CDS Curb | CV Events Additions

The following are data objects based on release 1.1.0 of the CDS Events API. It has been extended to support computer-vision based inferencing capabilities which are highlighted in orange. No modifications to the CDS release have been changes, only additional columns or values have been added.

Curb Event

A Curb Event is a record of activity that happens within the geographic bounds of a Curbs object.

A Curb Event is represented as a JSON object, whose fields are as follows:

Name	Type	Required/Optional	Description
`event_id`	UUID	Required	The globally unique identifier of the event that occurred.
`event_type`	Event Type	Required	The event_type that happened for this event.
`event_purpose`	Event Purpose	Conditionally Required	General curb usage purpose that the vehicle performed during the event. Required for sources capable of determining activity type for relevant event_types.
`event_location`	GeoJSON	Required	The geographic point location where the event occurred.
`event_time`	Timestamp	Required	Time at which the event occurred.
`event_publication_time`	Timestamp	Required	Time at which the event became available for consumption by this API.
`event_session_id`	UUID	Optional	May be provided to tie known connected `park_start` and `park_end` event types together by a unique session ID. If *not* confident of being able to determine a `park_end` event at some time after `park_start` is recorded (i.e., you cannot detect when a vehicle departs), then do *not* use

			session_id. This field may be most useful to payment companies who provide their source data as sessions (typical for transaction data). *Note also*: the use of the term "session" across CDS means the start and end of curb usage of a vehicle, not necessarily a financial or payment session or transaction.
`curb_zone_id`	UUID	Conditionally Required	Unique ID of the Curb Zone where the event occurred. Required for events that occurred at a known Curb Zone for ALL *event_types*.
`curb_area_ids`	Array of UUID	Conditionally Required	Unique IDs of the Curb Area where the event occurred. Since Curb Areas can overlap, an event may happen in more than one. Required for events that occurred in a known Curb Area, if known and used, for these event_types: *enter_area, exit_area, park_start, park_end*
`curb_space_id`	UUID	Conditionally Required	Unique ID of the Curb Space where the event occurred. Required for events that occurred at a known Curb Space, if known and used, for these event_types: *park_start, park_end, enter_area, exit_area*
`data_source_type`	Enum Source Type	Required	General category of the source creating the event.
`data_source_operator _id`	UUID	Conditionally Required	Unique identifier of the entity responsible for operating the event data source. IDs can identify the fleet operator sending a data feed, or

			the organization (company or city) operating the sensor. IDs for fleet operators are required and global and come from the [data_source_operators.csv](#) file, and optional for others. Read our [How to Get a Data Source Operator ID](#) guide. An agency at their discretion may allow a small, local company to simply provide a consistent `data_source_operator_name` string instead of this field, otherwise this field is required.
`data_source_operator_name`	String	Optional	Name of the provider responsible for operating the vehicle, device, or sensor at the time of the event. May be sent along with `data_source_operator_id` or on its own for small operators at the discretion of the city.
`data_source_device_id`	UUID	Required	Unique identifier of this event source, whether sensor, vehicle, camera, etc. Allows agencies to connect related Events as they are recorded by the same source. If coming from a provider, this is a generated UUID they use and not the same as the external `vehicle_id`. If this field is needed for your use cases, review our [Privacy Guidance](#).
`data_source_manufacturer`	String	Optional	Manufacturer of the data source hardware or vehicle reporting event data.
`data_source_model`	String	Optional	Model of the data source hardware or vehicle

