

OnlineOpinion® User Manual - Code Set Version 5.5.x

Table of Contents for OnlineOpinion® Code Set Version 5.5.x

- What is OnlineOpinion®?
 - How does it work?
 - Can I make customizations?
 - Client Support and the Primary Developer Contact
- Code Installation
 - Code Set Files
 - Installation Instructions
 - Standard Files and Associated Functions
 - OnlineOpinion® Engine File
 - Configuration Files
 - Disabling or Providing Alternative OnlineOpinion Object Configurations for Certain Browsers
 - CSS File Configuration
- Frequently Asked Questions (FAQs)
 - Inline Feedback Implementation
 - How do I place an Inline Feedback link?
 - How do I create separate "site" and "page" level feedback links?
 - Can I have different cards appear from the same page with one Inline feedback implementation?
 - Floating [+] Feedback Icon Implementation
 - What are the Basic Parts of the Floating [+] Feedback Icon?
 - How do I keep the Floating [+] Feedback Icon from overlapping my content?
 - Can the placement of the Floating [+] Feedback Icon be changed?
 - How do I get the Floating [+] Feedback Icon to appear over Flash elements on the page?
 - How do I get the Floating [+] Feedback Icon to disappear when clicked but always reappear when the page is refreshed?
 - How do I get the Floating [+] Feedback Icon to disappear after it has been clicked on and not reappear for a specific period of time on that page?
 - How can I make the Floating [+] Feedback Icon bigger?
 - How can I change the color of the Floating [+] Feedback Icon?
 - How do I change the text of the floating Feedback text?
 - How do I change the background color of the box around the floating Feedback icon?
 - How do I change the color and/or size of the floating Feedback text?
 - How do I disable the rollover state of the floating Feedback Icon?
 - Tab [+] Feedback Icon Implementation
 - How do I change the look of the Tab?
 - oo_style.css
 - Bar [+] Feedback Icon Implementation
 - How do I change the background color?
 - How do I change the text?
 - How do I change the height or width of the bar?
 - Event-Driven Feedback Implementation
 - What is the difference between user invoked vs. Event-Driven Feedback Implementations
 - How do I make sure that a user won't see the event driven card more than once?
 - How do I change how long it takes for a user to see a survey card a second time?
 - How do I adjust how many users will see the card?
 - How do I use On-Exit Event-Driven Feedback
 - How do I prevent an On-Exit Event from rendering when a user clicks on some links on my web pages?
 - Does "disableLinks" work for page anchors too?
 - How do I stop the card from rendering when a user hits the "Back" button?
 - How do I implement another event driven card somewhere else on my site?
 - How do I place an on click event feedback link?
 - How do I make sure that a user who has seen one of my first event driven surveys doesn't see one of the later surveys?
 - How do I delay an on entry card from rendering until after the page completely loads?
 - How can I get the survey to bypass pop-up blockers?
 - Invitation Feedback Implementation
 - What is the OnlineOpinion® Invitation Feedback?
 - How is Invitation Feedback different from Standard and Event-Driven Feedback Code?
 - When should Invitation Feedback be used?
 - Can you explain some example scenarios for Invitation Feedback?
 - How do I use Invitation Feedback
 - Inhibiting the prompt
 - Inhibiting the Card
 - How do you configure Invitation Feedback?

- Cookie Configuration
- On-Page Card Configuration
 - What is the difference between a Standard Pop up vs. On-Page Card
 - How do I change from a pop up card to an On-Page Card?
 - How do I get the On-Page Card to appear over the Flash elements on my page?
 - How do I change the background overlay color for the On-Page Card?
 - How do I close an On-Page Card?
 - How do I show a card is loading?
- Custom Variable Configuration
 - What are Custom Variables?
 - How do I add Custom Variables?
 - Why won't my Custom Variables work?
 - What is a "legacy custom variable" and how is it used?
- Referrer Rewrite Configuration
 - What are domain/URL referrer rewrites? Why are they used?
- New Window Resize Configuration
 - Why would I want to configure the size of the window for the Dynamic Comment Card?
- TeaLeaf Cookie Configuration
 - How do I integrate OnlineOpinion® Code with Tealeaf?
- Tunnel Configuration
 - What is a tunnel configuration
 - Can tunnel paths be made in all OCode types? How about Invitation Feedback?
 - What controls tunnels configurations?
 - Why won't my tunnel path work?
- Image and Verbiage Customization
 - Can I change the word "Feedback?"
 - Can I use my own image instead of the one OpinionLab® gave me?
- Mobile Solutions
 - Will OnlineOpinion work on my mobile device?
 - What will I see if my mobile device does not support Javascript?
 - How do I prevent OnlineOpinion from showing on all mobile devices?
 - How do I prevent OnlineOpinion from showing on all non-iOS (iPhone/iPad) mobile devices?
- Known Issues
 - Comment Card Support
 - Mobile Browsers
 - Internet Explorer Version 6
 - Internet Explorer Version 7
 - Internet Explorer Version 9
 - FireFox Version 3.5
 - On-Exit Event-Driven Feedback
 - Invitation Feedback
 - Tunnel Methodology
 - Custom Variables
 - On-Page Card
 - ClickTale Integration

What is OnlineOpinion®?

OnlineOpinion® is OpinionLab®'s method of collecting secure, customer feedback data for its clients. It is comprised of a code set that is typically installed on a web site to collect customer data via OpinionLab®'s online comment card. The standard code allows for various options described below.

An OnlineOpinion® code set is a set of files, be they JavaScript (JS), Cascading Style Sheets (CSS), Hypertext Markup Language (HTML), Portable Network Graphics (PNG), Joint Photographics Experts Group standard (JPG), Graphics Interchange Format (GIF), or text (TXT) files.

Within any OnlineOpinion® Version 5+ Code Set, an instance is a single OnlineOpinion® object, comprised of:

- The External JS Engine File (usually named as some variant of OO-engine.JS). This is the blueprint for instances of OnlineOpinion® objects (OOo.OCODE and OOo.INVITE).
- The "new" operator that specifies the configuration of the OnlineOpinion® Object. This is usually housed in a External JS Configuration File (Usually named some variant of oo_conf.js) or even instantiated within the page itself.

Typically, OpinionLab delivers a single instance for the feedback method purchased. The client can create multiple instances based off of this using the included User Manual or contract with OpinionLab® to build more. Multiple instances can be housed in one or several External JS Configuration Files, or even instantiated within the page itself, if the client so desires. Depending on the OnlineOpinion® configuration, this along with the proper include file may be all that is required, or additional supporting code and images may be included in the code set. For explanations on the file formats listed, please consult The World Wide Web Consortium at <http://www.w3.org>.

How does it work?

OnlineOpinion® code consists of a small JavaScript include statement that refers to a code set consisting of various JavaScript (JS), Cascading Style Sheet (CSS), Hypertext Markup Language (HTML) and/or image files, depending on the configuration selected. These files are stored locally by the client usually on the client's root directory and only communicate with the OpinionLab® servers when a Comment Card is invoked. The Comment Card may be invoked either by a user clicking on a specified feedback link (Inline Feedback link placed in the header/footer of a web page or the Floating [+] Feedback Icon, Tab [+] Feedback Icon, or Bar [+] Feedback Icon implementations of [+] Feedback icon), by an event-driven trigger (such as clicking on a specified page element, on entry to a page, or on exit from a page), or through Invitation Feedback prompt(described below).

The default code formats a call to the OpinionLab® Comment Card Server to dynamically serve comment cards according to the referring domain/URL where the user invokes the feedback icon. Data is stored by OpinionLab® and is accessed by client administrators via an online administrative toolkit.

Can I make customizations?

No additional coding or configuration is required by the client's developers to implement the core code. However, depending on the implementations purchased by the client, the code set can be customized for functionality and look-and-feel as described later in this document.

Additionally, Custom Variable fields are available as generic database fields to store user, page or session information as desired by the client. Instructions for clients on how to add Custom Variables are included in this documentation. The effort required on the client's part to implement the use of such fields is minimal.

The color of the images and text and the font size for certain implementations can be changed, but may require the use of CSS, HTML, and image editing software and/or additional assistance from OpinionLab®. Replacement or deletion of the [+] icon requires previous approval from OpinionLab®.

Client Support and the Primary Developer Contact

OnlineOpinion® implementations are not difficult for a client's Primary Developer Contact and IT team to configure and implement, but do require an understanding of the following:

- The client's business requirements.
- The goals that the implementation is designed to address.
- The pages/paths where the code is to be implemented and how these pages function (To identify any potential known issues early in the decision-making process).
- A basic understanding of Hypertext Mark-up Language (HTML), JavaScript (JS), and Cascading Style Sheets (CSS)
- Any restrictions on inserting certain types of code into the pages in question stemming from internal protocol or applications used to deploy content.
- The way the OnlineOpinion® code should be configured and how its core functions are used (Based on documentation and support from

OpinionLab®).

OpinionLab® will provide training and instructions to a single client developer contact who will then act as a conduit between OpinionLab® developers and client developer teams. Under this agreement, a client's Primary Developer Contact can be passed to another individual in the client's organization once annually. Additional support for client developers who are not the Primary Developer Contact and/or direct code implementation on the part of OpinionLab® developers will incur additional support charges (Sold at an hourly rate in 15 hour minimum blocks) outside of the standard support agreement and require a new agreement to be signed before any work can proceed.

Code Installation

Once OpinionLab® has confirmed your requirements and has configured the code, you will receive a code set package in the form of a WinZip file.

Code Set Files

You will receive a WinZip file called oinstallV5.zip (or a variation on this). Once unzipped, you will have an oinstallV5 directory that contains the following:

- "onlineopinionV5" directory - This directory folder includes the purchased code set (JavaScript, CSS, and/or image files) with any specified configuration applied. It will contain a JS engine file called oo_engine.min.js (Or oo_engine.js for an uncompressed engine) and additional JS configuration (or CONF) file(s). CSS, HTML, and/or image files may also be present depending on the specified configuration of the code that has been requested and the implementations purchased.
- "includeV5.txt" - This text file contains the default recommended include statement corresponding to the JavaScript code. The **includeV5** statement is pasted into the html of an individual web page or template. See below for example include statements.
- "readmeV5.pdf" - This printer friendly document contains a description of OnlineOpinion® and instructions for implementation.

The sections below show the typical files included in the onlineopinionV5 directory for standard implementation types. CONF file and image names may vary. If the implementation is configured for certain components, such as [+] Feedback Icon, Invitation Feedback, Event-Driven Feedback, On-Page Card, or Inline Feedback, the onlineopinionV5 directory will typically contain one engine file and all the other files associated with all of the component.

Standard Inline Feedback Implementation

- oo_engine.min.js
- oo_conf.js
- oo_icon.gif



Inline Feedback Implementation Note

For the Inline Feedback implementation, an onClick script is used. This can be used multiple times on a page. You will find an example onClick script in your include.txt file.

Standard Floating [+] Feedback Icon Implementation

- oo_engine.min.js
- oo_conf.js
- oo_style.css
- oo_float_icon.gif

Standard Bar [+] Feedback Icon Implementation in English for the United States and English and French for Canada

- oo_engine.min.js
- oo_conf_bar_EN-US.js
- oo_conf_bar_EN-CA.js
- oo_conf_bar_FR-CA.js
- oo_style.css
- oo_bar.png
- oo_bar_icon.gif

Standard Event-Driven Feedback Implementation

- oo_engine.min.js
- oo_conf.js



On-Click Event-Driven Feedback Implementation Note

For On-Click Events, an onClick script is used. This can be used multiple times on a page.

Standard Invitation Feedback Implementation

- oo_engine.min.js

- oo_conf.js
- oo_style.css
- oo_inv_monitor.html
- oo_inv_prompt.html
- oo_loading.gif
- oo_inv_opinionlab.gif

Implementation with On-Page Card, Inline Feedback, Tab [+] Feedback Icon and Event-Driven Feedback Components and Multiple Configuration Files

- oo_engine.min.js
- oo_conf_inline.js
- oo_conf_tab.js
- oo_conf_event.js
- oo_style.css
- oo_icon.gif
- oo_tab.png
- oo_tab_icon.gif
- oo_loading.gif
- oo_close.png
- oo_shadow.png

Full Code Set

- oo_engine.min.js
- oo_conf.js
- oo_style.css
- oo_icon.gif
- oo_float_icon.gif
- oo_tab.png
- oo_tab_icon.gif
- oo_bar.png
- oo_bar_icon.gif
- oo_loading.gif
- oo_close.png
- oo_shadow.png
- oo_inv_monitor.html
- oo_inv_prompt.html
- oo_inv_opinionlab.gif



Reducing Server Calls

OnlineOpinion® code can be integrated into existing JS and CSS files, **HOWEVER, THIS MUST BE DONE AT YOUR OWN RISK.** Modified code set files may not be supported by OpinionLab®, and code placement can effect performance and functionality. Integration may also hinder OpinionLab®'s ability to trouble shoot issues that might arise.

Installation Instructions

The recommended implementation is to save the onlineopinionV5 folder in the destination web site's root directory. The folder can be stored in an alternate location, but all files (except the configuration file, which CAN be stored in another location) for one implementation should be stored in the same folder.



Place onlineopinionV5 Directory in the Root

The onlineopinionV5 directory should be saved to the root directory of the server. The location paths for the engine, configuration and CSS files in the include statement are relative. So are the JS, HTML and image files that are referenced inside the CONF files. If the location of any files changes, the references in the CONF file(s) and/or include statement must be updated to reflect their new location. Requests for specialized file placement which require OpinionLab® resources may incur additional charges.

Additional issues may occur for the Invitation Feedback instantiation if the onlineopinionV5 directory is not placed in the same domain as the page that instantiates it. More information can be found in the Invitation Feedback Implementation FAQ section.

The includeV5 statement is inserted into the html of the web page or template, outside any <DIV> or <TABLE> tags and before just before the </BODY> close body tag. Its contents will vary depending on the configuration that has been requested.



OnlineOpinion Version 5.0 and Above Requires Modern (v3.6+) Dynamic Comment Cards for Proper Supported Functionality

OnlineOpinion JavaScript Code works in conjunction with OpinionLab's dynamic comment card (DCC) technology. For all features described in the user manual to be properly supported, the card called must be a compatible version. Lower card versions may still perform basic functionality and reporting, but supported features like referrer rewrites and custom variables may not be supported.

3.6.1 Comment Card Templates Will Not Work With "onPage" Card Functionality

If your implementation involves the "onPage" (as opposed to the traditional comment card in a new window) functionality on a comment card using the 3.6.1 template, you will notice that the card defaults back to the standard 3.6 template. The code will still work properly but these two cannot be used together. The correct card will still be presented, but it will use the 3.6 template.

Firefox browsers utilizing Firebug version 1.9b3 will break the comment card resize functionality

There was a bug discovered and acknowledged by Firebug that broke the global "window.resize" Javascript function. As of Firebug version 1.9b4 (released 12/16/11), the problem was resolved. To avoid this issue, simply upgrade Firebug to version 1.9b4 or greater.



Include Statement and Configuration File Call Placement

For some implementations (such as Floating Icon, Bar Icon, Tab Icon, and Invitation Feedback) the placement of the Include Statement or the Configuration File call outside any <DIV> or <TABLE> tags and before just before the </BODY> close body tag is vital. Some implementations depend on certain Document Object Model elements to be instantiated BEFORE they can execute properly. Improper placement may cause issues with rendering and/or proper functioning of certain implementations.



DOCTYPE Support

Online Opinion code relies on JavaScript (JS) and, in configurations like [+] Feedback icons, On-Page Card, or Invitation Feedback, Cascading Style Sheets (CSS). For JS and CSS to perform properly, a supported DOCTYPE must be declared at the beginning of a document (No code can precede this). If you intend to use Online Opinion code, especially for instances relying on CSS, please make sure you use a supported and properly formed DOCTYPE on your page. For more on information, please consult <http://www.w3.org>

Supported DOCTYPEs

- <!DOCTYPE HTML>
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
- <!DOCTYPE math PUBLIC "-//W3C//DTD MathML 2.0//EN" "http://www.w3.org/TR/MathML2/dtd/mathml2.dtd">
- <!DOCTYPE math SYSTEM "http://www.w3.org/Math/DTD/mathml1/mathml.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN" "http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math-svg.dtd">
- <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
- <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
- <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Basic//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic.dtd">
- <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Tiny//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny.dtd">

Unsupported DOCTYPEs (DO NOT USE)

- No DOCTYPE at all
- A malformed DOCTYPE
- <!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
- <!DOCTYPE>



Quirks Mode is NOT Supported

Quirks Mode is not officially supported by OnlineOpinion Code for Version 5 and up. Many more recent features of OnlineOpinion code either do not work or are reduced in functionality when running in Quirks Mode (Such as on-page card and floating icon placement). We recommend that you update any page running in quirks mode to a supported DOCTYPE to avoid any loss of functionality.

For inline and on click event configurations, an onclick JS call (onclick="oo_instance.show()") must be associated with an element or elements in a header or footer or other fixed location (Link, span of text, button, image, etc.). The "oo_instance" referred to in the example below calls the object name determined in the CONF file. The name of the OnlineOpinion object can be anything, but must correspond to the CONF file and should not conflict with any other JS used.



Unique Object Identifier Required

Each CONF file must have a unique identifier for its OnlineOpinion® Object. If two or more OnlineOpinion® Objects use the same identifier, the code will not work.

