Crash Location Resolution

User Manual for System Integrators

Send a confirmed crash point; get back its official location - route and reference post, or nearest cross street - and the roadway data for that spot, in one call.

Version: September 29, 2026
Bad Marine LLC

Contents

1. **Welcome to Crash Location Resolution**
   What it does, how it fits with your crash system, who uses what, a crash start to finish
2. **Getting connected**
   The two endpoints, conventions used everywhere, code values
3. **How a point is resolved**
   The four outcomes, search distances, the GIS answer wins
4. **Crashes on a state route**
   Route, reference post, mile-marker calibration, snap distance
5. **Crashes on a local road**
   Nearest feature, cross street, intersections, narrative, outside city limits
6. **Bridges**
   Over and under
7. **Placeholders and unresolved points**
8. **The roadway (MIRE) data**
   Every element, when it's filled, what's not available yet
9. **The single-crash endpoint**
   Request, response, validation errors
10. **The batch endpoint**
    CSV in, CSV out
11. **When a GIS source doesn't answer**
    Degraded answers, timeouts, retrying
12. **Configuration (System Admins)**
    GIS layers, search distances, timeouts
13. **How accurate it is**
    Measured against finalized state locations, where to look twice
14. **Quick reference**
    Code values, messages, glossary
15. **The online evaluation page**
    Trying the service without writing any code

1. Welcome to Crash Location Resolution

1.1 What Crash Location Resolution does

Crash Location Resolution is a web service. You send it a crash's confirmed latitude and longitude; it works out the crash's official location against your state's GIS data and sends back, in one answer:

- the **resolution method** - whether the point is on the state route system, on a local road, a placeholder, or on no road at all;
- on a state route: the **route**, **reference post** and **log mile**, calibrated between the surveyed mile markers on either side of the crash;
- on a local road: the **nearest street**, **intersection**, **bridge** or **railroad crossing**, the **cross street**, distance and direction, and a written **narrative**;
- **bridge context** when the crash is on or under a bridge;
- how far the crash is from the **nearest town** when it's outside city limits;
- the **roadway (MIRE) data** for that spot - speed limit, traffic volume, ownership, district, rural / urban and more - each one either filled or marked unavailable with the reason.

Every answer is worked out live against the GIS services, the same way every time, and nothing is stored.

1.2 How it fits with your crash system

The service is the location step of a crash system; it isn't a crash system itself. Your crash system (for example the Refinery) keeps the report, shows the officer's point, and records the point your review settles on. When a point is confirmed, the crash system sends it to this service, then stores the answer with the crash.

- **It plugs in; it doesn't take over.** One call in, one answer out. The crash system decides when to call and what to keep.
- **It stands on its own.** It doesn't need the Refinery or any other Bad Marine product, and a crash system that doesn't have it still works - the location fields are simply filled in by hand.
- **The state's GIS wins.** Where the GIS can supply a roadway value, that value is returned even if the officer reported something different. The officer's report is not the source for roadway data.
- **Nothing is stored.** No database, nothing to back up. Keeping the answer is the crash system's job.

> **Note:** the service resolves exactly the point it's given. It never moves a point or second-guesses a reviewer's placement - send the point you want resolved.

1.3 Who uses what

Who	What they use
A crash system (such as the Refinery)	The **single-crash endpoint** (section 9), called each time a reviewer confirms a point.
Analysts with a list of points	The **batch endpoint** (section 10): upload a CSV, get a CSV back.
Anyone evaluating the service	The **online evaluation page** (section 15) - the batch endpoint behind a simple upload screen.
System Admins	The service's **configuration** (section 12): which GIS layers it reads and

Who	What they use
	how far it searches.

1.4 A crash, start to finish

Here is how one crash moves through the Refinery. Any crash system that calls the service follows the same pattern.

1. The officer's report arrives with a latitude and longitude. The Refinery shows the point on its location screen.
2. The reviewer confirms the point or moves it to the right spot, then clicks **Map it**.
3. The Refinery sends the crash key and the point to the single-crash endpoint.
4. The service checks for a placeholder, tries the state route system, falls back to the local road network, looks for a bridge, and gathers the roadway data - usually in a few seconds.
5. The Refinery shows the route and reference post (or street and narrative) and fills in the roadway fields from the answer.
6. If the reviewer moves the point again, the old answer is marked out of date until **Map it** is clicked again. When the location is right, the reviewer completes the location step.