			reporting event data.
`sensor_status_is_com missioned`	Boolean	Optional	If a sensor was used to capture this event, the commissioned status at the time that the event was reported. Indicates whether the sensor is currently in a state where it should be reporting data.
`sensor_status_is_onl ine`	Boolean	Optional	If a sensor was used to capture this event, the online status at the time that the event was reported. Indicates whether the sensor is currently online and reporting data.
`vehicle_id`	String	Optional	A vehicle identifier visible externally on the vehicle itself. If this field is needed for your use cases, review our Privacy Guidance.
`vehicle_license_plat e`	String	Optional	The consistently placed vehicle license plate, usable by ALPR systems, when required for curb use. This field is potentially sensitive (depending on local, state, and national laws) and a data privacy framework is recommended for collecting, retention, deletion, obfuscation, and security. If this field is needed for your use cases, review our Privacy Guidance.
`vehicle_license_plate_jur isdiction`	String	Optional	Jurisdiction or state in which license plate is registered
`vehicle_license_plate_de tection_conf`	License Plate Detection Confidence	Optional	Confidence value of the license plate detection.
`vehicle_license_plate_re cognition_conf`	License Plate Recognition Confidence	Optional	Confidence value of the license plate recognition.

`vehicle_permit_number`	String	Optional	If applicable, the assigned permit number for this vehicle from the city agency.
`vehicle_length`	Integer	Conditionally Required	Approximate length of the vehicle that performed the event, in centimeters. Required for sources capable of determining vehicle length.
`vehicle_type`	Vehicle Type	Conditionally Required	Type of the vehicle that performed the event. Required for sources capable of determining vehicle type.
`vehicle_type_conf`	Vehicle Type Confidence	Optional	Confidence value of the vehicle type.
`vehicle_color`	Vehicle Color	Optional	Color of the vehicle that performed the event.
`vehicle_color_conf`	Vehicle Color Confidence	Optional	Confidence value of the vehicle that performed the event.
`vehicle_courier_type`	Vehicle Courier Type	Optional	Courier Type of the vehicle that performed the event.
`vehicle_courier_type_conf`	Vehicle Courier Type Confidence	Optional	Confidence value of the courier type label.
`run_id`	UUID	Optional	ID from Runs table containing information about models and model versions
`vehicle_propulsion_types`	Array of Propulsion Type	Conditionally Required	List of propulsion types used by the vehicle that performed the event. Required for sources capable of determining vehicle propulsion type.
`vehicle_blocked_lane_types`	Array of Lane Type	Conditionally Required	Type(s) of lane blocked by the vehicle performing the event. If no lanes are blocked by the vehicle performing the event, the array should be empty. Required for sources capable of determining it

			for the following event_types: *park_start*
`curb_occupants`	Array of Curb Occupant	Conditionally Required	Current occupants of the Curb Zone. If the sensor is capable of identifying the linear location of the vehicle, then elements are sorted in ascending order according to the start property of the linear reference. Otherwise, elements appear in no particular order. Required for sources capable of determining it for the following event_types: *park_start, park_end, scheduled_report*
`actual_cost`	Integer	Optional	If available from the source, the actual cost, in the currency defined in currency, paid by the curb user for this event. The currency type is sent in with the REST Endpoints JSON object. All costs should be given as integers in the currency's smallest unit. As an example, to represent $1 USD, specify an amount of 100 (for 100 cents).

Event Type

Curb Event Type `event_type` enumerates the set of possible types of Curb Event. The values that it can assume are listed below:

Name	Description
`comms_lost`	communications with the event source were lost
`comms_restored`	communications with the event source were restored
`decommissioned`	event source was decommissioned
`park_start`	a vehicle stopped, parked, or double parked
`park_end`	a parked vehicle leaving a parked or stopped state and resuming movement
`scheduled_report`	event source reported status at a scheduled interval

`enter_area`	vehicle enters the relevant geographic area
`exit_area`	vehicle exits the relevant geographic area

Source Type

Curb Data Source Type `data_source_type` enumerates the set of possible categories of sources that are sending this event. The values that it can assume are listed below:

Name	Description
`data_feed`	directly from a provider data feed sent to the agency
`camera`	video or static image processing source
`above_ground`	sensor deployed above ground
`in_ground`	sensor deployed in the ground
`meter`	a smart parking meter
`payment`	from payment system or app
`in_person`	an individual on site recording the event digitally or otherwise
`other`	sources not enumerated above

Vehicle Type

Type of vehicle `vehicle_type` similar to vehicle_type in MDS. For this CDS release the list will be developed independently here to accommodate CDS and MDS use cases, while still aligning to the MDS design principles. In the next major MDS 2.0 release and next CDS release, alignment between CDS and MDS vehicle types can occur.

