



# **ByAllAccounts Connect Suite - Account Summary with Account Linking**

MCSuite-3.11-20250424

© 2025 Morningstar. All Rights Reserved.

ByAllAccounts

Account Linking version: 3.11

Account Summary version: 1.8

Document Version: MCSuite-3.11-20250424

Guide generated on: 4/22/2025

Please report incorrect or missing information.

Telephone: 1-866-856-4951 Option 1

Byallaccounts-support@morningstar.com

Web: <http://byallaccounts.morningstar.com>

## Table of Contents

---

<b>About this document</b> .....	<b>1</b>
When to use this document .....	1
Changes in this version .....	1
<b>About ByAllAccounts Connect Suite</b> .....	<b>1</b>
How the components work together .....	2
<b>About Account Summary</b> .....	<b>2</b>
Optional features .....	2
<b>About Account Linking</b> .....	<b>3</b>
See it in action .....	3
<b>Known limitations</b> .....	<b>3</b>
User types .....	3
Browser support .....	3
<b>Integrating ByAllAccounts Connect Suite</b> .....	<b>3</b>
Authentication .....	4
Permissions to allow framing .....	4
Whitelisting requirements .....	4
Whitelisting and your development environment .....	4
Download and install the components .....	5
Account Summary component .....	5
Account Linking component .....	6
Incorporate the components into your parent page .....	7
About attributes .....	7
Instantiating ByAllAccounts Connect Suite .....	9
Example setting all attributes for Account Summary .....	11
Example setting all attributes for Account Linking .....	12
About events .....	13
Critical event only in Account Summary .....	14
Critical events in both components .....	15
Critical event in Account Linking .....	16

---

Standard events in Account Linking .....	16
Monitoring for events .....	17
Managing critical events .....	17
accountSetup .....	17
badAuthentication .....	17
badConfiguration .....	18
internalError .....	18
userExit .....	18
Managing standard events in Account Linking .....	18
dataChanged .....	19
dataChangedAsync .....	19
dataChangedAsyncAuthenticate .....	19
userDeclinedAgreement .....	19
<b>Customize the user interface .....</b>	<b>19</b>
Applying the user interface customizations .....	20
<b>Appendix A: Cumulative changes .....</b>	<b>1</b>

---

## About this document

This document describes how to implement the ByAllAccounts Connect Suite of components that provide access to the ByAllAccounts<sup>SM</sup> (BAA) aggregation service. The components are:

- Account Summary
- Account Linking

This document describes these components, steps for embedding them in a web application, methods of authentication, and ways to customize them.

## When to use this document

This guide describes integration of the custom elements with your application. The Account Linking component can be implemented without the other components if you have your own Account List implementation. The following guides describe how to implement the Account Linking component alone. Choose the guide according to the authentication method you use.

- Use [Introduction to ByAllAccounts Connect Account Linking](#) if you are using ByAllAccounts DataConnect API for single sign on (SSO).
- Use [Introduction to ByAllAccounts Connect Account Linking using Morningstar Authentication Instructions](#) if you are using Morningstar Authentication for single sign on (SSO).

## Changes in this version

The changes in this release are described in [Appendix A: Cumulative changes](#) along with important information about other recent changes.

## About ByAllAccounts Connect Suite

ByAllAccounts Connect Suite is comprised of components that are each independently implemented in the main page of the parent application, which is referred to in the rest of this guide as the parent page.

- Account Summary - this component shows the investor's account summary information and facilitates user maintenance of the accounts. It provides a summary of the linked accounts, balances, and connectivity status and allows investors to manage their accounts on an ongoing basis.
- Account Linking - this component is used by Account Summary to add new accounts and maintain account credential information. The component can be used by investors to link accounts across 15,000 financial institutions.

The components are each implemented in their own iframe and are independently sized and positioned in the parent page. The parent page manages authentication and instantiation of each component, monitors for and handles critical events in each component, and manages hiding and revealing the Account Linking custom element.

---

## How the components work together

The components are used together, in cooperation, to provide a complete solution. In a typical usage scenario, a user action in the Account Summary component will trigger an event that requires the Account Linking component to be launched by the parent page. Conversely, as the user performs certain actions in the Account Linking component that may change the user data displayed in Account Summary view, the Account Summary component needs to react to refresh its data. The parent page provides the coordination, or "glue", to enable this experience. The parent page controls when each component is visible and when they call each other. Typically Account Summary is implemented to always show in the parent page. Account Linking may be invoked as a popup by the parent page as needed to complete an action called for by an event in the Account Summary component.

## About Account Summary

The Account Summary component provides access to account data for the ByAllAccounts<sup>SM</sup> aggregation service. It is a W3C custom HTML element that implements the account summary functionality within a parent page of your application. The Account Summary component is used in conjunction with the Account Linking component. The Account Summary component provides a page of summary credential and account information for an authenticated investor, in groups such as:

- In Progress – financial institution credentials for which newly added accounts are currently being aggregated
- Needs Attention – financial institution credentials that need user action such as providing additional authentication information
- Main – all credentials and accounts for the investor
- Pending Requests – requests for new financial institution Financial Institution connections

The Account Summary component can be fully customized to integrate into your parent page, with customizations to terminology, styles, and features. This document describes steps for embedding it in a web application, methods of authentication, and ways to customize it.

To fully integrate Account Summary with Account Linking, you will need to work with a ByAllAccounts Implementation Manager who will help with the internal ByAllAccounts setups required including creating a Firm for you in our system, providing an administrator login and password to perform API calls, etc.