2. Getting connected

2.1 The two endpoints

Endpoint	Send	Get back
POST /api/v1/crash-locations	One crash as JSON.	One JSON answer (section 9).
POST /api/v1/crash-locations/batch	A CSV file of crashes (section 10).	A CSV file with one results row per crash.

The service's web address, and how callers sign in to it, are set when it's installed on your own systems. Neither endpoint changes anything, so a call can always be repeated safely.

```
POST /api/v1/crash-locations
Content-Type: application/json

{ "crashKey": "223000105", "latitude": 40.9458802, "longitude": -98.38349 }
```

The smallest complete request

2.2 Conventions used everywhere

- **Field names are camelCase** JSON - crashKey, resolutionMethod, linearReference.
- **Code values are numbers.** Fields such as resolutionMethod and direction come back as numbers, not words: resolutionMethod **0** means ReferencePost. Every list is in section 2.3.
- **Distances are in feet; reference posts and log miles are in miles.** Coordinates are decimal degrees.
- **Exactly one location block is filled.** linearReference for a state route, nearestFeature for a local road; the other is null. Both are null for a placeholder or an unresolved point.
- **Roadway data is always there.** mireElements holds all 20 elements on every answer; each is either filled or marked unavailable with a reason (section 8).
- **Trim text values.** Some route labels come from the GIS with a trailing space ("75 ").
- **Nothing is stored.** Calling twice with the same point gives the same answer (unless the GIS data has changed in between).

2.3 Code values

Field	Values
resolutionMethod	0 ReferencePost (state route) - 1 NearestReferenceFeature (local road) - 2 Unresolved - 3 KnownPlaceholder
nearestFeature.referenceType	0 Intersection - 1 StreetOnly - 2 Bridge - 3 RailroadCrossing
direction, **directionFromMilePost**, **northSouthDirection**, **eastWestDirection**	0 N - 1 S - 2 E - 3 W
bridge.relation	0 Over (the crash is on the bridge) - 1 Under (the crash is on the road beneath)

Batch files use words: the batch endpoint's CSV writes these values as words (ReferencePost, Over), not numbers.

3. How a point is resolved

Every point goes through the same steps, in the same order:

1. **Placeholder check.** If the point is one of the two known placeholder coordinates, it's reported as a placeholder and nothing else is looked up (section 7).
2. **State route.** If a state route centerline is within **100 ft**, the point is snapped to it and given a route, reference post and log mile (section 4). Result: **ReferencePost**.
3. **Local road.** Otherwise the service looks for the nearest street, bridge or railroad crossing within **300 ft**, widening to **1,000 ft** if it finds nothing (section 5). Result: **NearestReferenceFeature**.
4. **Nothing nearby.** If there's still nothing, the point is off the mapped road network - a field, a parking lot, private property. Result: **Unresolved**, with the reason (section 7).
5. **Bridge check.** For a state route or a street, the service checks whether that road is carried by, or passes under, a bridge within **1,000 ft** (section 6).
6. **Roadway data.** The MIRE elements are always gathered for the point, whichever way it was resolved (section 8).

The GIS lookups for a crash run side by side where they can, so one answer usually takes a few seconds (3 to 9 seconds measured). Give your client a timeout of at least 20 seconds.

> **By design:** the state route system is always tried first. A crash within 100 ft of a state route is placed on that route even if a local street is nearer, because the route and reference post are the official way to locate it.

4. Crashes on a state route


```
"resolutionMethod": 0,
"linearReference": {
  "routeId": "281",
  "highwayLabel": "281",
  "highwayType": "US",
  "highwayTypeDescription": "US Highway",
  "referencePost": 71.49978554256488,
  "logMile": 71.33978554256488,
  "snapDistanceFeet": 0.9,
  "nearestMilePostRefPost": 72,
  "distanceFromMilePostFeet": 2697,
  "directionFromMilePost": 1,
  "mileMarkerCrossChecked": true,
  "snappedPoint": { "latitude": 40.94588, "longitude": -98.38349 },
  "bridge": null
},
"nearestFeature": null
```


A real answer for a crash on a US highway (trimmed, some numbers rounded)

Field	What it holds
routeId	The route's ID in the state route system.
highwayLabel, **highwayType**, **highwayTypeDescription**	The route number, its type code (I, US and the state's own route types) and the type written out.
referencePost	The crash's position along the route, in miles. This is the official location.
logMile	The distance from the start of the route, in miles.
snapDistanceFeet	How far the point was from the route's centerline. A few feet is normal.
nearestMilePostRefPost, **distanceFromMilePostFeet**, **directionFromMilePost**	The nearest surveyed mile marker, and how far and which way the crash is from it - "2,697 ft S of mile marker 72".
mileMarkerCrossChecked	true when the reference post was calibrated between two surveyed markers (section 4.1).
snappedPoint	The point on the centerline the crash was snapped to.
bridge	Over / Under and the crossed feature, or null (section 6).

