Data Importer

Information related to the data importer.

- [Using the Database Importer](#)
- [Importer Field Descriptions](#)

Using the Database Importer

Introduction

The database importer allows the admin to import their existing information into the OPS-COM system. Given a CSV file, the importer will transfer the data into the selected destination table.

The database importer only allows data to be imported into a select number of tables.
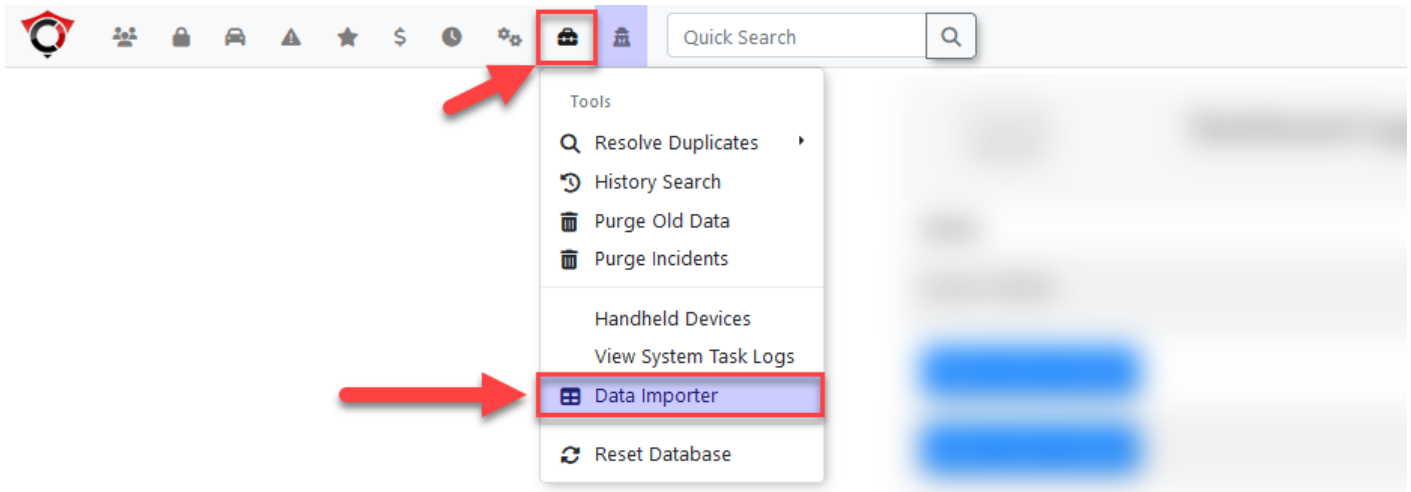
The currently allowed tables are as follows:

- UserProfile
- OffenceLocations
- Vehicle
- Permits
- Violations

Support for additional tables will continue to be added in the future.

Initial Setup

The database importer can be found listed under the tools tab.

If the database importer page is not visible, the user may not have the proper permissions. Before a user can import data, they must have the necessary permissions enabled on their account.

There are two permissions associated to table imports.

- Manage tables - allows the user to view the manage tables page, and use the uploads functionality, but NOT the ability to reset tables
- Reset tables - allows the user to reset the tables (empty them). Won't allow them to see the manage tables page.

The permissions can be found under the systems tab of the permissions management page. Enable them to allow a user access to the page.

For more information on permissions, refer to this article: https://wiki.ops-com.com/x/B4DjAQ

Importing Data into a Table

To start a new import, click on the upload button next to the destination table the information will be imported into.

Manage Administrator Roles

Editing Permissions: Tomahawk

👥 10   🔒 4   🚗 13   ⚠️ 12   ⭐ 25   📍 11 $ 9   📖 6   ⚙️ 23

✓ Manage Tables   The ability to manage data tables.

✓ Reset Tables   The ability to reset data tables.

Save Permissions

This will open a modal window prompting the user to select the CSV file they will be importing into the destination table. The imported CSV file can be comma-differentiated or semi-colon-differentiated.