## Optional features

There are optional Account Summary features that can be enabled for your Firm by ByAllAccounts.

- By default investor's data is displayed by Institution. An optional "Account Type" view can be enabled for your Firm to display the investor's data organized by Assets and Liabilities, as well as additional breakdown by account category.
- A "Net Worth Chart" can be enabled for your Firm. It displays a time-series chart consisting of historical data from aggregated accounts and calculates changes over time.

---

## About Account Linking

The Account Linking component provides access to the ByAllAccounts<sup>SM</sup> aggregation service. It is a W3C custom HTML element that implements the account setup functionality for aggregation within another application.

Account Linking supports adding and maintaining accounts within the aggregation system. The aggregated account data, including balance, holding, and transaction information, are available via ByAllAccounts APIs.

Account Linking can be fully customized to integrate into your parent page, with customizations to terminology, styles, and features.

To fully integrate Account Summary and Account Linking you will need to work with a ByAllAccounts Implementation Manager who will help with the internal ByAllAccounts setups required including creating a Firm for you in our system and providing an administrator login and password to perform API calls, etc.

## See it in action

To see Account Linking in action, click [here](#).

## Known limitations

There are currently no known limitations. The capabilities, interfaces, and behaviors of the ByAllAccounts components are subject to change as we make refinements based on customer feedback.

## User types

The components can only be used with BAA Investor users and those users must:

- have full Read-Write permission to their own data
- be Single Sign On (SSO) users
- be in a BAA Firm that is licensed to use Account Linking and Account Summary

## Browser support

Up-to-date versions of these browsers are fully supported:

- Chrome (including incognito mode), Firefox, Microsoft Edge, and Safari

## Integrating ByAllAccounts Connect Suite

The steps and processes that you need to implement and manage to integrate ByAllAccounts Connect Suite into your parent page include:

- Authentication
- Permissions to allow framing
- Downloading and installing both components
- Incorporating the components into your parent page

- 
- Handling events
  - Optionally, customizing the user interface of the components

## Authentication

Authentication is handled by the parent page before it invokes the components.

Authentication consists of invoking a DataConnect API method (<SESSIONAUTHRQ> or <SAMLAUTHRQ>) to authenticate the investor and obtain values for *jsessionId* and *csrfToken*. The *jsessionId* and *csrfToken* values must be passed to the components using the *auth-context* attribute.

For details you can refer to the Single Sign-on Operations section in the [DataConnect V4 Ultra User Guide](#).

If the parent page passes in read-only authorization for the user through *auth-context* to the component, the user can see data in the Institution view and Account Type view (if it is enabled) for a given investor and see various options that are available in the component, but the options are disabled. This capability enables Advisors to monitor information for their investors. Account Linking cannot be invoked when the user is in read-only mode.

## Permissions to allow framing

The Account Linking and Account Summary components use a Content Security Policy (CSP) that controls which domains are allowed to frame them. In order to allow a domain to frame the component, the parent page's domain(s) must be configured in the component's whitelist within the ByAllAccounts service. Your ByAllAccounts Implementation Manager will assist you with this configuration.

## Whitelisting requirements

For your domains to be whitelisted, they must meet the following requirements for ByAllAccounts Connect whitelist entries:

- Must be all lower case
- Must not have a trailing slash
- Can include a port number
- Cannot be "localhost"
- Must be https (not http)

Additionally

- Asterisk (\*) wildcards are supported

## Whitelisting and your development environment

In production, the ByAllAccounts components absolutely require proper whitelisting. However you may perform development on a system with a domain name that does not meet the whitelisting requirements (e.g. localhost). To work within these constraints, you can:

- 
- Use a mechanism to provide a domain name that meets the requirements, such as a proxy server or a third-party platform.
  - Consider available browser options to relax the browser security checking. This modification would be appropriate only for a development environment.

## Download and install the components

Each component must be downloaded and installed.

### Account Summary component

The *mstar-aggregation-consumer-accountsummary* component is built with Angular and can be used in frameworks such as React, and Vue, or in vanilla JS applications.

These instructions are for acquiring, customizing, and incorporating Account Summary. To fully implement it, work with your ByAllAccounts Implementation Manager.

#### Step 1

Download the packaged file for Account Summary from <https://www.byallaccounts.net/WebPortfolio/mstar-aggregation-consumer-accountsummary.tgz>.

#### Step 2

Install the package by running the following command in your development environment:

```
npm install mstar-aggregation-consumer-accountsummary.tgz
```

#### Step 3

Confirm the *mstar-aggregation-consumer-accountsummary* folder is under the *node\_modules* directory. The folder contains:

- *mstar-aggregation-consumer-accountsummary.js*, which is the JavaScript bundle for the Account Summary component.
- *as\_assets* folder, which contains various configuration files and images for the component.
- *package.json* file that describes the npm package.

The build system should ensure that the *as\_assets* folder and the main JavaScript file (*mstar-aggregation-consumer-accountsummary.js*) of the component are placed under the application's distribution (*/dist*) folder from where it is being served.

#### Step 4

Customize the user interface. See the "[Customize the user interface](#)" on page 19.

#### Step 5

---

Import the necessary JavaScript files into your HTML parent page and create an instance of the *mstar-aggregation-consumer-accountsummary* tag element.

### Step 6

Incorporate Account Summary into your parent page. See ["Incorporate the components into your parent page" on page 7](#).

**Note:** If you are installing Account Summary to an implementation that uses an existing Account Linking component, you must upgrade the Account Linking component to the most recent version as described in ["Account Linking component" on page 6](#).

