Customizing AccountView Consumer User Interface (CUI)

AccountView Version:	1.57
Document Version:	14
Document Issue Date:	July 19, 2018

Technical Support:	(866) 856-4951
Telephone:	(781) 376-0801
Fax:	(781) 376-8040
Web:	byallaccounts.morningstar.com

Table of Contents

Table of Contents

CUSTOMIZING ACCOUNTVIEW CONSUMER USER INTERFACE (CUI)

This document describes how to customize AccountView Consumer User Interface (CUI) which is a part of the Morningstar® ByAllAccounts℠ aggregation service. The AccountView CUI has a simple look and feel that may be customized by each firm in the system.

About this guide

This guide provides details about customizing CUI. It also provides additional information that those integrating CUI may find helpful:

- Appendix A: lists the error messages that can be generated by CUI.
- Appendix B: describes the postMessages analytics for the Add Account workflow.

Customization files

There are four files that you can get from ByAllAccounts, edit, and return to ByAllAccounts to deploy customizations for your own firm's CUI:

- CustomLoginCSS.css
- CustomCuiCSS.css
- CustomLoginDictionary.js
- CustomCuiDictionary.js.

These files are available here: https://www.byallaccounts.net/WebPortfolio/stylingTemplates.zip.

When you unzip the files, the .js files may have a .txt extension. The file extension is there to facilitate file transfer and editing.

If the firm chooses to have different interfaces for (SSO) users and non-SSO users, they can provide an extra pair of customization files.

If you want different custom styling for single-sign on (SSO) users

You can choose to have different custom styling for single-sign on (SSO) users. Provide ByAllAccounts with customized, renamed versions of the CustomCuiCSS.css and CustomCuiDictionary.js files:

- CustomCuiCSS_2.css
- CustomCuiDictionary_2.js

Note that login files are irrelevant for SSO users, so just provide these two files for SSO.

If you want mostly the same styling for both SSO and non-SSO users, you may choose to edit the base customization files first, make copies of them using the SSO naming structure, then further refine them.

Combinations of customizations

The firm can choose to use:

- the same custom styling for all users (return the four base files); this is the typical configuration
- use a separate custom stylings for SSO users and non-SSO users (return six files)
- custom styling for SSO users only (return two files)

What you can customize using the files

By providing customized files, you can control:

- **Styling** – By providing custom Cascading Style Sheet (.css) files you can redefine the look and feel of the login page and all other views and dialogs in the interface by adjusting fonts, colors, and backgrounds for each element in the display. Some settings affect individual elements and some are global settings. There are also styling options specific to narrow and mobile displays. You can also hide features.
- **Terminology** – By editing JavaScript (.js) files you can change text on views and buttons.
- **Options and features** – By editing the JavaScript files (.js) you can optionally include an initial landing page for users who have no account set up, adjust the header for the main accounts list view, add a full header to every page in the Add Accounts wizard, add a logo to the header(s), add custom sidebar content to most views, and change the footer button location to left-aligned.

Additionally, you can provide your own logo and graphics to include in the custom sidebars.

Overview of how to customize CUI

Here is an overview of the process for customizing the interface. These steps take into consideration that you may want to provide a different customized CUI for SSO users.

1. Download copies of the .css files and .js files that contain all the styles, terms, and option definitions currently used by the system. They are available here: https://www.byallaccounts.net/WebPortfolio/stylingTemplates.zip

   Note: When you unzip the files, the .js files may have a .txt extension. The file extension is there to facilitate file transfer and editing.
2. Use the developer tools in your browser to test and confirm the .css settings you want to use to customize the styles.
3. Edit style declarations in the .css files using the standard stylesheet language to reflect those settings.
4. Edit terminology in .js files to reflect any customizations you want to make to text in the user interface (UI) and on buttons.
5. Optionally, control other options and features, such as sidebar content and logo, by editing the .js file(s).
6. If you want to provide different customizations for SSO users:
   a. Save the CustomCuiCSS.css and CustomCuiDictionary.js files as CustomCuiCSS_2.css and CustomCuiDictionary_2.js, respectively.
   b. Refine the new files for SSO users.
7. Send the edited .css and .js files, logo, and any sidebar graphics to ByAllAccounts to be implemented for your firm in the AccountView CUI system.
8. If you include graphics in custom sidebars, ByAllAccounts will edit the .js file to reflect the local path.
9. Let ByAllAccounts know whether you want to apply custom styling to all users, just SSO users, or both. ByAllAccounts will adjust the settings for your firm for how you want these files used.
10. ByAllAccounts will test your files locally, deploy them to your vendor folder in the service, and enable custom styling for your firm. If you are customizing the login page, ByAllAccounts will give you a custom login URL.

