

# SP3-Batch Documentation v1.0.0

## Input File Layout

SP3 Batch input files must conform to a strict guideline so the inputs can be evaluated correctly. Please consider the following rules carefully when constructing your input files:

Input files **MUST**:

- Have a name with alphanumeric characters [0-9a-zA-Z] and can contain the "-" character.
- Be a CSV file with the **.csv** suffix.
- Conform to [RFC4180 Standards](#) with Unix end-of-line characters.
- Specify column headings in first line (See example below). Files with missing or invalid header lines will not process.
- Only contain rows with all required columns and correct value types.

Provided below are the required column names and the expected value types.

Please note that most column names have the format `<namespace>.<variable>`, this is by design and must be implemented. Also note that the values must conform to the value types specified.

### Required Values

Column Name	Value Type	Constraints	Description
id	string	Must be unique in context to the particular batch*	Input set identifier
site.lat	float	Latitude coordinate value	Latitude of the site to be analyzed
site.lng	float	Longitude coordinate value	Longitude of the site to be analyzed
building.year_of_construction	integer	Positive value only	Year date at which the building was constructed
building.num_stories	integer	Positive value only	Total number of stories above grade of the building being analyzed
building.building_type_id	string	Limited to the Building Structural Types table below	The structural system of the building being analyzed

\* We encourage using the [UUID v4 standard](#) for this purpose.

### Optional Values

Column Name	Value Type	Constraints	Description
site.vs30	float	Positive value only	Shear wave velocity of the soil at the site being analyzed in m/s
building.occupancy_id	integer	Limited to the Building Occupancy Types table below	The occupancy of the building being analyzed
building.total_area_ft	float	Positive value only	Gross area of the entire building being analyzed in square feet
building.building_ht_ft	float	Positive value only	Total height of the building above grade in feet

building.building_value	float	Positive value only	Total value of the building in US dollars
building.num_rooms	integer	Positive value only	Total number of bedrooms of a residential structure
building.num_baths	integer	Positive value only	Total number of bathrooms of a residential structure
building.construction_quality	float	Positive value ranging from 0 to 1.0	Ratio quantifying the quality of construction; 1.0 being excellent and 0.0 being poor
building.retrofit_str_year	integer	Positive value only	Year of structural retrofit if applicable
building.retrofit_str_level	float	Positive value ranging from 0 to 1.0	Ratio quantifying the extent of the structure retrofit; 1.0 being comprehensive and 0.0 being no effect
building.retrofit_ns_year	integer	Positive value only	Year of nonstructural retrofit if applicable
building.retrofit_ns_level	float	Positive value ranging from 0 to 1.0	Ratio quantifying the extent of the nonstructural retrofit; 1.0 being comprehensive and 0.0 being no effect
hazard.pre_shock_pga	float	Positive value only	Peak ground acceleration of the seismic event previous to aftershock events
hazard.return_periods	array	Array of positive values (eg. [72,475]), limit 5	Return periods in years of the seismic hazard level to be analyzed

## Building Structural Types

building.building_type_id	Building Type (Structural System)	Max Number of Stories
W1a	Single Family Dwelling Wood Light Frame	5
W2a	General Wood Light Frame	5
W2b	Wood Light Frame over Podium: RC Walls	7
W2c	Wood Light Frame over Podium: RC Frame	7
S1a	Steel Moment Frame Perimeter	25
S1b	Steel Moment Frame Space	25
S1c	Steel Moment Frame Perimeter SidePlate Connection	25
S2a	Steel Braced Frame (concentric)	25
S2b	Steel Braced Frame (eccentric)	25
S2c	Steel Braced Frame (brb)	25
S3a	Steel Light Frame	5
S3b	Steel Light Frame over Podium: RC Walls	7

S3c	Steel Light Frame over Podium: RC Frame	7
C1a	RC Moment Frame Perimeter	25
C1b	RC Moment Frame Space	25
C2a	RC Cantilever Shear Wall	25
C2b	RC Coupled Shear Wall	25
C2c	RC Shear Wall with RC Frame	25
PC1a	Single-Story Rigid-Wall Flexible Diaphragm (e.g. tilt-up): Precast Concrete Walls	1
PC2a	PRESS Frame	25
RM1a	Single-Story Rigid-Wall Flexible Diaphragm (e.g. tilt-up): Reinforced Masonry Walls	1
RM2a	Reinforced Masonry with Rigid Diaphragms	5
S4a	RC Shear Wall with Steel Frame	25

