

Scripting in BBOalert

BBOalert is developed using Javascript language. The users are enabled to add Javascript code to develop their own functionality. Javascript code can be provided with the data using the 'Script' records.

Three different ways of defining scripts are supported :

- Simple scripts contained in one line

```
Script,<script name>,<code>
```

- Medium size scripts entered direct in the data file

```
Script,<script name>
```

```
<code>
```

```
<code>
```

```
...
```

```
Script
```

- Complex scripts edited and debugged with Javascript development tools like Visual Studio Code. Lines beginning with // are interpreted as comments and therefore ignored by the tool. Otherwise the code is considered erroneous by the development tool.

```
//Script,<script name>
```

```
<code>
```

```
<code>
```

```
...
```

```
//Script
```

Scripts can serve two purposes :

- Dynamic formatting of text fields in the BBOalert data depending on the actual bidding, Typically :
 - Alert explanation text
 - Button and shortcut text

- BBO customization : users who are missing some functionality in BBO can add. Also the user interface can be personalized.

Scripts may create and use :

- Variables
 - Local : when initialized with the 'var' keyword
 - Global : without the 'var' keyword
- Functions
 - Local : function <name> (<arguments>) {<body>}
 - Global : <name> = function (<arguments>) {<body>;}

Scripts are executed in the same environment as BBOalert and global names should be chosen with care to avoid conflicts with the existing ones unless it is done on purpose to override the declaration of an existing function or variable. It is a safe practice when the user's global names are in capitals.

Scripts may be fragmented. Scripts with the same name are automatically concatenated inside BBOalert. This allows us to declare portions of code with different options.

Dynamic formatting of text

The name of the script should be enclosed between two % characters. Then the script will be executed and the script call will be replaced by the returned value. The script should use the predefined variables :

- **B : actual call (bid)**
- **BR : <call> field for matching the current bid**
- **CR : <context> field for matching the current auction**
- **C : actual bidding context(auction preceding the bid)**
- **R : string to be returned**

BBO event handlers

Users can define their own scripts to handle the BBO events. The template contains the possible event handling scripts :

Script,onAnyMutation
Script,onBiddingBoxCreated
Script,onBiddingBoxRemoved
Script,onBiddingBoxDisplayed
Script,onBiddingBoxHidden
Script,onAuctionBoxDisplayed
Script,onAuctionBoxHidden
Script,onNewAuction
Script,onNewActivePlayer
Script,onExplainCallDisplayed
Script,onExplainCallHidden
Script,onDataLoad
Script,onLogoff
Script,onAnyOpponentChange
Script,onNewDeal
Script

The onDataLoad event is called when BBOalert data is :

- **Imported**
- **Appended**
- **Retrieved from cache**
- **Retrieved from BBO convention card**

It is not strictly required but the onDataLoad should contain all global functions and variables declarations.

Another important event handler is onAnyMutation : it is called at every change in the BBO document. It should be used to implement the modifications in the user interface.

Some global BBOalert variables and functions may be used in the scripts.

Global variables

activePlayer

string : contains current active player direction and uid

auctionBoxDisplayed

boolean = true : during the auction

biddingBoxDisplayed

boolean = true : if bidding box is displayed

biddingBoxExists

boolean = true : if bidding box exists Bidding box element is created when the table is displayed

currentAuction

string : contains current auction

explainCallDisplayed

boolean = true : when opponents ask more info

lastDealNumber

string : contains current board number

LHOpponent

string : contains LHO user id

navDivDisplayed

boolean = true : if main BBO panel displayed

opponentChanged

string = 'L' when LHO changed or 'R' when RHO changed

RHOpponent

string : contains RHO user id

Global functions

areTheyVulnerable()

retrieve opponent's vulnerability tag from the auction box

areWeVulnerable()
retrieve our vulnerability tag from the auction box

bboalertLog(txt)
display text on the BBOalert blue panel

buttonOKvisible()
return true if OK button is visible

clearAlert()
Clear explanation text field

clickOK()
click OK button programatically

confirmBid()
when OK button appears, click it promatically

confirmBidsSet()
check if confirm bids switch is ON returns 'Y' 'N' or '' if not found

dragElement(elmnt)
make element draggable

elimine2Spaces(str)
elimine spaces and tabs from string str

findAlertText(context, call)
retrieve alert text for given context and call

getActivePlayer()
retrieve active player direction and user id

getAuctionBox()
retrieve the auction box element.

