QuestFields CSS

QuestObjects Tech Note 012 rev 2

⇒ This tech note describes how QuestFields use cascading style sheets to implement the "look" of the QuestField user interface. QuestFields come with various selectable "skins" that can change the look on a per-QuestField basis. Customers can create their own "custom skins" that override parts of the user interface.

1. About QuestField Client Technologies

QuestFields client software exists in various technical implementations. Some QuestFields are entirely implemented in DHTML (which inherently supports CSS styling). Some QuestFields are entirely implemented in Adobe Flash (older versions of which do not support CSS styling). "Hybrid" QuestFields use DHTML for the border of the QuestFields and Flash for the innards.

This tech note applies only to *styled QuestFields*, i.e. QuestFields that are part of a "normal" HTML web page for use in browsers that support CSS styling. See the `useStyles` specification attribute in the QuestFields Client Administration Guide.

You can configure QuestFields so that they automatically select the appropriate technology for the user's browser. In that case, CSS styling will be applied in modern browsers. And in older or non-compatible browsers, the QuestFields will revert to a "generic" look implemented in Flash. On devices that support neither full DHTML nor Flash, the QuestFields Server can revert to a "static" interface implemented as JSP pages on the QuestFields Server. See the `technology` specification attribute in the QuestFields Client Administration Guide.

⇒ QuestFields version 1.x support Flash with CSS skinning;
QuestFields version 1.3 provides beta support for full AJAX with DHTML;
QuestFields version 2.x are available in either Flash or AJAX/DHTML (licensed separately).

2. Embedded versus Full-Page QuestField Modes

"Embedded" QuestField Mode

When included into your own web pages loaded in desktop web browsers (such as Internet Explorer, Firefox, or Safari on Windows, Mac, or Linux), QuestFields are *embedded* so they become an integral part of the layout of the web page. Embedded QuestFields typically consist of an *input field* with a *pop-up area* that displays associated information, such as a list of results. The *pop-up area* is typically hidden while not in use, and will appear automatically as results are received from the server. AutoSuggest QuestFields have a *combo button* that is used to "toggle" the visibility of the *pop-up area*.

With QuestFields working in embedded mode, the QuestField CSS is responsible for the "look" of the QuestFields themselves only. Your own style sheets determine the look of other elements on the web page (and basics such as the default font).

The QuestField styling for embedded QuestFields is not platform-specific (embedded QuestFields typically look the same on Windows, Mac, and Linux), although you can influence the appearance of individual QuestField elements depending on the operating system by using style sheet rules in a custom skin (see section 5 below).

"Full-Page" QuestField Mode

On some platforms such as the iPhone, QuestFields can take over the full web page. Instead of a pop-up area, the QuestFields results list (or any other popup pane) takes over the entire browser screen. In this case, the entire page is styled by the QuestField CSS, and the "look" of the page can be quite different depending on the platform. For example, on the iPhone, there are no scroll bars, and the page header has a toolbar that uses typical "Cocoa Touch" gradients and buttons. On Windows Mobile devices, full-page QuestFields adopt the typical look of the Windows Mobile operating system.

QuestFields running in full-page mode will load platform-specific style sheets that provide this "native look".

⇒ Embedded QuestFields require a single license per technology (DHTML/AJAX, Flash) for all supported browsers and platforms; Full-page QuestFields are licensed separately, but share CSS styling so they do not require any additional work.

Cross-Domain Access

QuestFields also support full-page mode in "regular" web browsers, where they can function as a pop-up window above your website, or as a frame inside your web page. This allows AJAX technology-based QuestFields to work across domains without requiring a proxy. Flash technology-based QuestFields support cross-domain access even in embedded mode. See the QuestFields Client Administration Guide for more information.

3. QuestField Skins

QuestField Skins

Styled QuestFields use a "skinning" mechanism that allows CSS style sheets to set the shape and color of QuestField elements by specifying a single `skinName` specification attribute. By default, QuestFields use a generic skin identified by skin name "`qo-generic`". MasterObjects supplies various additional skins. You can also create your own "custom skins".

Using class names that are automatically applied to each QuestField, skin style rules can adjust the look of individual QuestFields depending on their height, the color of the background, and the operating system. See section 7 for details.

4. Skin-independent Colors

Some elements of a QuestField use colors that are independent of the chosen skin and color theme. This applies to elements whose colors are determined by the following QuestField specification attributes:

- The color of input text (and the wiggly underline under it) is determined by specification attributes `newColor`, `okColor`, `completeColor`, `placeholderColor`, and `errorColor`.
- The background color of the optional QuestField footer is determined by specification attribute `footerColor`.