## Account Linking component

**Note:** When adding Account Summary to an existing implementation of Account Linking, you must use these instructions to download a new copy of Account Linking.

The *mstar-aggregation-consumer-accountsetup* component is built with Angular and can be used in frameworks such as React, and Vue, or in vanilla JS applications.

These instructions are for acquiring, customizing, and incorporating Account Linking. To fully implement it, work with your ByAllAccounts Implementation Manager.

### Step 1

Download the packaged file for Account Linking from <https://www.byallaccounts.net/WebPortfolio/mstar-aggregation-consumer-accountsetup.tgz>.

### Step 2

Install the package by running the following command in your development environment:

```
npm install mstar-aggregation-consumer-accountsetup.tgz
```

### Step 3

Confirm the *mstar-aggregation-consumer-accountsetup* folder is under the *node\_modules* directory. The folder contains:

- *mstar-aggregation-consumer-accountsetup.js*, which is the JavaScript bundle for the Account Linking component.
- *assets* folder, which contains various configuration files and images for the component.
- *package.json* file that describes the npm package.

The build system should ensure that the *assets* folder and the main JavaScript file (*mstar-aggregation-consumer-accountsetup.js*) of the component are placed under the application's distribution (*/dist*) folder from where it is being served.

### Step 4

Customize the user interface. See the ["Customize the user interface" on page 19](#).

---

## Step 5

Import the necessary JavaScript files into your HTML parent page and create an instance of the *mstar-aggregation-consumer-accountsetup* tag element.

## Step 6

Incorporate Account Linking into your parent page. See ["Incorporate the components into your parent page" on page 7](#).

## Incorporate the components into your parent page

When the ByAllAccounts Connect Suite components are incorporated into a parent page, the Account Summary is visible and acts as the primary controlling entity for the Suite. The parent page must monitor for the *accountSetup* event issued by the Account Summary component and relay that to the Account Linking component to tell it which workflow to start when it exposes the component. That's the only information the parent page needs to relay between the two components, but the parent page must also listen for and manage critical events issued from both components. Additionally, the required attribute (*auth-context*) must be set for each component to manage authorization. Optionally, other attributes may be set for customizations.

This section includes:

- Descriptions of all attributes for the components
- Code for instantiating the components in your parent page
- Code samples for each component, showing all applicable attributes
- Descriptions of the critical and standard events, how to monitor for them, and how to manage them.

## About attributes

The following table describes the attributes for the Account Linking component and the Account Summary component. The only required attribute is *auth-context*. Refer to the table below for details.

Any of these attributes can be set (and reset) at any time but *translate-file-path*, *override-css-file*, and *ui-config-file* (for Account Linking) typically do not change during the lifetime of a parent page. An example of when *auth-context* might be reset during the lifetime of a parent page would be handling of an authorization expiring.

For information about events, see ["About events" on page 13](#).

Attribute	Req'd?	Parameters and Description
<i>auth-context</i>	Yes	<b>Possible Parameters</b> Must contain one of: <ul style="list-style-type: none"><li>▪ <code>{jsessionId: &lt;jsessionIdValue&gt;, csrfToken: &lt;csrfTokenValue&gt;}</code></li><li>▪ <code>{}</code></li></ul> <b>Description</b> The <i>auth-context</i> attribute is used for user authentication. It is required and may be reset

Attribute	Req'd?	Parameters and Description
		<p>any number of times during the lifetime of the parent page. For example, it would need to be reset after a timeout.</p> <p>It must contain the parameter pair <code>jsessionId/csrfToken</code> or an empty object.</p> <p>When the user is authenticated via SSO, the <code>jsessionId</code> and <code>csrfToken</code> pair must be included, using the form:</p> <pre>{jsessionId: &lt;jsessionIdValue&gt;, csrfToken: &lt;csrfTokenValue&gt;}</pre> <p>To prevent further use of the component by the user, use the empty value to clear the authentication data from the component:</p> <pre>{}</pre>
<i>route</i>	No	<p><i>Only applies to Account Linking.</i></p> <p><b>Possible Parameters</b></p> <p>The event detail strings that come from the Account Summary <i>accountSetup</i> event described in "<a href="#">About events</a>" on page 13.</p> <p><b>Description</b></p> <p>When any one of certain user actions is performed in the Account Summary component, a specific response is expected in a particular workflow of the Account Linking component. To effect the response, the parent page must be listening for the <i>accountSetup</i> event from the Account Summary component and pass its event detail string as the parameter on the <i>route</i> attribute. The parameter specifies the workflow to initiate in Account Linking. When a workflow completes, the Account Linking component tells the parent page that it is done.</p> <p>The <i>route</i> attribute must be set after the <i>auth-context</i> attribute because user authentication must be set before the component can attempt the operation. If the <i>auth-context</i> is not set before the <i>route</i> attribute, a <i>badAuthentication</i> event will occur.</p>
<i>translate-file-path</i>	No	<p><b>Possible Parameters</b></p> <p>The path to the custom <i>en.json</i> file.</p> <p><b>Description</b></p> <p>The <i>translate-file-path</i> attribute identifies the URL that is the path to the <i>en.json</i> files that you can create to provide custom terminology for certain text and labels. The path is relative to the base <i>href</i> of the deployed application. Note: Each component has its own <i>en.json</i> file with different terminology.</p>
<i>ui-config-file</i>	No	<p><i>Only applies to Account Linking.</i></p> <p><b>Possible Parameters</b></p> <p>Path and file name to your custom user interface (UI) configuration file.</p>

