

11 One-To-One Financial Portal JavaScript functions

The functions described in this chapter support the portal features of the One-To-One Financial application. You may use them to reduce the time and effort of developing your Web financial application.

The functions are defined in the files listed below, which can be found in the directory `$BVT01/finance/startup_scripts`.

JavaScript API file	Type	Page
<code>pt_favoritelinks.js</code>	JavaScript API file for favorite links	323
<code>pt_reminder.js</code>	JavaScript API file for reminders	329

The utility functions in these startup files perform operations that the BVSN Financial application uses in several places. Because they are only loaded once, at startup, they incur no overhead for repeated loading (unlike `#include` script files).

Favorite links functions

These functions support the favorite links management pages. The favorite links capabilities are part of One-to-One's portal features. These features allow site customers to create their own home page on the institution's site. In addition to seeing information about their financial accounts the customers can also save links to their favorite web sites. This chapter describes how to add, change, sort, and delete links from a list of favorite sites. These functions reside in the `pt_favoritelink.js` file.

The prefix at the beginning of these functions represents BroadVision (BV) JavaScript (JS) Portal (PT) Favorite Links (FL). Using a different prefix for any functions or constants that you add helps keep your code separate.

Favorite links constants

The constants shown in the table below are used by the functions that follow. The first constant in the table holds the name of the database table that stores the links information. The constants that start with Column are the names of the database columns. The constants that contain Field name the HTML fields in the form.

BVJS_PTFLTableName	Name of the database table that actually stores this information. "BVPTFL_Links"
BVJS_PTFLColumnID	Name of the database column that holds the ID for a favorite link. "LINK_ID"
BVJS_PTFLColumnURL	Name of the database column that holds the URL for a favorite link. "URL"
BVJS_PTFLColumnName	Name of the database column that holds the friendly name for a favorite link. "LINK_NAME"
BVJS_PTFLColumnRank	Name of the database column that holds the rank for a favorite link. "RANK"
BVJS_PTFLColumnFlags	Name of the database column that holds the flags for a favorite link. "FLAG"
BVJS_PTFLColumnUserString1	Name of the database column that holds the user-defined string column 1 for a favorite link. "STRINGCOL1"
BVJS_PTFLColumnUserString2	Name of the database column that holds the user-defined string column 2 for a favorite link. "STRINGCOL2"
BVJS_PTFLColumnUserInt1	Name of the database column that holds the user-defined integer column 1 for a favorite link. "INTCOL1"
BVJS_PTFLColumnUserInt2	Name of the database column that holds the user-defined integer column 2 for a favorite link. "INTCOL2"
BVJS_PTFLFieldWorkingID	Name of the HTML field in which BVJS_PTFLProcessForm() looks for the ID of the working link. "workingID"
BVJS_PTFLFieldWorkingURL	Name of the HTML field in which BVJS_PTFLProcessForm() looks for the URL of the working link. "workingURL"
BVJS_PTFLFieldWorkingName	Name of the HTML field in which BVJS_PTFLProcessForm() looks for the friendly name of the working link. "workingName"
BVJS_PTFLFieldWorkingRank	Name of the HTML field in which BVJS_PTFLProcessForm() looks for the rank of the working link. "workingRank"

The following functions are part of BroadVision's larger portal functionality that allows sites to implement personalized portal pages for their customers.

Favorite link methods	Description
BVJS_PTFLInit()	Sets the session variables.
BVJS_PTFLShow()	Displays the links as specified by a given format string.
BVJS_PTFLDelete()	Deletes a specified link.
BVJS_PTFLUpdate()	Changes the attributes of a given link.

BVJS_PTFLAdd()	Adds a new link.
BVJS_PTFLeditAll()	Formats the rows in the list of links for editing.
BVJS_PTFLSortManually()	Sorts the links per customer input.
BVJS_PTFLSortAlphabetically()	Sorts the links alphabetically.
BVJS_PTFLProcessForm()	Processes the HTML form.
BVJS_PTFLClearWorkingRow()	Clears the working link row.


BVJS_PTFLInit()

Sets the session variables. Use this function in writing a form and for editing links.

```
function BVJS_PTFLInit()
```

Parameters None

Return value Returns 0 for success and non-zero for an error.

 Any script that uses links must call this function once per session before it calls any other link related methods.

