines in the '.css' file by putting `/* ... */` around the line(s) as long as they do not also contain the comment markers.

When the user prints the web page, the browser will determine if the web page specified a '.css' file to be used for printing. If so, it gets the '.css' file and uses it to format the printed web page. People browsing using their PDA or cell phone will automatically use the file 'PDA.css'. This automatically occurs when your template has:

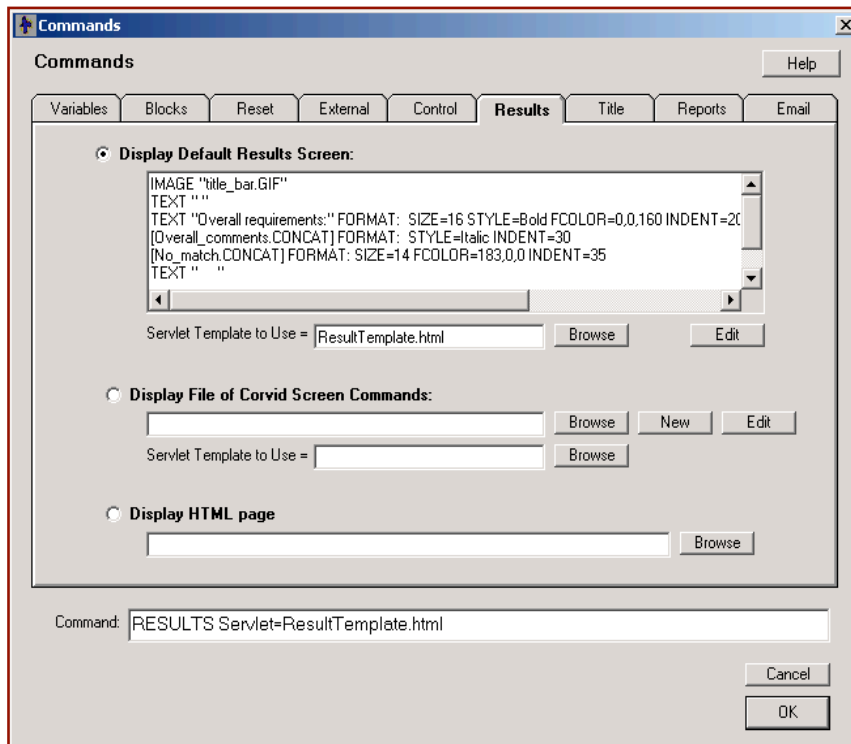
```
<link rel="stylesheet" type="text/css" href="print.css" media="print">
<link rel="stylesheet" type="text/css" href="PDA.css" media="handheld">
```

Using the Templates in Your Expert System

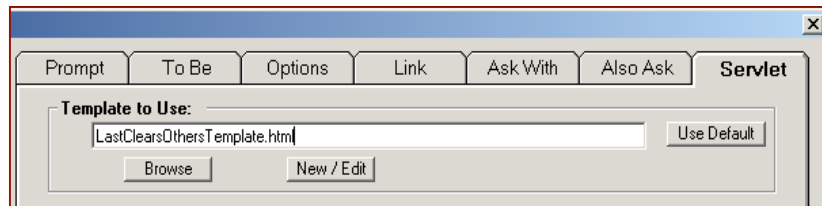
The CSS demo includes templates and a style sheet that you can use in your KB. They will get you started but you will want to customize them. Unzip 'GoesWithCVR.zip' and put the files in the same folder as the '.cvR' file. Unzip 'GoesInCorvidLinkBase.zip' and put the files in the folder pointed to by CORVID_LINK_BASE (which is specified under the 'Servlet' tab in the 'Properties' window). Under the 'Servlet' tab in the 'Properties' window, put 'QuestionTemplate.html' for the Default Question template. Put 'FinalTemplate.html' for the 'Final template'.



In the Command Block, edit your 'RESULTS' command and put 'ResultsTemplate.html' for the 'Servlet template'. (Notice there are two 'Servlet template' edit boxes. Make sure you use the correct one.)



The expert system can now be run using the Corvid Servlet Runtime. All questions will be asked using the 'QuestionTemplate.html' file. Other templates can be used to customize individual questions as needed by selecting the variable in the 'Variables' window and browsing to the desired template under the 'Servlet' tab.



CSS For Formatting

CSS classes can also be used to format content on a page. CSS classes can have specific sizes, borders, colors and positions. Every class or tag can have a specified background color, image, audio (to help the blind who use screen readers), bullets (for lists), fonts, and cursor image. This can be used to create a complex design with CSS that is used to format simple content generated in a Corvid report page.

CSS can be used to simplify building WINK (What I Need to Know) pages. (For more on WINK systems see the "WINK What I Need to Know Website Systems How To" or the Exsys Covid manual.

CSS can also be used to create pages that appear one way on the screen, but use a more "Printer Friendly" layout for printing.

Corvid templates using CSS can be edited in an HTML editor that supports CSS, such as Adobe® Dreamweaver®. However, care must be taken to make sure the Corvid commands, which are HTML comments, remain in the correct position relative to other code. Since HTML editors view these as comments, they may put other tags around them incorrectly.