4.1 Calibrated between surveyed mile markers

A route in the GIS carries only a start and an end reference post, and some routes run for hundreds of miles as a single piece. Stretching a reference post evenly across a piece that long drifts - by hundreds of feet in places. So the service finds the two surveyed mile markers on either side of the crash, on the same route, and works out the reference post between those two instead.

- That brought the median difference from the state's own finalized reference posts down from about 227 ft to about **5 ft**.

- Where there aren't two markers on either side - near the end of a route, or where markers are sparse - it falls back to the route's own start and end values, and **mileMarkerCrossChecked** is false.

4.2 Snap distance

A state route is matched only when its centerline is within 100 ft. Inside that, a larger **snapDistanceFeet** is still worth a look: a crash 80 ft from an interstate's centerline may really belong on a frontage road or a ramp.

5. Crashes on a local road

When no state route is within 100 ft, the crash is described the national MMUCC way - by distance and direction from the nearest reference feature.


```
"resolutionMethod": 1,
"linearReference": null,
"nearestFeature": {
  "referenceType": 0,
  "primaryFeatureName": "S 54TH ST",
  "crossStreetName": "PANAMA RD",
  "distanceFeet": 3.07,
  "direction": 3,
  "narrativeText": "3 ft W of the intersection of S 54TH ST & PANAMA RD",
  "referencePoint": { "latitude": 40.59592601, "longitude": -96.64852462 },
  "bridge": null
},
"outsideCityLimitsContext": {
  "municipalityName": "Hickman",
  "northSouthMiles": 2.0, "northSouthDirection": 1,
  "eastWestMiles": 0.97, "eastWestDirection": 3
}
```


A real answer for a crash at a rural intersection (trimmed, some numbers rounded)

5.1 The nearest feature

The service collects every street, bridge and railroad crossing within the search distance and picks the nearest. That becomes **primaryFeatureName**, and **referenceType** says what kind of feature it is.

- **Cross street** - the next-nearest road with a different name. It may be the street the crash is near, not one the primary street physically meets at that spot.
- **Intersection** - a crash within **5 ft** of the primary street is called an intersection crash. Whether a crash is intersection-related is ultimately a judgement the officer and reviewer make; treat this as a starting point.
- **Distance and direction** - how far and which way the crash is from the feature.

5.2 The narrative

narrativeText writes the location out. Its form depends on the feature. These are real answers:

Feature	Narrative
Street, with a cross street	S 9TH ST W of J ST (26 ft)
Intersection	4 ft S of the intersection of KOKOMO DR & FLETCHER AVE
Street, no cross street nearby	20 ft N of 56Th Road
Bridge	<distance> ft <direction> of the <bridge name> bridge
Railroad crossing	<distance> ft <direction> of the <crossing name>
On or under a bridge	<road> BRIDGE OVER <feature> / <road> UNDER <feature>

Street names are written exactly as the GIS street data writes them, capitals included.

5.3 Outside city limits

When the crash is outside every city's limits, **outsideCityLimitsContext** names the nearest town and gives the north-south and east-west miles from it to the crash - "2.0 mi S and 1.0 mi W of Hickman". It's null inside city limits, and when no town is found within 40 miles.

6. Bridges

Once a crash has been placed on a state route or a street, the service checks the bridges within 1,000 ft. If that road is the one a bridge carries, or the one it crosses, **bridge** is filled:

relation	Meaning	Written as
0 Over	The crash is on the bridge.	14 BRIDGE OVER BUFFALO CREEK
1 Under	The crash is on the road passing beneath it.	27TH ST UNDER I80

- **crossedFeatureName** is the other road, river or railroad, as the bridge data names it. Some bridge records carry a structure number or a generic name ("STREAM").
- Route names are matched loosely - "I-80", "I80" and "I 80" are the same road - but a bridge carrying two routes at once ("I180/US34") isn't matched to either.