BVJS_PTFLShow()

Iterates through the customer's links and displays them as specified by **formatString**.

```
function BVJS_PTFLShow(formatString, betweenString)
```

Parameters

formatString String specifying what each row looks like. Note that this parameter is passed unmodified to `BVI_Formatter::formatTable()`. See [BVI_Formatter](#) for more details.

betweenString String displayed between the rows.

Return value A string containing the generated HTML code, which is typically immediately output to the browser. If an error occurs, the return value is null.

Example The following code shows a simple example of using **formatString** to display links

```
var formatString = "<TR><TD><A HREF=\"<%= \" +
    BVJS_PTFLColumnURL + \" %s encoding:html:std%>\"> <+ \" +
    BVJS_PTFLColumnName + \" %s encoding:html:std%> </A></TD></TR>\\n\";
```

BVJS_PTFLDelete()

Deletes a given link.

```
function BVJS_PTFLDelete(id)
```

Parameters

id	Identity of the link to delete.
-----------	---------------------------------

Return value Returns zero for success and non-zero for an error.

BVJS_PTFLUpdate()

Changes the attributes of the link with the given id.

```
function BVJS_PTFLUpdate(id, newURL, newName, newFlags,  
                          newStringCol1, newStringCol2,  
                          newIntCol1, newIntCol2)
```

Parameters

id	Identity of the link to be changed.
newURL	The new URL for the given link.
newName	The new name for the given link.
newFlags	The new flags for the given link.
newStringCol1	The new user-defined first string column for the given link.
newStringCol2	The new user-defined second string column for the given link.
newIntCol1	The new user-defined first integer column for the given link.
newIntCol2	The new user-defined second integer column for the given link.

Return value Returns zero for success and non-zero for an error.

BVJS_PTFLAdd()

Adds a new link with a given URL, name, and flags. The rank is set to the current maximum rank +1. A unique identity (ID) is assigned and returned.

```
function BVJS_PTFLAdd(newURL, newName, newFlags,  
                      newStringCol1, newStringCol2,  
                      newIntCol1, newIntCol2)
```

Parameters

newURL	The URL of the new link.
newName	The friendly name for the new link.
newFlags	The flags for the new link.

newStringCol1	The user-defined first string column for the new link.
newStringCol2	The user-defined second string column for the new link.
newIntCol1	The user-defined first integer column for the new link.
newIntCol2	The user-defined second integer column for the new link.

Return value Returns a string, which is the id of the new link. Returns null if there is an error.

BVJS_PTFLEditAll()

Runs a loop through the customer's links in ascending order. The function formats each line as specified in **formatString**. The **formatString** is concatenated with another string that defines the hidden field names, and then passed directly to a `BVI_Formatter`. (The hidden field names are how the script page keeps track of the fields.) The formatting is specified by **formatString**. Note that this HTML code is only effective within a `<FORM>` tag, the `ACTION` attribute of the `<FORM>` tag will likely be the page displaying the form (though this is not required). That is, the page typically posts to itself. This string is usually output to the browser immediately.

```
function BVJS_PTFLEditAll(formatString)
```

Parameters

formatString	String describing how to format the rows of the link table. This parameter is concatenated with another string, then passed directly to a <code>BVI_Formatter</code> .
---------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Return value Returns a string containing HTML for editing each link. If an error occurs this function returns null.

Example The following code shows a simple example of using **formatString** to display links

```
var formatString = "<TR><TD><INPUT TYPE=submit NAME=\" " +
  "Edit<%= " + BVJS_PTFLColumnID + " %s " +
  "encoding:html:std %>\" Value=\"Edit\"></TD><TD><%= " +
  BVJS_PTFLColumnName + " %s encoding:html:std %></TD><TD><INPUT " +
  "TYPE=submit NAME=\"Delete<%= " + BVJS_PTFLColumnID +
  " %>\" Value=\"Delete\"></TD></TR>\n";
```

Remarks

For full functionality you should define the following HTML fields:

- **Edit[ID]** - sets the working link to the one given by [ID].
- **Delete[ID]** - deletes the link given by [ID].

BVJS_PTFLSortManually()

Sorts the links according to the customer's specification. See the `moveDown` and `moveUp` functions in ["Sorting links manually" on page 89](#) for an example of how to manipulate these hidden fields.

```
function BVJS_PTFLSortManually()
```

Parameters This function requires no parameters, however, it does make use of the hidden fields (**NAME=[ID]** **VALUE=[RANK]**) defined in the form from which this page was posted.