Most implementations will have a single Conf File, however in some cases (such as when a specialized configuration used in one section of the site or multiple languages), using multiple conf files may be more advantageous.

Example Include Statement with Single Configuration file for Floating, Inline and Event Driven Configurations

```
<!-- BEGIN: OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is
Copyright 1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product
and other products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1
and other patents pending. http://www.opinionlab.com -->

<!-- MAIN OL STYLESHEET -->
<link rel="stylesheet" type="text/css" href="/onlineopinionV5/oo_style.css" />

<!-- MAIN OL ENGINE; NECESSARY FOR ALL IMPLEMENTATIONS -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_engine.min.js"></script>

<!-- FEEDBACK CONFIGURATION -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_conf.js"></script>

<!-- END: OnlineOpinion v5.x.x -->
```

Example Include Statement with Multiple Configuration files for Floating, Inline and Event Driven Configurations

```
<!-- BEGIN: OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is
Copyright 1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product
and other products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1
and other patents pending. http://www.opinionlab.com -->

<!-- MAIN OL STYLESHEET -->
<link rel="stylesheet" type="text/css" href="/onlineopinionV5/oo_style.css" />

<!-- MAIN OL ENGINE; NECESSARY FOR ALL IMPLEMENTATIONS -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_engine.min.js"></script>

<!-- FEEDBACK CONFIGURATION ; INLINE OR FLOATING FEEDBACK -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_conf.js"></script>
<!-- INLINE CALL - onclick='oo_feedback.show()' -->

<!-- FEEDBACK CONFIGURATION ; INVITATION -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_conf_invite.js"></script>

<!-- FEEDBACK CONFIGURATION ; EVENT-DRIVEN FEEDBACK -->
<script language="javascript" type="text/javascript" charset="windows-1252" src=
"/onlineopinionV5/oo_conf_event.js"></script>

<!-- END: OnlineOpinion v5.x.x -->
```

Standard Files and Associated Functions

Each implementation will have code files appropriate to the client's stated requirements. The files below are the files used in various standard implementations. Some files may not be required for a particular client implementation. Some files should not be modified by the client. The total size of files varies per implementation, but the expected code footprint of each element is as follows:

Required Base Elements - JS Engine and Configuration Files

File Name	File Type	Size	Description
-----------	-----------	------	-------------

oo_engine.min.js OR oo_engine.js OR oo_engine_tiny.js OR oo_engine_isc.js	JavaScript	25K OR 40K OR <2K OR <25K	Standard compressed (oo_engine.min.js - 25K), uncompressed (oo_engine.js - 40K), tiny (oo_engine_tiny.js - 1.1K - may incur additional charges), or implementation specific condensed (oo_engine_isc.js - under 25K depending on components selected - may incur additional charges) engine file used for all implementations. This can be shared across implementations. This is where variables are declared and where core code resides. This is the main JavaScript engine file. <i>Do not alter or edit without consultation with or unless advised to do so by OpinionLab®.</i>
oo_conf.js	JavaScript	1K+	Configuration (CONF) file that passes values specific to the code implementation(s) to the engine file. CONF file is used to set the components used on a page and sometimes to override certain default values for a given components. Elements such as Feedback text and Cookie configurations can be modified here. There may be more than one CONF file in an implementation depending on desired behavior and language/region support: these CONF files will have an additional suffix added, such as "oo_conf_inline.js", "oo_conf_EN-US.js", or "oo_conf_inline_EN-US.js" <i>Edit only as advised by OpinionLab® or as instructed below.</i>

Optional Component Elements - CSS, HTML, and Image Files

File Name	File Type	Size	Description
oo_style.css	CSS	6K	Cascading style sheet that passes values specific to the implementation(s). This is the file that controls the overall style of many components. <i>Edit only as advised by OpinionLab or as instructed below.</i>
oo_icon.gif	Animated GIF	1K	Standard animated [+] image file for Inline Feedback. Default is standard black (HexDec #000000); Other colors may be available on client request. <i>Do not alter or remove.</i>
oo_float_icon.gif	Animated GIF	2K	Standard animated [+] image file for Floating [+] Feedback Icon. Default is standard black (HexDec #000000); Other colors may be available on client request. <i>Do not alter or remove.</i>
oo_bar_icon.gif	Animated GIF	2K	Standard animated [+] image file for Bar [+] Feedback Icon. Default is standard black (HexDec #000000); Other colors may be available on client request. <i>Do not alter or remove.</i>
oo_tab_icon.gif	Animated GIF	2K	Standard animated [+] image file for Tab [+] Feedback Icon. Default is standard black (HexDec #000000); Other colors may be available on client request. <i>Do not alter or remove.</i>
oo_loading.gif	GIF	6K	Standard animated loading image On-Page Card and Invitation Feedback components only. Image for interim between card request and card display. <i>Do not alter or remove.</i>
oo_bar.png	PNG	1K	For Bar [+] Feedback Icon component only. Background image. <i>Do not alter or remove.</i>
oo_tab.png	PNG	1K	For Tab [+] Feedback Icon component only. Background image. <i>Do not alter or remove.</i>
oo_close.png	PNG	1K	For On-Page Card component only. Image for the close button. <i>Do not alter or remove.</i>
oo_shadow.png	PNG	2K	For On-Page Card component only. Image for card drop shadow in CSS2. <i>Do not alter or remove.</i>
oo_inv_opinionlab.gif	GIF	2K	For Invitation Feedback component only. This is the OpinionLab® logo and is a required element for the prompt <i>Do not alter or remove.</i>
oo_inv_monitor.html	HTML	4K	For Invitation Feedback component only. This is the file that monitors the user's navigation to bring the survey to the front of the browser window when the user leaves the domain or defined "friendly" domains. <i>Do not alter or remove.</i>
oo_inv_prompt.html	HTML	2K	For Invitation Feedback component only. This is the file that contains the language used in Invitation Feedback. OpinionLab® recommends only changing this language and line breaks when absolutely necessary, since the prompt window is a fixed size and contains other graphic elements. These other elements may overlap and obscure text that does not fit within the same space and formatting as the original language used. <i>OpinionLab® will provide the initial English language markup file and will configure translated markup files (from translations provided by client) for multiple language implementations per signed agreement. OpinionLab® will not configure or troubleshoot markup files altered after the initial configuration.</i>

OnlineOpinion® Engine File

The JS engine drives all instances of OnlineOpinion®, and is configured by the CONF file or specific JavaScript. Clients will typically receive one engine file to support their particular implementation.



Engine should not be modified

Modified code set files may not be supported by OpinionLab®, and code placement can affect performance and functionality. Modifications may also hinder OpinionLab®'s ability to trouble shoot issues that might arise. **ANY CHANGES YOU MAKE MUST BE DONE AT YOUR OWN RISK.**

Compressed Engine

Most clients will receive the compressed version of the full engine. This engine file allows for all possible configurations described in this document, but has been minified to reduce the size of its footprint.

Uncompressed Engine

For clients who wish to make modifications to the OnlineOpinion® Code, an uncompressed engine file (oo_engine.js) can be made available. HOWEVER, modified code set files may not be supported by OpinionLab®, and code placement can affect performance and functionality. Modifications may also hinder OpinionLab®'s ability to trouble shoot issues that might arise. **ANY CHANGES YOU MAKE MUST BE DONE AT YOUR OWN RISK.**

Implementation-Specific Condensed JS Engine and CSS

For clients concerned with file size and code footprint, the engine and CSS file can be condensed further so that it accommodates only the specific implementation(s) and/or component(s) being used. If additional components are desired at a later date, another specialized engine will need to be produced. Requests for implementation specific condensed JS Engine and CSS require additional OpinionLab® resources and may incur additional charges.

Tiny Engine

For clients extremely concerned with file size and code footprint, the engine condensed to its smallest version at 1.1K (the Tiny Engine) can be used. This engine will support ONLY Event-Driven Feedback (On-Entry, On-Exit and On-Click). Additionally, it will ONLY support an absolute domain rewrite and specially formatted Custom Variables (Specialized implementation - see below). The Tiny Engine also depends on a shorter, specialized object instantiation.

For On-Click Implementations, we recommend using the oo_icon.gif animated GIF in conjunction with the word "Feedback" for this implementation. OpinionLab® reserves the right not to approve alternate images and text that do not reflect OpinionLab®'s brand image.

If additional components are desired at a later date, another engine would need to be produced. Requests for the Tiny Engine require additional OpinionLab® resources and may incur additional charges.

Tiny Engine Configuration Object Options

These are the ONLY options available in the Tiny Engine.

- **onEntry:** *Integer* || 0-100
Possibility of seeing a Dynamic Comment Card upon loading the page 0-100
- **onExit:** *Integer* || 0-100
Possibility of seeing Dynamic Comment Card upon leaving the page 0-100
- **tealeafid:** *String*
If a client has Tealeaf, this object will collect their Tealeaf cookie as a string. Default name of a Tealeaf object is 'TLTSID'; if this default name is changed in the client's code, it must match the name of the value declared here.
- **domain:** *String*
Will dynamically replace the current URLs' hostname with the string provided



Domain Rewrite and the Tiny Engine

Notice 'http://' is truncated in this URL string:

```
domain: 'www.opinionlab.com'
```

- **customVariables:** *String*

In the full engine this was an object that could pass any number of key: value pairs. In order to keep the footprint down this is now a string that you must format to look like a javascript object.



Custom Variables and the Tiny Engine

Notice the concatenated string with a replaced variable:

```
customVariables: "{myKey1: myValue1, myKey2: '+myJSVariable1+'}"
```

Tiny Engine Instantiation Examples

Tiny Engine Instantiation Example - On-Click

```
var oo_on_click = new OOo();

<!--call the show() method from the link-->
<a href='http://www.opinionlab.com' onclick='oo_on_click.show()'>Click me!</a>
```

Tiny Engine Instantiation Example - On-Exit with domain replacement and Custom Variables

```
var oo_on_exit = new OOo({
  onExit: 100
  , domain: 'www.opinionlab.com'
  , customVariables: "{text: 'Text', myVariable: '+myValue+'}"
});
```

Configuration Files

Code files are configured by OpinionLab® according to initial client requirements. The elements so noted below can be reconfigured by clients within the original files or copies of them. Requests for new code instances beyond the initial implementations indicated in the License Agreement which require OpinionLab® resources may incur additional charges.

The CONF file generally controls how a Comment Card renders: the type of link and its associated text and image, when and how Event-Driven Feedback and Invitation Feedback cards render, and if the card is an "On-Page" Comment Card. It also controls any URL/domain rewrites that may be required. Custom Variables are added in this file.

The CSS file controls the placement, color, font and size of elements of pertaining to the [+] Feedback Icons, Invitation Feedback, and the On-Page Card.

A typical CONF file is shown below. In this case, it is for a Floating configuration with an On-Page Card and three Custom Variables.

```
/* OnlineOpinion v5.x.x */
/* Released: x/x/2012 */
/* Components: Full
/* The following code is Copyright 1998-2012 OpinionLab, Inc. All rights reserved. Unauthorized use
is prohibited. This product and other products of OpinionLab, Inc. are protected by U.S. Patent No.
6606581, 6421724, 6785717 B1 and other patents pending. http://www.opinionlab.com */

/* Floating [+] Feedback Icon Configuration */
var oo_float = new OOo.Ocode({
  floating: {},
  onPageCard : {},
  customVariables :
  {text:'Text', numbers: 12345, pageTitle: document.title}
});
```

Configuration Types and Elements

The oo_conf.js file passes values to the oo_engine.min.js or oo_engine.js file.

The items in this section show the elements that can be modified for each configuration type. Note that elements of the "ENGINE" files are not included below as they should not be modified by clients unless directed to do so by OpinionLab®.

The fields defined in the CONF file are as follows:

OnlineOpinion® OCode Type Instantiation and Configuration

OCode covers all components except Invitation Feedback. This includes inline feedback, [+] Feedback Icons and Event-Driven Feedback. Specialized OCode components such as On-page Card, tunnels and Custom Variables are also available to be configured. Most options have default values, so specification is not required unless noted (as shown with **REQUIRED**) or desired. For a single instance:

```
new OOo.Ocode({
  /* Configuration object goes here */
});
```

For multiple instances:

```
var oo_instance_1 = new OOo.Ocode({
  /* Configuration object goes here */
});
var oo_instance_2 = new OOo.Ocode({
  /* Configuration object goes here */
});
```

cookie

This object will determine if the code runs or not based on the specified options. Cookies effect the rendering of [+] Floating Feedback Icon and Event-Driven Feedback.

- **name:** *String* || **null**
Determines the name of the cookie that will be set. The prefix "oo_" is recommended to help avoid conflicts and to trouble shoot.



Sharing Cookie Names

Unique cookie names are typically used for each object, but if the cookie name can be shared across event driven objects to prevent more than one event survey from rendering to a user.

- **type:** *String* || *null*

If the cookie is set to "**domain**", it will disable the code for the entire domain. When set to "**page**", it will disable the code on a per page basis.



"domain" vs. "page"

For Event-Driven Feedback configurations, "domain" to prevent the survey associated with one or multiple objects from appearing to users on a repeated basis. For [+] Floating Feedback Icon configurations, "page" type is usually used along with the "disappearOnClick" (see configuration element below) to prevent the icon from reappearing on that page until the cookie has expired: "domain" is not usually recommended for this configuration.

- **expiration:** *Integer* || *null*

Determines how long until the cookie expires in seconds.



Disappearing Icon Configuration

If a page cookie is set to '0' for [+] Floating Feedback Icon configuration and "disappearOnClick" (see configuration element below) is set to 'true', the icon will disappear after being clicked by a user, but will return when the user navigates away and returns, or refreshes the page.

referrerRewrite

Will rewrite the "referer" passed to the OpinionLab® server, **which is the basis for card activations and report groups**. If not used the code will simply pass the current URL.

- **searchPattern** *String* or *RegExp* || *null*

This will be the pattern that will search against the current URL. For example, you can search for the domain and then replace anything that follows it, or you can search for the domain and a certain level of sub-directory and replace the rest of the URL. **Note:** If you leave this option null, the replacePattern will replace the entire URL.

- **replacePattern** *String*

Will replace the matched string from the searchPattern or replace the entire URL if searchPattern is null.

disappearOnClick

If set to "true" the icon will disappear after clicking on it. Requires a cookie to be set if the icon is expected to not reappear upon page redraw. Does not work with Inline or Event Driven implementations.

floating

This object enables and controls the Floating [+] Feedback Icon attached to the page. Cannot be used with another [+] Feedback Icon in a single object. Additional customizations can be handled in the CSS.

- **id:** *String* || *null*

If the client needs to target the floating div for some reason they can attach an identifier for it here.

- **position:** *String* || *null*

Will set how the logo positions itself on the page. Default will simply position itself relative to the bottom right hand corner. Other options are "**rightOfContent**", "**fixedPreserveContent**" and "**fixedContentMax**".

- **contentId:** *String* || *null*

If you set a special position for the floating code you must tell the code what content div to respect otherwise it will default to bottom right hand corner.

- **caption:** *String* || "**Feedback**"

Text that will be displayed underneath floating logo. OpinionLab recommends that the standard text be used. Changes to create a longer text string may create problems with the positioning of the text relative to the [+] Floating Feedback Icon.

- **hoverCaption:** *String* || "**Click here to rate this page**"

Text that will be displayed when the logo is hovered over. Also used for the image alt attribute of the logo. Changes to create a longer text string may create problems with the positioning of the text relative to the [+] Floating Feedback Icon.

tab

This object enables and controls the Tab [+] Feedback Icon attached to the page. Cannot be used with another [+] Feedback Icon in a single object. Additional customizations can be handled in the CSS.

- **position:** *String* || '**right**'

Sets the tab to either on the right (default) of the page or the '**left**' if specified.

- **title:** *String* || "**Feedback**"

Changes the title tag associated with the tab.

bar

This object enables and controls the Bar [+] Feedback Icon attached to the page. Cannot be used with another [+] Feedback Icon in a single object. There are no options for this implementation: All customizations of the bar can be done in the css.

events

Controls whether or not a card will fire on entry or exit of a page, or clicking a particular link.

- **onEntry:** *Integer* || 1 to 100
The probability of a card popping on loading the page. Expressed as an integer: 50 for 50% chance
- **delayEntry:** *Integer*
The length of time in seconds before the card pops.



Delay entry only applies to an on-entry event. This will not work with on-click or on-exit events.

- **onExit:** *Integer* || 1 to 100
The probability of seeing a comment card pop on exit from a page or Browser close in certain browsers. Expressed as an integer: 50 for 50% chance
- **onSingleClick:** *Integer* || 1 to 100
The probability of the card launching when the OOo.Ocode.show method is called from a link. Expressed as an integer: 50 for 50% chance
- **disableLinks:** *RegExp*
When using a card that pops upon page exit you can specify <a> tag based links that will not activate the card using a regular expression. Any links that return a match for the regular expression will not pop a card.
- **disableFormElements:** *Boolean* || **false**
If set to **true** no form elements will pop a card on page exit.



Using disableLinks or disableFormElements

Other links and buttons on the page that do not fall into the disableLinks or disableFormElements definition should use the following script in the HTML of the page to prevent them from causing an Event Card to render on exit from a page:

```
oo_instance.interruptShow = true;
```

onPageCard

Will embed the Dynamic Comment Card on the page rather than being a new window. Do not use with an on exit event as it will never be shown.