> **By design:** the GIS has no elevation. At a stacked interchange a point can sit closer to the road above than the road below, or the other way round, and nothing in the data can tell them apart. Check bridge crashes against the crash diagram.

7. Placeholders and unresolved points

7.1 Placeholder coordinates

Some crash systems put a fixed, off-map coordinate on a crash whose location is unknown or on private property. Snapping those to a road would invent a location, so they're recognized and reported as they are:

Point sent	resolutionMethod	diagnostics.reason
43, -95	3 KnownPlaceholder	Unknown / Not Stated
0, 0	3 KnownPlaceholder	Private Property

Both location blocks are null, and every roadway element is unavailable with the same reason. These two coordinates are accepted even though they're outside the state.

7.2 Unresolved

When nothing on the road network is within 1,000 ft, the answer is **2 Unresolved**, and **diagnostics.reason** says: "No road found near this point -- it appears to be off the mapped road network entirely (e.g. private property, a field, or another non-roadway location)." The roadway elements that don't depend on a road - rural / urban, ownership, district - are still filled.

> **Unresolved or failed?** if a GIS source failed while the local-road search ran, the answer is still Unresolved but the reason is left blank and the source is named in failedSources instead - the point may well be on a road (section 11).

8. The roadway (MIRE) data

MIRE - the federal Model Inventory of Roadway Elements - is the standard list of roadway facts that safety analysis relies on. Every answer carries all 20 elements in **mireElements**, each in the same shape:


```
"speedLimitMph": {
  "value": 50,
  "isAvailable": true,
  "unavailableReason": null,
  "source": "Speed_Limits/0"
},
"accessControl": {
  "value": null,
  "isAvailable": false,
  "unavailableReason": "No access-control source for non-Interstate routes",
  "source": "n/a"
}
```


One filled element and one unavailable element

- **Check isAvailable, not value.** An unavailable number comes back as 0, which is not a real value.
- **source** names the GIS layer the value came from, so every value can be traced.
- **unavailableReason** says why an element is empty for this crash.

8.1 Every element

Element	Filled when	Value
speedLimitMph	A speed-limit record matches the route, or the matched local street has one.	Miles per hour.
aadt, **aadtYear**	A traffic count point on the same state route is within 2 miles of the reference post (6 miles on an Interstate, where count points are further apart).	Vehicles per day from that point's latest count, and the count year. Points not counted within 3 years of the newest nearby count are skipped.
governmentalOwnership	Always.	**Federal** for Interstate and US routes, **State** for the state's own routes; off the state system, **City/Municipal** inside city limits, otherwise **Local**.
dotDistrictNumber	The point is inside a mapped district.	The district number.
ruralUrbanDesignation	Always.	true = urban (inside city limits), false = rural.
routeNumber, **segmentIdentifier**	On a state route.	The route ID; route@reference post, e.g. 281@71.50.
routeStreetName	On a state route or a local street.	The route label or street name.
federalAidRouteType	On a state route.	The route type written out, e.g. Interstate.
accessControl	On an Interstate.	Full Control (Interstate).
oneTwoWayOperations	On a local street whose centerline	As the street data writes it.

Element	Filled when	Value
	record carries it.	

8.2 Not available yet

These elements are always unavailable in this release, because no reliable public GIS source has been confirmed for them: **surfaceType**, **functionalClass**, **medianType**, **numberOfThroughLanes**, **directionOfInventory**, **beginPointDescriptor**, **endPointDescriptor** and **segmentLengthMiles**.

> **Why unavailable, not a guess:** an unavailable element tells your crash system to fill it another way; a guessed value would look just as trustworthy as a real one.

9. The single-crash endpoint

9.1 Request

Field	Rule
crashKey	Required, up to 50 characters. Returned unchanged, so the answer can be matched to the crash.
latitude	Decimal degrees, inside the state's bounding box - unless it's a placeholder coordinate (section 7.1).
longitude	Decimal degrees (negative in the western hemisphere), inside the state's bounding box - unless it's a placeholder.

9.2 Response

Field	What it holds
crashKey	As sent.
resolutionMethod	How it was resolved (sections 2.3 and 3).
linearReference	The state-route location, or null (section 4).
nearestFeature	The local-road location, or null (section 5).
outsideCityLimitsContext	Nearest town and distance, or null (section 5.3).
mireElements	All 20 roadway elements (section 8).
diagnostics.failedSources	GIS sources that didn't answer for this crash; empty when all did (section 11).
diagnostics.reason	The placeholder type, or why the point is Unresolved.
diagnostics.ambiguousRouteMatch, **algorithmVersion**	Reserved; currently always false and 1.0.0.
resolvedAtUtc	When the answer was worked out.

