

# Wiki Help API

The Wiki Help API allows developers to create context driven help pages for each screen. The API exports screen and field information, which can be used to determine which help page to load. Wiki sites are often used for help, because they're designed for easy contribution. A Wiki is also useful with Profound UI driven screens, because you can load different help pages by passing different text to the URL to load different pages. However, not just Wikis, but any sites that accept URL parameter strings can be setup to host the help documentation.

The API allows you to create two different types of help: screen-level or field-level. The API works for Rich Display Files but not for Genie.

The Wiki Help API became available with Profound UI Version 5, Fix Pack 3.0.

## Screen-Level Help

Screen-Level help is intended to be used when your help website uses a different help page for each screen.

To setup help pages that describe an entire screen, not each specific widget, you might use the API function, `pui.wikihelp.getScreenInfo()`. `getScreenInfo()` returns an object with the following members:

- `formats` - An array of strings naming the screen's current record formats.
- `dspf` - The name of the display file.

If your help pages

## Field-Level Help

Field-level help is designed to be used when your help site uses a different page for each field on a screen. Also, some pages might show multiple fields on a help page, and passing some text to the URL causes the page to scroll to the appropriate field. For those pages, field-level help is useful.

The field-level help creates overlays for each field that the user clicks to load a help page. For this feature, you'll need to call API functions to enable and disable the help, write a click event handler, and add custom CSS for the overlay. You as the developer must provide the interface to toggle the help overlays on and off. Some example interfaces are given below.

You can optionally add some text or an image to the overlays.

## API Functions

To enable the field help overlays, call `pui.wikihelp.overlayOn()`. A "div" element is placed over each field on the page for the current record formats.

Call `pui.wikihelp.overlayOff()` to hide the overlays.

Some sample user interface code is included below. Place your custom JavaScript in the custom js folder in your Profound installation's IFS directory. For example, create a file named, `/profoundui/userdata/custom/js/wikiHelp.js`.

The first example places a toggle button in the top-right corner of the page. Clicking the button toggles the help overlay on or off.

## Toggle Button

```
pui.onload = function(parms) {
  var helpButton = document.getElementById("wikihelp-toggle");

  // Don't show help toggle if screen isn't sign-on, end-of-job, or timeout.
  if( helpButton && (parms.name === "SIGNONSCRN" || parms.name === "EOJSCRN" || parms.name ===
"TIMOUTSCRN") ) {
    helpButton.parentNode.removeChild(helpButton);
    helpButton = null;
  }
  else if(! helpButton ) {
    // The help toggle wasn't already added; create it.
    var helpButton = document.createElement("input");
    helpButton.id = "wikihelp-toggle";
    helpButton.value = "Help On";
    helpButton.type = "button";
    helpButton.helpOverlayOn = false;
    helpButton.style.position = "absolute";
    helpButton.style.zIndex = "1010";
    helpButton.style.top = "2px";
    helpButton.style.right = "2px";
    helpButton.style.width = "75px";
    document.getElementsByTagName( "body" )[0].appendChild(helpButton);

    helpButton.onclick = function() {
      if( this.helpOverlayOn ) {
        pui.wikihelp.overlayOff();
        this.value = "Help On";
      }
      else {
        pui.wikihelp.overlayOn();
        this.value = "Help Off";
      }
      this.helpOverlayOn = ! this.helpOverlayOn;
    };
  }
};
```

Note: pui.onload is called after the record format is rendered on the page.

The next example toggles the help overlay when the F1 key is pressed.

## Keyboard Toggle

```
var helpOverlayOn = false;
function toggleHelp() {
    if( helpOverlayOn )
        pui.wikihelp.overlayOff();
    else
        pui.wikihelp.overlayOn();
    helpOverlayOn = !helpOverlayOn;
}
// Internet Explorer
if("onhelp" in window) {
    window.onhelp = function(){
        toggleHelp();
        return false;
    }
}
// Other browsers.
else {
    addEvent(window, "keyup", function(){
        if( e.keyCode === 112 ) { // F1.
            toggleHelp();
        }
    });
}
```

## Skip Item

By default, help overlays are drawn for every visible widget. However, not every widget needs an explanation—some may exist for style. Thus, overlaying those widgets can clutter the help system and confuse the user.

