VA PANEL POC

High Level Process Summary

- Visitor Opens VA Panel
- Registering the Chat Button Div Container
- Embedded Chat Button is loaded with online/offline status
- If Required check button status and update FAQ copy accordingly
- Visit Clicks Chat Button
- Listen for Chat Session Started Event to hide VA Panel
- Listen for Chat Session Ended Event/Exit Survey Submitted Event to re-show VA Panel

Registering the Chat Button Div Container

```
var buttonId = "lpButtonDiv-va-panel"; // This will
be the unique div id for the embedded panel button
within the DOM


/**************
    REQUIRED
***************/

// register button div zone with LivePerson now div
exists within the DOM
var sdes = [{
  "type": "pagediv",
  "divId": buttonId
}];
if(lpTag && lpTag.sdes && lpTag.sdes.send) {
  lpTag.sdes.send(sdes);
} else if (lpTag && lpTag.sdes.push) {
  lpTag.sdes.push(sdes);
}
```

```
    // ^ The above will tell LP that the div now exists
    and is ready to receive the content -
    online/offline/busy etc...
```

Binding to Events

Embedded Chat Button is loaded with online/offline status

```javascript
var lpButtonEngagementId;

// make sure the function and objects exist
if(lpTag && lpTag.events && lpTag.events.bind) {
   lpTag.events.bind("LP_OFFERS",
"OFFER_IMPRESSION",function(e,d){

    var buttonState;
    if(e.state == "0") {
      buttonState = "n/a";
    } else if(e.state == "1") {
      buttonState = "a button was loaded - the state
is online";
    } else if (e.state == "2") {
      buttonState = "a button was loaded - the state
is offline";
    }

    if(e.engagementType == "5" && e.engagementId ==
'922185332') {
      // means embedded button and the unique id of
this demo POC embedded button
      // the id will be unique to your account and
will differ between DEV and PROD so your code will
need to store both values to compare
      lpButtonEngagementId = e.engagementId; // store
the value so we can use it later when we need to
```

```
    query the button state later in the process for an
    accurate value — this is in case the refresh timer
    has changed the state from online/offline and vice
    versa
        }
        console.log("button impression event
    ",e.state,e.engagementId,buttonState);
      });
    })
```

- Event to bind to: "LP_OFFERS", "OFFER_IMPRESSION"
- This event fires every time a button is displayed on the page
- You will need to check the e.engagementType value to see what type it is.
- 5 = embedded but there may be more than one embedded button on some pages that are NOT va panel related.
- so you will need to look at the e.engagementId property for the event and match it against a list you maintain to know if it is a va panel button or not
- otherwise the e.engagementName property may give some indication if named correctly on the liveperson side.

FYI -- Types and States

enums to help decipher the event data object values

```
var _TYPES = {
  0:'peeling_corner',
  1:'overlay_invite',
  2:'toaster',
  3:'slideout_invite',
  4:'invite',
  5:'embedded_button',
  6:'sticky_button'
};
```

```
var _eSTATES = {
    NA:0,
    ONLINE:1,
    OFFLINE:2,
    UNKNOWN:3,
    BUSY:4
  };
var _eRENDERINGTYPES = {
  CONTENT:0,
  CHAT:1
},
```

Chat Window Events

```
lpTag.events.bind("lpUnifiedWindow",
"state",function(e,d){
  if(e.state == "waiting" || e.state == "resume") {
    hideVaPanel();
  }
});
```

Conversation Status Events

```
lpTag.events.bind("lpUnifiedWindow","conversationInfo"
,function(eventData,appName){
  console.log('LivePerson conversationInfo event :
',eventData.state);

  // if the post chat exit survey has been submitted
then auto close the thank you window
  if(eventData.state == "applicationEnded") {
    closeThankyouWindow();
    showVaPanel();
```

```
    }

});
```

- applicationEnded : post chat survey submitted
- NOTE: if the chat configuration of the LivePerson Campaign does NOT have an exit survey attached, this event will NOT fire
- Instead you should trigger this code on the "ended" event instead.

```
...
  if(eventData.state == "ended") {
    // only use this conditional if you DO NOT have a
post chat exit survey configured
    // you would put the same code as above to close
the thank you screen
    closeThankyouWindow()
    showVaPanel()
  }
...
```

How to close the thankyou window after the survey is sent...

```
function closeThankyouWindow() {
  var closeBtn = document.querySelector('.lp_close');
// this is the class of the close button
  if(closeBtn) {
    closeBtn.click(); // if we find it, click to
close the thank you screen
  }
}
```