9.3 Validation errors

A request that breaks a rule gets **400** and a standard problem answer naming each bad field. Nothing is looked up.

```
HTTP 400
{
  "title": "One or more validation errors occurred.",
  "status": 400,
  "errors": {
    "Latitude": [ "Latitude must fall within Nebraska's bounds (39.5 to 43.5)." ]
  }
}
```

A point outside the state's bounding box

A GIS source failing is **not** an error: the answer still comes back **200**, with that source named in failedSources (section 11).

10. The batch endpoint

The batch endpoint takes a CSV of crashes and returns a CSV with one results row per crash, in the same order - for analysts with a list of points, or for loading a backlog in one go.

10.1 The file you send

Send the CSV as a multipart/form-data upload in a form field named **file**. It needs a header line and three columns, in any order; other columns are ignored:

Column	Headers it recognizes
Crash key	any header containing "crash" and "key", or "case" and "number", or exactly "id"
Latitude	any header containing "lat"
Longitude	any header containing "lon"

```
crash_key,latitude,longitude
223000101,40.83988549,-96.7399728
223000102,41.23462996952996,-96.1630662224083
```

The smallest valid batch file

10.2 How a batch runs

- Up to **5,000 rows** per upload.
- Rows are resolved **8 at a time**, each exactly as the single-crash endpoint would resolve it.
- Each row has **25 seconds**. A row that runs out of time gets an error row; the rest of the batch carries on.
- A row that can't be read, or breaks a validation rule, gets an error row too. **One bad row never fails the batch.**

10.3 The file you get back

resolved-crash-locations.csv, one row per uploaded row:

Columns	Contents
row_number, **crash_key**, **input_latitude**, **input_longitude**	What you sent.
error	Why the row wasn't resolved; blank for a good row.
resolution_method, **resolution_notes**	The method as a word, and diagnostics.reason.
route_id, **highway_label**, **reference_post**, **log_mile**	State-route location (reference posts to 3 decimals).
primary_feature_name, **cross_street_name**, **narrative_text**	Local-road location.
distance_feet, **direction**	Snap distance on a state route, distance from the feature on a local road; compass letter.
bridge_relation, **bridge_crossed_feature**	Over / Under and the crossed feature.
segment_identifier ... **dot_district_number**	The 20 roadway elements, in the order of the response;

Columns	Contents
	blank when unavailable. Rural / urban is written Rural or Urban.
failed_sources	GIS sources that didn't answer, separated by semicolons.

Not in the batch file: the nearest-mile-marker fields, the snapped point, the outside-city-limits context, and the reason for each unavailable element. Use the single-crash endpoint when you need those.

11. When a GIS source doesn't answer

Every answer depends on the GIS services being reachable. When one isn't, the service gives the best answer it can instead of failing:

- Each GIS request gets **8 seconds** per attempt and **20 seconds** in all. At most **6** run at once, so a batch doesn't flood the GIS.
- If the main route layer fails, backup copies of it are tried.
- If a source still doesn't answer, it's named in **diagnostics.failedSources** (failed_sources in a batch) and the answer is built from the rest. The elements that source would have filled come back unavailable.
- In a batch, a row that hangs for 25 seconds becomes an error row: "Resolution timed out after 25s (a GIS source may be unresponsive); try this crash again separately."

> **What to do:** an answer with anything in failedSources is incomplete, not wrong. Resolve that crash again later; if the same source keeps failing, the GIS service itself is likely down.

In the Refinery, the Doctor's spot-mapping check tells an administrator whether the service is reachable at all.

12. Configuration (System Admins)

The service is configured in the **ArcGis** section of its settings file, or by the matching environment variables. A change takes effect when the service restarts.

Setting	Default	What it controls
RouteSnapToleranceFeet	100	How close a state route must be to be matched (section 3).
StreetSearchRadiusFeet / **StreetSearchRadiusMaxFeet**	300 / 1,000	The local-road search, first and widened (section 5).
BridgeSearchRadiusFeet	1,000	How far to look for a bridge (section 6).
MunicipalitySearchRadiusMiles / **... MaxMiles**	10 / 40	How far to look for the nearest town (section 5.3).
AttemptTimeoutSeconds / **TotalRequestTimeoutSeconds**	8 / 20	Time allowed per GIS attempt and in all (section 11).
MaxConcurrentGisRequests	6	GIS requests running at once.
NativeSpatialReferenceWkid	102704	The projection distances are measured in (a state plane, in feet).
Layers	the state's public GIS	The address of each GIS layer: highways (plus backups), mile markers, speed limits, AADT, districts, municipal boundaries, street centerlines, bridges, railroad crossings.