Once a file has been selected, press the submit button to begin the process.
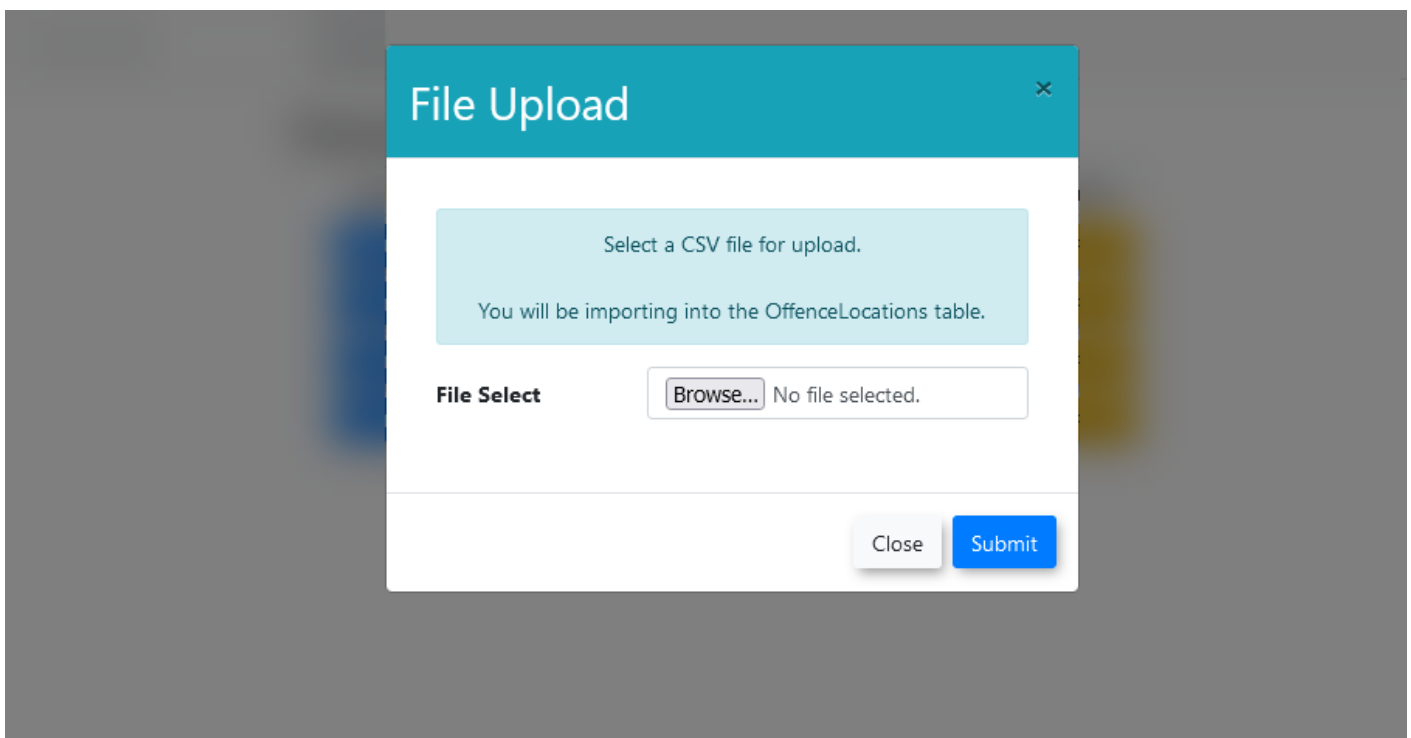