## Building Occupancy Types

building.occupancy_id	Occupancy Type
1	Commercial Office
2	Education [k-12]: Elementary Schools
3	Education [k-12]: Middle Schools
4	Education [k-12]: High Schools
5	Healthcare
6	Hospitality
7	Multi-Unit Residential
8	Research
9	Retail
10	Warehouse
11	Single-Unit Residential

## Example Header Line

```
id,site.lat,site.lng,building.year_of_construction,building.num_stories,building.building_type_id
```

### Example Value Line\*

```
"6d046874-89c6-11e7-bb31-be2e44b06b34",34.05,-118.25,2017,10,"C1a"
```

\*Matches the example header line

## Setup

### Email

An email address will need to be provided to HBRisk in order to receive all batch related notifications.

### Integrating with SP3 Batch

The application is hosted through AWS S3 which provides multiple ways to interact with SP3 Batch. The SP3 Batch can be accessed through the AWS CLI client, an AWS supported language-specific SDK, and through HTTP requests. Instructions for downloading and using the AWS CLI are located at <https://aws.amazon.com/cli/> and examples below make use of the CLI client.

The following examples are written using a Mac/Linux operating system.

```
# Configuring AWS defaults and credentials
$ aws configure
# Access Key Id provided by HBRisk
AWS Access Key ID [None]: INSERT-ACCESS-KEY-ID-HERE
# Secret Access Key provided by HBRisk
AWS Secret Access Key [None]: INSERT-SECRET-ACCESS-KEY-HERE
# The region in which SP3 batch is located
Default region name [None]: us-west-2
# The default output format
Default output format [None]: json
```

### Uploading Input Files

After a successful upload, a notification email will be sent out to the email provided explaining that the batch process has begun.

*Note: The examples below show the use of a variable labeled <client>. This will be provided by HBRisk.*

```
# Writing files
$ aws s3 cp ./input.csv s3://sp3-batch/<client>/inputs/<batch>.csv

# Here input.csv is a local input file and <batch> is a user-
# chosen name for both (i) the uploaded copy of the input file and
# (ii) the batch itself.
```

### Downloading Output Files

After the batch process has completed, a notification email will be sent out to the email provided explaining that the batch process has completed and files are available for download. List remote files by using the AWS S3 ls (list) command. Read remote files using the AWS S3 cp (copy) command:

```
# Listing output files
$ aws s3 ls s3://sp3-batch/<client>/outputs/<batch>/
2017-09-06 00:00:00      256 file_1.csv
2017-09-06 00:00:00      256 file_2.csv
2017-09-06 00:00:00      256 file_3.csv

# Notice that the output directory for a batch gets its name from the
# uploaded input file

# Reading files
$ aws s3 cp s3://sp3-batch/<client>/outputs/<batch>/out.csv ./dest.csv

# Here out.csv is the remote output file to download and dest.csv
# is a user-chosen name for the local copy of that file.
```

## Available SDKs

While using the CLI is fairly easy and scripts can be written to automate the process, using a language-specific SDK could prove easier to integrate into an existing code base. A list of popular SDK documentations are provided below:

- Java: <https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html>
- Php: <https://docs.aws.amazon.com/aws-sdk-php/v3/api/>
- Node.js: <https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/index.html>
- C#: <https://docs.aws.amazon.com/sdkfornet/v3/apidocs/Index.html>

## AWS User Permissions

In order to initiate the batch process, the CSV input file must be uploaded to S3. As mentioned above, HBRisk will provide a key prefix to prepend to all input and output files (denoted with the variable `<client>`).

- bucket: `sp3-batch`
- input file key prefix: `<client>/inputs/<input_file_name>.csv`
- output file key prefix: `<client>/outputs/<input_file_name>/<output_file_name>.csv`

The AWS User provided by HBRisk is allowed certain actions depending on the key prefix.

- List privileges have been granted for **all** directories under the `<client>/*` key prefix.
- Read privileges have been granted for both the `<client>/inputs/<input_file_name>.csv` and `<client>/outputs/<input_file_name>/<output_file_name>.csv` keys.
- Write privileges have been granted for the `<client>/inputs/` key prefix only.

Both the read and write privileges are available for csv files **exclusively**. All other file types are denied.

## What To Expect

The SP3 batch process is automated for ease-of-use. The process is simple, when an input file is uploaded to S3, a notification email will be sent out to the email provided. After the batch process has completed, another email will be sent notifying that the outputs are available.