Pointing at another GIS: layer addresses are settings, but the service reads each layer's own field names (route ID, reference posts, street name and so on). A GIS whose layers name those fields differently needs the matching layer readers adapted - a change to the service, not a setting.

13. How accurate it is

The service has been measured against thousands of crashes whose official locations the state had already finalized: each finalized point was sent through the service and the answer compared with the state's own.

Road type	Resolved	Agreement with the finalized location
State routes	97%	Same route on every resolved crash. Reference post within about 5 ft at the median, and within about 80 ft for 9 crashes in 10.
City streets	100%	Location type 78-87%, primary street 63-77%, cross street 66-70% - best in the largest cities, where street data is most complete.
Town and village streets	100%	Location type 84%, primary street 73%, cross street 67%.
County roads	100%	Location type 82%, primary street 52%, cross street 28%.
Bridges	65% detected	When a bridge is detected, Over / Under is right 95% of the time.

13.1 Where to look twice

- **County roads.** Many rural roads have two names - a mile-grid name in the crash system ("ROAD 60") and a different one in the public street data ("South 60th Road"). The point is placed on the right road; the name may not be the one you expect. Closing this needs a crosswalk between the two naming systems.
- **Intersection or not** (section 5.1).
- **Stacked interchanges and bridges** (section 6).
- **Large snap distances** on a state route (section 4.2).

> **Note:** the finalized locations were set against the road network as it was then; the service uses the network as it is now. A road realigned or renumbered since can show a genuine difference that isn't an error.

14. Quick reference

14.1 Code values

Field	0	1	2	3
resolutionMethod	ReferencePost	NearestReferenceFeature	Unresolved	KnownPlaceholder
referenceType	Intersection	StreetOnly	Bridge	RailroadCrossing
direction (all)	N	S	E	W
bridge.relation	Over	Under		

14.2 Messages

Message	What to do
Latitude must fall within Nebraska's bounds (39.5 to 43.5). / Longitude must fall within Nebraska's bounds (-104.5 to -95).	The point is outside the state. Latitude and longitude are often swapped, or the minus sign is missing.
'Crash Key' must not be empty.	Send a crash key.
Missing crash key (batch)	Add the key to that row.
Latitude '...' is not a valid number (batch)	Use decimal degrees - no degree signs or letters.
Row has fewer columns than expected (batch)	Fix that line of the CSV.
Resolution timed out after 25s ... (batch)	Resolve that crash again on its own (section 11).
No road found near this point ...	The point is off the road network. Check it.
Unknown / Not Stated, **Private Property**	A placeholder coordinate (section 7.1). Nothing to do.
No CSV file was uploaded (batch)	Send the file in a form field named file.
Batch has n rows; this endpoint accepts at most 5000 per upload.	Split the file.