Once the CSV file has been uploaded, the table import screen will appear.
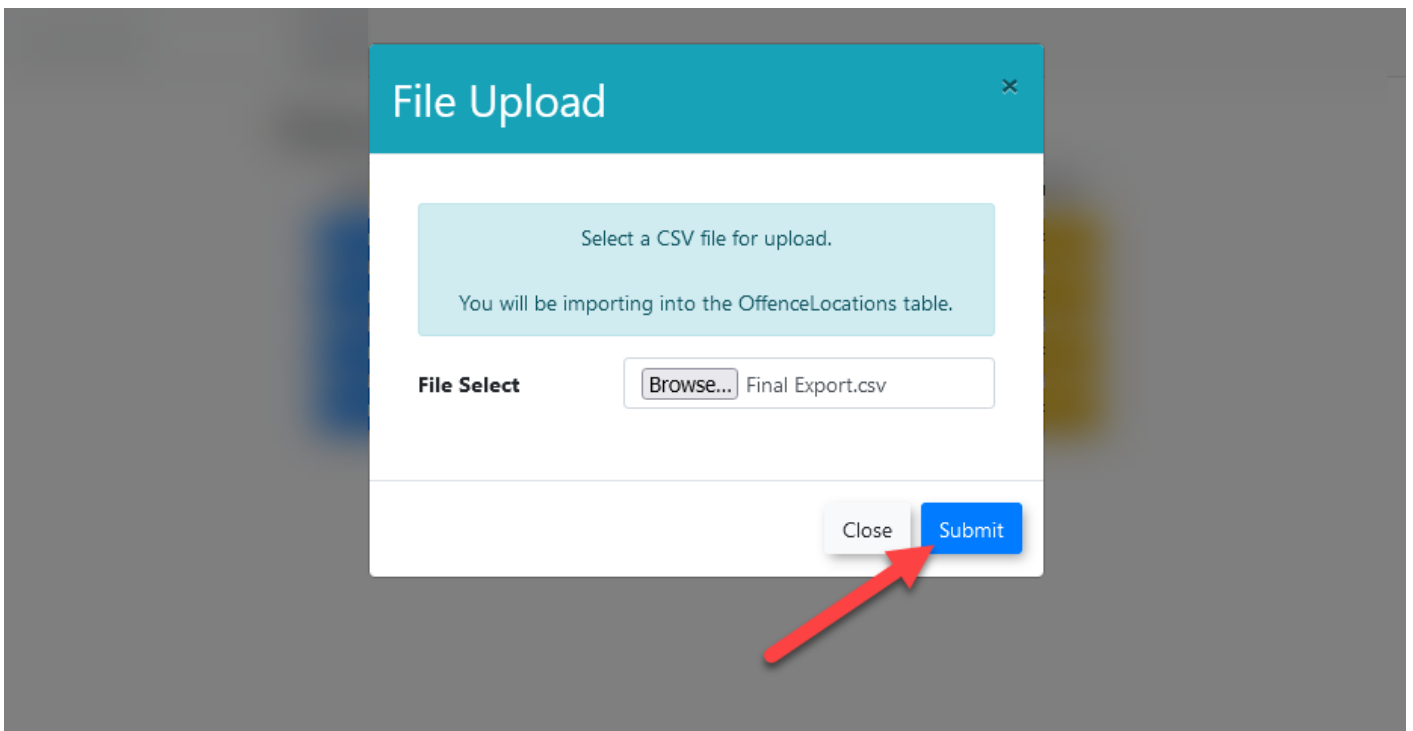
Column Matching

