HSBC CMB -- VA PANEL POC

High Level Process Summary

- Website page opens – already tagged with LivePerson code for monitoring etc (via Tealium)
- VA Panel is loaded onto page
    - at this point you should bind to all relevant events using bindToChatEvents() function as an example
- Visitor Opens VA Panel
    - Registering the Chat Button Div Container to load the hidden button -- see function injectLivePersonEmbeddedButtonContainer
    - Embedded Chat Button is loaded with online/offline status
    - VA Panel can show/hide custom HTML content responsive buttons
        - e.g.

```
 <p id="button-container">
  <a id="lp-va-panel-button-online" href="#" class="btn btn-success lp-va-panel-button hide-lp-button" onclick="triggerChatButtonClick();">Click to chat</a>
  <a id="lp-va-panel-button-offline" href="#" class="btn btn-warning lp-va-panel-button hide-lp-button">All Agents are OFFLINE/BUSY</a>
 </p>
```

- Visitor has conversation with Ask Andrew A.I. and can see the chat button at the bottom of panel
- If Visit Clicks Chat Button
  - call custom function to "fake" click the hidden button to start the chat -- see function triggerChatButtonClick
  - inside that custom function, feed in the preChatLines to the chat window -- see function addPreChatLinesToChat
- Listen for Chat Session Started Event to hide VA Panel
  - see lpTag.events.bind("lpUnifiedWindow", "state"...
- Listen for Chat Session Ended Event/Exit Survey Submitted Event to re-show VA Panel
  - see lpTag.events.bind("lpUnifiedWindow","conversationInfo"...

Configuration through _config

The poc.html file shows an approach to organise the config for all LP panel related code.

```
var _config = {
  USING_PROXY_BUTTON : true, // setting this to FALSE
will tell the code that you are NOT going to be
providing your own custom buttons on the page. This
will require an accompanying change on the LE2 admin
side to insert the HTML of the button content rather
than just empty HTML when using this setting set to
TRUE
  SEND_LAST_QUESTION_ASKED_INTO_CHAT : false, //
currently not in scope - follow the use of this to
see how you can add this feature to the code in the
future.
  EMBEDDED_BUTTON_ID_LOADED:null,
  EMBEDDED_BUTTON_TYPE: 5,
  EMBEDDED_BUTTON_IDS : [955221432,955231332], //
These are the unique button ids within our system
that correspond to your CLONE/PROD accounts
  // PLEASE NOTE ^^^ you should NOT need to change
the array above - I have preconfigured it with your
engagement ids for your clone/prod accounts
  EMBEDDED_BUTTON_DIV_CONTAINER_ID : "lpButtonDiv-
need-help-panel",
  VA_PANEL_EVENT_NAMESPACE : "VA_PANEL",
  VA_PANEL_EMBEDDED_BUTTON_IMPRESSION_EVENT_NAME :
"EMBEDDED_BUTTON_IMPRESSION",
  VIRTUAL_ASSISTANT_CONVERSATION_CHAT_LINES_CLASS :
"faq-chat-line" // replace with whatever class/logic
you might use to get the last question/ chat lines. I
suspect it will be completely different with the
actual AskAndrew and you will call an API to get that
data. this is just POC.
};
```

Which options should NOT be changed?

- `_config.EMBEDDED_BUTTON_ID_LOADED`

> used to store the id of the engagement button when loaded to be used later on in other supporting functions and methods

- `_config.EMBEDDED_BUTTON_TYPE`

> Maps to the correct type value for Embedded engagements with the LP system

- `_config.EMBEDDED_BUTTON_DIV_CONTAINER_ID`

> This is the unique div container ID where the `lpTag` object will attempt to load in the button when you register this destination with us via `injectLivePersonEmbeddedButtonContainer()` on panel

> expansion. it matches the value setup in the LivePerson admin system. Editing this value would cause the button not to load as we cannot find the expected destination within the DOM.

- `_config.VA_PANEL_EVENT_NAMESPACE`

> unique namespace for a custom event we trigger using the `lpTag.events` bridge to notify the panel code that the button in question has been loaded allowing you to react

- `_config.VA_PANEL_EMBEDDED_BUTTON_IMPRESSION_EVENT_NAME`

> the unique event name to correspond with the above namespace

Which options can I change?

- `_config.USING_PROXY_BUTTON`
  - `true` ==> tells the code you will be displaying your own HTML buttons for the online/offline state of the LivePerson Embedded button which will be loaded.
    - **PLEASE NOTE** This option is recommended for responsive requirements as any images deployed by the LivePerson system will not responsive. We can deploy HTML which you style via CSS if needed.
    - If using this option ensure that whatever HTML elements of your **ONLINE** button state have an onclick function call to `triggerChatButtonClick` (or equivalent code) which will start the chat window by calling our API and fake clicking the actual button loaded on the page – which will probably be hidden/have no actual viewable HTML content.
    - If you do not wish to handle the custom elements then suggest setting this to `false`
  - `false` ==> will exclude the following functions from execution ... `hideLivePersonButtonContainers` AND

refreshProxyButtonStatus ... this is because the LivePerson button shown will contain the HTML required for the online/offline states.

- _config.SEND_LAST_QUESTION_ASKED_INTO_CHAT
    - true ==> **ONLY HAS AN IMPACT IF _config.USING_PROXY_BUTTON == true** -- because default chat button clicks cannot be intercepted. Only using a proxy button allows this feature to be in scope.
        - enables the addPreChatLinesToChat example function to return some preChatLines to the chat window as part of the triggerChatButtonClick function call.
        - Follow this function's approach if you want to pass something like the last question asked to AskAndrew into the chat window for the agent and consumer to see in the header.
        - **PLEASE NOTE** this feature shows the key API calls you need to make but how you get the information in question from AskAndrew is down to you.
        - The only requirement is you pass it into the click call with the named parameter as an array [] of Strings.
    - false (default) ==> disables the behaviour

POC Styling related options...

The following are all style related for the purposes of the POC demo.

> You may/may not need them for your own implementation.

```
var LP_EXAMPLE_VA_PANEL_CSS_CLASS_NAME_FOR_MARKING_ELEMENTS = "lp-va-panel-button";

var LP_VISIBLE_ONLINE_BUTTON_CLASS_NAME_LIST = "btn
```

```
btn-success " +
LP_EXAMPLE_VA_PANEL_CSS_CLASS_NAME_FOR_MARKING_ELEMENTS;

var LP_VISIBLE_OFFLINE_BUTTON_CLASS_NAME_LIST = "btn
btn-warning " +
LP_EXAMPLE_VA_PANEL_CSS_CLASS_NAME_FOR_MARKING_ELEMENTS;

var LP_HIDDEN_BUTTON_CLASS_NAME_LIST = "btn btn-success hide-lp-button " +
LP_EXAMPLE_VA_PANEL_CSS_CLASS_NAME_FOR_MARKING_ELEMENTS;
```

Example page

For further details please refer to the comments within the poc.html file