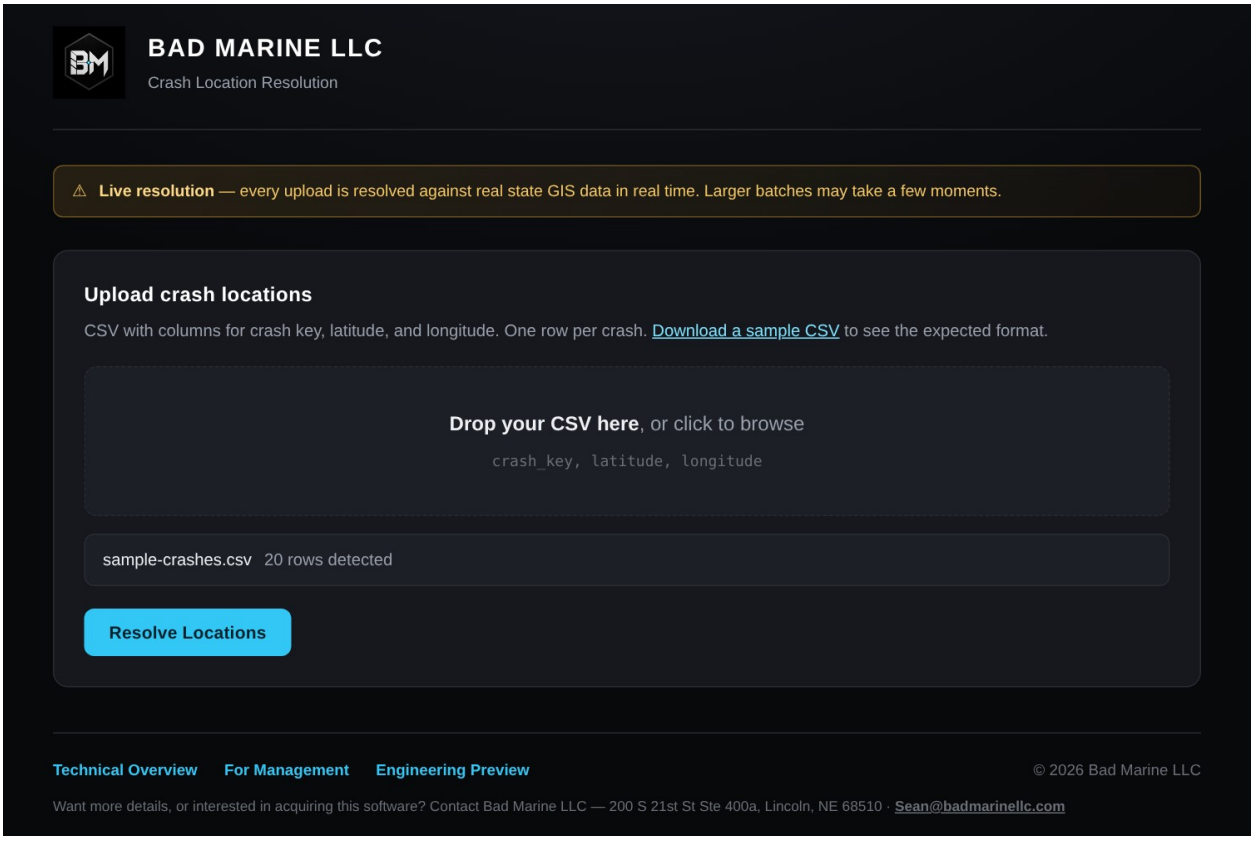
14.3 Glossary

Term	Meaning
LRS	Linear Referencing System - locating a point on a route by its distance along it.
Reference post	The crash's mile position along a state route, calibrated to surveyed mile markers.
Log mile	The distance along the route from its start.
Snap	Moving the point onto the nearest route centerline to measure along it. The point you sent is never changed.
MMUCC	Model Minimum Uniform Crash Criteria - the national crash-data standard, including how to describe a location with no reference post.
MIRE	Model Inventory of Roadway Elements - the national list of roadway facts used in

Term	Meaning
	safety analysis.
AADT	Annual Average Daily Traffic - the average number of vehicles a day.
GIS	Geographic Information System - here, the state's public map services.
Placeholder coordinate	A fixed coordinate standing in for an unknown or private-property location.
Crash key	Your identifier for a crash; returned unchanged.