Return value Returns zero for success and non-zero for an error.

BVJS_PTFLSortAlphabetically()

Sorts the links alphabetically by friendly name. (Names that start with numbers appear before names that start with letters. Names that begin with upper case letters appear before names with lower case letters.)

```
function BVJS_PTFLSortAlphabetically()
```

Parameters None

Remarks Returns zero for success and non-zero for an error.

BVJS_PTFLProcessForm()

Processes the HTML form created by the combination of a JSP file, the Session variables and constants defined in [BVJS_PTFLInit\(\)](#), and the HTML generated by [BVJS_PTFL>EditAll\(\)](#).

```
function BVJS_PTFLProcessForm()
```

Parameters None. However, this function makes use of the following HTML fields:

- All fields with names specified by constants starting with **BVJS_PTFLField**.
- All fields defined by the **formatString** in [BVJS_PTFL>EditAll\(\)](#).
- All fields defined by the automatic addition to the **formatString** in [BVJS_PTFL>EditAll\(\)](#).

Return value Returns zero for success or for a customer error that sets either **portal . linkError_blankURL** or **portal . linkError_updateInvalid**. All other types of errors return 1.

BVJS_PTFLClearWorkingRow()

Provides a clear row for the customer to use. Clears any unsaved contents of the current row by assigning it to a new empty object.

```
function BVJS_PTFLClearWorkingRow( )
```

Parameters None

Return value None

Reminders

These functions support the reminders management pages. The reminders capabilities are part of One-to-One's portal features. These features allow site customers to create their own home page on the institution's site. In addition to seeing information about their financial accounts the customers can also save reminders for important dates, such as anniversaries, birth dates, and tax accountant appointments. This chapter describes how to add, change, and delete reminders from a list of reminders. The functions described in this section reside in the `pt_reminder.js` file.

The prefix at the beginning of the functions and constants represents BroadVision (BV), JavaScript (JS), Portal (PT), Reminders (RM). Using a different prefix for any functions or constants that you add helps keep your code separate.

Reminders constants

The constants shown in the table below are used by the functions that follow. The first two constants in the table are for error message information. The rest of the message constants correspond to error messages. The errors are described in the `bvportal_msg.hh` file. The `BVJS_PTRMPersonalAlertType` is the alert type. The remaining constants all contain Field. These constants hold values where the `BVJS_PTRMProcessForm` method looks for information and actions to process. These are constants, so you should not change the values.

Reminders constants	Definitions and values
<code>BVJS_PTRMMessageSet</code>	The set of messages to use for logging. value = 35
<code>BVJS_PTRMMessageBase</code>	The offset for One-To-One Portal messages (to avoid collisions with other messages). value = 35000
<code>BVJS_PTRMMessageWrongNumAlertJobs</code>	Message number for incorrect number of alert jobs. value = 450
<code>BVJS_PTRMMessageInvalidReminder</code>	Message number for invalid reminder. value = 451
<code>BVJS_PTRMMessageMissingState</code>	Message number for missing state information. value = 21
<code>BVJS_PTRMMessageBadParam</code>	Message number for a bad parameter. value = 2
<code>BVJS_PTRMMessageMissingParam</code>	Message number for a missing parameter. value = 3
<code>BVJS_PTRMPersonalAlertType</code>	The <code>ALERT_TYPE</code> of personal reminders, as defined in <code>BV_ALERT_TYPES</code> . value = <code>bvptrm_personal_reminder</code>
<code>BVJS_PTRMFieldFriendlyName</code>	Name of the field in which <code>BVJS_PTRMJProcessForm()</code> looks for the friendly name of the new personal reminder. value = "New Friendly Name"
<code>BVJS_PTRMFieldMonth</code>	Name of the field in which <code>BVJS_PTRMJProcessForm()</code> looks for the month of a new personal reminder. value = "New Month"
<code>BVJS_PTRMFieldDay</code>	Name of the field in which <code>BVJS_PTRMJProcessForm()</code> looks for the day of the month of a new personal reminder. (Not the day of the week.) value = "New Day"
<code>BVJS_PTRMFieldAdd</code>	Name of the submit button for adding a new personal reminder. value = "Add New Reminder"
<code>BVJS_PTRMFieldDelete</code>	Name of the submit button for deleting a personal reminder. value = "DeleteReminder"

