

Certain Signal: Advanced Webhooks Integration Guide

This document is a guide to using **Certain Signal** to integrate with other applications by using **Advanced Webhooks**. This includes the use of nested JSON as well as everything available in the *standard* 'Webhooks' integration module.

If you are using that standard module, please see the separate guide to integrations using standard webhooks.



(Signal also integrates with Marketo, Eloqua, and Salesforce – see separate guides.)

Certain Signal for Advanced Webhooks is not included with Certain Platform by default. If you're interested in adding this or any other Signal integration to your instance of Certain, please email features@certain.com, including your account name and the Signal integration(s) you're interested in.

Contents

What is Signal? How does it work?	2
Prerequisites	2
• Data-Flow Considerations	2
• Credentials in Target Application.....	2
Overview of Setup Steps	2
Setting up Tags	3
• What Are Tags?	3
• Setting Tags Up for an Account.....	3
• Applying Tags in an Account.....	3
• Applying Tags in an Event.....	4
Opening Certain Signal	5
Setting up a Connection	5
• What are Connections?.....	5
• Adding a Connection	6
Setting Up Flows	7
• What is a "Flow"?	7
• The Flow List	8
• Configuring a Flow	8
Flow Data Source	9
• Available sources.....	9
• Activate for	9
Flow Filters	10
Flow Destination	10
• Setting up a Destination.....	11
• Mappings	11
• Example Mapping from Pasted JSON ...	15
Metrics Dashboard	19
• Account Insights.....	19
The Retry Queue	21
Replaying a Flow	22

What is Signal? How does it work?

Certain Signal processes data from your events in **real time**, passing it from Certain Platform to your target third-party application, which could be any app with webhook integration, such as Slack or Google Forms,

This real-time integration empowers sales and marketing teams to take intelligent, prompt action on the right event data.

Almost everything is set up at the **account** level. Signal processes information from the events in your account. For event-level information this is based on the custom **tags** you attach to data such as Registration Statuses, as explained in this guide.

Important: Signal processes **outbound** information, processing information *from* Certain and sending it to your target application.

Prerequisites

Data-Flow Considerations

- Do you capture data in registration questions that will be synced to the third-party app to which you are integrating?
 - If so, you'll need to apply tags to those questions: see page 5.
- Do you have different data mappings based on registration status, or attendee type?
 - If so, you'll need to apply tags to those items: see page 4.

Credentials in Target Application

For you to set up a Connection in Signal (see page 5), the administrator of your target third-party application will need to provide you with the information described under 'Adding a Connection' on page 6.

For example, if your chosen **Authentication Method** there is OAuth2 then they'll need to create an OAuth2 app in that system, and provide you with the following: **Endpoint**, **Client Id**, and **Client Secret**.

Overview of Setup Steps

Step	See Page
On Certain Platform	
1. Add tags in the account	3
2. Apply those tags	3 & 4
In Certain Signal	
3. Add a Connection	6
4. Configure a Flow	8

Setting up Tags

What Are Tags?

Tags are a way of identifying event-level data using labels you set at the account level.

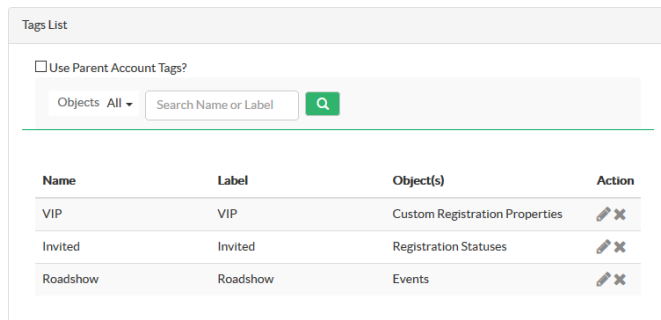
You can then apply those tags to generic items in events, especially custom registration statuses and custom registration properties for use in Certain Signal. (Tags can be used for other purposes as well, but this guide doesn't cover that.)







For example, your events may have several custom registration statuses in addition to the standard ones. You can apply the same Tags to more than one status, or you might choose to give each one its own Tag.

When you set up a Flow in Certain Signal to send data to your target application when an attendee's Registration Status changes, for example, you specify the *tags* applicable to those statuses, not the statuses themselves. That means the flow can apply to *any* event in your account.