Attribute	Req'd?	Parameters and Description
		<p><b>Description</b></p> <p>The <i>ui-config-file</i> attribute identifies the URL that is the path to and the name of the file that contains options for turning on/off UI sections in the Account Linking component. The path is relative to the base <i>href</i> of the deployed application.</p>
<i>override-css-file</i>	No	<p><b>Possible Parameters</b></p> <p>The path and file name for your custom cascading style sheet (CSS) file.</p> <p><b>Description</b></p> <p>The <i>override-css-file</i> attribute identifies the URL that is the path to and file name of a custom CSS file containing overrides to the default Morningstar Design System (MDS) styles. Note: Each component has its own CSS file.</p>
<i>custom-fonts</i>	No	<p><b>Possible Parameters</b></p> <p>The path and name for your font definitions.</p> <p><b>Description</b></p> <p>The <i>custom-fonts</i> attribute identifies the URL that is the path to and file name of your custom font definition file that will override to the default Morningstar Design System (MDS) fonts. Note: Each component has its own fonts file.</p>
<i>iframe-style</i>	No	<p><b>Possible Parameters</b></p> <p>Standard CSS inline style values. For example, <code>iframe-style="height:50vh; border:dashed;"</code></p> <p><b>Description</b></p> <p>The <i>iframe-style</i> attribute applies CSS inline styles to the iframe of the component. Clearing this attribute after it has been set reverts the iframe style of the component back to its original state.</p>

## Instantiating ByAllAccounts Connect Suite

In a typical usage scenario of the ByAllAccounts Connect Suite, the Account Summary component is displayed first and occupies a section on the parent page.

When the user performs an action in the Account Summary component, the Account Linking component is launched in a modal dialog where the user steps through the wizard flow to perform the associated operation.

When the user completes the operation in the Account Linking component, the user exits that component, the dialog is dismissed, and control is returned back to the Account Summary component.

These interactions between the Account Summary and Account Linking components can repeat multiple times during a user session, and at each time the Account Linking component needs to be launched in the dialog and then dismissed.

---

This launching of the Account Linking component can potentially be an expensive and lengthy operation as it involves network activity to load the component's JS bundles and to establish the session. To make this behavior as performant and smooth as possible, and to provide a good user experience to the user, it is recommended that both the Account Summary and the Account Linking components be preloaded and instantiated at the start of the parent application session, when the user logs in. The preloaded Account Linking component should then be placed in the modal dialog, ready to be used but kept hidden until it is needed.

The following code sample shows how to instantiate the components as recommended, including setting the *auth-context* attribute for each component for authentication purposes. You may optionally set additional attributes on the components as shown in the code examples in ["Example setting all attributes for Account Summary"](#) on page 11 and ["Example setting all attributes for Account Linking"](#) on page 12.

**Note:** Your parent page *must* additionally listen for and manage the critical events which are described in ["About events"](#) on page 13.

```

<!-- Load the main JavaScript bundle file of the Account Summary component -->
<script src="./mstar-aggregation-consumer-accountssummary.js"></script>
<!--Instantiate Account Summary (summaryCE) component -->
<script>
  summaryCE = document.createElement("mstar-aggregation-consumer-accountssummary");
  summaryCE.setAttribute("auth-context", JSON.stringify(
  { sessionId: "<jsession id>",
    csrfToken: "<csrf token>"
  }));

  // Add an event listener that listens to accountSetup events fired by the Account Summary
  (summaryCE) component, and when fired, display the Account Linking (setupCE) component
  summaryCE.addEventListener('accountSetup', (e) => {
    // Set the route attribute of the Account Linking (setupCE) component using the value
    passed in the event
    setupCE.setAttribute('route', e.detail);
    // Show the Account Linking (setupCE) modal dialog
    Document.getElementById('dialog').showModal();
  });

  content.appendChild(summaryCE);
</script>

<!--Preload the main JavaScript bundle file of the Account Linking (setupCE) component -->
<script src="./mstar-aggregation-consumer-accountsetup.js"></script>
<!--Instantiate Account Linking (setupCE) component and keep it hidden-->
<script>
  setupCE = document.createElement("mstar-aggregation-consumer-accountsetup");
  setupCE.setAttribute("auth-context", JSON.stringify(
  { sessionId: "<jsession id>",
    csrfToken: "<csrf token>"
  }));

  // Add an event listener that listens to userExit events fired by the Account Linking
  (setupCE) component, and when fired, hide the modal dialog
  setupCE.addEventListener('userExit', (e) => {
    // Hide the Account Linking (setupCE) modal dialog
    Document.getElementById('dialog').close();
  });

  // Add the Account Linking (setupCE) component to the dialog element and keep it hidden
  content = document.getElementById('dialog');
  content.appendChild(setupCE);
</script>

```

## Example setting all attributes for Account Summary

This section shows a code example for launching Account Summary with all applicable attributes.

This sample integration code:

- Loads the Account Summary component.
- Instantiates the component in the page.

- 
- Sets all common attributes. See ["About attributes" on page 7](#) for detailed information about attributes.
  - Makes Account Summary visible in the page.

Attributes are available to:

- customize the user interface

attributes: *translate-file-path, override-css-file, custom-fonts*

These attributes typically do not change during the lifetime of the parent page. We recommend setting these attributes first.

- provide authentication

attribute: *auth-context*

Authentication can change during the life of the parent page. When authentication expires or a new user is authenticated, you need to reset the *auth-context* attribute.