CSS and JS file hierarchy

The default style declarations and dictionary definitions for CUI are controlled by base files within the service. When you provide custom .css and .js files, the settings in them override the settings in the base files.

Consider giving ByAllAccounts files that contain only the overrides that you want to provide to the system. That way, it is easy to see what customizations you intend to make. If you intend to provide files that contain only your firm's overrides:

- Within .css files, you can provide individual property value declarations, without providing the entire declaration block. For example, you could include just the font-size for a style selector.
- Within the .js files, provide all entries for any variable (var) that you customize.

CUSTOMIZING STYLES

You can customize fonts, colors, and backgrounds by editing style declarations in these files:

- CustomLoginCSS.css – for elements in the login view. Note that the login view is not used by single sign-on (SSO) users.
- CustomCuiCSS.css – for elements in the views and dialogs.

Finding the element you want to edit in the .css file

Each view, dialog, and header displayed in CUI has a section associated with it within a .css file. The files contain comments to help guide you to the section of the code, and the names of the style selectors help identify what they control.

Naming conventions of style selectors in the .css files

The style selectors within the .css files use naming conventions to help you identify what area of the UI each one affects. For the login page (CustomLoginCSS.css), the convention is:

.*<view>*-*<generalDescription>*-*<widgetType>*

For example, the following style sets the font and color for the product name label on the login page.

```
.login-productName-label {
    color: gray;
      font-size: 25px;
      font-weight: 600;
}
```

For the CustomCuiCSS.css the convention is the same but the style selector names are prefixed with .cui:

.cui-*<view>*-*<generalDescription>*-*<widgetType>*

For example, the following style sets the background canvas color and border for the total accounts value in the account list view:

```
.cui-acctlist-totalAccountsValue-canvas {
   background-color: white;
   border-bottom: 4px solid darkgray;
}
```

In the style selectors, <*view*> is a descriptive name to identify the view, dialog, button, or header. For example, here are some of them:

- initialLanding
- fiSelect – financial institution selection
- fiSQA – financial institution security question and answer
- fiProgress – progress panel for connecting to the financial institution
- fiISAC – financial institution in-session activation codes view
- wizard – styles shared across all wizard views, such as button styles

And <*generalDescription*> provides a description of the area affected by the style. For example, here are some of them:

- mainTitle
- URL
- connectingMsg
- add, save, cancel

And <*widgetType*> identifies the part of the view, dialog, button, or header such as:

- button
- canvas – the background
- label
- link

Setting global styles

The style selectors with “wizard” as the view name are global settings that are standard throughout the interface, such as for Connect and Cancel buttons.

Setting styles for mobile devices

When the browser window is very narrow (or in portrait view on a mobile device), the view header automatically adjusts; a menu button containing Profile and Logout appears in upper left. If there is a logo, it is centered, and an Add icon (+) appears in upper right.

To control the styles for those elements, look in the .css files for the styles with “narrow” in the general description of the style selector name.

Setting styles for sidebar panels

Set the canvas styles for sidebar panels in the .css files. Note that the content and format of the sidebar panels is created using HTML in the .js files.

Hiding or disabling features

There may be features you want to hide, such as the Profile and Log Out links in the header or Financial Institution links and logos when adding an account. You can suppress those features by adding the declaration `display: none;` to their styles. Although anything can be hidden, do so sparingly as removing elements may affect the layout of widgets in a view resulting in an undesirable display.

Likewise, you can disable links such as the FI URL that appears below the FI name using a style setting.

Hiding Profile link and Log Out link

The following example shows how to hide the Profile link in the CustomCuiCSS.css file by setting display to none. Hiding Profile and Logout links requires an additional step of hiding them from the narrow display as well.

```
.cui-header-myProfile-link {
   font-family: Verdana,sans-serif; font-size: 11px;
   color:#66726F;
   text-decoration: none;
   font-weight:bold;
   white-space: nowrap;
   display: none;
```

When you hide the Profile link, you must also hide it in the narrow display (for mobile) by adding `display: none;` to `cui-header-narrowMyProfile-menuItem`.