Setting Tags Up for an Account

1. As an Administrator, go to **Account Settings > Management > Tags**.
2. Enter a Name and a Label for the tag.
3. Select the **Object(s)** to which the tag can apply; for example, 'Registration Statuses' and/or 'Custom Registration Properties'.
4. Click **Add**.
5. Repeat as required for as many tags as you need.
6. *Important:* Add enough tags to apply to all of the following that you will use in flows (see page 10):
 - a. **Registration Statuses**
 - b. **Custom Registration Properties**
7. Also add enough tags to apply to all of the following that you will use in filters for flows (see page 10):
 - a. **Attendee Types**
 - b. **Events**



Name	Label	Object(s)	Action
VIP	VIP	Custom Registration Properties	 
Invited	Invited	Registration Statuses	 
Roadshow	Roadshow	Events	 

Applying Tags in an Account

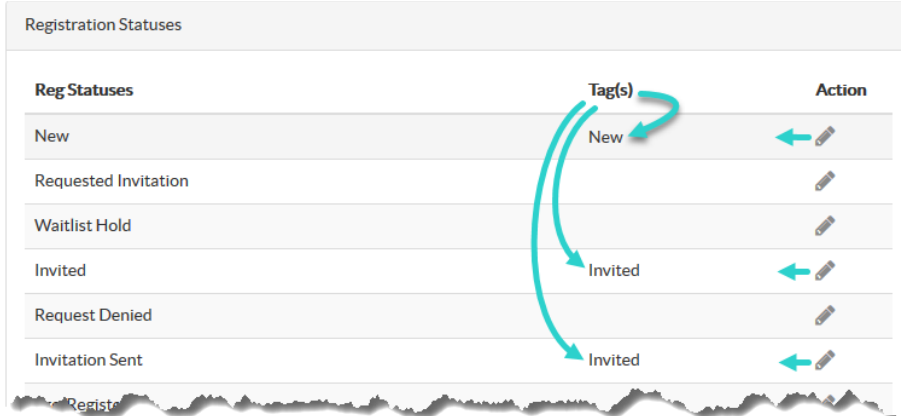
In each event from which you want information to flow through Certain Signal, apply tags to the relevant information: **Registration Statuses** and **Registration Custom Properties**.

(You can also tag Attendee Types and Events, so that you can filter registration records by attendee type or event: see **Filters** on page 10.)

Default Registration Statuses

These apply to all events, so an Administrator applies the tags at the *account* level.

1. Go to Account Settings > Management > Registration Statuses.
2. Select one or more Tags for each status.



Reg Statuses	Tag(s)	Action
New	New	← [edit icon]
Requested Invitation		[edit icon]
Waitlist Hold		[edit icon]
Invited	Invited	← [edit icon]
Request Denied		[edit icon]
Invitation Sent	Invited	← [edit icon]

Important: Even if you don't use any standard registration statuses, best practice is to set up tags for all of them, but it's *essential* to tag at least the 'New' status (which Certain uses "behind the scenes" when first processing each registration).

If you *do* use standard reg statuses, it's *essential* that you tag them all, so that you can use them in the Flows you configure in Signal: see page 8.

Applying Tags in an Event

Custom Registration Statuses

If any of the Flows you configure in Signal will watch or activate for Custom Reg Statuses (see page 9):

1. In each event, go to Plan > Event Setup > Custom Statuses
2. Select at least one tag for each status.

Custom Registration Properties

If any of the Flows you configure in Signal will watch or activate for Custom Reg Properties (see page 9):

1. In each event, go to Plan > Configure.
2. Under **Custom Registration Properties**, select at least one tag for each custom reg property in the event.

Standard Registration Properties (Automatic)

These tags are set up for you automatically, with names identical to the properties themselves: *Complete, Badge Printed, On To Do List, Invoice Generated, and Test.*

You only see them in Signal, where, just like Custom Reg Properties, you can activate Flows for them (see page 9). There's nothing to edit on Certain Platform.

Attendee Types

(Optional – for use with filters – see page 10.)

1. In each event, go to Plan > Event Setup > Attendee Types.
2. Select one or more Tags for each attendee type on which you may wish to filter registrations. (See **Filters** on page 10.)

Events

(Optional– for use with filters – see page 10.)

1. In each event you may wish to include in a filter (for example to ensure that only registrations for that event are passed to your target application):
2. Go to Plan > Event Setup > Details.
3. Select one or more Tags for the event.

Registration Questions

(Optional– for use with mapping Certain fields to fields your target application – see page 11.)

1. In each event in which you use registration question to capture data from attendees, and wish to pass those answers to your target application:
2. Go to Plan > Event Setup > Questions.
3. Select just *one* Tag for each question. (Selecting more could result in duplicate data in your target application.)

Opening Certain Signal

When Signal is activated for your account, the **Account Settings > Implementation** menu—available to Administrators—includes an extra option:

- **Signal Real-Time Data Integration**

Click that link to open Certain **Signal** in a separate window; it runs separately from Certain Platform.