- **closeWithOverlay:** *Boolean* || **false**
When **true** it will allow the user to close the onPageCard by clicking on the overlay.

tunnel

Will allow you to specify a number of steps that a user has to take in order for any of the code to run. Similar functionality is handled for Invite by "promptTrigger". No boolean logic is allowed.

- **path:** *Array*
An array of strings that define the steps. Must begin with a '/'.
Example: [/test/1.html, /test/1.html, /test/3.html]
- **cookieName:** *String* || **oo_tunnel**
Name of the cookie to be used for the tunnel

abandonment

Allows an "area" of a website to control when an a comment card will be shown. If you leave the area before you hit the specified end page a comment card will be shown on the subsequent page (on load) if the engine is on it and it passes any other conditions you have specified for that Ocode instance. The card only gets called on pages with code that abandon the tunnel: The user must hit a non-conforming page to abandon.



Example: The user engages the abandonment watch by hitting the start page, nothing should pop as long as they stay on the middle pages and then hit the end page. If the user "abandons" the middle pages before hitting the end page, it triggers the event card. Simply closing the browser will not cause a card to render.

- **startPage:** *String*
The page that a user must hit to activate the abandonment area.
- **middle:** *RegExp*
URLs that match the specified RegExp will be considered part of the area.
- **endPage:** *String*
The URL of the end page. If a user visits this page, there will be no comment card on the subsequent URLs.

tealeafCookieName

If the client has Tealeaf and their cookie is not 'TLTSID', pass the name of the cookie to this key as a string.

```
/* Inline \[+\] Feedback Icon configuration */
var oo_feedback = new OOo.Ocode({
  tealeafCookieName: "TL_Identifier"
});
```

sessionCookieName

If the client has a cookie that needs to be passed for specialized OpinionLab integration services, pass the name of the cookie to this key as a string. Contact Your Client services representative before using this.

```
/* Inline \[+\] Feedback Icon configuration */
var oo_feedback = new OOo.Ocode({
  sessionCookieName: "Cookie_Name"
});
```



In the configuration, "sessionCookieName" is incompatible with "tealeafCookieName". If both are used, "tealeafCookieName" will have priority.

clickTalePID

If the client has ClickTale®, pass the name of the PID to this key as a string.

```
/* Inline \[+\] Feedback Icon configuration */
var oo_feedback = new OOo.Ocode({
  sessionCookieName: "CTSID"
  clickTalePID : "****"
});
```

newWindowSize

An array that will define the initial width and height of the new window (If on-page card is not used). This will be influenced by the comment card's resize engine once the comment card loads. Example: [545, 700] the initial new window will be 545px wide by 700 pixels tall

disableMobile

If set to true it will prevent the code being run on the following mobile devices: Mobile Safari, Fennic, Mobile Opera, Mobile IE, Palm Pre and BlackBerry.

disableNoniOS

If set to true it will **only** allow code to run on a mobile device running on an iOS platform (iPhone, iPad) and prevent the code from running on any

Windows Mobile or Android based mobile device.

customVariables

This will pass any optional variables on the page to the comment card to be stored in the Custom Variable fields. Example:

```
{var1: 23, company: 'opinionlab'}
```

This will pass the Custom Variables to the comment card as custom_variables=23|opinionlab. Note that the names used for the values (var1, company) will not be written to the OpinionLab® database. They are only used for reference purposes so that it is easy to see what the value represents.



Custom Variable Values

If a value named as a custom variable is not available for a given page, the variable must be named and given a null value or else the OpinionLab code naming the variable as a custom variable will fail.

Example configurations for OCode Types

Please see examples below for sample configurations. Note that multiple configurations can be used in on CONF file by restating the opening

```
var oo_instance_x = new OoO.Ocode({
```

command and following with a new set of configuration values and properly closed.

This example will instantiate a basic Inline Feedback instance.

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Inline Feedback configuration */
var oo_instance = new OoO.Ocode({});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

This example will instantiate a Floating [+] Feedback Icon instance that:

- is positioned to the right of the content <DIV>
- has altered caption text
- utilizes an On-Page Card
- that closes on overlay click
- that contains Custom Variables.

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is
Copyright 1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This
product and other products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581,
6421724, 6785717 B1 and other patents pending. http://www.opinionlab.com */

/* Floating [+] Feedback Icon configuration */
var oo_float = new OOO.Ocode({
floating :
{position: "rightOfContent",
contentId: "content",
caption: "Rate Us"},
onPageCard :
{closeWithOverlay: true},
customVariables :
{specializedText: 'Text',
pageName: document.title}
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

This example will instantiate an implementation that:

- instantiates a Event-Driven On-Exit from a page set that:
 - is set to 30% probability
 - is disabled for links to <http://www.opinionlab.com>
 - has form elements disabled
 - has a cookie that applies to the domain for 1 hour
 - has a referrer re-write
 - will resize the window for the Dynamic Comment Card for the event instance (Which may again be overwritten by the card's resize engine)
- instantiates a basic Bar [+] Feedback Icon instance

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is
Copyright 1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This
product and other products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581,
6421724, 6785717 B1 and other patents pending. http://www.opinionlab.com */

/* Event-Driven Feedback configuration */
var oo_on_exit = new OOO.Ocode({
cookie:
{name: "oo_on_exit",
type: "domain",
expiration: "3600"},
event :
{onExit: 30,
disableLinks: /^(https:\\\\www\\.opinionlab\\.com)/i
disableFormElements: true},
referrerRewrite :
{replacePattern: 'http://www.opinionlab.com/exit_event.html'},
newWindowSize: [545, 600]
});

/* Bar [+] Feedback Icon configuration */
var oo_bar = new OOO.Ocode({
bar : {}
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

This example will instantiate an implementation that:

- is Event-Driven On-Entry with 100% probability
- has a 5 second delay
- uses an On-Page Card
- uses a cookie for the page that lasts 1 month
- has a tunnel path set up
- includes a Tab [+] Feedback Icon instance aligned to the left.
- has a Tealeaf Cookie association: cookie name "TL_Identifier" for both objects

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is
Copyright 1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This
product and other products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581,
6421724, 6785717 B1 and other patents pending. http://www.opinionlab.com */

/* Event-Driven Feedback configuration */
var oo_on_entry = new OOO.Ocode({
  cookie:
    {name: "oo_on_entry_tunnel",
      type: "page",
      expiration: "2592000"},
  event :
    {onEntry: 100,
      delayEntry: 5},
  onPage : {},
  tunnel :
    {path: ['/test/1.html', '/test/2.html', '/test/3.html'],
      cookieName: "oo_on_entry_path"},
  tealeafCookieName: "TL_Identifier"
});

/* Tab [+] Feedback Icon configuration */
var oo_tab = new OOO.Ocode({
  tab :
    {position: "left"},
  tealeafCookieName: "TL_Identifier"
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

OnlineOpinion® Invitation Type Instantiation and Configuration

Invitation type covers only Invitation Feedback. Specialized OCode components such as tunnels and Custom Variables are also configured. Most options have default values, so specification is not required unless noted (As shown with **REQUIRED**) or desired.

```
new OOO.Invitation(
  /* Configuration object goes here */
);
```

Required Configuration

pathToAssets is the only *required* configuration that must be defined in the configuration file. However if the optional configurations are not defined, they will default to the values listed.

- **pathToAssets:** *string* **REQUIRED**
This is the location of the onlineopinionV5 folder and its contents. The path must begin with a '/' or 'http://' (the assets, specifically oo_inv_prompt.html, must be hosted on the same domain as the URL that the Invitation Feedback prompt will fire on to avoid cross-domain issues) and end with a '/'. All assets (Images, styles, JavaScript and markup files) must be in the same place. This is a required field.



Multiple Implementations

If you require multiple implementations (for example, for multiple language implementations), you will need unique code sets for each. Files will not be shared across implementations.

Optional Configurations

The options shown below are optional fields with default values. Each CONF file is then configured to fit a particular implementation. If these fields are not necessary, you can omit them from the CONF file or comment them out (by adding // before them). However, if omitted they will behave with default behavior.

- **companyLogo:** *String* || **null**

If a logo is to be added to the top of the prompt, insert the URL location of the image that will be embedded in the prompt here. The URL should begin with a '/' or 'http://'



Logo size

The containing block for the company logo is 355 pixels wide by 55 pixels tall. Most modern browsers will accommodate and appropriately size image dimensions smaller than this, but MS Internet Explorer may skew the image to fit the available space. Because of this, recommended best practice is to have the logo image sized at 355 pixels by 55 pixels using a white (or the appropriate color) background.

- **responseRate:** *integer* || **50**

Percentage of users to be prompted expressed as an integer. If this value is not indicated, the default percentage is 50%.



Using the Tunnel Configuration

For implementations where a tunnel or RegExp is used to define when the prompt appears, the percentage of users who receive the prompt is based on the first page where they encounter the installed code (e.g. the homepage or another landing page), and not on the pages in the defined tunnel or matching the regular expression. In these scenarios, the percentage should be set higher to account for the fact that some of the users who fall within the percentage that should get the prompt may not continue to the pages where the prompt renders.

- **repromptTime:** *integer* || **604800**

Number of seconds until the prompt can reappear to the same user. Defaults to the equivalent of 7 days (One week).



Immediate Re-prompting

A value of -1 will cause a cookie to still be set, but the prompt will continuously appear.

- **asm:** 1 || **null**

This field is used to tell the engine that the survey to be launched is an "ASM" survey and requires a larger window. The value you should enter for an ASM survey is '1'. If the survey is NOT an ASM survey, the value should be null.

- **asm:** 2 || **null**

Passing a value of 2 instead of 1 (see above) will remove the iframe from around the survey, open the new window with scroll bars and allow the client to also pass the `newWindowSize` parameter to launch a survey in any size window they want. If 2 is passed for "asm", then a "newWindowSize" must also be specified.

- **popupType:** *String* || **'popunder'**

This option accepts one of two values. Either 'popup' or 'popunder'. 'Popup' will bring the Comment Card to the front as soon it renders. 'Popunder' will render a monitor window behind the current window. When the user leaves the site (as defined by the Friendly Domains below), closes the browser or when the user focuses the monitor window it will load the comment card for the URL where the prompt appeared.



Known Issue - Opera 10 and above

There is a known issue with Invitation Feedback popunder functionality where the new window will not launch below the current window: It will launch above. There is no known fix for this issue.

- **promptTrigger:** *Array* or *RegExp* || **null**

If you specify an array it will be a tunnel so a user must follow the steps exactly to be prompted. The URLs can be entered as full or relative paths contained within ' ' characters (e.g. 'http://www.opinionlab.com/'), separated by commas. A Regular Expression (RegExp)

will test the URL where the user has landed and if there is a match to the value entered, the prompt will render. This field is optional. If no value is specified, the code will be active on all pages where it is referenced. Incompatible with pagesHit. Similar functionality is handled for OCode Event implementations by "tunnel".

- **pagesHit:** *Integer* || 0
Determines number of confirming page hits (loads) or refreshes that must occur before a prompt will render. Incompatible with promptTrigger.



Use of "pagesHit" Within a Section or Site Wide

For the pagesHit, the code needs to be installed site-wide or throughout a particular section to properly fire. The number of pages hit can only be incremented when the code is on a particular page (no code, no incrementation). If '3' is specified as the pages that need to be hit, then when a user gets to the third page the rest of the code will execute.



So "typical" or simple prompt trigger that has a regular expression for multiple matching URLs like "VcartV", only has one URL or it, or you can have multiple entries with commas as in:

```
[ 'http://www.mystore.com/example/step1.html',  
  'http://www.mystore.com/example/step2.html',  
  'http://www.mystore.com/example/step3.html'  
]
```

in which case the user gets the invitation if they hit these pages in sequence.

- **area:** *RegExp* || null
If the monitor window is running in the background this regular expression will test the url for a match. If there is no match in the url the monitor window will come forward.
- **goal:** *String* || null
If the string provided has any partial match in the current url the user is on the monitor window will close silently in the background.
- **referrerRewrite:** *Object* || null
If you need to rewrite the referrer being sent to the comment card use this object with the following options:
 - **searchPattern** *String* or *RegExp* || null
This will be the pattern that will search against the current URL. For example, you can search for the domain and then replace anything that follows it, or you can search for the domain and a certain level of sub-directory and replace the rest of the URL.
Note: If you leave this option null, the replacePattern will replace the entire URL.
 - **replacePattern** *String*
Will replace the matched string from the searchPattern or replace the entire URL if searchPattern is null.
- **friendlyDomains:** *String* or *Decimal* || null
This will define the area that is considered the "site" for site exit or end of visit surveys. It can be a full URL string or an IP address. If no value is defined, the default behavior is for a popunder survey to come forward when the user leaves the domain where the prompt rendered. For example, if the user is on the domain <http://www.mystore.com> when the prompt appears, the popunder survey will come forward when they navigate to any other domain. If a value of "https://checkout.mystore.com" is entered, if the user clicks into a checkout path on this alternate sub domain, the survey will not come forward until the user has navigated to a domain that is not <http://www.mystore.com> or <https://checkout.mystore.com>;



Code must also be on Friendly Domains

The Invitation Feedback code must be installed on the friendly domain (and refer to the same file location) in order for this value to keep the survey from rendering until the user also leaves this domain. Just take the exact same object from one domain to the other: don't change any of the settings, otherwise it might result in unexpected behavior.

- **promptDelay:** *Integer* || 0
This will control the number of seconds that pass before the prompt appears on the page that it will render on.
- **neverShowAgainButton:** *Boolean* || false
Determines if a "never show again" button will be on the prompt for the survey. If the value is set to "true" and if the user clicks yes they will never see the prompt again (until they clear their cookies or 5 years have passed). This uses the same cookie (oo_inv_reprompt) as is used to control the prompt reappearing in general.
- **customVariables:** *Obj* || null
This will pass any optional variables on the page to the comment card to be stored in the Custom Variable fields. Example:

```
{var1: 23, company: 'opinionlab'}
```

This will pass the custom variables to the comment card as customVariables=23|opinionlab. Note that the names used for the values

(var1, company) will not be written to the OpinionLab® database. They are only used for reference purposes so that it is easy to see what the value represents.

- **newWindowSize:** *Obj || null*
An array that will define the initial width and height of the new monitoring window and subsequent Dynamic Comment Card. This will be influenced by the Dynamic Comment Card's resize engine once the comment card loads. Example: [545, 700] the initial new window will be 545px wide by 700 pixels tall
- **chromeMainWinPrompt:** *String || Please fill out the form behind this window when you are finished.*
The text that will be used by the JavaScript alert when the invite is set to 'popunder' and viewed in Google Chrome Just before the monitoring window is sent to the back. Default is as listed.
- **chromeSurveyPrompt:** *String || Please fill out the form.*
The text that will be used by the JavaScript alert when the invite is set to 'popunder' and viewed in Google Chrome Just before the monitoring window is brought to the prompt upon the appropriate trigger. Default is as listed.
- **promptMarkup:** *String || "oo_inv_prompt.html"*
Used to modify the file name for the prompt markup. Defaults to oo_inv_prompt.html. Only put the file name here do not add any slashes.
- **monitorWindow:** *String || "oo_inv_prompt.html"*
Used to modify the file name for the monitor window. Defaults to oo_inv_monitor.html. Only put the file name here do not add any slashes.

Cookie Configurations

These are internal variables that are exposed but should not be changed unless necessary. The only time the cookie names should be changed is when there is the possibility of conflict between multiple code sets used on the same domain. For example, if you have two Invitation Feedback surveys running concurrently and you want users to potentially see both, you should have different cookie names. You should also make sure that unique code instances that each use the oo_inv_tunnel cookie for the prompt that appears in a defined tunnel have different cookie names. *Cookies cannot be used across domains or sub-domains, so for example, if the code is installed on one domain and also on another sub-domain, users may see a prompt on each.*

- **tunnelCookie:** *string || 'oo_inv_tunnel'*
Names the cookie to be used if a tunnel is specified



promptTrigger and tunnelCookie

Note that this cookie is used in combination with a "promptTrigger" value (series of URLs) for the promptTrigger field above. The cookie is set on the first page where the user encounters the installed code. It is re-set if the user enters the tunnel but does not follow the URL path sequentially.

- **repromptCookie:** *string || 'oo_inv_reprompt'*
Names the cookie that will be used to know if and when to subsequently render the prompt or not. This is also the cookie used when the neverShowAgainButton option (described above) is used.
- **percentageCookie:** *string || 'oo_inv_percent'*
When a user first encounters the code on a site, the JavaScript will calculate whether or not the user should have a chance at seeing the prompt. This value is stored in this cookie. If no URL is defined for where the prompt should render, the prompt will then render immediately for users that fall into the "prompt" group. For RegExp and Tunnel prompt definitions, a user who is deemed to have the possibility of seeing the prompt will see it render when they arrive at the specified page or follow the defined tunnel. Note that depending on where the user first encounters the code, there may be users that are in the "prompt" group that may never reach the tunnel or pages where the prompt appears, resulting in a lower percentage of the user audience being polled than is actually defined in the CONF file. In these cases, it is recommended that the percentage be set to a higher value to account for "drop-off".