On this page, the columns in the imported file are matched to the columns existing in the destination table.

This is to let the program know where the data from the CSV file belongs in the destination table.

If the CSV file contained recognized column names, they will be automatically selected.

For more information on what each column is used for, or whether a column is required or optional, refer to this guide here: https://opscom.wiki/books/for-administrators-staff/page/importer-field-descriptions

Table Import

Is Unique	Notes	OffenceLocations Table Column ❓	Temp Table Column ❓
✓		LocationName	- Not Mapped -
		WriterVisible	- Not Mapped -
		GisNo	- Not Mapped -
	On finding a duplicate	Ignore	

Process

One the columns have been properly matched and the process button has been pressed, the user will be redirected back to the main page while the import is completed in the background.

If an email address is setup on the importing user's account, they will be informed on the progress of the import by emails sent to their email address.

Another import cannot be started until the current one is either completed or has been terminated. Terminating the progressing import will delete the temp tables created during the import, and isn't recommended unless the import has become stuck.

An existing import is already in progress. You must wait until it has been completed before starting another import.
If you would like to terminate the import in progress, click here

Post-Import Processing

After the initial import of the data into the base table, some tables run additional processing on the information.

These processes run separately from the import process, and are not affected by the duplicate settings chosen for the import.

After this process has been completed, the user will be sent a second email containing the final details of the import, including how many records were imported, updated, and any post-processing details.

Permits

- If the user association column is matched, the created permit will have a booking automatically created for the associated user through the creation of a PermitJoin record.
- If the option was selected, the newly-created permits that were booked to users will be automatically marked as paid. Otherwise, they will be located in the users' carts and the users must pay for them.

UserProfile

- When users are imported, if they aren't provided a LoginSource, it will be automatically set to OPSCOM.
- Users that are created are set to enabled automatically.

Vehicle

- If the user association column is matched, the created vehicle will be automatically associated to the user through the creation of a VehicleJoin record.
- If the alert column is matched, an alert will be created and automatically attached to the associated vehicle through the creation of an AlertComments records. The vehicle will also be flagged.

Violations

- Any violations created that don't have an Issued date and Due date will have one created for them at the time of the import.

OffenseLocations

- This table has no post-processing.

Additional Settings

Additional details on import settings.

Order of Operations

Imports that contain related information have a general order they should be done in, as some of the tables contain information from another table.

The imports are able to be done out of order, but the records cannot be associated correctly to each other when done out of order.

In general, the imports the table requires should be imported before it is.

Table Import	Requires	Notes
UserProfile		Should be the first thing imported.
Vehicles	• UserProfile	Having the user record created before the vehicle is what allows the user to be associated to the vehicle by a vehicleJoin record created during the post-import processing.

OffenceLocations		Doesn't require any data beforehand, but should be included when creating a violation.
Permits	• Vehicles ○ UserProfile	Having a vehicle record created before the permit is what allows the permit to be associated to the vehicle by a permitJoin created during the post-import processing. If there is also an associated user record, the permit is able to be marked as paid by joining the user and permit through a payment record.
Violations	• Vehicles ○ UserProfile • OffenceLocations	A violation requires the existence of a vehicle record beforehand to have the violation be created at all. Offence locations should be imported before violations in order for the violation to have its location properly marked.

User-Association Settings

Some tables have additional processing that is run after the import is completed to associate the newly-created records to existing users.

For this, the record identifying the uniqueness of the user must be selected.

It can either be UserUUID or Email, so make sure the same value is selected for both sections.

The tables that currently have this as an option are:

- Vehicles
- Permits

Mark Permits as Paid

Permit importing has an additional option for if you would like to have the booked permits that are created automatically marked as paid.

Otherwise, when they are created, they will be located in the associated users' carts.

Vehicle Alerts

Vehicle importing has an additional option for setting up an alert on a vehicle automatically when they are created.

Include a column containing vehicle alert comments within the imported CSV file and select the column in the alert section of the vehicle import page.

Alerts will be automatically associated to the proper vehicle after import.

The list of vehicles being marked with alerts may also be called a hotlist.

Unique Identifiers & Duplicate Settings

This is the column that is used to tell if the information in the record is unique. If the information in this column is duplicated in the file or already exists in the system, it will be ignored or overwritten based on the selected setting.

- Ignoring a duplicate will have the system do nothing with the record.
- Overwrite existing will take the information in the new record and replace the existing record with it.

The number of columns that mark a record as unique varies from table to table.