Note: To return from Signal to Certain Platform at any time, click



Setting up a Connection

What are Connections?

A **Connection** in Certain Signal specifies how to connect to your target application.

You can actually have multiple connections, including to other third-party applications. (Marketo, Eloqua and Salesforce are covered in separate guides.)

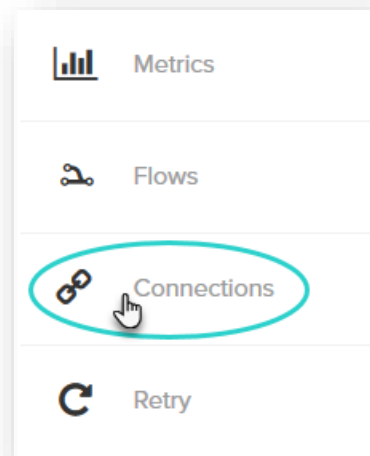
Each Flow (see page 7) requires a Connection. Multiple flows may use the same Connection.

You can set up a Connection before configuring your first Flow, but you also have the option to do so *while* configuring a Flow. This guide assumes you're setting up the Connection first.

Adding a Connection

As an Administrator you may set up one or more Connections for your account. You need only do so once – you can then use them in the Flows you set up (see page 7, below).

1. Go to **Account Settings > Implementation > Signal Real-Time Data Integration**.
2. As noted above, Certain Signal opens in a separate window.
3. Click **Connections** in the left navigation panel.



4. Click **Add A Connection** in the **Connection List** page that opens.
5. Enter the details in the **Connection Setup** screen that opens.
 - **Target:** Use the pre-selected value – **Advanced Webhook**.
 - **Connection Name:** Enter a name of your own choice. This could be just the name of the application with which you are integrating, for example.
 - **Authentication Type:** Select the authentication type to be used from the options available. (Your choice determines the other fields displayed for you to complete.)

Basic Authentication	Open / No Auth	API Key / Token	OAuth2
<ul style="list-style-type: none"> • URL • Content Type • Request Method • User Name • Password 	<ul style="list-style-type: none"> • URL • Content Type • Request Method 	<ul style="list-style-type: none"> • API Key / Token 	<ul style="list-style-type: none"> • Grant Type • Client ID • Client Secret • Authorization URL • Access Token URL • Refresh Token URL • Scope • Test Connection URL

Basic Authentication

- **URL:** The endpoint to which Signal must send data.

- **Content Type:** Select one of the two options.
 - application/json
 - x-www-form-urlencoded
- **Request Method:** Select 'POST' or (if appropriate) 'GET'.
- **User Name:** The user in the target system must have the necessary minimum permissions in that system if any are required.
- **Password:** That user's password.

Open / No Auth

The lowest level of security

- As for Basic Authentication above, but without a User Name or Password.

API Key / Token

- **API Key / Token:** To be provided by the administrator of your target application.

OAuth2

All these values must be provided by the administrator of your target application.

- **Grant Type:** Select 'Client Credentials' or 'Authorization Code'
 - **Client Id** and **Client Secret:** These two long strings of characters are unique to the OAuth2 app your administrator has set up, as explained under 'Credentials ...' on page 2.
 - **Authorization URL**
 - **Access Token URL**
 - **Refresh Token URL**
 - **Scope:** Depending on your target app, your requirements, and your security policies, this might be blank, a keyword, or a URL, for example.
 - **Test Connection URL**
- **Is this a primary connection** –Leave clear. This check box is displayed for all connections, but is only relevant to Eloqua, Marketo and Salesforce integrations requiring backwards compatibility.

6. Click **Save & Test**.

7. If the test is successful, click **Close**. If it's not, check that the values in step 5 are all correct.

Setting Up Flows

What is a "Flow"?

A flow is a configuration to manage the flow of data from Certain to your target application. You create Flows from the landing page in Signal: see 'Configuring a Flow', below. You may configure several flows for an account, which might all use the same Connection (explained above).

You only need to configure a flow once at the account level. When a flow is complete, it will start picking up data for each event in that account within about a minute.

The minute's delay is because Signal runs independently from Certain Platform. So if you *edit* a flow then the same slight delay occurs before that change takes effect in the processing of the registrations.

The Flow List

As an Administrator in Certain Platform, go to Account Settings > Implementation > **Signal Real-Time Data Integration**.

As noted above, **Certain Signal** opens in a separate browser window.