getBiddingBox()
retrieve bidding box element

getChatDiv()
returns div element containing chat dialog

getChatInput()

retrieve chat text input element

getChatMessage()
retrieve actual chat message input text

getContext()
Get actual bidding context

getCurrentChatDestination()
retrieve current chat destination

getDealNumber()
retrieve current board number

getExplainCallBox()
retrieve call explain box element

getExplainCallInput()
retrieve explain call box text input element

getExplainInput()
retrieve bidding box text input element

getMyHand()
retrieve my hand into a string

getNavDiv()
get main BBO panel div element

getNow(secs)
get formatted actual date and time if secs=true resultion up to seconds

getPartnerHand()
retrieve partner's hand if visible

getScript(scriptName)
retrieve script text by script name

getSeatNr()
get seat number tag of the opener

getVisibleMessageInput()
retrieve visible chat input element

isAlertON()

retrieve Alert button state (true = ON)

isVisible(e)

Check if element e is visible

makeRegExp(s)

transform string s into RegExp object

matchContext(refContext, actContext)

Check if actual bidding context matches reference context from the table

myOpponent(lho)

retrieve opponent's user id.

myPartner()

retrieve my partner's user id

mySeat()

retrieve my direction from the auction box 'S' 'W' 'N' 'E' or " if not found

ourVulnerability()

retrieve our vulnerability tag

preparePageReload()

reverseString(str)

reverse characters in the string

sendChat()

send chat message programmatically

sendMessageToOpponents(text)

sen chat text to both opponents

sendPrivateChat(uid, text)

send chat message to specific user id

setAlert(on)

set alert button state (on=true = ON)

setButtonPanel(on)

display button panel if on=true.

setChatDestination(t)
set chat destination to t

setChatMessage(msg, send)
send chat message text to msg If send=true send it immediately

setChatToTable()
set chat destination to table

setExplainCallText(txt)
set explain call box input text

setExplainText(txt)
set bidding box explanation text

setOptions(on)
display BBOalert panel if on=true.

setOptionsOff()
hide BBOalert panel

setOptionsOn()
display BBOalert panel

stripContext(ctx)
Strip context ctx from leading passes

toggleOptions()
toggle BBOalert panel display

undoCommand()
generate UNDO menu command programmatically

userScript(S, CR, C, BR, B)
execute script

whoAmI()
returns current BBO user-id

writeToClipboard(txt)
copy string txt to the clipboard

Sample data

//BBOalert

//Script,onAnyMutation

LARGE_BIDDING_BOX();

HOVER_BBOALERT_TAB();

SET_PROFILE_TEXT_LENGTH();

SWAP_BIDDING_BUTTONS();

DEAL_TOOLBAR_RIGHT();

//Script,onBiddingBoxCreated

console.log('Script onBiddingBoxCreated');

//Script,onBiddingBoxRemoved

console.log('Script onBiddingBoxRemoved');

//Script,onBiddingBoxDisplayed

console.log('Script onBiddingBoxDisplayed');

LARGE_BIDDING_BOX();

//Script,onBiddingBoxHidden

console.log('Script onBiddingBoxHidden');

setButtonPanel(false);

//Script,onAuctionBoxDisplayed

console.log('Script onAuctionBoxDisplayed');

//Script,onAuctionBoxHidden

console.log('Script onAuctionBoxHidden');

//Script,onNewAuction

console.log('Script onNewAuction ' + getContext());