- **UserProfile** uses one identifier, which can either be UserUUID or Email. For this table, it is one or the other, with UserUUID taking precedence over Email if both columns are supplied.
- **OffenceLocations** uses one identifier, which is LocationName.
- **Vehicle** uses one identifier, which is Plate.
- **Permits** uses two primary identifiers, PermitNo and LotNameID. Both are required to make a unique record, meaning identical PermitNo can be in use if they have different LotNameIDs.
- **Violations** uses one identifier, which is Ticket.

Foreign Lookup Columns

Some information in one table comes from another table and is stored as an ID in the destination table.

The data in these columns should be entered as normal, then the system will automatically look to see if there is a match in the corresponding table and input the correct ID value into the destination table.

For instance, one of the foreign keys for the Vehicle table is Colour, stored in the VehicleColour table. If any of the values in the column imported for Colour exist in the VehicleColour table, it will match them by name and store the proper ID in the Vehicle table.

If it doesn't find a matching value in the lookup table, it will enter a Null as the value instead.

A list of the import tables are listed below.

UserProfile Table

Inserting Into	Sourced From Table
prov	Provinces
UserTypeID	UserTypes

Vehicle Table

Inserting Into	Sourced From Table
ProvID	Provinces
ColourID	VehicleColours
MakeID	VehicleMake
TypeID	VehicleType
PlateTypeID	VehiclePlateType

Permits Table

Inserting Into	Sourced From Table
LotNameID	LotName

StateID	PermitState

Violations Table

Inserting Into	Sourced From Table
VehicleID	Vehicle
TicketType	TicketCategory
LocationID	OffenceLocations

Status Emails

Over the course of the import process, a total of two emails will be sent to the user who initiated the import:

- The first email is sent after the data from the CSV has been inserted into the temp table, and records the number of rows that were successful, and the rows that failed to be imported because they were malformed.
- The second email is sent after the data has been inserted into the base table. It contains the number of records that were updated in the base table, the number of records inserted into the base table, and any relevant information from the post-import processing that happened.

Table Reset

In addition to table imports, the page also allows for the purging of a table.

Click on the button that says Reset next to the table to be emptied.