To hide the Log Out link, add `display: none;` to `.cui-header-logout-link`. Then also hide it in the narrow display (for mobile) by adding `display: none;` to `cui-header-narrowLogout-menuItem`.

Note: If you hide both the Profile link and Log Out link, be sure to hide the menu button used for both commands when the display is narrow (`cui-acctlist-narrowMenu-button`).

Hiding Financial Institution links and logos

To remove the URL from the list of institutions that show when a user is adding an account, add `display:none;` to these styles in the CustomCuiCSS.css file:

- `.cui-fiSelect-URL-link`
- `.cui-fiSelect-URL-linkOver,`
- `.cui-fiSelect-URL-link:hover`
- `.cui-fiSelect-URLnothot-link`

Note that if you turn off the display of the FI URL, you should also change how the FI search is done so that it does not include the URL in the search. Set the `includeURLinSearch` parameter to false in the CustomCuiDictionary.js file.

To remove institution logos, find `var cui_fiSelect` in the CustomCuiDictionary.js file and set `includeLogosInDisplay : "false",`

Make the FI URL displayed below the FI name inactive

To deactivate the FI URL displayed below the FI name in the list of FIs, add `pointer-events: none;` and `cursor: default;` to the `.cui-fiSelect-URL-link` style in the CustomCuiCSS.css file. The URL will still be displayed, but it will not be an active link.

CUSTOMIZING TERMINOLOGY

You can customize text on views and buttons by editing these files:

- CustomLoginDictionary.js – Controls terminology for the login view. Note that the login view is not used by single sign on (SSO) users.
- CustomCuiDictionary.js – Controls terminology for views and dialogs.

Finding the element you want to edit in the .js file

Each view, dialog, header, and footer displayed in CUI has a section associated with it within a .js file. The .js files contain comments to help guide you to the section of the code, and the names of the variables help identify what they control.

Naming conventions of variables in .js files

The variable names used within the .js files help you identify what area of the UI each variable affects. Each variable uses the same descriptive name as used for style selectors in the .css file to identify the view, dialog, button, or header in order to help you coordinate changes. Refer to Naming conventions of style selectors in the .css files, page 3.

CUSTOMIZING OPTIONS AND FEATURES

You can edit options and features for CUI, such as adding a logo or custom sidebars and turning on and off features, by editing the .js files:

- CustomLoginDictionary.js – Controls options and features for the login view. Note that the login view is not used by single sign on (SSO) users.
- CustomCuiDictionary.js – Controls options and features for CUI. The same principles apply for CustomCuiDictionary_2.js.

Refer to Table 1 for information on how to implement these customizations.

Table 1. Customization options and features

What you can do	What this change affects	How to do it
Add logo to login view	The login view.	Notify ByAllAccounts Support and provide the logo. ByAllAccounts will place the logo in your vendor resources file and add the relative path to your CustomLoginDictionary.js file. See Requirements for logo, page 8.
Add logo to view headers	Header in main Account View list and also throughout the Add/Edit Account view if full headers are enabled.	By default, a logo is not displayed in headers. To show a logo, in CustomCuiDictionary.js, find `var cui_header` and set `isLogoDisplayed : "true",` Provide the logo file to ByAllAccounts. If you want the logo to also appear in all headers in the Add/Edit Accounts wizard, enable full headers. In either case see Requirements for logo, page 8.

What you can do	What this change affects	How to do it
Enable full headers	By default, simple headers are used throughout the Add/Edit Account views. You can enable full headers that show the same utilities that are enabled for the main Account View header such as profile, exit, logout, etc.	To enable full headers in the Add/Edit Accounts view, set the following element in the CustomCuiDictionary.js file: within `var cui_header`, set `isFullHeaderDisplayedInWizard : "true",`
Add initial landing page	Optionally adds an initial landing page for users who have no account set up.	To add an initial landing page, set the following element in the CustomCuiDictionary.js file: within `var cui_initialLanding` set `isViewDisplayed : "true",` Use HTML within the .js file to define the content and format. Provide any images to ByAllAccounts. See Requirements for custom sidebar graphics, page 8.
Add custom sidebar panels	In most views, you can optionally define a custom sidebar panel.	If you want add custom sidebar panels, define the content and format using HTML within the .js files. Each variable for which you can create a sidebar contains an element that has "SBContent" within the name, such as `FILoginSBContent`, to indicate where to enter the HTML. In the corresponding .css file, define the background canvas for the sidebar. If you are including graphics in your sidebar panels, provide them to ByAllAccounts Support. See Requirements for custom sidebar graphics, page 8.
Change the footer button alignment in views and dialogs	Footer action buttons are right aligned by default. When you change them to left aligned, the button order also changes.	For the login page, find `var LoginPage` in CustomLoginDictionary.js and set `buttonLayoutOrder : "left",` For the rest of the UI, find `var cui_footer` in CustomCuiDictionary.js and set `buttonLayoutOrder : "left",`
Disable searching for institutions by URL	By default, users can search for an institution by name or URL when adding an account. You can change that so they can only search by name.	Find `var cui_fiSelect in` CustomCuiDictionary.js and set `includeURLinSearch : "false",` If you disable searching by URL, you may want to also remove the URL from the list of institutions by editing the .css file; refer to Hiding or disabling features, page 4.