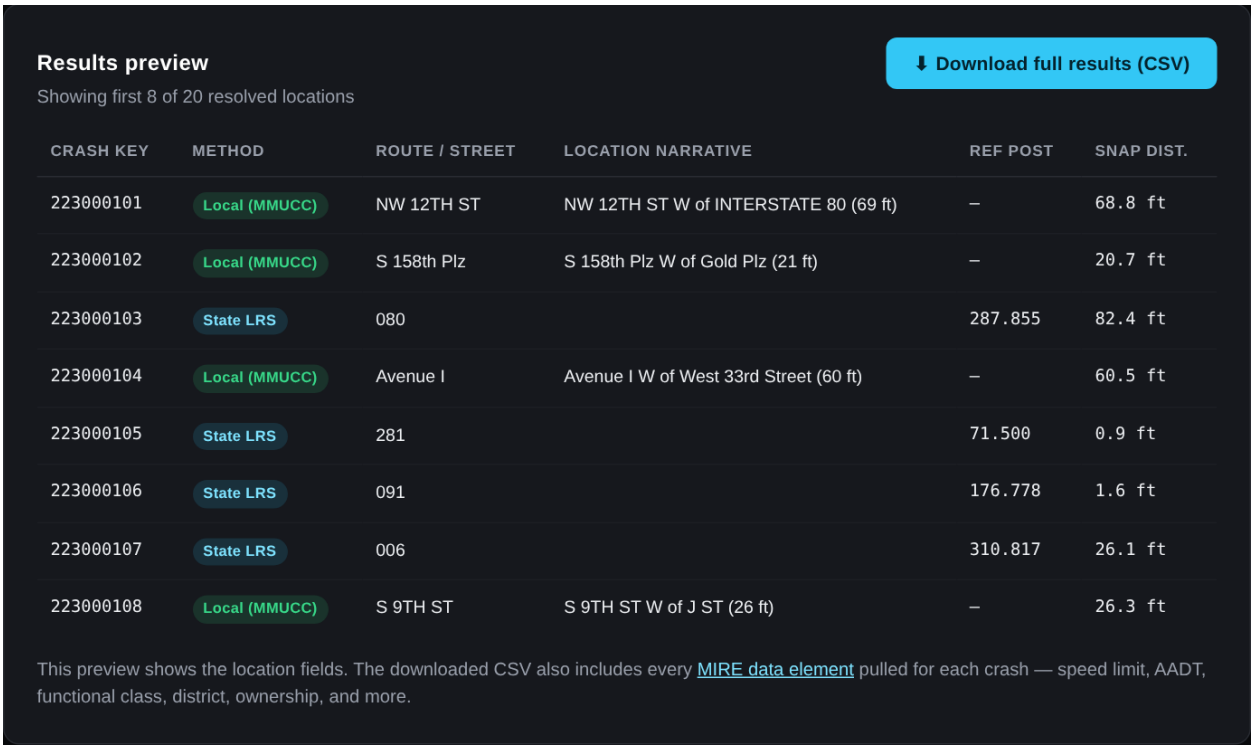
15. The online evaluation page

The evaluation page is a simple upload screen in front of the batch endpoint, so anyone can try the service without writing code. It resolves against the live GIS exactly as described in this manual.



The evaluation page

1. Drop a CSV on **Drop your CSV here**, or click to browse. **Download a sample CSV** gives you 20 example crashes to try.
2. Check the file chip - it shows the file name and how many rows were detected.
3. Click **Resolve Locations** and wait for the progress bar (roughly a third of a second per crash).
4. Read the **Results preview** - the first 8 crashes with their method, route or street, narrative, reference post and snap distance.
5. Click ⬇ **Download full results (CSV)** for every crash with every column (section 10.3). **Start over** clears the page for another file.



Results preview

Showing first 8 of 20 resolved locations

⬇ Download full results (CSV)

CRASH KEY	METHOD	ROUTE / STREET	LOCATION NARRATIVE	REF POST	SNAP DIST.
223000101	Local (MMUCC)	NW 12TH ST	NW 12TH ST W of INTERSTATE 80 (69 ft)	–	68.8 ft
223000102	Local (MMUCC)	S 158th Plz	S 158th Plz W of Gold Plz (21 ft)	–	20.7 ft
223000103	State LRS	080		287.855	82.4 ft
223000104	Local (MMUCC)	Avenue I	Avenue I W of West 33rd Street (60 ft)	–	60.5 ft
223000105	State LRS	281		71.500	0.9 ft
223000106	State LRS	091		176.778	1.6 ft
223000107	State LRS	006		310.817	26.1 ft
223000108	Local (MMUCC)	S 9TH ST	S 9TH ST W of J ST (26 ft)	–	26.3 ft

This preview shows the location fields. The downloaded CSV also includes every MIRE data element pulled for each crash — speed limit, AADT, functional class, district, ownership, and more.

The Results preview

Label	Means
State LRS	ReferencePost
Local (MMUCC)	NearestReferenceFeature
Placeholder	KnownPlaceholder
Unresolved	Unresolved
Error	The row has an error; the reason is in the narrative column.

- **Batch limits:** 20 crashes per batch, or 150 with an unlock code from Bad Marine LLC (**Have an unlock code?**). 150 is a hard ceiling for the evaluation, set by what the public GIS handles reliably; the batch endpoint itself takes 5,000.
- **Nothing is kept.** Download the results before you click Start over or close the page.
- The footer links to a **Technical Overview**, a short summary **For Management**, and an **Engineering Preview** of the code.

To evaluate a larger dataset, or to ask about acquiring the software, contact Bad Marine LLC at **Sean@badmarinellc.com**.