  Note: In Internet Explorer 6 or lower, the background of the footer is always set to light gray (`#F2F2F2`).
- The status area on the right of the input text uses symbols and animations in fixed colors. Animations are drawn against a white circle that looks good on any light background color.

  Note: In Internet Explorer version 6 or lower, the background of QuestField input elements is always forced to white, regardless of the skin.

5. Platform-specific Styling

QuestFields embedded into your website

QuestFields running in "embedded mode" were designed to run in a wide range of standard web browsers on any platform or operating system. For a non-exhaustive listing of supported browsers, see `http://www.questobjects.com/browsers.html`. In the latest browsers, QuestFields can use AJAX technology, requiring no plug-ins whatsoever. Using Flash technology, AutoSuggest QuestFields can even run in old browsers (IE 5.5) on outdated operating systems such as Windows 98.

QuestFields running in embedded mode typically use a common visual style on any browser platform: You select a single "skin" (or create a custom skin) that blends in with the look of your website. QuestFields running in embedded mode use the same CSS style sheets on any platform. Using operating system-specific class selectors, you can still optimize the look of individual embedded QuestFields for a specific operating system (e.g., Windows, Mac, Linux).

Full-Page QuestFields

Although browsers on the latest mobile platforms provide improved support for standard DHTML and AJAX, the layout of typical AJAX websites does not lend itself well for devices with a small screen. Typically, the screen of mobile devices is too small to show the "dynamic content" displayed by AJAX applications. This also applies to dynamic content displayed by QuestFields, such as the results list. For this reason, QuestFields support a full-page mode (licensed separately) that provides a native "look" for specific platforms, such as Apple's iPhone OS X (implementing the "Cocoa Touch" look and feel), Microsoft's Windows Mobile, Blackberry OS, Palm Pre, Symbian OS, or Google's Android.

QuestFields running in full-page mode use an additional style sheet for the current platform. On regular browsers on Windows, Mac OS, and Linux, the full-page QuestField uses a "default" platform style sheet. For specific portable devices such as the iPhone, a specific platform style sheet is loaded.

6. QuestFields Style Sheets

Directory Location

QuestFields use multiple style sheets that are embedded automatically into the web page(s) in which they are used. Therefore, it is not necessary to include any of the QuestField style sheets described below into the header of your web pages.

QuestField style sheets are provided in the styles/ subdirectory of the /questobjects/ directory. The /questobjects directory is always present on (and served by) the QuestFields Server. By default, however, the client files in this directory are expected to be at the root of the local web server, thus /questobjects/styles. By copying the QuestFields client files (and styles) to the local web server, page load time is typically enhanced. Also, the page will build normally, even when the QuestFields Server is unavailable temporarily. See the QuestFields Client Administration Guide for more information.

It is possible to override the location of the QuestField client files by setting QO_config.rootPath. For example, if you do not have a copy of the styles directory hosted on your local web server, you can add the following to the header of your web page to load the client files (and styles) from the QuestFields Server URL:

```
<script type="text/javascript">
// <![CDATA[
        var QO_config =
                {
                        rootPath: "http://your.questobjects.server/questobjects"
                };
// ]]>
</script>
```

Styles Directory Structure

This is an overview of the essential QuestFields style directories:

/styles/shared/

This directory contains styles that are common to all QuestFields. It contains style sheet qo-common.css, which is loaded before any of the other style sheets.

/styles/questlet/

This directory contains style sheets and images that determine the look of QuestFields using DHTML technology, as well as the "skins" that are used by styled QuestFields.

/styles/questlet/dhtml/

This directory contains images that are used by QuestFields using DHTML/AJAX technology.

/styles/questlet/skins/

This directory contains subdirectories for standard skins provided by MasterObjects. See section 7 below.

/styles/platforms/*{platform}*

These directories (e.g., /styles/iphone/) contain the platform.css style sheet and images loaded for full-page QuestFields used on a specific platform.

⇒ Note: either the platform.css style sheet for the specific platform *or* platform.css for the default platform is loaded, so each platform.css contains a full set of style rules.

Platform.css is loaded for any DHTML QuestFields, regardless of whether they are running in embedded or full-page mode. Custom QuestField layouts (such as custom list layouts) may therefore use classes defined in the platform.css file, and will work both in embedded and full-page mode.