Name	Description
`bicycle`	A two-wheeled mobility device intended for personal transportation that can be operated via pedals, with or without a motorized assist (includes e-bikes, recumbents, and tandems)
`cargo_bicycle`	A two- or three-wheeled bicycle intended for transporting larger, heavier cargo than a standard bicycle (such as goods or passengers), with or without motorized assist (includes bakfiets/front-loaders, cargo trikes, and long-tails)
`car`	A passenger car or similar light-duty vehicle
`scooter`	A standing or seated fully-motorized mobility device intended for one rider, capable of travel at low or moderate speeds, and suited for operation in infrastructure shared with motorized bicycles
`moped`	A seated fully-motorized mobility device capable of travel at moderate or high speeds and suited for operation in general urban traffic

`motorcycle`	A seated mobility device capable of travel at high speeds and suited for operation in general urban traffic or expressways
`truck`	A light or heavy duty 4 wheeled truck
`van`	A van with significant interior cargo space
`freight`	A Box Truck or large delivery truck with attached cab
`other`	A device that does not fit in the other categories
`unspecified`	Unspecified
bus	A bus

Vehicle Type Confidence

Vehicle_type_conf is a softmax confidence score where 0 is the lowest confidence and 100 is the highest. It is used to show the level of confidence in the detected vehicle type.

Value	Description
`0.00-1.00`	Confidence level

Propulsion Type

Propulsion type `vehicle_propulsion_types` of the vehicle, similar to propulsion_type in MDS. For this CDS release the list will be developed independently here to accommodate CDS and MDS use cases, while still aligning to the MDS design principles. In the next major MDS 2.0 release and next CDS release, alignment between CDS and MDS propulsion types can occur.

Name	Description
`human`	Pedal or foot propulsion
`electric_assist`	Provides power only alongside human propulsion
`electric`	Contains throttle mode with a battery-powered motor
`combustion`	Contains throttle mode with a gas engine-powered motor

A vehicle may have one or more values from the `vehicle_propulsion_types`, depending on the number of modes of operation. For example, a scooter that can be powered by foot or by electric motor would have the `vehicle_propulsion_types` represented by the array `["human", "electric"]`. A bicycle with pedal-assist would have the `vehicle_propulsion_types` represented by the array `["human", "electric_assist"]` if it can also be operated as a traditional bicycle. A hybrid vehicle may use `["combustion", "electric"]`.

Vehicle Color

vehicle_color is the color for the vehicle that we detected during the event. We currently support 10 colors.

Name	Description
`black`	
`white`	

`red`	
`yellow`	
`blue`	
`green`	
`purple`	
`orange`	
`beige`	Includes brown
`gray`	Includes silver

Vehicle Color Confidence

Vehicle_color_conf is a softmax confidence score where 0 is the lowest confidence and 100 is the highest. It is used to show the level of confidence in the detected color

Value	Description
`0.00-1.00`	Confidence level

Vehicle Courier Type

Courier type of vehicle detected. Currently there are 6 courier types available. Many others can be added as necessary.

`Amazon`	Amazon Delivery
`UPS`	UPS Delivery
`FedEx`	FedEx Delivery
`USPS`	Unites States Postal Service Delivery
`OnTrac`	OnTrac Delivery
`Aramark`	Aramark Delivery

Vehicle Courier Type Confidence

Vehicle_courier_type_conf is a softmax confidence score where 0 is the lowest confidence and 100 is the highest. It is used to show the level of confidence in the detected courier type

Value	Description
`0.00-1.00`	Confidence level

Event Purpose

General event purpose `event_purpose` that the vehicle performed during its event, discernible by observation, sensors, or self-reported in company data feeds. New event purposes MAY be generated to reflect local curb uses, but when possible, the following well-known recommended values should be used. It may not always be knowable, but where it is possible this information should be conveyed. If multiple purposes apply, then use the more descriptive/specific value.