Example Configurations for Invitation Type

Please see examples below for sample configurations. Note that multiple configurations can be used in on CONF file by restating the opening

```
new OOo.Invitation({
```

command and following with a new set of configuration values. This approach can be used when the prompt that appears to users is the same across implementations, but where you wish to vary the percentage or other conditions under which the prompt and survey appear you must instantiate different objects. If you are implementing prompts in multiple languages, you must have a unique markup.html file, and since the other related files (aside from the CONF file) must be contained in a single directory for each implementation, you will require multiple sets of the other files.

This prompt will render:

- on every page where the code is installed.
- will launch an ASM survey immediately in front of the browser window.

- a prompt that will immediately reappear if the user returns to one of the pages where the code is installed.

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOO.Invitation({
/* REQUIRED fields */
  pathToAssets: 'http://mystore.com/onlineopinionV5i/'

/* Optional fields */
  , companyLogo: 'http://www.mystore.com/images/logo.gif'
, responseRate: 100
  , repromptTime: -1
  , popupType: 'popup'
  , asm: '1'
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

The example below shows a tunnel where:

- the prompt will only render when the steps are followed sequentially, to 80% of those who encounter the code.
- the survey will render behind the browser and will come forward when the user exits the domain.
- users who have seen the prompt and return to the pages where the code is installed will not see the prompt again for 7 days.

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOO.Invitation({
/* REQUIRED fields */
  pathToAssets: 'http://mystore.com/onlineopinionV5i/'

/* Optional fields */
  , companyLogo: 'http://www.mystore.com/images/logo.gif'
, responseRate: 80
  , repromptTime: 7
  , popupType: 'popunder'
  , promptTrigger: ['http://www.mystore.com/example/step1.html',
'http://www.mystore.com/example/step2.html',
'http://www.mystore.com/example/step3.html'
]

});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

The example below shows:

- a prompt that will render based on a regular expression.
- a prompt that will render 3 seconds after a URL matching the RegEx is loaded.

```

/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOo.Invitation({
/* REQUIRED fields */
  pathToAssets: 'http://mystore.com/onlineopinionV5i/'

/* Optional fields */
  , companyLogo: 'http://www.mystore.com/images/logo.gif'
, responseRate: 100
  , promptDelay: 3
, repromptTime: -1
  , neverShowAgainButton: true
  , popupType: 'popup'
  , promptTrigger: /products/
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */

```

Disabling or Providing Alternative OnlineOpinion Object Configurations for Certain Browsers

In some cases, it is desirable to disable or provide an alternate OnlineOpinion Object instance to either collect feedback from a specific audience or to account for particular known issues.

Using the if/else statement below will allow you to disable an OnlineOpinion Code instance for Microsoft Internet Explorer Version 6 and Version 9 and Mozilla Firefox Version 4. In the example below, an on-entry feedback event will trigger an on-page card if the user is using any supported browser except Internet Explorer 6, Internet Explorer 9, and Firefox 4.

```

if (OOo.Browser.IE && OOo.Browser.Version == 6) {
  // Affect IE 6 or leave empty
}
else if (OOo.Browser.IE && OOo.Browser.Version == 9) {
  // Affect IE 9 or leave empty
}
else if (OOo.Browser.Gecko && navigator.userAgent.indexOf('Firefox/4') != -1) {
  // Affect Firefox 4 or leave empty
}
else {
  // Instantiate normally for all other browsers
/* Entry configuration */
  var oo_entry = new OOo.Ocode({
  cookie: {name:"oo_entry", type:"page", expiration:"0"},
  onPageCard : {}
  });
}

```

If a single browser version is desired to be disabled, simply leave off of the other conditionals. In the example below, an invite feedback prompt will trigger if the user is using any supported browser except Firefox 4.


```
if (OOo.Browser.Gecko && navigator.userAgent.indexOf('Firefox/4') !== -1) {
    // Affect Firefox 4 or leave empty
}
else {
    // Instantiate normally for all other browsers
/* Run the Invitation instance */
    new OOo.Invitation({
        pathToAssets: '/onlineopinionv5/',
    });
}
```

If an alternate configuration is desired, simply leave off of the other conditionals and add the object configuration. In the example below, all supported browsers will display a Floating Feedback Icon that will launch an on page comment card except Internet Explorer 6 which will launch the card in a new window.

```
if (OOo.Browser.IE && OOo.Browser.Version == 6) {
    // Affect IE 6 or leave empty
/* [+] Floating Icon configuration */
    var oo_floating = new OOo.Opcode({});
}
else {
    // Instantiate normally for all other browsers
/* [+] Floating Icon configuration */
    var oo_floating = new OOo.Opcode({
        onPageCard : {}
    });
}
```

Additionally keep in mind the Mobile browser disabling features (disableMobile and disableNoniOS) outlined in the configuration section of the User Manual.

For code to affect specific supported browsers not mentioned here, Please contact your OpinionLab Client Services Representative.

CSS File Configuration

The **oo_style.css** contain all the CSS selectors for the OCode type implementations and many selectors for the Invitation type.

Floating [+] Feedback Icon Based Selectors

```
#oo_feedback_fl_spacer
.oo_feedback_float
.oo_feedback_float .olUp
.oo_feedback_float .olUp img
.oo_feedback_float .oo_transparent
.oo_feedback_float: hover .olUp
.oo_feedback_float .fbText
.oo_feedback_float .olOver
.oo_feedback_float .olOver span
.oo_feedback_float: hover .olOver
```



Common Modifications

- Modify ".oo_feedback_float" to change the text styling.
- Modify ".oo_feedback_float .oo_transparent" to change background color and opacity for the off state.
- Modify ".oo_feedback_float: hover .olOver" to change background color for the over state.

On-Page Card Based Selectors

```
#oo_overlay
#oo_overlay.no_loading
.oo_cc_wrapper
.oo_cc_wrapper.iwrapper
.oo_cc_wrapper.iframe
.oo_cc_wrapper.oo_cc_close
.oo_cc_wrapper.oo_shadows
.oo_cc_wrapper.oo_shadows div
.oo_cc_wrapper.oo_shadows.oo_top
.oo_cc_wrapper.oo_shadows.oo_body
.oo_cc_wrapper.oo_shadows.oo_bottom
```



Common Modifications

- Modify "#oo_overlay" to change the loading graphic, background color, and opacity for overlay.

Bar [+] Feedback Icon Based Selectors

```
#oo_bar
#oo_bar:hover
#oo_bar span
```



Common Modifications

- Modify "#oo_bar" to change the background color and graphic for the icon.

Tab [+] Feedback Icon Based Selectors

```
#oo_tab
#oo_tab span
#oo_tab a
.oo_tab_left
.oo_tab_left span
.oo_tab_left a
.oo_tab_left:hover
.oo_tab_right
.oo_tab_right span
.oo_tab_right a
.oo_tab_right:hover
.oo_tab_ie_right
```



Common Modifications

- Modify "#oo_tab" to change the background color and graphic for the icon. You will need to modify oo_tab.png as well.

Invitation Feedback Based Selectors

```
#oo_overlay
#oo_overlay.no_loading
#oo_container
#oo_invitation_prompt
#oo_ol_brand
#oo_invitation_prompt a
a#oo_launch_prompt
#oo_invitation_prompt a#oo_launch_prompt:hover
#oo_invitation_prompt a:hover
#oo_close_prompt:hover
```

Print Based Selectors (@media print)

#oo_bar, .oo_feedback_float, #oo_tab - Hides [+] Feedback Icons from printers

Frequently Asked Questions (FAQs)

Below are an assortment of frequently asked questions regarding specific implementations, configurations, and customizations.



Additional OpinionLab® Services

Please note that where OpinionLab® services are required for additional configurations or customizations mentioned below, chargeable blocks of time will apply.

Inline Feedback Implementation

How do I place an Inline Feedback link?

After your code has been properly installed, simply apply an onclick Javascript call to the desired element or elements on your page:

```
onclick="oo_instance.show()"
```

We recommend using the oo_icon.gif animated GIF in conjunction with the word "Feedback" for this implementation. OpinionLab® reserves the right not to approve alternate images and text that do not reflect OpinionLab®'s brand image.

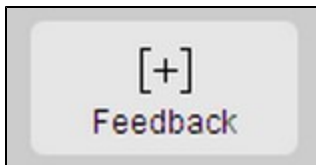
How do I create separate "site" and "page" level feedback links?

If you want the comment cards to be different for each link, and if you want to track the data separately for each, you will need two separate OnlineOpinion® objects in the CONF file. In the onClick script referenced above, "oo_instance" should be changed to reflect the OnlineOpinion® object for the corresponding conf file. One of the OnlineOpinion® objects should also include a domain rewrite (see below) to change the URL for one of the links to distinguish it for card activation and reporting.

Can I have different cards appear from the same page with one Inline feedback implementation?

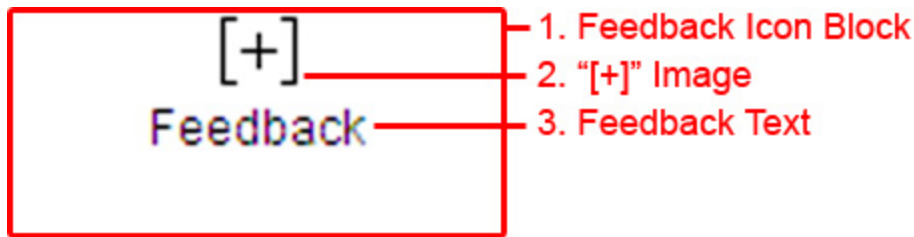
(Yes, but you will need to instantiate a new OnlineOpinion® object for each instance and apply a referrer rewrite to one line to differentiate the referer value being used. Contact your customer service representative for more information.

Floating [+] Feedback Icon Implementation



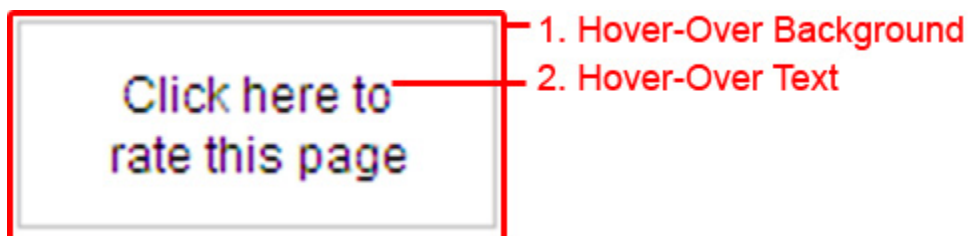
What are the Basic Parts of the Floating [+] Feedback Icon?

Default (Inactive) State



1. **Feedback Icon Block** - Reference is controlled in [configuration file](#) by "floating -> id". Look and Feel is controlled in [CSS file](#) by ".oo_feedback_float".
2. **"[+]" Image** - Reference is controlled in [CSS file](#) by ".oo_feedback_float .olUp img".
3. **Feedback Text** - Reference is controlled in [configuration file](#) by "floating -> caption".

Hover-over (Active) State



1. **Hover-Over Background** - Look-and-Feel is controlled in [CSS file](#) by ".oo_feedback_float .olOver".
2. **Hover-Over Text** - Reference is controlled in [configuration file](#) by "floating -> hoverCaption".

How do I keep the Floating [+] Feedback Icon from overlapping my content?

If your content is "liquid", the default code will adjust the position of the icon relative to the adjusted page content: You can add some padding to the right of your content using CSS. For other content alignment types, create a <DIV> layer around the content of your web page and give it the ID "content". Then change the float_style in the CONF file to "rightOfContent" or "fixedPreserveContent".

Can the placement of the Floating [+] Feedback Icon be changed?

Yes to a degree. The Floating [+] Feedback Icon reference is controlled in the [configuration file](#) by "floating -> position", which controls the placement, and by "floating -> contentId:", which refers to content container the placement is relative to. The vertical position can be modified to an extent through the CSS, however OpinionLab® **strongly discourages** changing the vertical positioning of the icon.



Identified Content Container Required for Floating + Feedback Icon Placement

Clients must have a containing DIV or TABLE with an ID specified in the CONF file ("float -> contentId") encompassing the entirety of the page content for any placement besides the default (None). Failure to properly identify this containing element sets positioning to the default position. Failure to properly place this containing element will cause icon rendering issues for any configuration except the default.

Options for "float_style" are as follows:

null (Default)



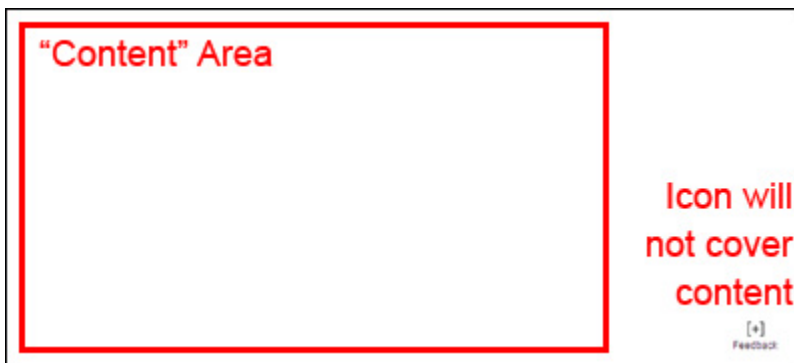
Icon hovers in the lower-right corner of the window. This configuration respects no content container and will render on top of existing content if the window is resized or the content is not set up with a margin or padding on the right side.

"rightOfContent"



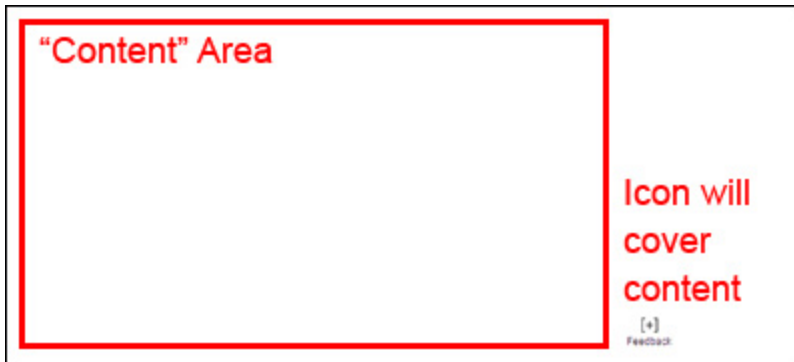
Icon hovers outside the lower-right corner of the content container. This configuration respects the content container and will not render on top of existing content if the window is resized.

"fixedPreserveContent"



Icon hovers in the lower-right corner of the window. This configuration respects the content container and will not render on top of existing content if the window is resized.

"fixedContentMax"



Icon hovers outside the lower-right corner of the content container. This configuration respects the content container **UNTIL** the window is resized to below the width of the container, when it will render the icon on top of the content.

How do I get the Floating [+] Feedback Icon to appear over Flash elements on the page?

By default, Flash elements take precedence over all other elements on the page. If a significant margin or padding exists for the page, the icon can be made to float in that margin as described above. If you wish to have the default floating code and have it float over Flash elements, those elements must be exported in a fashion that allows them to be used with z-indexing. You can then set the z-index accordingly for the OpinionLab code and your Flash elements so that the OpinionLab® code takes precedence. Please see the article at http://kb2.adobe.com/cps/155/tn_15523.html for guidance on exporting Flash elements.

How do I get the Floating [+] Feedback Icon to disappear when clicked but always reappear when the page is refreshed?

In the [configuration file](#), add "disappearOnClick" and set it to "true" to cause the Floating [+] Feedback Icon to disappear after it has been clicked.

```
/* Floating [+] Feedback Icon configuration */
var oo_float = new Oo.Ocode({
    floating : {},
    disappearOnClick: true
});
```

How do I get the Floating [+] Feedback Icon to disappear after it has been clicked on and not reappear for a specific period of time on that page?

In the [configuration file](#), add "disappearOnClick" and set it to "true" to cause the Floating [+] Feedback Icon from disappearing after it has been clicked. Then in the same instantiation, add "cookie -> name" set to the desired cookie name (usually "oo_float"), "cookie -> type" set to "page" (or "domain" to disable it for the entire domain), and "cookie -> expiration" set to whatever time period (in seconds) desired (example: for one hour, set to "3600").

```
/* Floating [+] Feedback Icon configuration */
var oo_float = new Oo.Ocode({
    cookie:
        {name: "oo_float",
         type: "page",
         expiration: "3600"},
    floating : {},
    disappearOnClick: true
});
```

How can I make the Floating [+] Feedback Icon bigger?

You can use an image editing program to adjust the image file you have been given, or you can ask OpinionLab® to supply you with a larger

image. The size of the entire Feedback block (.oo_feedback_float in the oo_style.css file) may need to be adjusted to allow the larger image to fit. OpinionLab® reserves the right not to approve alternate images that do not reflect OpinionLab®'s brand image.

How can I change the color of the Floating [+] Feedback Icon?

You can use an image editing program to change the color of the file you have been given, or you can specify the hex# and ask OpinionLab® to supply you with a customized icon. OpinionLab® reserves the right not to approve alternate images that do not reflect OpinionLab®'s brand image.

How do I change the text of the floating Feedback text?

Change the text in "floating -> caption" to whatever text is desired in the CONF file. OpinionLab® reserves the right not to approve alternate text that does not reflect OpinionLab®'s brand image.

How do I change the background color of the box around the floating Feedback icon?

Indicate the hex# for the "background" color of the ".oo_feedback_float .oo_transparent" element in the oo_style.css file.