BVJS_PTRMFieldEdit	Name of the submit button for editing a personal reminder. value = "EditReminder"
BVJS_PTRMFieldIsYearly	Name of the field in which BVJS_PTRMJProcessForm() looks for the flag indicating that this reminder is to repeat yearly. value = "IsYearly"
BVJS_PTRMFieldInterval	Name of the field in which BVJS_PTRMJProcessForm() looks for the reminder interval for the new or working personal reminder. value = "NewInterval"
BVJS_PTRMFieldWorkingID	Name of the field in which BVJS_PTRMJProcessForm() looks for the ID of the working reminder. value = "WorkingReminderID"
BVJS_PTRMFieldUpdate	Name of the submit button for updating the working personal reminder. value = "UpdateWorkingReminder"
BVJS_PTRMFieldDaysBefore	Name of the field in which BVJS_PTRMJProcessForm() looks for the number of days prior to the event that an alert should be triggered. value = "DaysBeforeEvent"
BVJS_PTRMFieldDeliveryType	Name of the field in which BVJS_PTRMJProcessForm() looks for the delivery type of the reminder being updated or added. value = "DeliveryType"

Reminders functions

These functions are part of BroadVision's larger portal functionality that allows sites to implement personalized portal pages for their customers.

Reminders functions	
BVJS_PTRMInit()	Determines the jobID of the personal reminders alert job.
BVJS_PTRMGetPersonalInbox()	Returns any messages in the customer's inbox that were generated by a personal reminder.
BVJS_PTRMDisplayPersonal()	Displays the customer's personal reminders.
BVJS_PTRMEditPersonal()	Edits the customer's personal reminders.
BVJS_PTRMAddPersonal()	Adds a personal reminder to the customer's list of alerts.
BVJS_PTRMUpdatePersonal()	Updates a personal reminder in the customer's list of alerts.
BVJS_PTRMDeletePersonal()	Deletes a personal reminder from the customer's list of alerts.
BVJS_PTRMProcessForm()	Processes the HTML form performing any actions specified.
BVJS_PTRMClearWorkingReminder()	Clears the working reminder.

BVJS_PTRMInit()

Determines the **jobID** of the personal reminders alert job, which is saved for later use by other reminders functions. Clears the portal session error flags.

```
function BVJS_PTRMInit()
```

Parameters None.

Return value Returns zero for success and non-zero for an error.

```
Example          function BVJS_PTRMInit()
                 {
                 var portal = Session . portal;

                 if(portal == null)
                   Session . portal = portal = new BVI_Properties();

                 portal . personalReminderError = 0;
                 portal . personalReminderError_dateInvalid = 0;
                 portal . personalReminderError_editInvalid = 0;
                 portal . personalReminderError_blankName = 0;
                 portal . personalReminderError_invalidInterval = 0;
```

BVJS_PTRMGetPersonalInbox()

Returns messages in the customer's inbox that were generated by a personal reminder.

```
function BVJS_PTRMGetPersonalInbox()
```

Parameters None

Return value A **BVI_Table** containing the portion of the customer's alert inbox generated by personal reminders. Returns null on an error.

BVJS_PTRMDisplayPersonal()

Displays the customer's personal reminders by calling a **BVI_Formatter** to format the output as specified by the given **formatString**.

```
function BVJS_PTRMDisplayPersonal(formatString, betweenString)
```

Parameters

formatString	String describing how to format the buttons and fields of the reminders page. This parameter is passed without being modified directly to a BVI_Formatter .
betweenString	String passed unmodified to a BVI_Formatter .

Return value A string. The result from the formatter. This string is typically output to the browser immediately. Returns null on an error. Returns the empty string if there is no data to display.

BVJS_PTRMEditPersonal()

Calls a **BVI_Formatter** to format the HTML form as specified by the given **formatString**. Sets some of the hidden variables as necessary.

```
function BVJS_PTRMEditPersonal(formatString, betweenString)
```

Parameters

formatString	String describing how to format the buttons and fields of the reminders page. This parameter is passed unmodified directly to a BVI_Formatter .
betweenString	String passed unmodified to a BVI_Formatter .

Return value A string representing the result from the formatter concatenated with any hidden fields that are set. This string is typically output to the browser immediately. Returns null on an error.