Complete POC HTML Example Code

This is the POC code in full connected to jbeadle's test account for demo 90233546

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title></title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-
width, initial-scale=1">
        <script type="text/javascript">
            /*
                basic tag setup for the demo
                will be handled via Tealium deployment
of liveperson code or direct on page deployment
                purely for demo
            */
            window.lpTag = window.lpTag || {};
            lpTag.section = ["pws", "english",
"cmb","uk"];

        </script>
        <script>

            window.lpTag=window.lpTag||
{};window.lpTag.autoStart = false;
            if(typeof
window.lpTag._tagCount==='undefined'){window.lpTag=
{site:'90233546'||'',section:lpTag.section||'',autoSt
art:lpTag.autoStart===false?
false:true,ovr:lpTag.ovr||
{},_v:'1.6.0',_tagCount:1,protocol:'https:',events:
{bind:function(app,ev,fn){lpTag.defer(function()
{lpTag.events.bind(app,ev,fn);},0);},trigger:function(
```

```
app,ev,json){lpTag.defer(function()
{lpTag.events.trigger(app,ev,json);},1);}},defer:func
tion(fn,fnType){if(fnType==0){this._defB=this._defB||
[];this._defB.push(fn);}else if(fnType==1)
{this._defT=this._defT||
[];this._defT.push(fn);}else{this._defL=this._defL||
[];this._defL.push(fn);}},load:function(src,chr,id)
{var t=this;setTimeout(function()
{t._load(src,chr,id);},0);},_load:function(src,chr,id)
{var url=src;if(!src){url=this.protocol+'//'+
((this.ovr&&this.ovr.domain)?
this.ovr.domain:'lptag.liveperson.net')+'/tag/tag.js?
site='+this.site;}var
s=document.createElement('script');s.setAttribute('ch
arset',chr?chr:'UTF-8');if(id)
{s.setAttribute('id',id);}s.setAttribute('src',url);d
ocument.getElementsByTagName('head').item(0).appendCh
ild(s);},init:function(){this._timing=this._timing||
{};this._timing.start=(new Date()).getTime();var
that=this;if(window.attachEvent)
{window.attachEvent('onload',function()
{that._domReady('domReady');});}else{window.addEventL
istener('DOMContentLoaded',function()
{that._domReady('contReady');},false);window.addEvent
Listener('load',function()
{that._domReady('domReady');},false);}if(typeof(window
._lptStop)=='undefined')
{this.load();}},start:function()
{this.autoStart=true;},_domReady:function(n)
{if(!this.isDom)
{this.isDom=true;this.events.trigger('LPT','DOM_READY'
,{t:n});}this._timing[n]=(new
Date()).getTime();},vars:lpTag.vars||
[],dbs:lpTag.dbs||[],ctn:lpTag.ctn||
[],sdes:lpTag.sdes||[],ev:lpTag.ev||
[]};lpTag.init();}else{window.lpTag._tagCount+=1;}
            window.lpTag._delayLoad=function()
{setTimeout(function()
{window.lpTag.start()},1)},window.attachEvent?
```

```
window.attachEvent("onload",function()
{window.lpTag._delayLoad()}):
(window.addEventListener("load",function()
{window.lpTag._delayLoad()},!1),window.addEventListen
er("load",function(){window.lpTag._delayLoad()},!1));

        </script>
        <link rel="stylesheet" type="text/css"
href="https://netdna.bootstrapcdn.com/bootstrap/3.1.1
/css/bootstrap.min.css">
        <style type="text/css">
          .container { margin-top: 20px; }
          #slideout { background: #fff; box-shadow: 0
0 5px rgba(0,0,0,.3); color: #333; position:
absolute; top: 100px; right: -520px;
          width: 500px; -webkit-transition-duration:
0.3s; -moz-transition-duration: 0.3s; -o-transition-
duration: 0.3s; transition-duration:
          0.3s; }

          #slideout form { display: block; padding:
20px; }

          #slideout textarea { display:block; height:
100px; margin-bottom:
          6px; width: 250px; }

          #slideout.on { right: 0; }

          .vertical-text {
            transform: rotate(-90deg);
            transform-origin: left top 0;
            display: inline-block;
          }

          .need-help {
            position: relative;
            right: 50px;
            top: 100px;
```

```html
            }
        </style>
    </head>
    <body>
      <div class="container">