This will open a new modal window, prompting the user to confirm they wish to delete the contents of the table. This action is permanent, and once the data is deleted, it cannot be recovered.
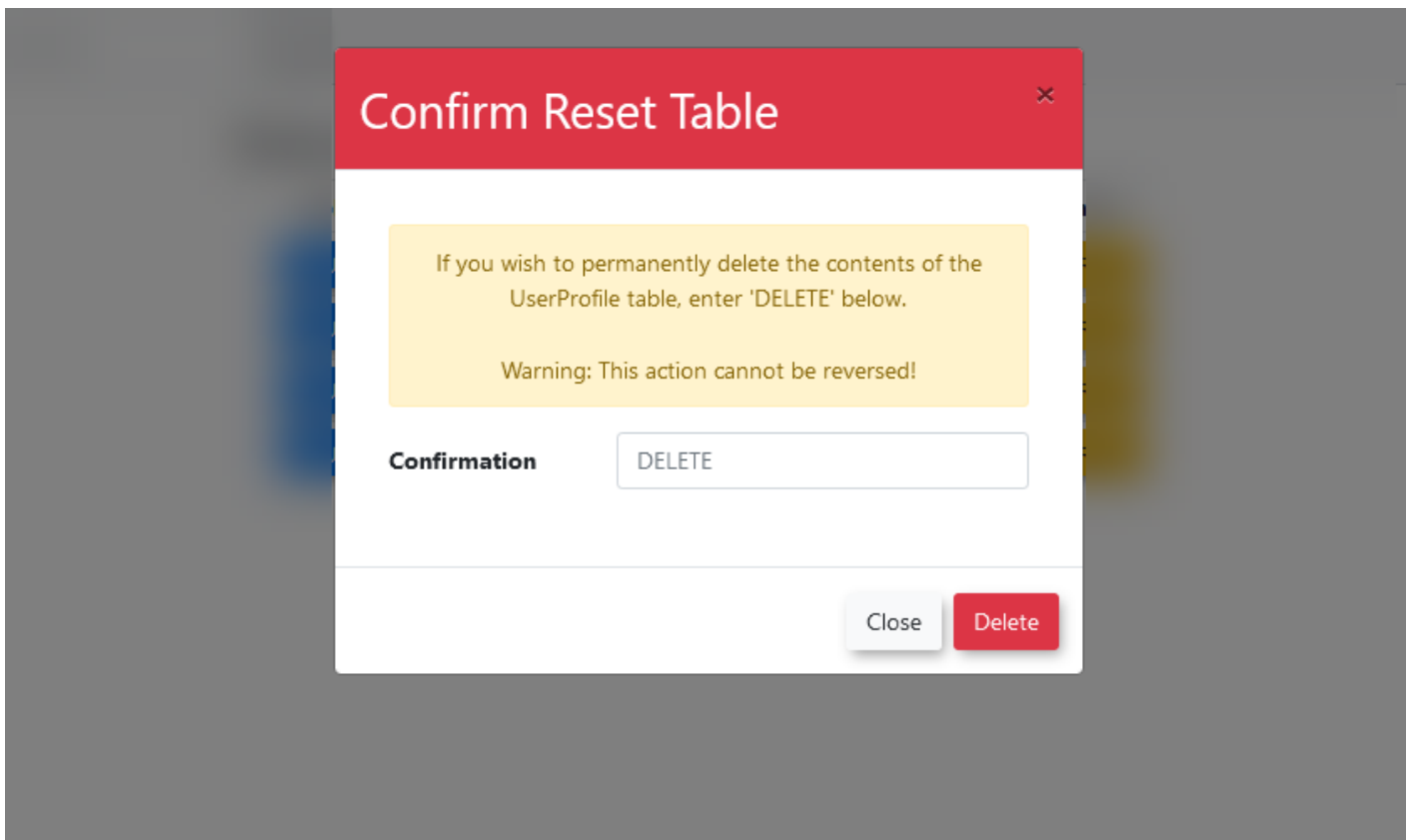
If it is certain the data should be deleted, enter DELETE into the confirmation text field and press the delete button.

The selected table will then be purged of all records.

Some tables are closely connected to the records of a related table, and removing the data they contain will also purge the contents of the related table.

- Purging the Vehicle table will also purge the VehicleJoin table.
- Purging the Permits table will also purge the PermitJoin table.

Note that only the table contents are deleted, and the table itself remains intact.

Related Pages

- https://opscom.wiki/books/for-administrators-staff/page/importer-field-descriptions

Importer Field Descriptions

Introduction

A list of all the columns used in the importer and a description of what the field is used for in the system.

- **Unique ID** fields are required, and are used to identify a record as being a unique entity in the system before it has been imported and an ID can be assigned to it.
  - They contain the bare-minimum information needed to import the record.
- **Recommended** fields are not technically required, and a record can be made without them, but they contain information that make the record a useful entity in the system.
  - This is information like the first and last names of a user or the value of the fine on a violation.
  - A record is able to exist in the system without these fields, but it won't do much without them.
- **Optional** fields are not required, and a record without them can still function as a useful entity in the system.
  - They contain supplemental information that can be filled in later or aren't needed for the basic functionality of the record, such as the year a vehicle was made.

Importer Columns

UserProfile Table

Column Name	Internal Name	Requirement	Type	Description

User Unique ID (UUID)	UserUUID	Unique ID • This or Email is Required	Standard	The primary identifier. If UUID is not supplied, email will be used as the key identifier instead. This value identifies the user as a unique record and it or the email is required to import a record.
Email Address	email	Unique ID • This or UUID is Required	Standard	The secondary identifier. If UUID is not supplied, email will be used as the key identifier instead. This value identifies the user as a unique record and it or the UUID is required to import a record.
Username	username	Recommended	Standard	The username of the user.
First Name	firstName	Recommended	Standard	The first name of the user.
Middle Name	middleName	Optional	Standard	The middle name of the user.
Last Name	lastName	Recommended	Standard	The last name of the user.
Password	password	Optional	Standard	The password of the user.
User Type	UserTypeID	Recommended	Foreign Key • UserTypes Table	The user type of the user. Attaches to the UserTypes table. The values supplied by the user in the uploaded file are expected to be values from the TypeName column of the UserTypes table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the TypeName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The UserTypeID from the UserTypes table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Street Address	street	Optional	Standard	The street address of the user.
City	city	Optional	Standard	The city of the user.