//Script,onNewActivePlayer

```
console.log('Script onNewActivePlayer ' + getActivePlayer());
```

```
//Script,onExplainCallDisplayed
```

```
console.log('Script onExplainCallDisplayed');
```

```
setExplainCallText("");
```

```
getExplainCallInput().click();
```

```
//Script,onExplainCallHidden
```

```
console.log('Script onExplainCallHidden');
```

```
//Script,onDataLoad
```

```
DEAL_TOOLBAR_RIGHT = function () {
```

```
    var nd = getNavDiv();
```

```
    if (nd != null) {
```

```
        var dt = nd.querySelector('.dealViewerToolbarClass');
```

```
        if (dt != null) {
```

```
            dt.style.left = "";
```

```
            dt.style.right = '4px';
```

```
            var cc = nd.querySelector('.coverClass');
```

```
            if (cc != null) {
```

```
                cc.style.left = '0px';
```

```
                var ds = nd.querySelector('.dealScreenDivClass');
```

```
                if (ds != null) {
```

```
                    cc.style.width = (ds.getBoundingClientRect().width - dt.getBoundingClientRect().width) +
```

```
'px';
```

```
                }
```

```
            }
```

```
        }
```

```
};
```

```
HOVER_BBOALERT_TAB = function () {
```

```
    var t = document.getElementById('bboalert-tab');
```

```
    if (t.onmouseover == null) t.onmouseover = function () {
```

```
        setOptions(true);
```

```
    };
```

```

var rd = document.getElementById('rightDiv');
var vt = rd.querySelector('.verticalTabBarClass');
tabs = vt.querySelectorAll('.verticalClass');
for (var i = 0; i < tabs.length; i++) {
    if (tabs[i].textContent.search('BBOalert') == -1) {
        if (tabs[i].onmouseover == null) tabs[i].onmouseover = setOptionsOff;
    }
}
};

LARGE_BIDDING_BOX = function () {
    if(getPartnerHand() != "") return;
    if (biddingBoxDisplayed) {
        if (auctionBoxDisplayed) {
            var elAuctionBox = getAuctionBox();
            elAuctionBox.style.height = '40%';
            elAuctionBox.style.top = '0px';
        }
        var elBiddingBox = document.querySelector(".biddingBoxClass");
        var elBiddingButtons = elBiddingBox.querySelectorAll(".biddingBoxButtonClass");
        elBiddingBox.style.top = '40%';
        elBiddingBox.style.left = '0px';
        elBiddingBox.style.height = '35%';
        elBiddingBox.style.width = '100%';
        for (var i = 0; i < 17; i++) {
            elBiddingButtons[i].style.height = '30%';
            if (i < 12) elBiddingButtons[i].style.fontSize = '8vh';
            elBiddingButtons[i].style.width = '8%';
        }
        elBiddingButtons[0].style.left = '15%';
        elBiddingButtons[1].style.left = '25%';
        elBiddingButtons[2].style.left = '35%';
        elBiddingButtons[3].style.left = '45%';
        elBiddingButtons[4].style.left = '55%';
        elBiddingButtons[5].style.left = '65%';
        elBiddingButtons[6].style.left = '75%';
    }
}

```

```

    elBiddingButtons[7].style.left = '20%';
    elBiddingButtons[8].style.left = '30%';
    elBiddingButtons[9].style.left = '40%';
    elBiddingButtons[10].style.left = '50%';
    elBiddingButtons[11].style.left = '60%';
    elBiddingButtons[11].style.width = '18%';
    elBiddingButtons[7].style.top = '40%';
    elBiddingButtons[8].style.top = '40%';
    elBiddingButtons[9].style.top = '40%';
    elBiddingButtons[10].style.top = '40%';
    elBiddingButtons[11].style.top = '40%';
    elBiddingButtons[13].style.top = '33%';
    elBiddingButtons[14].style.top = '33%';
    elBiddingButtons[15].style.top = '65%';
    elBiddingButtons[16].style.left = "";
    elBiddingButtons[16].style.right = '4px';
}

};

SWAP_BIDDING_BUTTONS = function () {
    if (biddingBoxDisplayed) {
        var elBiddingBox = document.querySelector(".biddingBoxClass");
        var elBiddingButtons = elBiddingBox.querySelectorAll(".biddingBoxButtonClass");
        for (var i = 12; i < 15; i++) {
            elBiddingButtons[i].style.left = "";
            elBiddingButtons[i].style.right = '4px';
        }
        elBiddingButtons[16].style.left = '4px';
        if (getExplainInput().getAttribute('maxlength') != '254')
            getExplainInput().setAttribute('maxlength', '254');
    }
};

SET_PROFILE_TEXT_LENGTH = function () {
    otherInput = $('otherInputClass')[0];
    if (otherInput.getAttribute('maxlength') != '2540')
        otherInput.setAttribute('maxlength', '2540');
};

```

```
};  
//Script,onLogoff  
console.log('Script onLogoff');  
//Script,onAnyOpponentChange  
console.log('Script onAnyOpponentChange ' + opponentChanged);  
setChatMessage(findShortcut('ww'), true);  
//Script,onNewDeal  
console.log('Script onNewDeal');  
//Script
```