In this example, *hostname* should be replaced with your host name for those files.

```
<!-- Load the main JavaScript bundle file of the Account Summary component -->
<script src="./mstar-aggregation-consumer-accountsummary.js"></script>
<script>
  mstarWebComp = document.createElement("mstar-aggregation-consumer-accountsummary");
  mstarWebComp.setAttribute("ui-config-file", "https://hostname/customassets/config/ui-
config.json");
  mstarWebComp.setAttribute("translate-file-path", "https://hostname/customassets/i18n/");
  mstarWebComp.setAttribute("override-css-
file", "https://hostname/customassets/css/corporatestyle.css");
  mstarWebComp.setAttribute("custom-
fonts", "https://fonts.googleapis.com/css?family=Lobster");
  mstarWebComp.setAttribute("auth-context", JSON.stringify(
    { sessionId:
"212764322EB9DBEBED880425DE3216EB.s1a",
    csrfToken:
"DBC63BDE361C192B3CEF641B4C551DD8AC27B4A0276E91"
  }));
  content.appendChild(mstarWebComp);
</script>
```

## Example setting all attributes for Account Linking

This section shows a code example for launching Account Linking with all applicable attributes.

This sample integration code:

- Loads the Account Linking component.
- Instantiates the component in the page.
- Sets all common attributes. See ["About attributes" on page 7](#) for detailed information about attributes.
- Makes Account Linking visible in the page.

Attributes are available to:

- 
- customize the user interface

attributes: *ui-config-file*, *translate-file-path*, *override-css-file*, *custom-fonts*

These attributes typically do not change during the lifetime of the parent page. We recommend setting these attributes first.

- provide authentication

attribute: *auth-context*

Authentication can change during the life of the parent page when authentication expires or a new user is authentication. If you reset the *auth-context* attribute, you must then reset the *route* attribute.

In this example, *hostname* should be replaced with your host name for those files.

```
<!-- Load the main JavaScript bundle file of the Account Linking component -->
<script src="./mstar-aggregation-consumer-accountsetup.js"></script>
<script>{
  // This event listener listens to accountSetup events fired by the Account Summary
  // component, and when fired it launches the Account Linking component
  summaryCE.addEventListener('accountSetup', (e) => {
    mstarWebComp = document.createElement("mstar-aggregation-consumer-accountsetup");
    // Set attributes
    mstarWebComp.setAttribute("ui-config-file", "https://hostname/customassets/config/ui-
    config.json");
    mstarWebComp.setAttribute("translate-file-path", "https://hostname/customassets/il8n/");
    mstarWebComp.setAttribute("override-css-
    file", "https://hostname/customassets/css/corporatestyle.css");
    mstarWebComp.setAttribute("custom-
    fonts", "https://fonts.googleapis.com/css?family=Lobster");
    // Pass in the auth context of the logged in user to the mstar component
    mstarWebComp.setAttribute("auth-context", JSON.stringify(
      { sessionId: "212764322EB9DBEBED880425DE3216EB.s1a",
        csrfToken: "DBC63BDE361C192B3CEF641B4C551DD8AC27B4A0276E91"
      }
    ));
    content.appendChild(mstarWebComp);
  });
}</script>
```

## About events

The Account Summary component (*mstar-aggregation-consumer-accountsummary*) and the Account Linking component (*mstar-aggregation-consumer-accountsetup*) can issue two types of events, which are caused by a trigger action.

How an event is resolved depends on the type of event:

- Critical events must be monitored for and managed by the parent page. Typically, these events signal that there is an error that the component cannot resolve or that the user has completed their work. Additionally, the Account Summary component can issue an event that signals there is a parameter value that the parent page must pass to Account Linking.

- Standard (informational) events occur in Account Linking only and provide information for the parent page. Informational events may be monitored, but are managed by the component. For example, Account Linking handles any problems that occur when the credentials entered for a financial institution cannot be used to successfully authenticate at that institution.

"Monitoring for events" on page 17 provides a code sample for monitoring for a specific event. "Managing critical events" on page 17 describes ways in which the parent page may need to handle the critical events. "Managing standard events in Account Linking " on page 18 describes managing the informational events.

## Critical event only in Account Summary

The following critical event only occurs in Account Summary.

Trigger	Name	Event Detail String	Description
A user action in the Account Summary component requires the parent page to provide <i>route</i> information to the Account Linking component	accountSetup	A string to pass as a parameter value to Account Linking on the <i>route</i> attribute .	<p>This event is triggered by a user action in Account Summary. The parent page must pass the resulting event detail string as the parameter on the <i>route</i> parameter for Account Linking to tell Account Linking what workflow to open.</p> <p>Examples that can cause the event to fire include when user clicks:</p> <ol style="list-style-type: none"> <li>1) "Link More Accounts" button.</li> <li>2) "Connect Accounts" menu option and selects "Enter New Credential" button in the dialog.</li> <li>3) "Connect Accounts" menu option and selects "Link More Accounts" button in the dialog.</li> <li>4) "Re-Authenticate" menu action.</li> <li>5) The edit pending request action.</li> <li>6) The account name.</li> </ol> <p>For information about managing this event, refer to <a href="#">"accountSetup" on page 17.</a></p>

## Critical events in both components

The following critical events must be monitored in both components.