BVJS_PTRMAddPersonal()

Adds a personal reminder to the customer's list of alerts, given all of the parameters, that is, inserts a row in **BV_ALERT_SPEC** table.

```
function BVJS_PTRMAddPersonal(friendlyName, date, daysBefore, interval,  
                              isYearly, deliveryType)
```

Parameters

friendlyName	A string of up to 255 bytes, the name of the personal reminder.
date	A BVI_DateTime , the date of the personal reminder event.
daysBefore	A non-negative integer, the number of days before the event that reminders must be sent.
interval	A non-negative integer, the number of days between reminders. That is, reminders must be sent every interval number of days. Zero is a special case. Only two reminders will be sent one on the date, and one daysBefore date (if daysBefore is not zero).
isYearly	An integer, one if the year of this reminder should be incremented when the event occurs, otherwise zero.
deliveryType	A string of up to 255 bytes, the name of the delivery mechanism to be used. If null, bv_inbox_delivery will be used.

Return value Returns zero for success and non-zero for an error.

▶ If the **friendlyName** is blank, the **portal . personalReminderError_blankName** is set and success is returned. If the **interval** is negative, the **portal . personalReminderError_invalidInterval** is set and success is returned. If any error flags are set, **portal . personalReminderError** (the master error flag) is also set.

BVJS_PTRMUpdatePersonal()

Updates a personal reminder in the customer's list of alerts with a given ID, that is, updates a row in the BV_ALERT_SPEC table.

```
function BVJS_PTRMUpdatePersonal(ID, originalName, friendlyName, date,
                                daysBefore, interval, isYearly,
                                deliveryType)
```

Parameters

ID	An integer, the ALERT_ID of the reminder to be updated. The ALERT_ID column in the BV_ALERT_SPEC table.
originalName	A string of up to 255 bytes, the original name of the personal reminder. If the customer is changing the name of a reminder the script needs to track the original name and the new name.
friendlyName	A string of up to 255 bytes, the name of the personal reminder.
date	A BVI_DateTime, the date of the personal reminder event.
daysBefore	A non-negative integer, the number of days before the event that reminders must begin being sent.
interval	A non-negative integer, the number of days between reminders. That is, reminders must be sent every interval number of days. Zero is a special case. Only two reminders will be sent one on the date, and one daysBefore date (if daysBefore is not zero).
isYearly	An integer, one if the year of this reminder should be incremented when the event occurs, otherwise zero.
deliveryType	A string of up to 255 bytes, the name of the delivery mechanism to be used. If null, the bv_inbox_delivery will be used.

Return value

Returns zero for success and non-zero for an error.

▶ If the **friendlyName** is blank **portal . personalReminderError_blankName** is set and success is returned. If the **interval** is negative, **portal . personalReminderError_invalidInterval** is set and success is returned. Even when there is a customer input error, the **workingReminder** information in **Session . portal** is still updated. If any error flags are set, **portal . personalReminderError** (the master error flag) is also set.

BVJS_PTRMDeletePersonal()

Deletes a personal reminder of a given ID and name from the customer's list of alerts, that is, deletes a row in the BV_ALERT_SPEC table.

```
function BVJS_PTRMDeletePersonal(ID, Name)
```

Parameters

ID	An integer, the ALERT_ID of the reminder to be updated. The ALERT_ID column in the BV_ALERT_SPEC table.
name	A string of up to 255 bytes, the name of the personal reminder to delete.

Return value Returns zero for success and non-zero for an error.

BVJS_PTRMProcessForm()

Updates the HTML form by reposting to itself. Uses the information in `Session . portal` and the `Request` object to extract information from the HTML form that was most recently posted (if any) and perform any action the HTML form specifies. The possible actions are:

- Add a personal reminder.
- Update a personal reminder.
- Delete a personal reminder.

The add, update, and delete methods do not need to be called directly. They can be called through this `BVJS_PTRMProcessForm()` method.

```
function BVJS_PTRMProcessForm()
```

Parameters None

Return Value Returns zero for success and non-zero for an error. If any error flags are set, `portal . personalReminderError` (the master error flag) is also set.

Remarks This method processes only one action at a time. Therefore, you should design your user interface so that customers can select only one action at a time.

BVJS_PTRMClearWorkingReminder()

Clears the working reminder.

Parameters None

Return values None