What you can do	What this change affects	How to do it
Hide financial institution (FI) logos when selecting an FI	By default, logos for a few FIs are displayed for quick selection. You can remove them.	Find `var cui_fiSelect` in CustomCuiDictionary.js and set `includeLogosInDisplay : "false",`
Remove elements from the display or inactivate links	Removing elements may affect layouts so use sparingly.	Deactivate links or remove elements by hiding them using style settings in the .css file. Refer to Hiding or disabling features, page 4.

Requirements for logo

The logo file must be Portable Network Graphics (PNG) format and must fit in a space of 55h by 180w. Note that the measurement is in pixels, and the filename must have the .png extension. Be sure that the background of the logo is transparent if you want to display it on a non-white background.

Requirements for custom sidebar graphics

There are currently no standard requirements for the custom sidebar graphics, although a .png format is recommended. If you use graphics, ByAllAccounts will edit the .js file(s) to reflect the local path. ByAllAccounts will not edit your graphics.

Appendix A: CUI Error Messages

This appendix lists the error messages that can be generated by AccountView Consumer User Interface (CUI) and the reason for each. In some cases, the comparable production error code is shown.

These error messages are not customizable.

Persistent messages for static state of an account

Error Message displayed in main view	Account's state
Cannot connect. Missing <login> and/or <password>.	Credential is missing the login and/or password.
One of the following based on the FI's support: ▪ Cannot connect. The <login> or <password> is incorrect. ▪ Cannot connect. The <login>, <password>, or answer to a security question, or <activation code> is incorrect. ▪ Cannot connect. The <login>, <password>, or <activation code> is incorrect. ▪ Cannot connect. The <login>, <password>, answer to a security question, or <activation code> is incorrect. Note: <login>, <password>, and <activation code> are replaced with the financial institution's own terminology.	Error code is 1007: Due to bad login, password, security question and answer (SQA), or in session activation code (ISAC).
We are having trouble accessing this institution. Please check back later.	Error code is not 1007 (failed to login or failed to find accounts or failed account creation or failed aggregation).
Cannot verify credentials due to institution limitation.	No error code: No connection has been attempted on the credential because FI does not support test credential (created by advisor in AccountView).
No connection attempt made yet	No error code: No connection has been attempted on the credential because no connect performed (created by advisor in AccountView).
Institution requires manual account setup.	Error code is 1006 but no accounts on the credential because FI does not support account discovery.
Institution is not yet supported.	Institution is a FI request and has all necessary data (FI name, FI URL, login, password, and account number). Note: Message is available for firms that allow advisors or clients to make FI requests.
Institution is not yet supported. A login, password, and account number are required before support can be processed.	Institution is a FI request and does not have all necessary data (created by advisor in AccountView). Note: Message is available for firms that allow advisors or clients to make FI requests.