Trigger	Name	Event Detail String	Description
A request to the server finds user's authentication context is not valid (resulted in a 401 Unauthorized)	badAuthentication	"Unauthorized access"	<p>User's authentication context is either not valid, the session timed out, or their authentication expired. The user will no longer be able to interact with the component after a <i>badAuthentication</i> (401) event until the component receives changes to the <i>auth-context</i> attribute that provide valid authentication data.</p> <p>The user's authentication could end at any time. Whenever the parent page receives a <i>badAuthentication</i> event it must re-authenticate and invoke a <i>setAttribute auth-context</i> with the refreshed header information before the user can proceed with the operation they had started.</p> <p><b>Note:</b> When authentication fails for either component, it is recommended that authentication is reestablished for both.</p> <p>For information about managing this event, refer to <a href="#">"badAuthentication" on page 17</a>.</p>
Component is not licensed	badConfiguration	"User not licensed"	<p>If the component is not licensed at initiation, there will be a blank area in the parent page where the component is expected. If the configuration changes while the component is in use, the component will be in a blocked state and greyed out on the parent page.</p> <p>For information about managing this event, refer to <a href="#">"badConfiguration" on page 18</a>.</p>
A request to the server resulted in an error other than 401 or an internal code error occurred	internalError	One of: "Forbidden access" "Service unavailable" "Unknown server error" "Unknown	<p>An unexpected error occurred during the user's interaction with the component. The parent page must remedy the problem and possibly display a message to the user.</p> <p>Some causes include:</p> <p>Forbidden access: User does not have the necessary permissions; user may have read only permissions.</p> <p>Service unavailable: Possibly an intermittent service</p>

Trigger	Name	Event Detail String	Description
		error" "Resource not found"	interruption. Resource not found: Could be a misnamed path or file. For information about managing this event, refer to <a href="#">"internalError" on page 18.</a>

## Critical event in Account Linking

The following critical event must be monitored in Account Linking.

Trigger	Name	Event Detail String	Description
User completes action and/or exits the operation	userExit	"User exit"	User has done one of the following: completed the account add, completed the account edit, canceled out of the workflow, or declined the user agreement.  For information about managing this event, refer to <a href="#">"userExit " on page 18.</a>

## Standard events in Account Linking

The following standard events can occur in Account Linking. These events that may be monitored by the parent page but are handled by the Account Linking component

Trigger	Name	Event Detail String	Description
User makes a change to credential or account	dataChanged	"Data changed"	User makes a change to a credential or account that may result in synchronous changes to any or all of the following: authentication status, accounts linked to the credential, and/or financial data for those accounts.
User initiates an aggregation operation	dataChangedAsync	"Async Data change"	User initiates the service to run the asynchronous process of composite "aggregate" which can be discover/add/aggregate, add/aggregate, or just aggregate.
User initiates an authenticate operation	dataChangedAsyncAuthenticate	"Async Authenticate Data change"	User initiates an authentication operation, which causes the service to run the asynchronous process of attempting to authenticate.
User declines	userDeclinedAgreement	"User	User does not accept the terms of the latest user

Trigger	Name	Event Detail String	Description
user agreement		declined user agreement"	agreement when prompted.

## Monitoring for events

This code sample shows how to listen for a specific event. Both components must be monitored for critical events.

```
mstarWebcomp.addEventListener('userExit', function (event) {
  console.log(' @@@@ userExit event is called. detail: ' + event.detail);
});
```

## Managing critical events

This section describes managing the critical trigger events listed in ["About events" on page 13](#). These critical events must be monitored and managed by the parent page.

### accountSetup

*Applies to Account Summary only.*

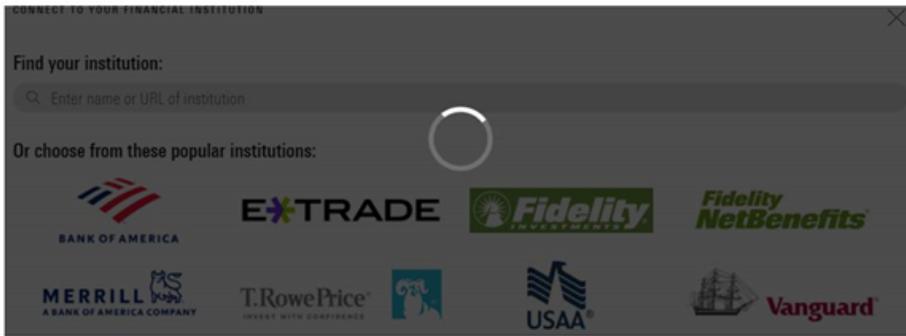
Some user actions in the Account Summary component cause it to fire its *accountSetup* event which the Account Linking component should respond to. To effect that response, the parent page must monitor for that event and provide the glue logic to handle it. The parent page must pass event detail string of the *accountSetup* event as the parameter on the *route* attribute when invoking the Account Linking component to tell the Account Linking component where in its workflow to open.

### badAuthentication

A badAuthentication event can be caused by:

- timeout: the user stops interacting with a component for a period of time and the session for the *auth-context* times out.
- authentication ended: the authorized connection ended.

For badAuthentication events, the parent page must reauthenticate the user and set a new *auth-context*. Until the reauthentication happens, the component is in a blocked state and is greyed out.



When the authentication is reestablished, the user may proceed from where they left off.

**Note:** When authentication fails for either component, reestablish authentication for both.

### **badConfiguration**

A badConfiguration event is caused by an unlicensed component.

If the component is not licensed at initiation, there will be a blank area in the parent page where the component is expected. If the configuration changes while the component is in use, the component will be in a blocked state and greyed out as it is for a badAuthentication event. The parent page needs to recognize that error and remedy the problem.

### **internalError**

The internalError event can happen for several reasons. For example, a "Resource not found" error would occur if the parent page provided an invalid credential identifier when requesting a *route* to */edit-credential*.