Province/ State	prov	Optional	Foreign Key • Provin ces Table	The province or state of the user. Attaches to the Provinces table. The values supplied by the user in the uploaded file are expected to be values from the ProvName column of the Provinces table, as these are the values that will be matched against. These are the long names of the province and not the abbreviation; as in, Ontario and not ON. The values in the uploaded file must be an exact match to the values in the ProvName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The ProvID from the Provinces table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Postal Code	postal	Optional	Standard	The postal or ZIP code of the user.
Cellphone Number	phonecell	Optional	Standard	The cellphone number of the user.
Employee Number	employNo	Optional	Standard	The employee number of the user.
Student Number	studentNo	Optional	Standard	The student number of the user.
Secondary Phone Number	sPhone	Optional	Standard	The secondary phone number of the user.
Date of Birth	DOB	Optional	Standard	The date of birth of the user.
Preferred Name	preferredname	Optional	Standard	The preferred name of the user.
Public Comment	publicComment	Optional	Standard	The public comment for the user. Visible to the user.
Private Comment	privateComment	Optional	Standard	The private comment for the user. Not visible to the user.
Login Source	loginSource	Optional	Standard	The method by which the user logs into the system. If a login source is not supplied for a user, it will be set to OPSCOM by default.

OffenseLocations Table

For additional information on this table, refer to this article: https://ops-com.atlassian.net/wiki/x/9ZYL

Column Name	Internal Name	Requirement	Type	Description
Location Name	LocationName	Unique ID • Required	Standard	The name of the location being imported. This value identifies the location as a unique record and is required to import a record.
Writer Visible	WriterVisible	Optional	Standard	Whether the writer of a violation of a ticket made in the location is visible to the user.
GIS Number	GisNo	Optional	Standard	The GIS number is a geographic location code and is only used by certain clients. Associated with GIS maps (https://www.gismaps.org/).

Vehicle Table

Column Name	Internal Name	Requirement	Type	Description
Licence Plate	Plate	Unique ID • Required	Standard	The license plate number of the vehicle. This value identifies the vehicle as a unique record and is required to import a record.
Vehicle Year	Year	Optional	Standard	The year of the vehicle.
Active Vehicle	active	Optional	Standard	Indicates whether the vehicle is active or not. If not supplied, vehicles will be set to active by default.

| Plate Type Name | PlateTypeID | Optional | Foreign Key
• VehiclePlateType Table | The plate type of the vehicle. Attaches to the VehiclePlateType table.

The values supplied by the user in the uploaded file are expected to be values from the TypeName column of the VehiclePlateType table, as these are the values that will be matched against.

The values in the uploaded file must be an exact match to the values in the TypeName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match.

The TypeID from the VehiclePlateType table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found. |
| Province/State | ProvID | Optional | Foreign Key
• Provinces Table | The province or state of the user. Attaches to the Provinces table.

The values supplied by the user in the uploaded file are expected to be values from the ProvName column of the Provinces table, as these are the values that will be matched against. These are the long names of the province and not the abbreviation; as in, Ontario and not ON.

The values in the uploaded file must be an exact match to the values in the ProvName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match.