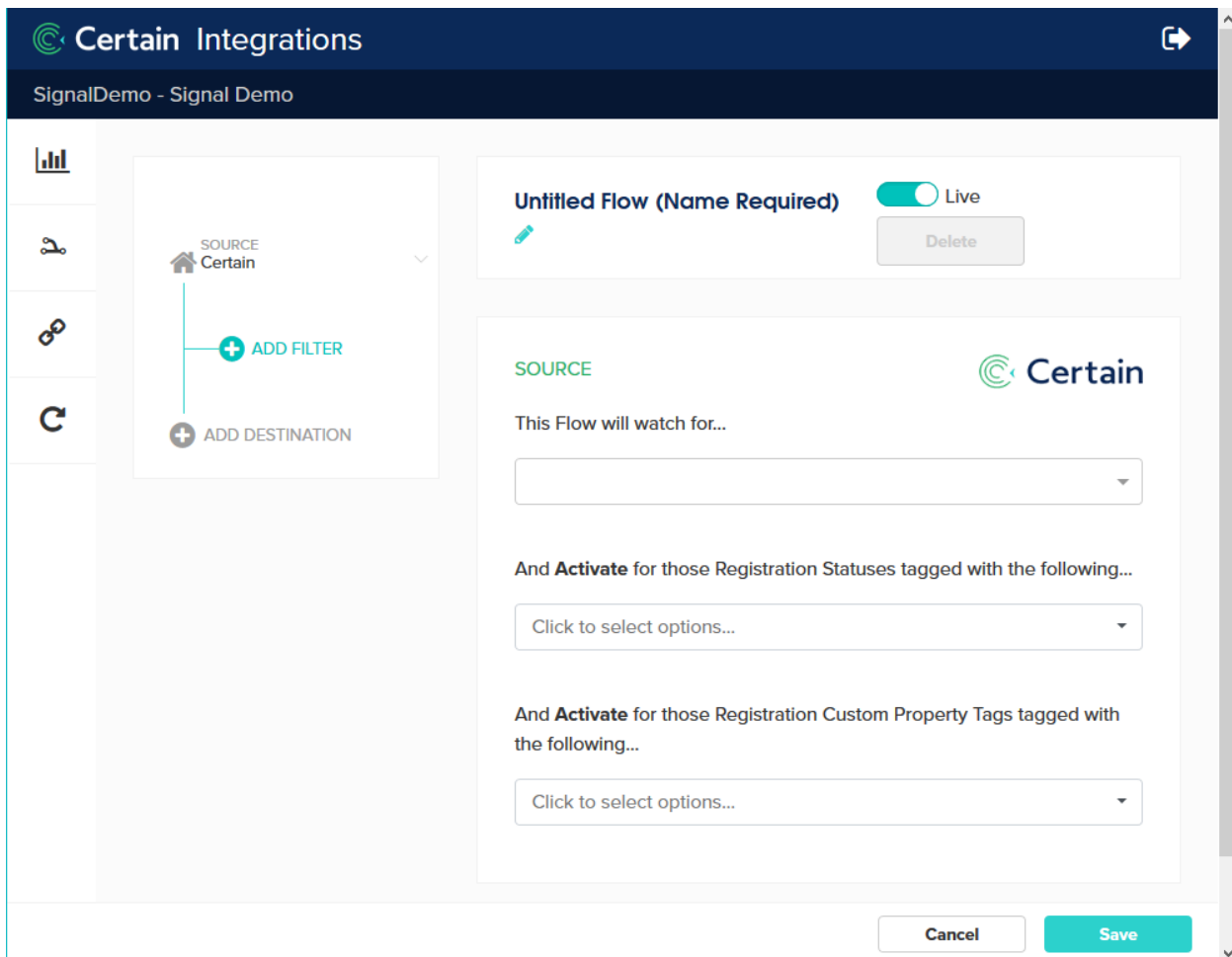
The main screen in Signal is the **Flow List**, which lists all flows.

The **Status** column shows whether a flow is completely set up.

The **Active** column shows whether the flow is running. Click the toggle button to change a flow from Active to Inactive, or vice versa.

Configuring a Flow

Click **ADD A FLOW** to start setting up a new flow – see screenshot below.



Certain Integrations


SignalDemo - Signal Demo

SOURCE
Certain

+ ADD FILTER

+ ADD DESTINATION

Untitled Flow (Name Required) Live

SOURCE 

This Flow will watch for...

And **Activate** for those Registration Statuses tagged with the following...

Click to select options...

And **Activate** for those Registration Custom Property Tags tagged with the following...

Click to select options...

The configuration consists of:

- **Name**
- **'Live' or 'Test'** status. (See immediately below.)
- **Source:** What information the Flow will look for, and what it will activate for in your events. (See page page 9.)
- **Filters:** Optional filters to narrow down that information. (See page page 10.)
- **Destination:** Where and how that information goes into your target application. (See page 10.)