When an internalError event occurs, the user interface for the component remains static. The parent page needs to recognize that error and remedy the problem. The parent page can pass a message to the user, describing the type of error.

### **userExit**

*Applies to Account Linking only.*

A component sends the userExit event when the user exits it, such as by completing adding an account, completing editing an account, canceling out of the workflow, or declining the user agreement.

Upon an userExit event, the parent page could hide Account Linking or activate it again. To activate it again, if the authorization context is still valid, the parent page can set the *route* attribute to reopen Account Linking at a set point in the workflow.

## **Managing standard events in Account Linking**

*These events apply to the Account Linking component only.* This section describes managing the standard (informational) trigger events listed in ["About events" on page 13](#). These events that may be monitored by the parent page but are handled by the Account Linking component. You may want to monitor standard events to update your own back end.

---

### **dataChanged**

The `dataChanged` event lets the parent page know something has changed in Account Linking. The parent page should refresh any display of account, credential, position, and transaction data. The event is emitted each time one of these data changes occurs, so the parent page could receive this event multiple times during a single session of the Account Linking component.

### **dataChangedAsync**

The `dataChangedAsync` event lets the parent page know the user submitted a composite "aggregate" in Account Linking that triggered the service to run the complex and long-running process asynchronously. The composite "aggregate" can be `discover/add/aggregate` or just `aggregate`. When the parent page receives this event it should poll for asynchronous activity completion and then query the API to obtain the changed data.

### **dataChangedAsyncAuthenticate**

The `dataChangedAsyncAuthenticate` event lets the parent page know that an authentication was initiated in Account Linking. The asynchronous process can be complex and long-running. When the parent page receives this event, it can poll for asynchronous activity completion and then query the API to obtain the authentication status.

### **userDeclinedAgreement**

A `userDeclinedAgreement` event occurs when a user does not accept the terms of a user agreement when it is presented in Account Linking.

## **Customize the user interface**

There are multiple types of customizations you can optionally provide for the components to seamlessly blend their appearance into your parent page. Customizations must be explicitly set individually on each component. Apply these customizations using the files provided in the `assets` folder for Account Linking and the `as_assets` folder for Account Summary.

- **Terminology** – the `i18n/en.json` translation file contains all the static text displayed on pages, dialogs, and buttons within the interface. Edit this file to customize text and terminology. Configure the component to use your custom terminology file by setting the `translate-file-path` attribute. Note that the translation file for each component has a different set of terms.
- **Styles** – define an optional CSS file to override the default Morningstar Design System (MDS) styles and adjust fonts, colors, and backgrounds for elements in the display. Configure the component to use your custom CSS file by setting the `override-css-file` attribute. To use fonts other than the MDS Univers font, you can add a global entry like this to your CSS file:

```
* {
  font-family: YourFont, Verdana, sans-serif !important;
}
```

- **Fonts** – define an optional fonts file to override the default Morningstar Design System (MDS) fonts for elements in the display. Configure the components to use your custom fonts by setting the `custom-fonts`

---

attribute. These new fonts are then available to be set using the CSS file specified using the *override-css-file* attribute.

- **Optional features** – *For Account Linking only*, the config/ui-config.json file contains options that enable you to turn on/off UI features in Account Linking. Configure Account Linking to use your custom configuration file by setting the *ui-config-file* attribute. Only a subset of the options currently applies to the component. These options are:
  - "fiLogosVisible" - controls whether a page of popular financial institution (FI) logos is shown in the FI selection step. If true, the logos are shown; if false, then only a search control and FI list are shown.
  - "includeURLinSearch" - controls whether the FI URLs are included in the search criteria when the user is searching for an institution.

For example:

```
"cui-fi-select": {  
  "fiLogosVisible": false,  
  "includeURLinSearch": false  
}
```

## Applying the user interface customizations

Use these instructions to customize terminology, styles, fonts, and optional features.

1. Edit the files provided in the *assets* and *as\_assets* folders. These files and edits are described in "[Customize the user interface](#)" on page 19.
2. Modify your web server to allow our origin ([www.byallaccounts.net](http://www.byallaccounts.net)) to access those files. We suggest the following:
  - Header set Access-Control-Allow-Origin "[www.byallaccounts.net](http://www.byallaccounts.net)"
  - Header set Access-Control-Allow-Headers "Cache-Control,Content-Type,Accept,Referer,User-Agent,Sec-Fetch-Dest"
  - Header set Access-Control-Allow-Methods "GET,OPTIONS"

If this access is not allowed, then the components will not be able to access the customization files and each will have to use its own version of those files.

3. Set the appropriate attribute to identity each custom file. For attribute information see "[About attributes](#)" on page 7.

---

## Appendix A: Cumulative changes

This appendix keeps a running list of released versions and a summary of the changes in each component, starting with when Account Summary as introduced in version 1.0.

**WARNING:** When a release is required, you *must* use the latest version of the component. If you do not, any project you have that uses the component will fail because the new version has changes to the underlying API.