7. QuestField Skins

A QuestField skin is a named subdirectory in the `/styles/questlet/skins/` directory, containing style sheets and images that define the visual aspects of the QuestField. QuestFields always load skin `qo-generic`, followed by the custom skin (if one is specified using the `skinName` specification attribute).

Each skin contains the following style sheets:

`questlet.css`

> Contains style rules for any styled QuestFields, including those that use Flash technology for the "internals". This includes styles for the QuestField border, the optional combo button, the optional submit button, and the border around the QuestField pop-up area.

`questfield.css`

> Contains the style rules defined in `questlet.css`, plus those used by elements *inside* of DHTML QuestFields. These include the styles used by the result list, the "information pane", and the "about box".

`fullpage-default.css`

> Contains the rollup of the style rules from `questlet.css` and `questfield.css`, plus any full-page QuestField style rules for the default platform that override `platform.css`.

`fullpage-{platform}.css`

> `fullpage-default.css` is only loaded for the default platform (i.e., regular web browsers on Windows, Mac, Linux). For separately licensed platforms, a separate style sheet `fullpage-~{platform}.css` is used. For example, on the iPhone, `fullpage-iphone.css` is loaded.

The above style sheets apply to any size QuestField using any color theme. It is possible to create style rules to override the look of QuestField elements for specific QuestField heights, platforms, color themes, and background colors.

QuestField Height

To reflect the height of the QuestField input element, the QuestField applies a `qoH{height}` class to the QuestFields outer element.

So, to define a style rule that applies only to QuestFields of height 20 pixels, use selector `.qoH20`.

Platform

To reflect the current platform (and, for the default platform, the operating system), the QuestField applies a `qo{Platform}` class to the QuestFields outer element (in the case of full-page QuestFields, to the `<body>` element).

For the default platform, the class name reflects the operating system (`qoWin`, `qoMac`, or `qoLin`). For separately licensed platforms, the platform reflects the capitalized platform name (e.g., `qoIphone`).

So, to define a style rule that applies only to QuestFields running on the Mac, use selector `.qoMac`.

Color Theme

The `<body>` element of QuestField-enhanced web pages automatically receives a class whose name reflects the color scheme, starting with characters `mo`. Standard QuestFields set class name `moBlue` on the body tag.

So, to implement style rules that apply to the orange color scheme, use selector `.moOrange`.

Background Color

Some QuestField elements need to adjust their look depending on the color of the background (the color of the elements that are displayed behind it). To this end, QuestFields allow setting the `bgColor` and `popUpBgColor` specification attributes, as described in the QuestFields Client Administration Guide.

You can apply specific style rules to QuestField elements for approximate QuestField background colors. To this end, QuestFields set a `qoC`*`{RGB color}`* class name on the outer element nested into the background. The approximate color value consists of three hexadecimal character values (one each to represent Red, Green, and Blue). The class is set for any background color in the range of `#RxGxBx`.

Skin Style Sheet Suffix

CSS style sheets of skins that support many combinations of heights, platforms, color themes, and background colors tend to become quite large. For these skins, it is possible to create duplicates of the `questlet.css`, `questfield.css`, and `fullpage-default.css` files that include only the selection rules for a specific combination of height, platform, color theme, and/or background color. The specific versions of the skin style sheets should then be named using a specific suffix. After adding the suffix to the `skinName` specification attribute of the QuestField, the suffix is appended to the standard file names.

The suffix is added to the `skinName` QuestField specification attribute as follows:

```
        skinName = "qo-rounded?suffix={suffix}"
```

For example, after setting `skinName` to `qo-rounded?suffix=789`, the QuestField will load style sheet `questlet-789.css`, `questfield-789.css`, or `fullpage-`*`{platform}`*`-789.css`.

Note: the generic skin (`qo-generic`) does not use the suffix, so it always loads `questlet.css`, `questfield.css`, or `fullpage-`*`{platform}`*`.css`.

THIS DOCUMENT REPRESENTS THE VIEWS OF MASTEROBJECTS ON THE TOPICS DISCUSSED IN THIS TECH NOTE AS OF THE DATE OF PUBLICATION. THE INFORMATION IN THIS DOCUMENT SHOULD NOT BE INTERPRETED AS A COMMITMENT ON THE PART OF MASTEROBJECTS. THIS TECH NOTE IS PROVIDED FOR INFORMATION PURPOSES ONLY.