

Address Standardization Script for Engaging Networks

This project integrates your Engaging Networks form with the Address Standardization service.

How to use

1. Add the script below to the Engaging Networks page:

```
<script
  defer="defer"
  src="{YOUR_EN_URL}/as-en.js"
  data-cid="{YOUR_AS_ID}"
></script>
```

2. The script will look for the address fields following the Engaging Networks default field names. You need to provide a valid **cid** (Address Standardization Client ID) as a data attribute. You can also customize every field via data attributes.

Options

Every option can be set as a data attribute on the **script** tag.

- **cid** - Address Standardization Client ID (**required**).
- **as_record** - Address Standardization Record. Set the field name to store the Address Standardization Record Data. Usually a hidden field.
- **as_date** - Address Standardization Date. Set a field name to store the date. Every time the Address Standardization service is called, this field will be updated.
- **as_status** - Address Standardization Status. Set a field name to store the status of the last Address Standardization attempt.
- **address1** - Address 1 Field Name.
- **address2** - Address 2 Field Name.
- **city** - City Field Name.
- **region** - State Field Name.
- **postalCode** - Postal Code (zipcode) Field Name.
- **country** - Country Field Name.

Default Options

This is an example of the `script` tag with the default options:

```
<script
  defer="defer"
  src="{YOUR_EN_URL}/as-en.js"
  data-cid="0"
  data-as_record=""
  data-as_date=""
  data-as_status=""
  data-address1="supporter.address1"
  data-address2="supporter.address2"
  data-city="supporter.city"
  data-region="supporter.region"
  data-postalCode="supporter.postcode"
  data-country="supporter.country"
></script>
```

IMPORTANT: This project only works on Engaging Networks Forms. The `window.enOnSubmit` event listener is used to change the fields.

Development

Your js code must be on the `src/app` folder. Right now, this project has no styling changes, but the structure is ready for you to add your own on `src/scss`.

Install Dependencies

1. `npm install`

Deploy

1. `npm run build` - Builds the project
2. `npm run watch` - Watch for changes and rebuilds the project

It will create a `dist` folder, where you can get the `as-en.js` file and publish it.

Currently it's published for tests on:

<https://aaf1a18515da0e792f78-c27fdabe952dfc357fe25ebf5c8897ee.ssl.cf5.rackcdn.com/1874/as-en.js>