Date	Version	Required?	Summary of Changes
April 17, 2023	Account Linking 3.4 Account Summary 1.1	Yes *	<p>This release introduces the Account Summary component.</p> <p>ByAllAccounts Connect component has been renamed to Account Linking.</p> <p>* Implementations continuing to run the existing Account Linking component alone can continue to run it without changes. However, anyone implementing the Account Summary component with it must download a new version of the Account Linking component.</p>
May 25, 2023	Account Linking 3.4 Account Summary 1.1	No	<p>Updates for Account Linking:</p> <ul style="list-style-type: none"><li>■ Several status information messages have been updated.</li><li>■ An unnecessary scrollbar that appeared when running in Firefox has been removed.</li></ul> <p>Updates for Account Summary:</p> <ul style="list-style-type: none"><li>■ Errors that are externally actionable by users are now included in the Needs Attention group.</li><li>■ Several status information messages were updated.</li><li>■ Several user interface refinements were made in the Needs Attention including improving some credential behavior.</li></ul>

Date	Version	Required?	Summary of Changes
June 15, 2023	Account Summary 1.1	No	The Account Summary component can have an optional "Account Type" view enabled for the Firm to display the investor's data organized by Assets and Liabilities, as well as additional breakdown by account category.
July 6, 2023	Account Summary 1.1	No	The Account Summary component can now be enabled in read-only mode by the parent page passing in read-only authorization to the component. This capability enables Advisors to monitor information for their investors. The Advisor can see data in the Institution view and Account Type view (if it is enabled) for a given investor and see various options that are available in the component, but the options are disabled. Account Linking cannot be invoked when the user is in read-only mode.
July 27, 2023	Account Linking 3.5 Account Summary 1.2	No	Added the <i>iframe-style</i> attribute to customize styling for the iframe of the component.
August 17, 2023	Account Linking 3.6 Account Summary 1.3	No	Upgraded to the underlying Angular technology. Minor usability improvements.
September 7, 2023	Account Linking 3.6 Account Summary 1.3	No	Updates for Account Linking: <ul style="list-style-type: none"> <li>■ The language on the User Agreement page and button has been updated.</li> <li>■ On the landing page, the financial institutions (FIs) displayed are filtered based on whether or not the Firm is enabled for open authorization (OAuth). The default FI list includes Open Banking FIs only for the firms enabled for OAuth and</li> </ul>

Date	Version	Required?	Summary of Changes
			<p>the FI list for the firms not enabled for OAuth has been modified.</p> <ul style="list-style-type: none"> <li>■ The list of popular financial institutions (FIs) has changed. Goldman Sachs - Private Wealth Management FTP replaces Goldman Sachs Private Wealth Management - Quicken. For firms that are enabled for open authorization (OAuth), Fidelity Open Banking and Empower Open Banking have been added to the popular FIs list.</li> </ul> <p>Updates for Account Summary:</p> <ul style="list-style-type: none"> <li>■ The language on the User Agreement page and button has been updated.</li> </ul>
September 28, 2023	Account Linking 3.6	No	<p>There are a few financial institutions (FIs) that will require a license to use going forward. If your Firm does not have a license for those financial institutions, then you will see the following message when interacting with accounts or credentials for them:</p> <p>"If you are interested in connecting to {FI Name}, please contact your aggregation service provider."</p>
November 9, 2023	Account Linking 3.6 Account Summary 1.3	No	<p>Web Content Accessibility Guidelines (WCAG 2.1) compliance changes are complete for Account Summary, so the Morningstar Connect components are now fully compliant.</p>
November 30, 2023	Account Linking 3.6 Account Summary 1.3	No	<p>Buttons and fonts have been enlarged for design consistency.</p>
December 21, 2023	Account Linking 3.6 Account Summary 1.3	No	<p>Changes in this release were made for maintenance purposes.</p>
January 18, 2024	Account Linking 3.6	No	<p>Changes in this release were made for</p>

<b>Date</b>	<b>Version</b>	<b>Required?</b>	<b>Summary of Changes</b>
	Account Summary 1.3		maintenance purposes.
February 8, 2024	Account Linking 3.6	No	Changes to Account Linking support different Referrer Policy settings in customer environments.
February 29, 2024	Account Linking 3.7	No	Changes in this release were made for maintenance purposes.
	Account Summary 1.3		
March 21, 2024	Account Linking 3.7	No	Changes in this release were made for maintenance purposes.
	Account Summary 1.3		
April 11, 2024	Account Summary 1.3	No	Changes in this release were made for maintenance purposes.
May 2, 2024	Account Linking 3.7	No	Documentation was updated to clarify details about whitelisting.
	Account Summary 1.4		An optional Net Worth Chart feature has been added for the Account Summary component.
June 13, 2024	Account Linking 3.7	No	Wait times for open authorization (OAuth) have been improved for the Account Linking component.
August 15, 2024	Account Linking 3.8	No	Changes in this release were made for maintenance purposes.
	Account Summary 1.5		
September 5, 2024	Account Linking 3.8	No	Changes in this release were made for maintenance purposes.
	Account Summary 1.5		
September 26, 2024	Account Linking 3.9	No	The underlying Angular technology has been upgraded.
	Account Summary 1.6		
October 17, 2024	Account Linking 3.9	No	Changes to the components in this release were made for maintenance purposes.
	Account Summary 1.6		Documentation was updated to reflect that names are now prefixed with ByAllAccounts.
November 7, 2024	Account Linking 3.10	No	Changes to the components in this release were made for maintenance purposes.
	Account Summary 1.7		
December 12, 2024	Account Linking 3.10	No	Changes to the components in this

---

Date	Version	Required?	Summary of Changes
	Account Summary 1.7		release were made for maintenance purposes.
February 20, 2025	Account Linking 3.10 Account Summary 1.7	No	Changes to the components in this release were made for maintenance purposes.
April 24, 2025	Account Linking 3.11 Account Summary 1.8	No	The underlying Angular technology has been upgraded to Angular 19.