'Live' or 'Test'

The **Live** toggle switch determines whether your Flow is Live or Test:



A **Live Flow** will pick up all *live* registrations in live events. It will ignore test registrations, even in live events.



A **Test Flow** picks up all test registrations: that's all registrations in *test* events, plus any registrations marked as 'Test' in *live* events."

Best Practice: Set a new flow up as Test—and test it—before setting it to Live.

Flow Data Source

Next, specify the **Source** of data for the flow (optionally applying [Filters](#)).

The Source of a Flow is what the Flow will watch for in your data in Certain and when it will activate, based on that data.

For example, it might watch for any change to a Registration Status, and activate if an attendee's status has changed to a status tagged as 'Registered'.

Available sources

You set a flow to watch for any one of the following:

- **Registration Create Update:** When a registration is created or updated.
- **Registration Status Change:** When a registration's status changes.
- **Session Registration Status Change:** When a registration's session registration status changes.
- **Event Create Update:** When an event is created or updated.

Note: You can always save an incomplete Flow and complete it at a later date. As soon as a Flow is complete, it will start picking up data after the usual minute's delay.

Activate for ...

Choose what the flow should activate for by selecting one or more tags in each appropriate object's drop-down list

As shown in the screenshot above, you can activate for tags applied to **Registration Statuses** and/or **Registration Properties**. Other options may be added depending on what the Flow is watching for:

- If the Flow is watching for Registration Status Change, you *must* activate for **Registration Statuses**.
- If it's watching for Session Registration Status Change, you *must* activate for **Session Registration Statuses**.
- And if it's watching for Event Create Update, you *must* activate for **Event Statuses**.

The tags available for selection are those set up for that object; see page 3.

For example, the tags for Registration Statuses include Registration Status tags, which you can apply to:

- standard registration statuses at account level (see page 4)
- custom registration statuses at event level (see page 4)

Flow Filters

You can filter the data going into a flow by selecting fields in any of these three filter types:

Event fields, Profile fields, and Attendee Type tags.

The flow will only include a registration if it meets the rule(s) specified in the filter.

- **Event**
Available fields:
 - standard event fields (e.g., Event Code)
 - custom event fields
 - event tags
- **Profile**
Available fields:
 - standard profile fields (e.g., Position)
 - custom profile fields
- **Attendee Type Tags**
Available fields:
 - tags that can be applied to Attendee Types

Note: For custom fields you can only select “enumerated” questions – those that have pre-configured answers; that is, questions that are of types Select, Multi-select, Checkbox, or Radio.

Flow Destination

Select **Advanced Webhooks** from the integrations set up by Certain for your account. (You may have this and others to choose from.)



Setting up a Destination

1. Give the Destination a **Name** of your choice,
2. Select the **Connection** to use.

Note: You can instead click **New Connection** to add a connection; the process to set one up is the same as described on page 5.

3. Select **Advanced Webhook** as the **Action** for this connection.
4. Select or add the mappings to use, as described next,

Mappings

Select or set up a set of mappings. You can have two: one for the **Payload** and (if required) one for the **Http Header**.

The setup process is the same for each:

A mapping specifies how each *target* field in your third-party application matches a *source* field in Certain.

Select a mapping from the drop-down list.

If nobody has set any mappings up yet, or you need something other than one of the existing mappings, click **New Mapping** to add one. The **Mapping Setup** screen opens:



Mapping Setup

New Mapping

Mapping Name:

Select Root Element Type:

NOTE: The easiest way to start setting up a Webhooks+ destination is to obtain a sample JSON from the source. Paste that below and we'll parse out the structure you'll need to support it.

Sample JSON from Destination...

- Mapping Name:** Give the mapping a name of your choice.
Best Practice: For multiple mappings, used for different flows, make the names self-explanatory. For example, 'Lead Mapping' and 'Form Mapping'
- Select Root Element Type:** In the great majority of cases, leave this set to the default value, **Object**. Select **Array of Objects** only if your JSON payload begins with a square bracket: [
- Hint:** As it says on screen, the easiest way to start setting up an Advanced Webhooks destination is to obtain a sample JSON from the source. Paste that into the text box, and Signal will "parse out the structure you'll need to support it"; that is, it will work out what the source fields are and save you having to type them in yourself.

(If you'd prefer to enter them manually instead, click 'Let me set them up myself' and proceed to step 10.b.)

Caution: Only do this if you are completely comfortable with JSON syntax, as explained on https://www.w3schools.com/js/js_json_syntax.asp, for example.)

4. 'Parsed Structure' Option

Continue here if you're going the route of pasting in a sample JSON and letting Signal parse it to determine the data structure. (Otherwise, skip ahead to step 10.)