How do I change the color and/or size of the floating Feedback text?

Indicate the hex# for the .oo_feedback_float .fbText element in the oo_style.css file.

How do I disable the rollover state of the floating Feedback Icon?

Simply comment out ".oo_feedback_float:hover .olOver" and ".oo_feedback_float:hover .olUp" from the oo_style.css file.

Tab [+] Feedback Icon Implementation



How do I change the look of the Tab?

To change the look of a Tab [+] feedback Icon (color, size, text, etc.) from it's base implementation you must modify one or all of the following:

- The configuration file
- oo_style.css
- oo_tab.png
- oo_tab_icon.gif

The Configuration file

With the appropriate engine file, the configuration file instantiates the tab implementation. The configuration file can be used to place the tab on the left or the right of the page and to change the title text used by screen readers. The tab icon, by default, positions to the right side of the page and has a title of "Feedback". To modify these add the following to the tab call in the configuration file:


```
tab: {
  position: left,
  title: 'My Custom Title'
}
```

oo_style.css

The style sheet controls a lot about the look and feel of the Tab icon. Items that can be effected in the CSS are image references, vertical placement, background color, rounded corners for Webkit and Gecko browsers, tab height, tab width, hover state, and more. All tab elements contain "oo_tab" within their selector name. You'll need to know a little about CSS to do anything extravagant, but here are some basics:

Modify the vertical position of the Tab Icon

Find "#oo_tab" selector and, if you want to keep the Tab Icon positioned based from the top simply modify the attribute "top" to the desired percentage. If you want the Tab Icon position based from the bottom, change the attribute "top" to "bottom" and change the percentage.

Modify the height or width of the Tab Icon

Change the "height" or "width" attributes found in the "#oo_tab" selector.

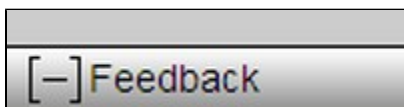
oo_tab.png

This is the image file used to create the background and image text used in the Tab Icon. It is an PNG-8 file using an Alpha Transparency. You will note that gradients are present on either side: this allows the PNG to be used for either right or left positioning in the configuration file. To modify the text or background simply open the file make your updates and save the file in the same format. To effect the text positioning within the tab, modify "#oo_tab a" selector on the oo_style.css.

oo_tab_icon.gif

This is basically the same OpinionLab Icon used in the Floating or inline implementations put on it's side. Modify it as you would those files. To effect it's positioning within the tab, modify "#oo_tab span" selector on the oo_style.css.

Bar [+] Feedback Icon Implementation



How do I change the background color?

The background color is controlled in the CSS by the "#oo_bar" and "#oo_bar a" selectors and the oo_bar.png image. You must effect these selectors in the CSS and this image to update the background color.

How do I change the text?

The text is controlled in the oo_bar.png image and "title" element of the OnlineOpinion® Object. You must effect these this image and the OnlineOpinion® Object in the Conf file to update the tab text.

How do I change the height or width of the bar?

The background color is controlled in the CSS by the "#oo_bar" selector. You must effect this selector in the CSS and this image to update the background color.

Event-Driven Feedback Implementation

What is the difference between user invoked vs. Event-Driven Feedback Implementations

The standard model for the code is **user invoked**. The signature **[+] Feedback Icon Implementation** of the OpinionLab® icon [+] remains in a fixed position relative to the user's browser and is always visible as they scroll up and down the page. Additionally, the **Inline Feedback implementation** of the user invoked code is inserted into a table in the html of a page to appear as a link in a header and/or footer or another fixed location on the page.

Alternatively, the code can be configured to be "**Event-Driven**", meaning a card will render automatically when a user enters or exits a given page, or when they click on a given element on the page. We recommend that Event-Driven Feedback Implementations be used in a limited area of a web site and for a limited period of time to avoid annoying users.

Event driven cards triggered by a user exiting a page can be configured so that the card does NOT render on exit to specified domains or URLs. The code can also be configured to have a domain cookie to prevent the survey from rendering more than once to the same user for the specified period of time. A "referrer rewrite" is typically included in event driven code files to rewrite the referring URL to include a domain prefix such as Event-Driven Feedback or "survey". This allows for two different but simultaneous cards to be activated for the same web page; one for the user-invoked Feedback card and one for the event driven card. It also allows data from the event driven card to be easily segregated in the Toolkit reports.



Known Issues for Event Cards

- Pop-up blockers (Which are commonly enabled in a user's browser) will prevent the On-Entry and On-Exit Pop-up Comment Cards from rendering for a user. On-Entry card may avoid this by using the On-Page Card Component: The On-Page Card component does **NOT** function with On-Exit Events.
- If using On-Exit Event-Driven Code, please contact your Company's Primary Developer Contact. They have received specialized training for dealing with the intricacies and limitations this implementation as described in the "Disclaimer for OnlineOpinion® (Version 5+) On-Exit Event-Driven Feedback Implementation". Additional support for client developers who are not the Primary Developer Contact and/or direct code implementation on the part of OpinionLab® developers will incur additional support charges at an hourly rate outside of the standard support agreement and require a new agreement to be signed before any work can proceed.

How do I make sure that a user won't see the event driven card more than once?

Event driven cards display based on probabilities (onEntry, onSingleClick, onExit): After the card has displayed once based on the set probability, the probability for that event **automatically drops to zero until the cookie (if instantiated) expires**. Additionally, there is no cookie instantiated by default: One must be instantiated by indicating its values in the OnlineOpinion® Object. See the example below:

```
/* Event-Driven Feedback configuration */
var oo_on_entry = new OoO.Ocode({
  cookie:
    {name: "oo_on_exit_rewrite",
     type: "domain",
     expiration: "3600"},
  event :
    {onExit: 100}
});
```

How do I change how long it takes for a user to see a survey card a second time?

In the conf file instantiation, add "cookie -> name" set to the desired cookie name (usually "oo_event" or a variant thereof), "cookie -> type" set to "domain" (or "page" to disable it just for that page rather than the entire domain), and "cookie -> expiration" set to whatever time period (in seconds) desired (example: for five minutes, set to "300").

How do I adjust how many users will see the card?

Adjust the onEntry (for on Entry trigger), onExit (for on Exit trigger) or onSingleClick (for on Click trigger) to reflect the percentage of page loads for which the code will trigger. The percentages are expressed as integers (0 to 100).

How do I use On-Exit Event-Driven Feedback

Enabling

OnlineOpinion® (Version 5+) On-Exit Event Driven Feedback presupposes the following to ensure proper its expected behavior.

Properly Formatted Page Code on the Web Site

OnlineOpinion® (Version 5+) expects its implementation to be on properly formatted HTML pages. This includes, but is not limited to, the following:

- DOCTYPE and HTML Support for Javascript and Cascading Style Sheets (CSS)
- Properly functioning internal or Third-Party JavaScript that has not modified DOM elements
- Properly opened and closed Tags

For more please consult <http://www.w3.org>

Proper Inclusion of and Reference to the OnlineOpinion® Code Set

The placement of the Include Statement that calls the code files should be outside any <DIV> or <TABLE> tags and before just before the </BODY> close body tag is vital. Include Statement elements should also be called in the order specified. Some implementations depend on certain Document Object Model elements to be instantiated **before** they can execute properly. Improper placement (Such as placing of some code elements within a <DIV> or <TABLE> tag) may cause issues with rendering and/or proper functioning of certain implementations.

Proper Configuration of the OnlineOpinion® Code Set

The proper configuration of the OnlineOpinion® code is outlined in a descriptive "Readme" PDF distributed along with the code set at the time of code delivery. This PDF can be requested in advance at any time from your OpinionLab® Client Support or Sales representative. It is OpinionLab®'s expectation that the client's Primary Developer Contact will review this PDF to ensure proper understanding of how the code functions.

Most configurable elements of the code set are done by OpinionLab® prior to providing the code to the client. OpinionLab® then provides direction to the client's Primary Developer Contact as to the elements that can/should change from the client's Quality Assurance and/or User Acceptance Testing process to the live installation.

OpinionLab® will provide guidance as to the use of regular expression (Regex) patterns designed to disable the On-Exit trigger for links on the page as indicated below. It is the client's Primary Developer Contact that will be responsible for determining:

- **What links should be disabled.**
- **Patterns in the links that can be used in a Regex to control them.**
- **Entering and testing the regular expression (Regex) patterns to ensure they adequately control the On Exit trigger.**
- **Ensuring that any new links added to the pages after the release of the OpinionLab® code to the site match the regular expression (Regex) pattern previously entered OR modifying the regular expression (Regex) pattern accordingly.**

No Support for Unauthorized Customizations

OpinionLab® provides support before, during and after deployment of the OnlineOpinion® event code for the standard code files OpinionLab® has provided. OpinionLab® strongly advises against changing any functions in the engine or configuration files. If the client makes changes to the code provided, or if the client implements code dependencies that impact the core functions of the OnlineOpinion® code, OpinionLab® cannot warranty or support the resulting, modified, code without charging additional fees.

It is the client's understanding that should their Primary Developer Contact and IT team implement changes directly to the code files, or implement custom or third-party code that interacts with the OnlineOpinion® code, the client's Primary Developer Contact and IT team is responsible for understanding the potential impact on the OnlineOpinion® code. The client's Primary Developer Contact and IT team must make the choice between reverting to the original code for OpinionLab® support or resolving their own code issues.

Disabling

Clients will likely want to disable On-Exit Event-Driven Feedback for certain links on their pages to other pages within their site or a specific path (e.g. a user clicking a link in a checkout path). The following is a list of methods for interrupting the expected behavior that triggers on certain exit events.

For HTML Links and Forms: Properly Formatted Regular Expression Applied to the Activated Disable Links Feature

When using a card that pops upon page exit you can specify <a> tag based links that will not activate the card using a regular expression (Regex). Any links that return a match for the regular expression will not trigger a Dynamic Comment Card. The client's Primary Developer Contact is responsible for determining what links require a match and entering the specific regular expression (Regex) patten that apply to such links. OpinionLab® can provide guidance as necessary.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Event-Driven Feedback configuration */
var oo_on_exit = new OOO.Ocode({
  cookie:
    {name: "oo_exit",
      type: "domain",
      expiration: "3600"},
  event :
    {onExit: 100,
      disableLinks: /^(https:\\/\\/www\\.opinionlab\\.com)/i}
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

For JavaScript Driven Links: Apply Interrupt Show Feature Script On-Click

Other links and buttons on the page that do not fall into the disableLinks or disableFormElements definition should use the following script in the HTML of the page to prevent them from causing an Event Card to render on exit from a page:

Interrupt Show Script

```
onclick="oo_exit.interruptShow = true;"
```

Or, it can be added to another DOM Element:

Interrupt Show Script

```
myDomElement.onclick = function(e) {
  //code that does whatever is specified
  oo_exit.interruptShow = true;
}
```

The client's Primary Developer Contact is responsible for determining what links or functions require these scripts and implementing them in their pages.

For Form Elements: Activate Disable Form Elements Feature

If disableFormElements is set to "true", no form elements will trigger card on page exit.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Event-Driven Feedback configuration */
var oo_on_exit = new OOO.Ocode({
  cookie:
    {name: "oo_exit",
      type: "domain",
      expiration: "3600"},
  event :
    {onExit: 100,
      disableFormElements: true}
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

For Browser Close or Browser Actions: Cannot be Disabled

See "Known Issues" in this document.

How do I prevent an On-Exit Event from rendering when a user clicks on some links on my web pages?

Use the "disableLinks" section of the CONF file to enter "friendly" domains, paths or URLs to which the user may exit but where you do NOT want the card to trigger. OpinionLab® may provide assistance with RegEx patterns to apply wildcard elements and avoid individual URL entry (additional charges may apply), however it is the clients responsibility to determine the regular expression pattern to be employed. This ONLY effects <a> tag links: JavaScript or other methods WILL NOT be effected by "disableLinks".

```
disableLinks: /^(.*domain.*)/i
```

```
disableLinks: /^(https:\\\\www\.domain\.com)/i
```

```
disableLinks: /^(http:\\\\www\.domain1\.com\/cart\.jsp|http:\\\\www\.domain2\.com)/i
```

Additionally you can disable the JavaScriptInstance by applying the following code to an onclick an referring to the proper instance name (in this case "oo_instance"):

```
oo_instance.interruptShow = true;
```

For example:

```
<a href="http://www.opinionlab.com" onclick="oo_instance.interruptShow = true;">Link to OpinionLab</a>
```

OR:

```
<button type="submit" onclick="oo_instance.interruptShow = true;">Search</button>
```

Or, if you added an onclick event to some DOM Element:

```
myDomElement.onclick = function(e) {
    //code that does whatever is specified
    oo_instance.interruptShow = true;
}
```

Does "disableLinks" work for page anchors too?

Yes. As long as the clickable item is a link element, then "disableLinks" will effect it.

How do I stop the card from rendering when a user hits the "Back" button?

If you have an event driven card that is triggered on exit from a page, **there is no way to prevent the card from rendering when a user exits a page via the Back or Refresh buttons.** The survey will render only once for a user, and then a cookie will be set that will prevent it from rendering on subsequent use of the Back or Refresh buttons. An alternative is to use an "abandonment" survey, which will render when a user exits a defined tunnel (but not on browser exit).

How do I implement another event driven card somewhere else on my site?

Copy the original CONF file and rename it. Change the OnlineOpinion® Object name so that it is unique. On the pages where you want the new event driven card to render, use the original include statement, updating the CONF file name to the new name. Within the CONF file, rename the cookie so it is unique, and edit any elements such as the cookie expiration, domain rewrite, survey probability percentage (onEntry, onSingleClick, onExit) and disableLinks according to the new requirements.

How do I place an on click event feedback link?

Simply apply an onclick Javascript call to the desired element or elements on your page:

```
onclick="oo_on_click.show()"
```

We recommend using the oo_icon.gif animated GIF in conjunction with the word "Feedback" for this implementation. OpinionLab® reserves the right not to approve alternate images and text that do not reflect OpinionLab®'s brand image.

How do I make sure that a user who has seen one of my first event driven surveys doesn't see one of the later surveys?

In the new CONF files corresponding to the new event driven surveys, use the same cookie name as in the original file. Then the cookie will apply across your implementations for the duration of the expiration period of the cookie.

How do I delay an on entry card from rendering until after the page completely loads?

Enter the number of seconds you want the card to be delayed for the onEntryDelay variable in the CONF file.

How can I get the survey to bypass pop-up blockers?

If you are using a pop-up card that is Event-Driven Feedback, it will be blocked by enabled pop-up blocker settings for On-Entry and On-Exit events. If you are using an On-Entry or On-Click trigger, you can combine this with an On-Page Card configuration to bypass pop-up blockers, but you cannot use this approach for on Exit triggers, because the user has left the page in question.

Invitation Feedback Implementation



What is the OnlineOpinion® Invitation Feedback?

The OnlineOpinion® Invitation Feedback is designed to be an "intercept" card that solicits user feedback after that user has finished their visit to a site or has performed a given task on the site. It uses an on-page prompt that invites users to opt into a survey based on predefined parameters, such as a percentage of users to visit a page, the path a user follows on a site or an area of the site visited by the user.

How is Invitation Feedback different from Standard and Event-Driven Feedback Code?

The standard OnlineOpinion® icon is typically installed site-wide to enable a user-invoked Comment Card. At any time, users can select the signature Floating [+] Feedback Icon or inline Feedback link to rate pages, offer their comments and answer optional questions. An Event-Driven Feedback version of the code is also available as an "intercept" card to automatically render a Comment Card on entry to a page, exit from a page, or on click of a specified element.