Name	Description
construction	Construction of hard assets including buildings and roadside infrastructure
delivery	General delivery of parcels, goods, freight
emergency_use	Includes ambulance, fire truck, police
parking	Vehicle parking, charging, or stopping
passenger_transport	Picking up and/or dropping off of human passengers
special_events	Includes unloading equipment for concerts, theatre, street events
waste_management	Retrieval/disposal of waste
device_maintenance	Includes scooter pickup, drop off, battery swapping
autonomous	Autonomous vehicle use
ems	Emergency medical vehicle use
fire	Emergency fire vehicle
food_delivery	Delivery of food items ready for consumption to an end consumer
parcel_delivery	Delivery of parcels, including bulk food goods to a restaurant or other business
police	Use by a police vehicle
public_transit	Includes large or small buses or paratransit.
ride_hail	Includes privately run ride hailing services
road_maintenance	Includes pothole patching, striping, snow plowing, street sweeping
service_vehicles	Includes private sector activity like some utilities
taxi	Traditionally licensed taxi services
utility_work	Includes public sector activity like sewer, water, telecoms
vehicle_charging	Parking for electric vehicles to charge
vehicle_parking	Includes private or commercial vehicle free or paid/metered parking
vending	Mobile vending or food truck curb uses
unspecified	Unknown or unspecified activity type
double_park	Parking or loading event occuring within travel lane

Lane Type

Type(s) of lane used or blocked `vehicle_blocked_lane_types` by the vehicle performing the event, outside of curb zones. E.g., double parking.

Name	Description
`travel_lane`	A standard vehicle travel lane.
`turn_lane`	A dedicated turn lane.
`bike_lane`	A lane dedicated for usage by cyclists.
`bus_lane`	A lane dedicated for usage by buses.
`parking`	A lane used for parking, not allowed for travel.
`shoulder`	A portion of the roadway that is outside (either right or left) of the main travel lanes. A shoulder can have many uses but is not intended for general traffic.
`median`	An often unpaved, non-drivable area that separates sections of the roadway.
`sidewalk`	A path for pedestrians, usually on the side of the roadway.
`unspecified`	Unspecified
`center_turn_lane`	A center lane available for left turns in both directions. This lane is sometimes used for courier parking for loading activity.

[Top](#)

Curb Occupants

A Curb Occupant `curb_occupants` object represents a specific vehicle's occupancy in a curb region at a specific point in time. Curb Occupant objects contain the following fields:

Name	Type	Required/Optional	Description
`type`	[Vehicle Type](#)	Required	The vehicle type of the occupant. When the event source is not capable of distinguishing vehicle type, this property must take the value "unspecified".
`length`	Float	Conditionally required	The approximate length in centimeters of the vehicle. Required when the event source is capable of determining vehicle length.

`linear_location`	Array of Float	Conditionally required	A two-element array that specifies the start and end of the occupant's linear location relative to the start of the Curb Zone in that order. Required when the event source is capable of determining the linear location of occupants.

Status

The Curb Status is the current status of sensors that are monitoring curb places.

A Curb Status is represented as a JSON object array of all deployed sensors, whose fields are as follows:

Name	Type	Required/Optional	Description
`data_source_device_id`	UUID	Required	Unique identifier of this event source, whether sensor, vehicle, camera, etc.
`data_source_type`	Enum Source Type	Required	General category of the source creating the event.
`data_source_operator_id`	UUID	Conditionally Required	Unique identifier of the entity responsible for operating the event data source. Can be global from data_source_operators.csv or defined per city.
`sensor_status_is_commissioned`	Boolean	Optional	If a sensor was used to capture this event, the commissioned status at the time that the event was reported. Indicates whether the sensor is currently in a state where it should be reporting data.
`sensor_status_is_online`	Boolean	Optional	If a sensor was used to capture this event, the online status at the time that the event was reported. Indicates whether the sensor is currently online and reporting data.