- Obtain a sample JSON payload from your target application. This will include the target fields that you will map to Certain fields in step 10.c.
- Paste that into the **Sample JSON from Destination** text box.
- Click **Parse JSON**.
- Signal will determine the target fields in that JSON and display them on screen as if you had entered them yourself. (See an example on page 16.)

9. Proceed to step 10.c below to map those fields to source fields in Certain Platform.


10. 'Manual Setup' Option (and mapping parsed fields)

For each of the target fields in your target application:

If you chose to let Signal parse a JSON (per steps 3 to 9), skip the next step and go straight to step 10c – the list of target fields will be prepopulated for you.

(As mentioned in step 8 above, see an example screenshot of The screenshot below shows how Signal would parse this JSON.

- a. Parsed JSON on page 16.)
- b. If you chose *not* to let Signal parse a pasted JSON to determine the data structure, so clicked '**Let me set them up myself**' in step 3, then:
 - i. Enter the name of the target field and click **Add**.
 - ii. Select the data type for that field in the adjacent drop-down list of JSON data types:
String, Number, Boolean, Object, Array of Strings, Array of Numbers, Array of Objects, or Array of Booleans.
Caution: The type *must* map the type of data in the JSON payload.

- c. Click  to select the "source" field in Certain that matches that target field.
For example: if the target field is 'Title' in your target application, the source field might be 'Position', meaning the value of the Position field in Certain will populate the Title field in your target application.



The Certain fields you can choose from as the source of the data going into the target fields in your target application will include some of the following, depending on what your flow will activate for. (If, for example, it will activate for Event Statuses, then you can only map *event* fields.)

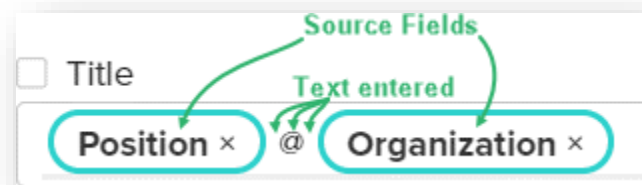
- Profile Standard Fields
- Profile Custom Fields
- Registration Standard Fields
- Registration Custom Question Tags – one field for the question, and one for its answer
(See page 5 about not using the same tag for more than one question.)
- Registration Standard Properties and Tags
- Array/List supported fields
(e.g., Registration Question List > *Question Name*,
Registration Question List > *Question Code*, and Registration Question List > *Question Answer*)
- Account Standard Fields (Account Code)
- Event Standard Fields (e.g., Event Code and Event Name)

- Event Custom Fields
- Array/List supported fields
(e.g., Event Question List > *Question Name*,
Event Question List > *Question Code*, and Event Question List > *Question Answer*)
- Flow Fields (Flow Name)
- Macros (e.g., Current Date)

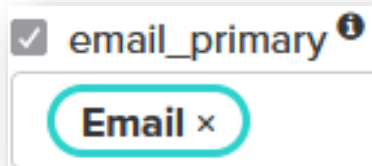
Note: Unlike a standard Webhooks integration (see separate guide), an *Advanced Webhooks* integration can include **arrays**, so the Certain source fields you can select from include lists such as Registration Question List, as listed above.

Note: You can concatenate multiple source fields for the same target field, and even type fixed text.

For example, for target field 'Title' you could choose source fields 'Position' and 'Organization', separated by two spaces and "@":



- d. To make a target field a **required** field, select the checkbox next to it.



- If a required field is missing, there'll be a validation error when the flow tries to update your target application. This is not normally fixable, so would not go into The Retry Queue.

- e. To **delete** a source field, click the **x** after its label.

11. In the second column you can select a transformation option for each field.

The default of *no* selection means the data will be sent to your target application unchanged.

The options are:

- lower case
- Proper Case
- UPPER CASE
- Trim (This removes extra spaces.)

Note: You can select more than one transformation for a field. For example, you could change it to Proper Case *and* trim it.

12. Continue adding more target/source field pairs as necessary.

If you've selected a mapping then two other buttons are enabled, with self-explanatory labels:

- **Edit Mapping**
 - **Preview Mapping**
13. **Save** the mapping.

Example Mapping from Pasted JSON

If you chose the '**Parsed Structure**' option in Step 4, the next three pages illustrate the three steps involved:

1. An example JSON you might paste in for Signal to parse.
2. A screenshot showing how Signal parsed that JSON

3. A screenshot showing those source fields mapped to target in CertainExample JSON

```
{
  "attachments": [
    {
      "author_name": "Sample Event Survey",
      "color": "#FF8C00",
      "pretext": "*Event Name:* My Test Event `Event Survey Response`",
      "text": "Survey responses below:",
      "fields": [
        {
          "short": "true",
          "title": "Testing",
          "value": "Correct"
        }
      ],
      "title": "Attendee: Matt Cash (75876-4308060-1172)"
    }
  ]
}
```

The screenshot below shows how Signal would parse this JSON.