The ProvID from the Provinces table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found. |

Vehicle Make Name	MakeID	Optional	Foreign Key • VehicleMake Table	The make of the vehicle. Attaches to the VehicleMake table. The values supplied by the user in the uploaded file are expected to be values from the MakeName column of the VehicleMake table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the MakeName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The MakeID from the VehicleMake table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Vehicle Type Name	TypeID	Optional	Foreign Key • VehicleType Table	The type of the vehicle. Attaches to the VehicleType table. The values supplied by the user in the uploaded file are expected to be values from the TypeName column of the VehicleType table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the TypeName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The TypeID from the VehicleType table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Vehicle Make Name	MakeID	Optional	Foreign Key • VehicleMake Table	The make of the vehicle. Attaches to the VehicleMake table. The values supplied by the user in the uploaded file are expected to be values from the MakeName column of the VehicleMake table, as these are the values that will be matched against.

Vehicle Colour Name	ColourID	Optional	Foreign Key • VehicleColours Table	The colour of the vehicle. Attaches to the VehicleColours table. The values supplied by the user in the uploaded file are expected to be values from the ColourName column of the VehicleColours table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the ColourName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The ColourID from the VehicleColours table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
VIN Number	vin	Optional	Standard	The Vehicle Identification Number (VIN) of the vehicle.

Permits Table

Column Name	Internal Name	Requirement	Type	Description
Permit Number	PermitNo	Unique ID • Required	Standard	This field is the first primary identifier. Both it and LotNameID must be provided to create a new permit This value identifies the permit as a unique record and is required to import a record.

Lot Name (Long)	LotNameID	Unique ID • Required	Foreign Key • LotNames Table	The long name of the lot associated with the permit. Attaches to the LotNames table. This field is the second primary identifier. Both it and PermitNo must be provided to create a new permit. This value identifies the permit as a unique record and is required to import a record. The values supplied by the user in the uploaded file are expected to be values from the LotName column of the LotNames table, as these are the values that will be matched against. These are the long form version of the lot name, and not the lot short name. The values in the uploaded file must be an exact match to the values in the LotName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The LotNameID from the LotNames table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Visible	visible	Optional	Standard	Indicates whether the permit is visible or not.
Permit Status	status	Optional	Standard	The status of the permit.
Permit State	StateID	Optional	Foreign Key • PermitState Table	The state of the permit.

Violations Table

Column Name	Internal Name	Requirement	Type	Description
Ticket Number	Ticket	Unique ID • Required	Standard	The unique identifier for the violation. This value identifies the violation as a unique record and is required to import a record.

Licence Plate	VehicleID	Recommended	Foreign Key • Vehicle Table	The license plate number of the associated vehicle. Attaches to the Vehicle table. The values supplied by the user in the uploaded file are expected to be values from the Plate column of the Vehicle table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the Plate column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The VehicleID from the Vehicle table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Ticket Type Name	TicketType	Recommended	Foreign Key • TicketCategory Table	The type of the ticket. Attaches to the TicketCategory table. The values supplied by the user in the uploaded file are expected to be values from the TicketTypeName column of the TicketCategory table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the TicketTypeName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The TicketTypeID from the TicketCategory table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.
Fine Amount	Fine	Recommended	Standard	The amount of fine associated with the ticket.
Towing Amount	Towing	Optional	Standard	The amount of towing charges associated with the ticket.
Tax Amount	taxAmount	Optional	Standard	The amount of tax associated with the ticket.
Issued Date	Issued	Recommended	Standard	The date the ticket was issued.
Due Date	Due	Recommended	Standard	The due date for payment of the ticket.
Ticket Writer	Writer	Optional	Standard	The admin who wrote the ticket.

Violation Location Name	LocationID	Recommended	Foreign Key • OffenceLocations Table	The name of the location where the violation occurred. Attaches to the OffenceLocations table. The values supplied by the user in the uploaded file are expected to be values from the LocationName column of the OffenceLocations table, as these are the values that will be matched against. The values in the uploaded file must be an exact match to the values in the LocationName column, matching the capitalization, spelling, and spacing exactly, or else it won't be counted as a match. The LocationID from the OffenceLocations table matching the name will inserted into the record in the base table. The value will be left blank in the imported record if no matches are found.