Possible messages during Add or Edit

Error Message on connection page	Short Error Message for 'Connection Progress' table at end of Add operation	Error Case during Add or Edit Account
Cannot find your institution	N/A	When attempting to add an account the Financial Institution cannot be found.
Cannot verify credentials due to institution limitation.	Institution limitation	Connection failed because FI does not support Test Credential.
Credentials verified but unable to discover accounts at this institution.	Institution limitation	Connection failed because FI does not support Account Discovery.
No accounts found at this institution.	Accounts not found	Connection failed to find any accounts. (During Add Account only)
We are having trouble accessing this institution. Please check back later.	We failed, check later	Connection failed with a non-1007 error (system failure during login, account discovery, or account creation).
Cannot connect. The <login> or <password> is incorrect. Note: <login> and <password> replaced with the financial institution's own terminology	Incorrect credentials	Connection failed with a 1007 error when connecting on the login and password page. User needs to change input.
Cannot connect. The answer to the security question is incorrect.	Incorrect credentials	Connection failed with 1007 error when connecting on the security question and answer (SQA) page. User needs to change input.
Cannot connect. The information you provided is incorrect.	Incorrect credentials	Connection failed with 1007 error when connecting on the "in session activation code" (ISAC) page. User cannot change input so only option is to cancel and start over.

Appendix B: CUI postMessages for Add Account workflow

Consumer UI provides analytical information about the user experience in Consumer UI when Consumer UI is contained in a customer's web page. Consumer UI generates messages during the user's Add Account process that can be consumed by the parent frame. The messages are sent via the window.postMessage() method to the parent frame. The parent frame may listen for these messages via a window.addEventListener() method that listens for an event of type "message" whose origin is https://www.byallaccounts.net.

These messages are generated during the Add Account process. They are not generated during an Edit Account. Each message consists of two parts:

- the numeric workflow identifier, which indicates the 'Add' process in which the message was generated.
- the text that describes the event, as listed in Events and messages, page B-2.

The numeric identifier is needed because the user may start a second 'Add' process before the first has completed its connection. The identifier ties together all the messages caused by the successful (or unsuccessful) creation and aggregation of an account.

Example postMessages

Here are example postMessages:

```
1:openAccountView
1:selectSearchedFinancialInstitution
1:addAccountUsernamePassword
1:connectionInProgress
1:connectionCompleteWithSuccess
2:selectSearchedFinancialInstitution
2:addAccountUsernamePassword
2:addAccountSQAA
2:connectionInProgress
3:selectPopularInstitution
3:cancelAddAccountUsernamePassword
2:connectionCompleteWithSuccess
```

These messages were generated by these events:

- Successfully added account (workflow 1)
- Successfully added an account that required an answer to a security question (workflow 2)
- Started to add another account (before the 2nd account finished aggregating) and then canceled (workflow 3)

Events and messages

The following table lists all messages generated, and the event that triggers them, during the Add Account workflow.

When...	Message that is posted	Actual event that triggers message
Account View is loaded	openAccountView	First Consumer UI view is displayed (independent of which view)
User searches for an FI in the search bar	selectSearchedFinancial-Institution	User selects an FI from the text list
User clicks on one of the popular institutions	selectPopularInstitution	User selects a popular FI tile
User enters account's login/password	addAccountUsername-Password	User invokes 'Connect' in the Login/pw view
User decides to cancel out of login/pw view	cancelAddAccountUsername-Password	User invokes 'Cancel' in the Login/pw view
User served error on the login/pw view	errorAddAccountUsername-Password	System displays connection error message in the Login/pw view
User provides activation code	addAccountISAC	User invokes final 'Connect' in the ISAC view
User decides to cancel out of ISAC (activation code) view	cancelAddAccountISAC	User invokes 'Cancel' in the ISAC view
User served error on the ISAC view	errorAddAccountISAC	System displays connection error messagein the ISAC view
User provides answer to security question	addAccountSQAA	User invokes 'Connect' in the SQAA view
User decides to cancel out of SQAA view	cancelAddAccountSQAA	User invokes 'Cancel' in the SQAA view
User served error on the SQAA view	errorAddAccountSQAA	System displays connection error message in the SQAA view
Connection is in progress	connectionInProgress	System displays 'connection in progress' view
Connection completed	connectionCompleteWith-Success	System determines final connection state of account is successful, even if 'in progress view' no longer displayed
Connection failed	connectionCompleteWith-Failure	System determines connection results in a "Fix it" state
User credentials are not correct, clicks "Fix It"	fixIt	User selects "fixIt" in the progress view
User clicks "Done", exiting out of 'Add Account' wizard	doneAccountView	User clicks "Done" (depending on how ConsumerUI was entered this could result in leaving Consumer UI or returning to main Consumer UI account list)
User clicks "X", exiting out of 'Add Account' wizard	xAccountView	User clicks "X" (from any view) to finish adding accounts (again this could result in leaving Consumer UI or returning to main Consumer UI account list)