Parsed JSON

Enter Target Field and Click on Add Button

```
{}  
  attachments {}  
  fields {}  
  title {}
```

The interface shows a tree view of the JSON structure. Each field is represented by a text input, a menu icon, and a dropdown menu. The dropdown menus are currently set to "Click to select options...".

- attachments**:
 - author_name**: Click to select options...
 - color**: Click to select options...
 - pretext**: Click to select options...
 - text**: Click to select options...
- fields**:
 - short**: Click to select options...
 - title**: Click to select options...
 - value**: Click to select options...
 - Add a Field**: String x Add
- title**: Click to select options...
Add a Field: String x Add

Add a Field: String x Add

Save **Cancel**

See the screenshot below for an example of how you might map these fields.

Parsed JSON with each target field mapped to a source field in Certain

```
attachments [
  {
    author_name "Event Name"
    color "#36a64f"
    pretext "First Name Last Name Registration Code"
    text "Question ansered by Attendee"
    fields [
      {
        short true
        title "Profile Question List > Question Name"
        value "Registration Question List > Question Answer"
      }
    ]
    title "Registration Status"
  }
]
```

Registration info for Event: **Event Name** Click to select options...

color: **#36a64f** Click to select options...

pretext: **First Name** **Last Name** **Registration Code** Click to select options...

text: **Question ansered by Attendee** Click to select options...

fields:

- short: **true** Click to select options...
- title: **Profile Question List > Question Name** Click to select options...
- value: **Registration Question List > Question Answer** Click to select options...

Add a Field String Click to select options... Add

Add

title: **Registration Status** **Registration Status** Click to select options...

Add a Field String Click to select options... Add

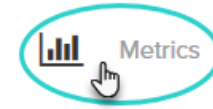
Add

Add a Field String Click to select options... Add

Save **Cancel**

Metrics Dashboard

To see the statistics available in Signal, click **Metrics** in the left navigation panel when looking at flows.



The choices in the new navigation panel you then see for Metrics depend on the flows and their targets.

The first one will be **Insights**, as illustrated and described below.

Other links may include, for example, **Webhook Posts**; all work in the same way as Account Insights.

Account Insights

Select whether you want to see **Live Flows** or **Test Flows**.

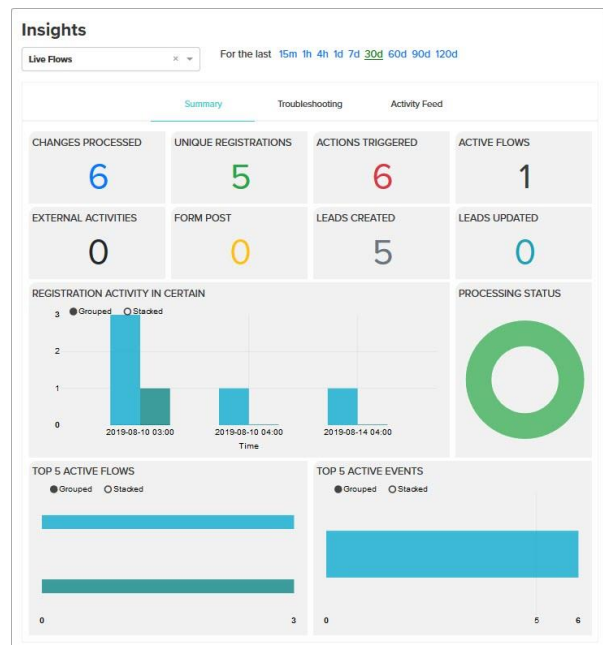
Select the period for which you want to see the data; for example, **for the last 15 minutes**, 1 hour or 4 hours, or a number of days.

There are three tabs: [Summary](#), [Troubleshooting](#), and [Activity Feed](#).

• Summary tab

This is the default tab on the **Insights** page, as shown here. The figures shown depend on the flows and actions. For some of them you can click the number to drill down further – for example, click **Unique Registrations** to see the registrations processed by any flows in your selected time frame.

You can filter or search for records when you've drilled down. For example, filter on an Event Code to see only the registrations in that event.



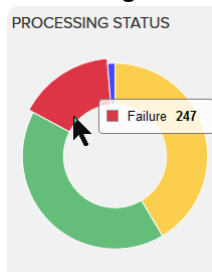
The figures listed below are shown. Remember these are for the whole account: all events and registrations, and for all flows in your account, which might include some which are not Webhooks integrations. (Note that some of the figures shown may therefore not be applicable to webhooks.)