Both the Event-Driven Feedback and the Invitation Feedback implementations provide an "intercept" card that renders to users meeting certain conditions. However, the OnlineOpinion® Invitation Feedback has several important differentiating features:

Avoids Pop-up Blockers

Standard Event-Driven Feedback code, when used as a pop-up card, is blocked by many pop-up blocker services. By providing an opt-in prompt that renders within a page, Invitation Feedback allows users to actively opt in or out of the survey. Because users select whether to take the survey, the survey is not blocked by standard pop-up blockers.&

True End-of-visit Survey

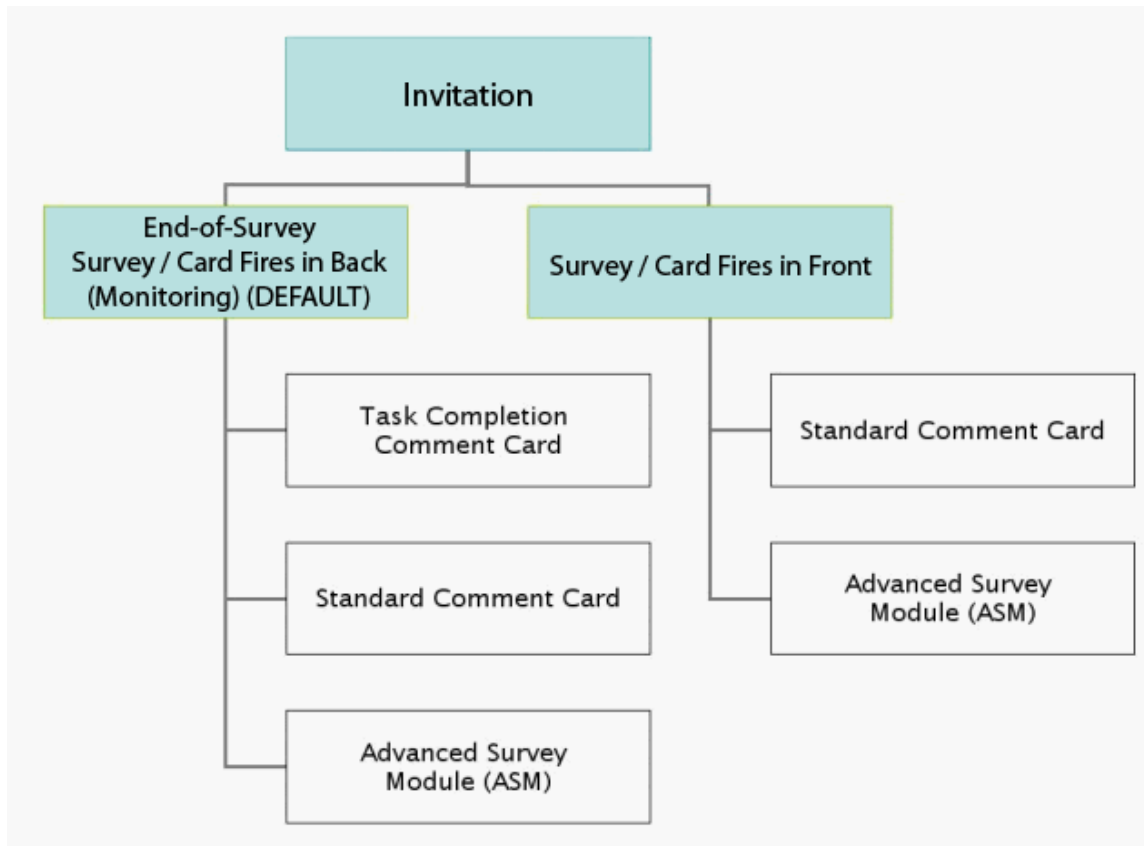
Invitation Feedback allows you to specify whether the survey should render immediately or if it should render behind the main browser until the user leaves a site. For a survey that renders immediately, the questions can be specific to the area of the site where the survey is enabled or a specific task that was completed. For a survey that renders in the background, questions can be geared to the overall site and user sentiment on exit.

Less Intrusive

Invitation Feedback uses an opt-in prompt so that the user can decide whether or not they want to take an intercept survey. A cookie is placed on the user's browser once the prompt renders so that a user is not asked to answer the survey more than once in the specified period of time.

When should Invitation Feedback be used?

Depending on the purpose, Invitation Feedback should (or should not) be used in different ways. additionally, different comment Card/ASM implementation can be used with Invitation Feedback in different situations.



Keep in mind that some surveys are more useful **after a process has been completed**. Additionally, longer surveys are more likely to be filled out when they **do not interrupt an immediate process**.

Can you explain some example scenarios for Invitation Feedback?

Home Page Survey

A user arrives at the home page of your site. You can prompt them several seconds after they arrive, allowing them to take an initial look at your home page design, before you solicit feedback on the homepage itself. The survey can render immediately before the user navigates to the rest of the site.

Note that if you are trying to exclusively survey home page "bouncers", or those who leave your domain after visiting the homepage, an Event-Driven Feedback Comment Card (EDCC) may be more suitable. An EDCC will allow you to render a survey on exit from the homepage when the user is not navigating to a "friendly" URL or domain that you have specified (in other words, the rest of your site).

Site Experience Survey

A user arrives to your site, either to the home page or another bookmarked/linked page. The user receives a prompt to take a survey when they have finished navigating the site. Once the user leaves the site, the survey comes forward for them to provide their feedback on their overall experience.

Support Area Assessment

A user is navigating your site and goes to your "Support" area to find articles that will assist in troubleshooting your site or products/services. They receive a prompt when they first arrive to the Support area asking them to provide feedback once they have finished. The user searches for articles, reads various resulting information, and navigates back to another area of your site. The survey renders when the user has left the Support area.

Bill Payment Task Feedback

A user arrives at your Bill Payment page. They enter their payment information and submit the payment. On arriving at a confirmation page, they

see the prompt render. If the user responds "yes" to the prompt, the survey can render immediately, or it can render when the user leaves the site. Note that this will only capture feedback of those who have completed the task, in this case, a Bill Payment.

Alternatively, you could have the prompt appear at the opening page for the Bill Payment process so that you capture feedback from those who do complete the process but also those who do not. Note that the standard invitation code will render the Comment Card based on where the prompt has rendered, in this case at the beginning of the Bill Payment process, so it means that both those who complete and those who abandon the process will see the same Comment Card. If you wish to exclusively survey those who do not complete a task or process ("task abandonment") an Event-Driven Feedback Comment Card may be more suitable.

Checkout/Shopping Cart Abandonment

A user places items in their shopping cart and proceeds to the checkout area. An Invitation Feedback prompt appears at the beginning of the checkout process. The user either completes their purchase or abandons the checkout area before completing their purchase. The survey can come forward when they leave the checkout area, domain/sub-domain or the overall site. Note that this will capture feedback from both those who do and do not complete a purchase using the same Comment Card. If you want to exclusively survey those who do not complete a purchase, an Event-Driven Feedback Comment Card may be more suitable.

How do I use Invitation Feedback

Enabling

OnlineOpinion® (Version 5.1+) Invitation Feedback Implementation presupposes the following to ensure proper its expected behavior.

Properly Formatted Page Code on the Web Site

OnlineOpinion® (Version 5.1+) expects its implementation to be on properly formatted HTML pages. This includes, but is not limited to, the following:

- DOCTYPE and HTML Support for Javascript and Cascading Style Sheets (CSS)
- Properly functioning internal or Third-Party JavaScript that has not modified DOM elements
- Properly opened and closed Tags
- Properly referenced code set files per instruction found in the User Manual



To avoid cross-domain issues the path to the assets directory in the CONF file must be on the same domain as the page that has Invitation Feedback include code applied to it.



Using a Consistent Engine File

OpinionLab requires that a consistent OnlineOpinion engine file be installed globally, on all pages, across any given domain for proper functioning of the Invitation implementation. Using the same file would be best, but at a minimum, the engine should use the same version across the domain. Doing so will avoid issues with friendly domains, pages hit, safe areas and other aspects of the Invitation implementation.

For more please consult <http://www.w3.org>

Proper Inclusion of and Reference to the OnlineOpinion® Code Set

The placement of the Include Statement that calls the code files should be outside any <DIV> or <TABLE> tags and before just before the </BODY> close body tag is vital. Include Statement elements should also be called in the order specified. Some implementations depend on certain Document Object Model elements to be instantiated **before** they can execute properly. Improper placement (Such as placing of some code elements within a <DIV> or <TABLE> tag) may cause issues with rendering and/or proper functioning of certain implementations.

Proper Configuration of the OnlineOpinion® Code Set

The proper configuration of the OnlineOpinion® code is outlined in a descriptive User Manual PDF distributed along with the code set at the time of code delivery. This PDF can be requested in advance at any time from your OpinionLab® Client Support or Sales representative. It is OpinionLab®'s expectation that the client's Primary Developer Contact will review this PDF to ensure proper understanding of how the code functions.

Most configurable elements of the code set are done by OpinionLab® prior to providing the code to the client. OpinionLab® then provides direction to the client's Primary Developer Contact as to the elements that can/should change from the client's Quality Assurance and/or User Acceptance Testing process to the live installation.

OpinionLab® will provide guidance as to the use of regular expression (Regex) patterns designed to disable the Invitation Feedback trigger for links on the page as indicated below. It is the client's Primary Developer Contact that will be responsible for determining:

- Identifying URL patterns used to inhibit the prompt (if any).
- Identifying site paths used to inhibit the prompt (if any).
- Identifying domains should be identified as "Friendly" (if any).
- Patterns in the prompt triggers and/or Friendly Domains that can be used in a RegEx to control them.
- Entering and testing the regular expression (RegEx) patterns or paths to ensure they adequately control the prompt trigger.
- Ensuring that any new pages or Friendly Domains added to the site after the release of the OpinionLab® code to the site match the regular expression (RegEx) pattern or path previously entered OR modifying the regular expression (RegEx) pattern or path accordingly.
- Ensuring that appropriate controlling code is installed properly on all relevant sites.

No Support for Unauthorized Customizations

OpinionLab® provides support before, during and after deployment of the OnlineOpinion® event code for the standard code files OpinionLab® has provided. OpinionLab® strongly advises against changing any functions in the engine or configuration files. If the client makes changes to the code provided, or if the client implements code dependencies that impact the core functions of the OnlineOpinion® code, OpinionLab® cannot warranty or support the resulting, modified, code without charging additional fees.

It is the client's understanding that should their Primary Developer Contact and IT team implement changes directly to the code files, or implement custom or third-party code that interacts with the OnlineOpinion® code, the client's Primary Developer Contact and IT team is responsible for understanding the potential impact on the OnlineOpinion® code. The client's Primary Developer Contact and IT team must make the choice between reverting to the original code for OpinionLab® support or resolving their own code issues.

Disabling

Clients will likely want to disable Invitation Feedback for certain links on their pages to other pages within their site or a specific path (e.g. a user clicking a link in a checkout path). The following is a list of methods for interrupting the expected behavior.

Inhibiting the prompt

URL Pattern Based

In this instance, the prompt renders when the URL that a user visits matches a regular expression (RegEx) defined in the code. For example, you could configure the prompt to render to users who visit a page containing "/product" or "/support" in the URL. This is controlled by the "promptTrigger" element.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOo.Invitation({
/* REQUIRED fields */
  pathToAssets: '/onlineopinionV5/'

/* Optional fields */
  , promptTrigger: /products/
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

Path/Tunnel Based

A Path/Tunnel Configuration allows you to specify a number of steps that a user has to take in order for any of the code to run. For example, you can configure an Invitation Feedback to render to those who follow a 3-page bill payment process in sequence. This is controlled by the "promptTrigger" element and is incompatible with the Conforming Page Hit feature.

Associated Cookie

Note that this cookie is used in combination with a "promptTrigger" value (series of URLs) for the promptTrigger field above. The cookie is set on the first page where the user encounters the installed code. It is re-set if the user enters the tunnel but does not follow the URL path sequentially.

Why won't my tunnel path work?

There could be several reasons.

1. You have not cleared your browser cache after your updates.
2. You have a specialized engine that does not support tunnel path (Tiny or Specialized Instance Condensed).
3. Your tunnel path are not configured correctly.
4. You have conflicting JavaScript code affecting your OnlineOpinion® instance.

If none of these items are the cause of your Custom Variables not working, please contact OpinionLab® for additional support

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOo.Invitation({
/* REQUIRED fields */
  pathToAssets: '/onlineopinionV5/'

/* Optional fields */
  , promptTrigger: ['http://www.mystore.com/example/step1.html',
'http://www.mystore.com/example/step2.html',
'http://www.mystore.com/example/step3.html'
]

});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

Conforming Page Hit

The prompt renders when the browser has hit and/or refreshed a conforming URL the amount of times specified. . This is controlled by the "pagesHit" element and is incompatible with the Path/Tunnel Based feature.

Use of "pagesHit" Within a Section or Site Wide

For the pagesHit, the code needs to be installed site-wide or throughout a particular section to properly fire. The number of pages hit can only be incremented when the code is on a particular page (no code, no incrementation). If '3' is specified as the pages that need to be hit, then when a user gets to the third page the rest of the code will execute.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOo.Invitation({
/* REQUIRED fields */
  pathToAssets: '/onlineopinionV5/'

/* Optional fields */
  , pagesHit: 3

});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

Browser based

If you have a need to disable the invitation prompt for any given browser, you can add the following conditional statement to check for and

eliminate specific browser types:

```
if (navigator.userAgent.indexOf('Firefox') == -1) {
    // Include invitation configuration for any browser besides Firefox here
}
else {
    for (var x = 1; x <= 3; x++) {
        if (OOo.Browser.Gecko && navigator.userAgent.indexOf('Firefox/'+x) != -1) {
            // Include invitation configuration for Firefox versions previous to 4 here.
        }
        // This code disables invitation for any Firefox version 4+.
    }
}
```

Inhibiting the Card

Closing the card without bringing it to the front

In certain cases it is desirable to close the comment card without bringing it to the front upon the user reaching an appropriate "Goal" page. To specify a "Goal" page use the "goal" parameter: This will close the monitoring window upon reaching a conforming match URL.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OOo.Invitation({
/* REQUIRED fields */
    pathToAssets: '/onlineopinionV5/'

/* Optional fields */
    , goal: "completion.html"
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

Preventing a card from coming to the front for only a section of your domain

There are cases where it is desirable for a card to come to the front before a user leaves the domain or closes their browser. In these cases use a regular expression (RegEx) statement with the "area" parameter to identify the section(s): non conforming pages will cause the Monitoring window to come to the front and trigger the specified card.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OoO.Invitation({
/* REQUIRED fields */
  pathToAssets: '/onlineopinionV5/'

/* Optional fields */
  , area: /^(sectionRegEx)/i
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

The "Popunder" Monitoring Window and Cross-Domain Implementations

OpinionLab® has developed certain features to help address cross-domain issues, however the Invitation Feedback's "popunder" monitoring window configuration is not specifically designed to function in a cross-domain environment due to the security issues inherent in this type of implementation and its function to gather information specifically regarding the current website. The level of complexity to implement increases **dramatically** when these features are engaged and may incur additional support charges. Clients wishing to utilize the Invitation Feedback across multiple domains and/or sub-domains may encounter rendering issues. Special considerations include, but are not limited to:

- Full list of identified "Friendly Domains" that will be affected
- Domain Cookies
- Specially Configured OnlineOpinion® (Version 5.1+) Invitation Feedback Code to be places on landing pages of Friendly Domains

Code must also be on Friendly Domains

The Invitation Feedback code must be installed on the Friendly Domain (and refer to the same file location) in order for this value to keep the survey from rendering until the user also leaves this domain. Just take the exact same object from one domain to the other: don't change any of the settings, otherwise it might result in unexpected behavior.

Prompt Inhibiting Cookies Do Not Cross Over to Friendly Domains

Cookies cannot be used across domains or sub-domains, so for example, if the code is installed on one domain and also on another sub-domain, users may see a prompt on each.

Example Code Configuration

```
/* OnlineOpinion v5.x.x - Released: xx/xx/xxx - Components: Full - The following code is Copyright
1998-2010 OpinionLab, Inc. All rights reserved. Unauthorized use is prohibited. This product and other
products of OpinionLab, Inc. are protected by U.S. Patent No. 6606581, 6421724, 6785717 B1 and other
patents pending. http://www.opinionlab.com */

/* Run the Invitation Feedback instance */
new OoO.Invitation({
/* REQUIRED fields */
  pathToAssets: '/onlineopinionV5/'

/* Optional fields */
  , friendlyDomains: ['http://www.friendlydomain1.com', 'http://extranet.friendlydomain2.com']
});

/* OnlineOpinion v5.x.x, Copyright 2010 OpinionLab, Inc. */
```

How do you configure Invitation Feedback?

After consulting with clients to determine their needs, OpinionLab® provides a code package that has been configured to address a particular intercept scenario.

The main configuration options control when/where the initial user **prompt** renders, when the **survey** itself renders, what **type of survey** appears, and how the **cookie** prevents the user from being prompted on a repeated basis.

Prompt Options and Functionality

Prompt Timing

A prompt can be triggered in one of 3 ways:

1. **Percentage (Default)** - In this instance, the prompt renders based on a defined percentage of visitors to a page where the code is installed. For example, you can configure the prompt to render to 5% of those who visit your homepage.
2. **Tunnel/Path** - In this instance, the prompt renders based on a percentage of users who have followed a specific path of sequential pages. For example, you can configure the prompt to render to 20% of those who follow a 3-page bill payment process in sequence. this Feature is incompatible with the Conforming Page Hit feature.
3. **Regular Expression (RegExp) Match** - In this instance, the prompt renders when the URL that a user visits matches a "regular expression" defined in the code. For example, you could configure the prompt to render to 75% of users who visit a page containing "/product" or "/support" in the URL.
4. **Conforming Page Hit** - In this instance, the prompt renders when the browser has hit and/or refreshed a conforming URL the amount of times specified. This feature is incompatible with the Tunnel/Path feature.



For any of the prompt options above, the specified percentage of users who visit the defined pages will receive the on-page prompt. If no percentage is defined, a default value of 50% is used. Those users who have received the prompt previously and who still have the associated cookie will not see the prompt (see cookie description below).

The prompt can also be delayed so that users have a chance to view the page they are loading before the survey invitation appears.

Once the prompt appears, if the user selects the "no thanks" button or the close button ("x" at the top of the prompt), the prompt will disappear and the user will not receive a survey.