To solve the cluttered appearance, you can limit which widgets get an overlay by defining a **pui.wikihelp.skipItem** function. This function gets certain widget information passed by parameter, and it should return true when the widget should not get a help overlay.

The object passed to *skipItem* includes the following properties:

- *dspf* - the name of the display file that is currently rendered.
- *recfmt* - the Record Format name currently active.
- *fieldId* - the ID attribute of the widget.
- *field type* - the widget field type; e.g. grid, textbox, menu, hyperlink, etc.
- *fieldName* - If the widget's *value* property is bound, then *fieldName* is the name of the bound RPG field. If the value property is not bound, then *fieldName* is an empty string.

*skipItem* is called when the overlays are being generated: on the first call to `pui.wikihelp.overlayOn`. *skipItem* is called on each widget except layouts, panels, and hidden elements.

## Example skipItem

```
pui.wikihelp.skipItem = function(parms){
    if (parms["field type"] == "grid") return true;

    //Don't overlay any lines, which have IDs starting with "Line".
    if (parms["fieldId"].indexOf("Line") == 0) return true;

    if (parms.fieldName == "") return true; //Only allow overlay on bound fields.
    return false;
};
```

(skipItem is available with Profound UI Version 5, Fix Pack 7.0 and later.)

## Click Event Handler

To handle click events, create a function named, **pui.wikihelp.onclick**, and add it to the custom js folder in your Profound installation's IFS directory. For example, create a file named, **/profoundui/userdata/custom/js/wikiHelp.js**.

The function has one parameter, an object with the following members:

- dspf - The display file name.
- recfmt - The record format of the field that was clicked.
- fieldId - The ID of the clicked field.
- fieldName - The name of a bound variable for the clicked field, or an empty string if the field is unbound.
- event - The JavaScript mouse event from the clicked page element.

An example handler function is provided below.

### pui.wikihelp.onclick

```
pui.wikihelp.onclick = function( params ) {  
  // Construct the URL for a wiki where each display file name has its own page.  
  // The sections of the page are named after the record format, "_", and  
  // the field name or fieldId.  
  var url = encodeURI("http://localhost/dokuwiki/doku.php?id=" + params.dspf + "#" +  
    + params.recfmt + "_" +  
    + ( params.fieldName.length > 0 ? params.fieldName : params.fieldId )  
  );  
  
  // The wiki loads into another tab/window.  
  window.open(url, "WikiHelp");  
};
```

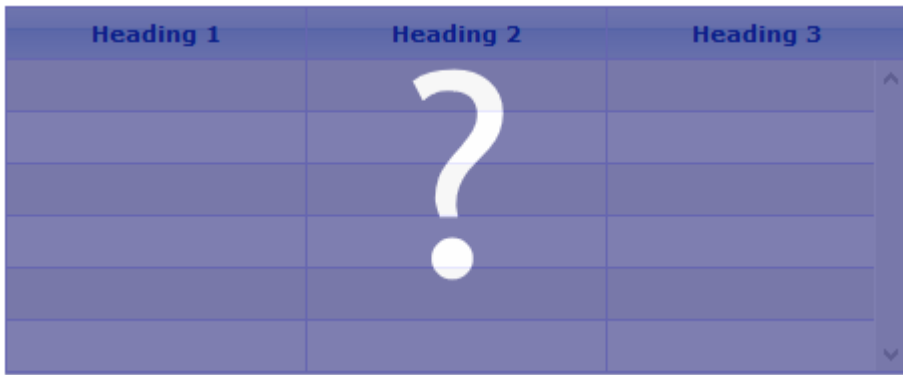
## Overlay Style

By default the overlays have no style, so they are transparent. To style the overlays, add a stylesheet to the custom css folder in your Profound installation's IFS directory. For example, create **/profoundui/userdata/custom/css/wikiHelp.css**. Each overlay has the html class name, "pui-wikih-overlay", so create a CSS rule for that class. Some example CSS is included below:

### Blue Transparent Overlay Example

```
.pui-wikih-overlay {  
  cursor: help;  
  padding: 0;  
  text-align: center;  
  color: white;  
  opacity: 0.50;  
  filter: alpha(opacity=50); /* for IE<9.*/  
  background-color: #000064;  
}  
.pui-wikih-overlay:hover {  
  background-color: #6496FF;  
  opacity: 0.90;  
  filter: alpha(opacity=90);  
}
```

Blue Transparent overlay on a subfile grid:



### Dashed Border Example

```
.pui-wikih-overlay {
  cursor: help;
  padding: 0;
  text-align: center;
  color: white;
  border: dashed blue 2px;
  box-sizing: border-box; /* Note:unsupported for IE<8.*/
  opacity: 0;
  filter: alpha(opacity=0);

  /* You must have a background-color for IE; without background IE doesn't capture mouse events.
  */
  background-color: #000000;
}
```

Note: for Internet Explorer (IE) browsers, the overlay must have a background-color style. Otherwise, IE cannot capture mouse clicks on the overlay, and the feature doesn't work.

## Adding Text to the Overlay

### pui.wikihelp.overlayText

By default the overlay has no text. You can specify a text string that appears in each overlay box. In your custom js file, set **pui.wikihelp.overlayText** to some string. For example:

```
/profoundui/userdata/custom/js/wikiHelp.js
```

```
pui.wikihelp.overlayText = "?";
```

### pui.wikihelp.fontScale

By default, the string's font size changes proportionally to the overlay box size. You can disable automatic font re-sizing by setting **pui.wikihelp.fontScale** to false.

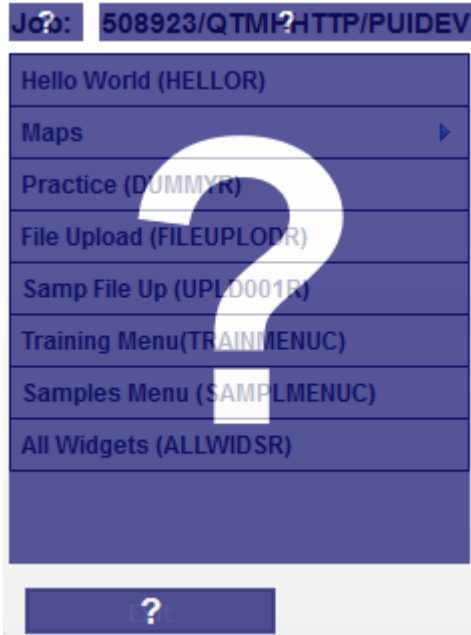
Alternately, you may want to adjust the scaling factor if your text is bleeding outside the overlay. This may happen for non-latin characters:

`/profoundui/userdata/custom/js/wikiHelp.js`

```
pui.wikihelp.fontScale = 0.85;  
pui.wikihelp.overlayText = "";  
  
pui.wikihelp.fontScale = 1.4;  
pui.wikihelp.overlayText = "";
```

The default scaling factor (1.56) works well for Latin alphabet strings 1 to 4 characters long in Helvetica font.

Font size is proportional to widget size:



Default font scale is too large for wide characters:



Font scale of 1.0 fixes this particular string:



## Adding an Image to the Overlay

`pui.wikihelp.image`

To add an image to the help overlay, set `pui.wikihelp.image` to a path string. For example,

```
/profoundui/userdata/custom/js/wikiHelp.js
```

```
pui.wikihelp.image = "/profoundui/userdata/images/comment_help.png" ;
```



`pui.wikihelp.imageSquare`

By default the image resizes into square dimensions. The smallest side of the overlay box determines the image size. If you want the image to use the same rectangular dimensions as the overlay box, then set `pui.wikihelp.imageSquare` to false.

Job: 508923/QTMFH/HTTP/PUIDEV



Hello World (HELLOR)
Maps
Practice (DUMMYR)
File Upload (FILEUPLOADR)
Samp File Up (UPLD001R)
Training Menu (TRAINMENC)
Samples Menu (SAMPLMENC)
All Widgets (ALLWIDSR)