        <div id="slideout">
          <p class="btn btn-primary need-help
vertical-text" id="need-help">Need Help?</button>
          <form>
            <p>FQA content goes here...</p>
            <p><div>Lorem ipsum dolor sit amet
consectetur adipisicing elit. Accusamus eum dolor
repellendus amet ab quod cupiditate cumque eaque
tempore, sit dolores, harum nam suscipit sunt magni
magnam optio? Architecto, nesciunt?</div></p>
            <p> <button type="button"
onclick="checkButtonState()" class="btn btn-
info">Click me to check the button state</button>
</p>
            <HR>
            Actual embedded chat button container is
below where the button would sit in the VA Panel
            <p id="button-container">
            </p>
          </form>
        </div>

      </div>
      <script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jq
uery.min.js"></script>
      <script>

        var lpButtonEngagementId;
        var lpButtonState;
```

```javascript
        function bindToChatEvents() {

            if(lpTag && lpTag.events &&
lpTag.events.bind) {

                lpTag.events.bind("lpUnifiedWindow",
"state",function(e,d){
                    console.log("LivePerson Chat Window
Event Detected: state == ",e.state);
                    if(e.state == "waiting" || e.state ==
"resume") {

                        hideVaPanel();
                    }

                });

lpTag.events.bind("lpUnifiedWindow","conversationInfo"
,function(eventData,appName){
                    console.log('LivePerson
conversationInfo event : ',eventData.state);

                    // if the post chat exit survey has
been submitted then auto close the thank you window
                    if(eventData.state ==
"applicationEnded") {
                        console.log(eventData.state,' event
fired means post chat survey has been submitted');
                        closeThankyouWindow();
                        showVaPanel();
                    }
                });

                lpTag.events.bind("LP_OFFERS",
"OFFER_IMPRESSION",function(e,d){

                    var buttonState;
                    if(e.state == "0") {
                        buttonState = "n/a";
```

```javascript
                } else if(e.state == "1") {
                    buttonState = "a button was loaded –
the state is online";
                } else if (e.state == "2") {
                    buttonState = "a button was loaded –
the state is offline";
                }

                if(e.engagementType == "5" &&
e.engagementId == '922185332') {
                    lpButtonEngagementId =
e.engagementId; // store the value so we can use it
later when we need to query the button state later in
the process for an accurate value – this is in case
the refresh timer has changed the state from
online/offline and vice versa
                }
                console.log("button impression event
",e.state,e.engagementId,buttonState);
            });
        }

    }

    function closeThankyouWindow() {
        var closeBtn =
document.querySelector('.lp_close'); // this is the
class of the close button

        // if we find it, click to close the thank
you screen
        if(closeBtn) {
            closeBtn.click();
            console.log('auto closing the thank you
screen');

        }
    }
```

```
        function togglePanel() {
          $('#slideout').toggleClass('on');
        }

        function checkButtonState() {

          if(lpButtonEngagementId) {
            var buttonState =
lpTag.taglets.rendererStub.getEngagementState(lpButto
nEngagementId).state; // use the number you grabbed
earlier when the button was loaded inside the panel
            console.log('the button id
',lpButtonEngagementId,' has the state of
',buttonState);
          }

        }

        function hideVaPanel() {
          console.log("LivePerson Chat window is open
... hiding NEED HELP panel and button...");
          $('#slideout').removeClass('on');
          $('#need-help').hide();
        }
        function showVaPanel() {
          console.log("LivePerson Chat session has
ended ... bringing back NEED HELP button...");
          $('#need-help').show();
        }


        // click function for the panel tab
        $('.need-help').click(function(){
          togglePanel(); // show/hide the panel

          var buttonId = "lpButtonDiv-va-panel"; //
This will be the unique div id for the embedded panel
button within the DOM
          // only do this once! we only need to
```

```
create the div tag once when first showing the panel
            if(!document.getElementById(buttonId)) {
                // if the div id has NOT already been
created, then make one and add to the panel in the
required position with the required id
                var buttonContainer =
document.getElementById('button-container');
                var buttonDiv =
document.createElement('div');
                buttonDiv.id = buttonId;
                buttonContainer.appendChild(buttonDiv);

                /***************
                    REQUIRED
                ****************/

                // register button div zone with
LivePerson now div exists within the DOM
                var sdes = [{
                  "type": "pagediv",
                  "divId": buttonId
                }];
                if(lpTag && lpTag.sdes &&
lpTag.sdes.send) {
                    lpTag.sdes.send(sdes);
                } else if (lpTag && lpTag.sdes.push) {
                    lpTag.sdes.push(sdes);
                }
                // ^ The above will tell LP that the div
now exists and is ready to receive the content -
online/offline/busy etc...
            }

        });

        bindToChatEvents();  // bind to events
    </script>
  </body>
</html>
```