Invitation Feedback triggers on a wildcard for any subdomain of a domain (any subdomain) of opinionlab.com

```
new OOo.Invitation({
  pathToAssets: <depends on what domain the code is on>,
  responseRate: 100,
  promptTrigger: '\\\\[OnlineOpinion@ Version 5.1 Instructions^.]\\.\.opinionlab\\.com/'
})
```

Invitation Feedback triggers on a wildcard for any variation on domain (anything containing 'opinionlab' in the domain)

```
OOo.Invitation({
  pathToAssets: <depends on what domain the code is on>,
  responseRate: 100,
  promptTrigger: /opinionlab/
})
```

Invitation Feedback triggers on a wildcard for a domain (or sub-domain) and a specific subdirectory (e.g. example.opinionlab.go.com/section/area*)

```
new OOo.Invitation({
  pathToAssets: <depends on what domain the code is on>,
  responseRate: 100,
  promptTrigger: /example\\.opinionlab\\.go\\.com\\/section\\/area*/
});
```

Prompt Cookie

The Invitation Feedback code includes a cookie that prevents the prompt from reappearing to an individual user who has previously seen the prompt render. The cookie can be set to expire after the preferred number of days, or the cookie can be set to expire immediately, meaning that the same user may see the prompt again (depending on the percentage option selected for the prompt) upon their next visit to the page in question. Of course, the cookie will only prevent the prompt from reappearing to a user if the user has not subsequently cleared their cookies from their browser.

An additional prompt option that you can make available to users is a "Never show me this prompt again" button, which when clicked, will set a cookie that will last 5 years. As long as the user does not clear their cookies, they will not see the prompt during this time period.

Legacy Variables - available for v5.3.3 +

When configuring Legacy Variables with the Invitation code set, position 1 will be reserved for Tealeaf by default and any additional legacy variables you pass will be appended starting at position 2. However, if ClickTale is being used, it will occupy position 2 and any additional legacy variables will append starting at position 3. To append Legacy Variables to position 2 or 3 (dependent on ClickTale), you can place the following configuration option within the Invitation instance:

Legacy Variables (no override of Tealeaf or ClickTale)

```
legacyVariables: {
  vars: 'myValue1|myValue2'
}
```

Additionally, you can add an override option that will entirely replace the custom_var value with any string you define with the "vars" parameter. Add the following boolean value:

Legacy Variables (override of Tealeaf and ClickTale)

```
legacyVariables: {
  vars: 'myValue1|myValue2'
  , override: true
}
```

Prompt Style and Language

The file that contains the markup for the prompt is called oo_inv_prompt.html. It contains non-semantic html elements with inline styles. The reason for this is that if the prompt were to use semantic markup, the host website would overwrite several styles that the prompt would use, making it very difficult to make the prompt appear as desired on various websites.

The default text/language has been selected to take in account browser height and brevity in order to prompt the user to accept the Dynamic Comment Card or Advanced Survey Module. This text can be modified, but please take into account modern browser heights and make sure that special character text is presented in Unicode format. It is STRONGLY encouraged to keep the prompt text as brief as possible (under 140 characters in total length) and to take variable browser heights into account (as this may effect the usability of the prompt for shorter browser windows).



Alternate Languages

Opinionlab® recommends leaving the default language and styles in the markup file and related css file. Because of the interrelated elements that make up the prompt, any edits made to these files may cause errors that make the prompt unclear or inoperable. OpinionLab® is not responsible and will not troubleshoot markup and/or css files that have been changed by clients. Opinionlab® may perform modifications at the request of the client and at an additional cost.

For multiple language implementations, OpinionLab® will configure a unique markup file for each language, based on the translation provided by the client. Translations should be provided in a text (.txt) file, in Unicode format.

You can add your company logo to the top of the prompt by adding a path to the companyLogo field in the oo_conf.js file. No changes to the markup file itself are necessary.



Non-inline Style Selectors

There are some styles that can't be passed inline or are easier to define externally. These styles are defined via sass (<http://sass-lang.com/>) which is then compiled to the minified css.

Card Options and Functionality

Once the user answers "yes" to the invitation to participate in a survey, the timing of the survey, as well as its content, can be controlled in the ways described below.

Survey Timing

1. **(Monitoring) Card will render behind the current window and come to front on domain exit** (Default) - In this instance, a loading page will render in a new window behind the current window, so that the user can continue using the site. The loading page monitors the user activity so that when the user leaves the current domain, closes the browser or selects the loading page window, it will come to the front to load the specific comment card.
2. **Card will render in front of the current window** - In this instance, the card will render immediately in front of the current window, similar to an Event-Driven Feedback card. If a user closes their browser before the prompt appears (for example with a delayed prompt), no survey will appear.

Survey Content

When you use the Invitation Feedback code, the survey that launches inside the window can follow one of the 3 formats below. Once your Invitation Feedback code is installed, you do not have to make code changes to change the survey that launches inside the window. Instead, the survey is activated against a domain and/or specific URL via the OnlineOpinion® Toolkit.

1. **Task Completion Card:** This card is similar to a standard OnlineOpinion® Comment Card, but has been standardized to benchmark data across industry verticals and the OpinionLab® client base. These will eventually allow clients to compare themselves to others in their vertical (anonymously) who are asking the same questions. The Task Completion answer list can be customized for each Comment Card.
2. **Standard Comment Card:** The OnlineOpinion® Comment Card contains standard Content, Design, Usability and Overall Page rating scales, as well as a standard Comments box. Various templates are available with different layouts. Alerts can be triggered based on the standard rating scales, comments entered or if the user has entered their email address via an optional email entry field.
3. **Advanced Survey Module (ASM):** The Advanced Survey Module creates surveys that typically contain more questions than can be contained by the one-page Comment Card. Conditional Branching and Piped-in Responses are also available in surveys created in the Advanced Survey Module. When you launch an ASM survey with your Invitation Feedback code, this should be indicated in the CONF file so that the window that opens is taller and accommodates the size of a typical ASM survey.

Notes on Comment Card Options

When deciding which type of Comment Card Option (see Survey Content above) should be used, you should take into account the user scenario you are intercepting, the type of feedback you wish to gather and the type of reports and alerts you will rely on to analyze your feedback.

Keep in mind that the Task Completion Comment Card is concise and asks standardized questions that will eventually allow you to benchmark your user responses with those of others in your industry. Because this is based on a standard Comment Card, you will also be able to trigger a monitor based on an NPS score, include a Tealeaf Session ID in an alert (where you have done a Tealeaf integration), and trigger alerts based on keywords used (if you have added an open Comments field). This card is ideal for "Task Completion" and "Site Experience" surveys.

The standard Comment Card offers flexibility in terms of card layout and customized language. All standard reports, including the O-Trend report and Business Intelligence Reports, as well as alerts, work well with these cards, especially when the standard "Comments" box is used. The standard Content, Design, Usability and Overall Page Ratings may provide valuable information to you, but may not be appropriate where you are

soliciting feedback on a multiple page or site experience.

The Advanced Survey Module (ASM) offers functionality like Conditional Branching, which may be useful when you would like to segment your audience and ask a different set of questions according to initial user responses. The ASM also allows you to filter your reports based on user responses, again allowing for audience segmentation. However, ASM surveys are not integrated with other OpinionLab® reports such as the Business Intelligence Report or with the Export Tool, Keyword Search or Alerts.

Cookie Configuration

Can OnlineOpinion® cookies cross domains?

Yes, to a point. OnlineOpinion® cookies can cross sub domains (For instance the cookie for an event on www.opinionlab.com would prevent that event with the same cookie name from occurring on blog.opinionlab.com). To prevent this simply name each cookie instance uniquely from sub-domain to sub-domain. Cookies cannot cross full domains (For example www.opinionlab.com to www.dialogcentral.com).

On-Page Card Configuration

The screenshot shows a web-based configuration interface for an 'On-Page Card'. The card has a header with the OpinionLab logo. Below the header, there are two main sections: 'Page Comments' and 'Page Ratings'. The 'Page Comments' section includes a dropdown menu to 'Choose a topic for your comments...', a text input field for 'Please enter your comments about this page. (1000 character limit)', and a link for customer service issues. The 'Page Ratings' section features a table with five rows: 'Content', 'Design', 'Usability', and 'Overall', each with five radio buttons for rating from '--' to '++'. Below these sections is an 'Email address' field (not required) and an 'Optional Questions' section with a question 'Would you recommend our products to others?' and 'Yes/No' radio buttons. The footer contains the OpinionLab logo, links to 'Privacy Policy' and 'About this system', a copyright notice, and a 'Submit' button.

What is the difference between a Standard Pop up vs. On-Page Card

The standard Comment Card renders as a small "pop-up" browser window that appears over the user's main browser window. The advantage to this implementation is that the code will not interfere with the functioning of the core code of the page.

The code can also be configured to render an "on-page" card so that the card appears within a dynamically rendered iframe on the page. The advantage to using one of these cards is that it will not be blocked by standard pop-up blockers. However, because the code generates the card in a dynamically generated iframe, there is always a possibility of conflict with other elements on the page, and should be tested thoroughly by the client.

Additionally, please keep in mind that the On-Page Card is element within a page. No "on exit" event is possible with an On-Page Card. Users with windows that are not maximized may have to resize their browsers when an On-Page Card is used.

On-Page Card can be used with any implementation type: Floating, Inline or Event. The variables and elements On-Page Card configuration will

be contained in the conf and CSS file for the implementation.



Intelligent Support links within an On-Page card

If presented with an Intelligent Support link within an On-Page card, the link will open in a new browser window.



Thank You Page Auto-close does not work with On-Page Cards

Comment Cards utilizing thank you pages which are set up in the toolkit to close after a certain period of time will remain open until the user takes action to close the instance when using the On-Page Card implementation.



Flash Z-index Layering and On-Page Card

Flash elements will take precedence over the On-Page Card, interfering with its visibility, unless the Flash elements are configured to use z-indexes to specify element layering. See article at http://kb2.adobe.com/cps/155/tn_15523.html for more details on using Flash elements and z-indexes.



Taller Cards May Not Show Submit Buttons with On-Page Card

Browsers set to smaller resolutions (below 1024x768px) and/or having multiple toolbars in their browser chrome may encounter some issues with taller comment cards. Taller cards may render with their submit buttons below the bottom of the browser window with an additional scroll bar to allow users to reach the submit button, but inhibiting feedback submission. To avoid this issue, try to keep On-Page Cards small. Contact OpinionLab® to set up optional area scrolling within your comment cards or to make other recommendations.

How do I change from a pop up card to an On-Page Card?

If, in your CONF file, add "onPage" to the desired instantiations. You have a specialized engine that does not On-Page Card (Tiny or Specialized Instance Condensed), you must request an On-Page Card implementation of the CONF file from OpinionLab® which may incur additional cost.

How do I get the On-Page Card to appear over the Flash elements on my page?

You must use the z-index in the style CSS to set the ordering of elements on the web page so that the On-Page elements take precedence over the Flash (and other) elements. See the article at http://kb2.adobe.com/cps/155/tn_15523.html for information on setting the z-index for Flash elements.

How do I change the background overlay color for the On-Page Card?

You can update the "oo_overlay" selector in the oo_style.css.

How do I close an On-Page Card?

The best way is to fill out and submit the card. Another way is to the close button located at the upper right of the card. Finally the On-Page Card implementation can be configured to close by clicking on the overlay background layer in your CONF file, add "onPage" to the desired instantiations by setting "onPage -> closeWithOverlay" to "true".

How do I show a card is loading?

A graphic named oo_loading is automatically set up to display in the overlay while the card is loading.

Custom Variable Configuration

What are Custom Variables?

Custom Variables are generic fields in the OpinionLab® database that are available to store additional information about users, sessions, pages

or products that may be relevant to a user's ratings and comments on a given page.

```
customVariables :
  {clientId: 1234, countryCode: 'usa'}
```

How do I add Custom Variables?

Uncomment the sample Custom Variable lines (and replicate if additional fields are to be used) in the CONF file. Add the name of the variable that contains the values that should populate a given custom variable field. For example,

```
var oo_instance = new OOo.Ocode({
customVariables :
  {page: s.pagename,
  page_location: window.location.href,
  title: document.title}
});
```

Values must be available as variables in the html of the pages themselves or dynamic languages such as ASP can be used to retrieve values:

```
customVariables :
  {clientId: <%=clientId%>}
```

Why won't my Custom Variables work?

There could be several reasons.

1. You have not cleared your browser cache after your updates.
2. You have a specialized engine that does not support Custom Variables (Tiny or Specialized Instance Condensed).
3. Your Custom Variables are not configured correctly.
4. The items referenced by your Custom Variables are not instantiated correctly.
5. You have conflicting JavaScript code affecting your OnlineOpinion® instance.

If none of these items are the cause of your Custom Variables not working, please contact OpinionLab® for additional support.

What is a "legacy custom variable" and how is it used?

A "legacy custom variable" is a custom variable that was previously used in V4 code and now captured in V5 code/reporting. In order to capture this variable, "legacyVariables" is used in the configuration file. As many as eight legacy custom variables may be used, slotted from positions two through nine (position one is reserved for Tealeaf and Clicktale, position ten is reserved for "iframe").

Referrer Rewrite Configuration

What are domain/URL referrer rewrites? Why are they used?

The default code uses the URL of the page to serve the correct comment card and report on associated data. In some cases, the URLs are not meaningful for report purposes, or they contain extraneous information that make card activation and reports inefficient. In these cases, a URL rewrite can be included in the CONF file to change the URL to something more meaningful (for example with a page name appended to the domain) or efficient (stripping out parameters that are not relevant to card activations or reports). For event driven cards, a domain rewrite typically adds a prefix such as Event-Driven Feedback to distinguish calls for the event card from calls for the regular Feedback card. It also serves to segregate event data.

Most typical domain/URL rewrite variations

Prepend the domain - Add 'event' or 'survey' as prefix to actual domain

```
referrerRewrite:
{searchPattern: /\:\/\/g ,
replacePattern: '://event.'}
```

Result: "http://www.opinionlab.com/index.html" would trigger the card for and report against "http://event.www.opinionlab.com/index.html"

Replace the Domain - Replace actual domain with hard coded "event.domain.com" (retaining actual page information--this is to use domain cookie for EDCCs)

```
referrerRewrite:
{searchPattern: /\:\/\/\[^\]\]*\/ ,
replacePattern: '://event.domain.com' }
```

Result: "http://www.opinionlab.com/index.html" would trigger the card for and report against "http://event.domain.com/index.html"

Replace the URL - Hardcode URL to something like 'event.domain.com/checkout.html'

```
referrerRewrite:
{searchPattern: /\:\/\/\.*\/g ,
replacePattern: '://event.domain.com/checkout.htm' }
```

Result: "http://www.opinionlab.com/index.html" would trigger the card for and report against "http://event.domain.com/checkout.htm"

Replace the Entire URL - Completely replace domain/URL without using RegEx Search Pattern

```
referrerRewrite:
{replacePattern: 'https://www.domain.com' }
```

Result: "http://www.opinionlab.com/index.html" would trigger the card for and report against "https://www.domain.com"

Append the URL - Add text to the end of the URL

```
referrerRewrite: {
replacePattern: location.href + '_special'
}
```

Result: "http://www.opinionlab.com/index.html" would trigger the card for and report against "http://www.opinionlab.com/index.html_special"

New Window Resize Configuration

Why would I want to configure the size of the window for the Dynamic Comment Card?

Usually this is not a concern: Most Dynamic Comment Cards invoked by the OnlineOpinion® Code Set are properly resized by the card's own resize engine. However for some specific instances in certain implementations using certain browsers: Specifically the On-Exit (Browser close only) Event-Driven and Invitation Feedback code for Gecko (like Firefox) and webkit browsers (like Safari or Chrome). The card's resize engine is ignored. Additionally specific client customizations may require a specialized size not accounted for by the resize engine. In these cases, the newWindowSize element should be configured to the appropriate size.

For example, a client has a comment card that is larger than the default size of the comment card window (545px width by 325px height) and, as a result, the bottom section of the card is cut off. By utilizing the "newWindowSize" parameter in the CONF file to define the minimum width and

height of a comment card window (if on-page card is not used).

```
/* Event-Driven Feedback configuration */
var oo_on_exit = new OOO.Ocode({
  cookie:
    {name:"oo_on_exit",
     type:"domain",
     expiration:"3600"},
  event :
    {onExit: 100},
  newWindowSize: [545, 700]
});
```

The developer configuring the CONF will need to determine the exact length of the card needed to create a proper fit. If this parameter is used, every comment card that utilizes that CONF file will be resized. If there are multiple cards of different sizes within a client's site, you will need to use one (separate) CONF file for the taller comment card and one CONF for all other default sized cards.

TeaLeaf Cookie Configuration

How do I integrate OnlineOpinion® Code with Tealeaf?

This version of OnlineOpinion® OCode automatically integrates with Tealeaf. If your Tealeaf cookie is named "TLTSID", the Tealeaf integration is automatic with OCode. If not, simply configure your instance to look for the Tealeaf cookie name specified in the domain with the tealeafCookieName.

Tunnel Configuration

What is a tunnel configuration

A Tunnel Configuration allows you to specify a number of steps that a user has to take in order for any of the code to run. For example, you can configure Invitation Feedback prompt to render to those who follow a 3-page bill payment process in sequence.

Can tunnel paths be made in all OCode types? How about Invitation Feedback?

Yes, tunnel configurations are possible in both OCode and Invitation (when page hit is not in play) Types.

What controls tunnels configurations?

For OCode Type, "tunnel" engages the tunnel feature, "path" controls the steps, and cookieName "changes the Tunnel path's Cookie Name. For Invitation type, "promptTrigger" controls both the tunnel and regular expression match.

Why won't my tunnel path work?

There could be several reasons.

1. You have not cleared your browser cache after your updates.
2. You have a specialized engine that does not support tunnel path (Tiny or Specialized Instance Condensed).
3. Your tunnel path are not configured correctly.
4. You have conflicting JavaScript code affecting your OnlineOpinion® instance.