For each of these you can click the number to drill down to details.

- **Changes Processed:** The number of registrations processed. Increments each time a registration is created or updated in Certain and then processed by a flow in Signal.
Note: This will often include the same registration more than once. For example, creating a registration and then updating it twice would total three “changes”.
- **Unique Registrations:** The number of registrations processed, counting any single registration only once. You may find this the most relevant figure on the page.

It's likely to be lower than **Changes Processed**. For example, if Joe Citizen's registration was processed three times, then Changes Processed will have increased by **3**. But since those changes were all for the same registration, the Unique Registrations will have increased by only **1**.

- **Actions Triggered:** The number of actions triggered by flows. If you only have one flow with one action, then this could be the same as Changes Processed. The more flows you have, and the more actions configured in those flows, the more actions may be triggered.
- **Actions Not Triggered:** Only displayed if registrations were processed by flows without actions being triggered. This could be caused, for example, by a registration having an untagged status.
- **Active Flows:** The number of flows actually processing registrations during the selected period. This is *not* related to whether the flow are shown as "Active" or "Inactive" on the [Flow List](#).
- **Leads Created:** The number of leads created in your target application by flows.
- **Leads Updated:** The number of leads updated in your target application by flows.
- **Registration Activity in Certain:** Overview information.
- **Processing Status:** A pie chart comparing the numbers of failures and successes.



Drilling down into the failures provides a high-level view for troubleshooting.

For example, you can see why an action has failed. If that's something you can address, then once you've done so you can expedite the fix in Signal by going back to Flows and clicking **Retry** in the left navigation panel — see [Retry Queue](#) on page 21.

• Troubleshooting tab

The second tab on the **Insights** page shows information you may find useful for troubleshooting.

For example, if some registrations are not processed because a Registration Status isn't tagged, this gives you the chance to rectify that so that the registrations can be processed on the next retry.

The numbers shown are as follows. Click a number to drill down to see details of the actual records,

- **In Retry Queue:** If an action fails, it joins the [Retry Queue](#)—see page 21—and will be tried again. (Maximum automatic retries per action = 3.)
- **Total Retried:** The number of retries. If an action was retried twice, that would have added 2 to this number.
- **Retried Abandoned:** The actions that failed three retries.
- **Validation Errors:** No retries possible because failed validation. For example, a lead that could not be created because a mandatory target field had no value in the source field. (See 'Mappings' on page 11.)
- **Retry Activity:** A chart showing retries by time.

- **Retry Processing Category:** A chart showing retries by category – “General”, “System”, “Config”, “Connection”, etc.
- **Connection Activity:** A chart showing activity per connection over time.

• Activity Feed tab

The third tab on the **Insights** page lists the registrations processed, noting success or failure, Registration Code, Event, Flow, etc.

This is a rolling history, by date. Think of it as another way to access lower-level data you could see by drilling down in the Summary or Troubleshooting tabs.

The Retry Queue

When an action fails it usually joins the “**Retry Queue**”, where it takes its turn to run again. (Failures that can’t be resolved, such as missing mandatory fields, are exceptions: they don’t join the queue.) An action can be retried up to three times, after which it does not rejoin the queue.

To see the Retry Queue, click **Retry** in the left navigation panel on the Flows page:



Causes of failure include ones under your control as a planner, such as a registration with a status that hasn’t been tagged, and technical ones such as a connection being down.

Where you can resolve the cause of a failure, such as tagging a registration or setting a flow back to being active, that’s all you need do. The action should succeed when retried.

For other failures that don’t resolve themselves, you probably need to contact your administrator or ask Certain for help.

The interval between retries depends on the severity of the reason: the more serious the reason the sooner the retry.

Filtering the Queue

You can filter the records shown in the Retry Queue using these three filters:

Integration: Probably only ‘Webhooks’ or ‘Advanced Webhooks’ unless you have multiple integrations set up.

Status: ‘All Statuses’, ‘Retry’, ‘Error’, ‘Failed’, or ‘Done’.

Category: ‘All Categories’, ‘General’, ‘System’, ‘Config’, ‘Certain API’, ‘Connection’, etc.

Submitting to the Queue

When you click an item in the Retry Queue, you see its full details.

If that knowledge is enough for you to solve the problem, then once you have done so you can click **Submit to Retry Queue**, which adds the item straight to the front of the queue.

[Submit to Retry Queue](#)[Cancel](#)

Replaying a Flow

If you change an aspect of a flow while it has been running for some time, such as changing its filters, you may want to replay that flow for the same registrations as before, as if the changes had been made earlier.

This is not something that you can do directly, but you can ask Certain to arrange it for you. You may be able to specify a date range, or even an event.



----- End of document -----