If none of these items are the cause of your Custom Variables not working, please contact OpinionLab® for additional support

Image and Verbiage Customization

Can I change the word "Feedback?"

The word "Feedback" can be changed for any of the [+] Feedback Icons and Inline Feedback implementations. However, alternate text must be sent to OpinionLab® for prior approval and OpinionLab® reserves the right not to approve alternate text that does not reflect OpinionLab®'s brand image.

Can I use my own image instead of the one OpinionLab® gave me?

You can use an alternate image but you must send it to OpinionLab® for approval prior to implementing. OpinionLab® reserves the right not to approve alternate images that do not reflect OpinionLab®'s brand image.

Mobile Solutions

Will OnlineOpinion work on my mobile device?

Mobile devices are not supported by OnlineOpinion. However, OpinionLab has made considerable efforts to have OnlineOpinion degrade gracefully. If you are interested in a fully-supported mobile solution, please contact your Customer Service Representative for options regarding our Mobile Optimized solutions.

What will I see if my mobile device does not support Javascript?

OnlineOpinion requires Javascript to function properly, unless activated using "Static Activation".

How do I prevent OnlineOpinion from showing on all mobile devices?

Your configuration (CONF) file has an optional object attribute called "disableMobile" which will disable your OnlineOpinion code on all mobile devices.

```
disableMobile: 'true'
```

How do I prevent OnlineOpinion from showing on all non-iOS (iPhone/iPad) mobile devices?

Your configuration (CONF) file has an optional object attribute called "disableNoniOS" which will disable your OnlineOpinion code on all mobile devices.

```
disableNoniOS: 'true'
```

Known Issues

The following are known issues with OnlineOpinion® (Version 5+) and are considered unavoidable. OpinionLab® is currently unable to surmount these issues with most modern browsers. If the issue stems from a particular browser it is possible to disable or provide an alternate OnlineOpinion object instance: see the section on configuring your OnlineOpinion files for more.

Comment Card Support

OnlineOpinion Version 5.0 and Above Requires Modern (v3.6+) Dynamic Comment Cards for Proper Supported Functionality

OnlineOpinion JavaScript Code works in conjunction with OpinionLab's dynamic comment card (DCC) technology. For all features described in the user manual to be properly supported, the card called must be a compatible version. Lower card versions may still perform basic functionality and reporting, but supported features like referrer rewrites and custom variables may not be supported.

3.6.1 Comment Card Templates Will Not Work With "onPage" Card Functionality

If your implementation involves the "onPage" (as opposed to the traditional comment card in a new window) functionality on a comment card using the 3.6.1 template, you will notice that the card defaults back to the standard 3.6 template. The code will still work properly but these two cannot be used together. The correct card will still be presented, but it will use the 3.6 template.

Firefox browsers utilizing Firebug version 1.9b3 will break the comment card resize functionality

There was a bug discovered and acknowledged by Firebug that broke the global "window.resize" Javascript function. As of Firebug version 1.9b4 (released 12/16/11), the problem was resolved. To avoid this issue, simply upgrade Firebug to version 1.9b4 or greater.

Mobile Browsers

OnlineOpinion® Code is developed specifically for screen based web browsers. While accommodations have been made to address the use of OnlineOpinion® instances on standard web sites that may be displayed on mobile browsers (such as some basic touch screen upgrades and the "disableMobile" feature), many mobile browsers do not support the minimum requirements to support OnlineOpinion® Code in regard to JavaScript and CSS support. It is the responsibility of the client to determine whether to apply or disable OnlineOpinion® instances on mobile browsers.

OpinionLab® does have solutions to specifically address mobile browsers and the challenges they present. For more information, contact your OpinionLab® Sales or Client Services Representative.

On-Page Card "closeWithOverlay" will not work with Mobile Browsers

"closeWithOverlay" has been disabled for browsers identified as Mobile to avoid touch screen based usability issues.

Android based mobile devices may not work properly with Mobile Browsers

If you find that your code is not performing as expected on Android based mobile devices, please refer to the "disableNoniOS" object attribute available for your configuration file to disable the code on any device besides iPhone/iPad.

Internet Explorer Version 6

Rendering issues will occur with Invitation, Floating Icon, Bar Icon, and On-Page card when using Internet Explorer 6 In Quirks mode

When Internet Explorer 6 renders a page in Quirks mode, CSS support is extremely limited. These limitations will effect the rendering of the above mentioned implementations and configurations. We suggest using a [conditional comment](#) to use an alternate instance or implementation (Inline Feedback, Tab Icon, or Event Driven with a new window card) when IE6 is in Quirks mode.

The hover states for Feedback Icons will not render in Internet Explorer 6

The hover-over states for Internet Explorer 6 does not display for floating icons: Because of Internet Explorer 6's lack of support for CSS 2, the Hover over state for the icon implementations will not activate. OnlineOpinion® Code deprecates gracefully to accommodate for this.

On page scrolling for On-Page cards and some Feedback Icons will not render in Internet Explorer 6

The absolute position some feedback elements is not respected by Internet Explorer 6: Because of Internet Explorer 6's lack of support for CSS 2, these elements will remain fixed in relation to the page. This includes, On-Page Card and Tab Feedback Icon.

The animated [+] icon GIF stops rotating after an inline link is clicked.

Internet Explorer 6 assumes that the clicking of a link enacts a new navigation where the current page contents will be replaced. As part of the preparation process for this, it halts the code that animates the GIFs.

Form elements handled by the Browser or operating system are beyond the control of JavaScript

Internet Explorer 6 handles some form elements differently than most browsers which may lead to cosmetic or functional issues with the interface where z-index is involved. This issue cannot be overcome through the use of JavaScript.

Internet Explorer Version 7

HTTPS to HTTP security warning feature may cause functional issues with DCC Thank You windows

When security feature to warn if crossing from https to http is enabled in browser (not a default setting), "close this window" and auto close features in DCC Thank You windows do not work correctly. They load the original page within the card window.

Bar Feedback Icon occasionally misplaces its location when screen is maximized in Internet Explorer 7

An inconsistent behavior has been documented where the Bar Feedback Icon implementation occasionally mis-aligns its positioning when the browser goes from not being maximized to being maximized. This positioning issue rectifies itself once the user begins scrolling again. The issue is caused by the way Internet Explorer 7 interprets the height and width of the browser window. This is considered a minor issue and there is currently no known fix.

Internet Explorer Version 9

On-Page card does not render properly in Internet Explorer 9 when using versions of OnlineOpinion lower than 5.1.10

In Internet Explorer 9 only, the On-Page Card will not render due to a change on the code from Internet Explorer 8 to Internet Explorer 9. This is considered a major issue for clients using on-page card and OpinionLab recommends either upgrading to OnlineOpinion Version 5.1.10 (or higher) or changing the implementation to a new window to address the issue.

Firefox Version 3.5

Firefox 3.5 and below will not support the external CSS @media attribute which is implemented within oo_style.css to hide the floating icon when a page is printed. There is a solution to this issue. The client can remove the line below from oo_style.css and place the block of code below within the header of any page with the floating icon.

```
@media print { (hash)oo_bar, .oo_feedback_float, (hash)oo_tab { display: none; }}

<style media="print">
  (hash)oo_bar, .oo_feedback_float, (hash)oo_tab { display: none; }
</style>
```

This is simply another way to define the media type of "print" that is still supported by most major browsers today.

** After copying the above code, you will need to replace (hash) with the actual symbol. Confluence mishandles the symbol and converts it into a link.

On-Exit Event-Driven Feedback

On-Exit Event-Driven Feedback is Affected by Browser and Toolbar Pop-up Blockers

Pop-up blockers (which are commonly enabled in a user's browser) will prevent both the On-Entry and On-Exit Event-Driven Comment Cards from rendering for a user in a new pop up window. The On-Entry card can avoid this by using the On-Page Card implementation; the On-Page Card component does NOT function with On-Exit Events.

On-Exit Event-Driven Feedback Cannot Be Disabled for Browser Based Navigation Actions and Certain Specialized Links

If the survey is set to render On-Exit (onExit > 0), it will render based on the percentage for all browser based actions that trigger the Unload event for a page. This includes:

- Any Page Refresh
- Browser Back Button
- Browser Forward Button
- Browser Close Button ("X")
- New URL Enacted in Browser Address Bar
- Browser Bookmarks
- Third Party Toolbar/Add-on Buttons

Additionally, specialized links on a page that are **not** described under the "To Disable the On-Exit Event-Driven Feedback Implementation for Specific Instances" section of this document (explained below) may also trigger the survey on the Unload event for a page.

On-Exit Event-Driven Feedback Does Not Function on Certain Browsers

On-Exit Event-Driven Feedback functions as expected on most major web browsers (Microsoft Internet Explorer 6+, Mozilla Firefox 3.6+, Apple Safari 4.1+, Google Chrome 4+). However, issues have been encountered with some specific browsers. These browsers are listed below:

- Opera
- Konqueror

Invitation Feedback

Multiple Implementations

If multiple implementations are required (for example, for multiple language implementations), unique code sets will be needed for each language. Files will not be shared across implementations. Due to this, these situations may incur additional support charges.

Cross Domain Issues

Invitation Feedback utilizes elements in the assets directory that will experience cross-domain issues if improperly installed on other domains or sub-domains. To avoid this, OpinionLab recommends the installation of any code set using Invitation Feedback to be installed on the same domain as the pages that are intended to use this feature.

Required Use of a Consistent Engine File

OpinionLab requires that a consistent OnlineOpinion engine file be installed globally, on all pages, across any given domain for proper functioning of the Invitation implementation. Using the same file would be best, but at a minimum, the engine should use the same version across the domain. Doing so will avoid issues with friendly domains, pages hit, safe areas and other aspects of the Invitation implementation.

No Dynamic Configuration Selection

If multiple implementations of the Invitation Feedback are to be used, the client must provide the methodology used to select the proper configuration file. If OpinionLab® performs additional work to dynamically select a configuration additional charges will apply.

Custom Configurations of the Prompt and Monitoring Window

We recommend leaving the default language and styles in the markup file and related css file. Because of the interrelated elements that make up the prompt, any edits made to these files may cause errors that make the prompt unclear or inoperable. OpinionLab® will not troubleshoot markup and/or css files that have been changed by clients.

- Inline and External Style Selectors Used in the Prompt and Monitoring Window - Most Style Selectors are defined inline in the Prompt and Monitoring Window HTMLs. There are some styles that can't be passed inline or are easier to define externally. These styles are defined via sass (<http://sass-lang.com/>) which is then compiled to the minified css.
- No Translation Services for Prompt or Monitoring Window - If alternate languages are required, the client must provide the specific text to be used for each translation. OpinionLab® will charge additional fees for translations and each iteration of translation performed. Dynamic Comment Cards will be translated as specified in the original agreement.
- Logo size - The containing block for the company logo is 355 pixels wide by 55 pixels tall. Most modern browsers will accommodate and appropriately size image dimensions smaller than this, but MS Internet Explorer may skew the image to fit the available space. Because of this, recommended best practice is to have the logo image sized at 355 pixels by 55 pixels using a white (or the appropriate color) background.

Invitation Feedback May Have Limited or Modified Functionality for Certain Browsers

Invitation Feedback functions as expected on most major web browsers (Microsoft Internet Explorer 6+, Mozilla Firefox 3.6+, Apple Safari 4.1+). However, issues have been encountered with some specific browsers. These browsers are listed below:

- Firefox 4 and above - The Invitation monitoring window will not move to the back due to disabled the ability to blur or focus windows. There is no known solution for this issue.
- Google Chrome - Modified functionality for the "Popunder" monitoring window feature: JavaScript alerts are used to assist in triggering events that are seamless in other browsers due to Chrome's unique configuration. Additionally, the Dynamic Comment Card Resize engine has some issues firing in Chrome, which may require using the "newWindowSize" feature depending on the card's default height. In OnlineOpinion Versions lower than 5.1.10, there may be a timing issue that will cause the default alert message to render rather than the configured message (This has been fixed in Version 5.1.10). See the section on translation regarding how to handle translating these alerts if required.
- Opera - There is a known issue with the Invitation Feedback "Popunder" monitoring window feature functionality where the new window will not launch below the current window: It will launch above. There is no known fix for this issue.

Tunnel Methodology

Two page tunnels will not respect the tunnel methodology in Online Opinion Versions prior to 5.1.10

If the implementation of OCode Or Invite is configured for only a two (2) page tunnel, any page with the that instance of OnlineOpinion code enacted on it will trigger the instantiation and not respect the specification of the second page of the tunnel. this is considered a minor issue and OpinionLab recommends an update to OnlineOpinion 5.1.10 or higher to resolve this issue.

Custom Variables

Clients upgrading to Version 5+ will see custom variables reporting differently

For clients upgrading from a previous version, the version 5+ custom variables report as different fields. This will affect Web Service, KANA integrations or any custom exports.

Version 5+ custom variables are not available for analytics integrations

Only Legacy Custom Variables can be used for analytics integrations, and sometimes (as would be the case with non-Tealeaf integrations) only with custom override script. Custom Variables for Version 5+ will not integrate.

On-Page Card

Auto Close feature of DCC will not work with On-Page Card in Quirks mode

When using on-page card in Quirks mode, you can't use the auto close feature, only "close this window".

"Time On Page" data

OnlineOpinion code versions previous to 5.3 do not support "timeonpage" for comment cards utilizing "onPage" functionality. This data is included in all OnlineOpinion V5.3+ engine files. If you are using an engine file previous to 5.3, please consult your Client Service representative to get a copy of the latest version in order to upgrade and allow this functionality.

Sizing of the On-Page window is determined by the height of the comment card

When using the on-page implementation, the height of the opening for the on-page window is dynamically set to match the height of the comment card that is being presented. Since this is true, the user will be presented with scrollbars if the size of the thank you window is taller (in pixels) than the size of the comment card.

ClickTale Integration

Recommended Settings for ClickTale®

When projects are set up in ClickTale®, a tracking code is required in order to record sessions for the project site. The page will ask for an approximate total for the domain's daily site traffic, which will calculate a recommended recording ratio, and will also request what types of pages will be recorded (HTTP or HTTPS). If the recording ratio value is less than 100%, only a sampling of the sessions will be recorded and any sessions not recorded will return a value equal to "NULL". As a result, the OnlineOpinion® code will not store a value for the ClickTale® session.

As stated previously, if the recording ratio is less than 100% or less than 1.0 in the code, then the call to ClickTale® will result in a value equal to NULL.

Recommended Settings for the OnlineOpinion® Configuration File

To integrate ClickTale® with the OnlineOpinion® code, add the ClickTale code to the very end of the "oo_conf.js" file as shown in the following codeset. A new instantiation of the OnlineOpinion code must be created within the function **WRoninit()** before the ClickTale function is called. The parameters for **sessionCookieName** and the **clickTalePID** must be part of the settings or the ClickTale custom variable will not be populated. The parameter for **sessionCookieName** must be set to "CTSID" and the **clickTalePID** parameter must be set to your assigned **ClickTale Project ID (PID)**. The following is an example:



In the configuration, "sessionCookieName" is incompatible with "tealeafCookieName". If both are used, "tealeafCookieName" will have priority.

1. Before the OnlineOpinion Code is instantiated, add the ClickTale Code

```
/* Begin ClickTale Integration */
var WRInitTime=(new Date()).getTime();
with (document) {
  write('<div id="ClickTaleDiv" style="display: none;"></div>');
  write(unescape("%3Cscript%20src='"+
    (document.location.protocol=='https'? 'https:
//clicktale.pantherssl.com/' : 'http://s.clicktale.net/') +
"WRb6.js'%20type='text/javascript'%3E%3C/script%3E"));
}
```

2. In the oo_conf.js file, add the following at the bottom of the page

```
function WRoninit() {
var oo_ClickTale = new Ooo.Ocode({
sessionCookieName : "CTSID",
clickTalePID : ####
});
}
var ClickTaleSSL=1;
if(typeof ClickTale=='function') ClickTale(4841,1,"www03");
/* End ClickTale Integration */
```

"####" Represents the assigned ClickTale® Project ID (PID)

If an On-Entry Event-Driven Card is used, it is recommended to add a delay greater than ten seconds to the activation. This will allow extra time for the ClickTale® API to resolve and pass the required variables to OnlineOpinion®.

Recording

When a user opens a page, there are currently no known triggers to indicate (onEntry or onLoad) if the ClickTale® API will record the session and there is no guarantee that the session will be recorded. A value of "NULL" will be returned if the session is not being recorded and as a result, that instance of OnlineOpinion® code will not store a value for ClickTale®.

If the session is being recorded, there is a trigger that indicates the API has started recording the session and will most likely will be recording during an event-driven action, however this is totally dependent on the ClickTale® API.

Information about ClickTale's APIs can be viewed at http://wiki.clicktale.com/Article/JavaScript_API

Known ClickTale® Browser Incompatibility

ClickTale® supports recordings of Internet Explorer and Firefox on all major operating systems. ClickTale®'s policy is to support all browsers with a market share of over 10% (Referenced from [ClickTale®'s Wiki](#)). ClickTale® is known to be incompatible with the following browsers:

- Google Chrome
- Apple Safari
- Opera

Where Can I Get More Information on ClickTale®?

You can go to [ClickTale®'s Wiki](#) for more detailed information.