Address Override Mechanism for TMA ASAP-to-PSAP

February 2026

Contents

Overview

Certain Authorities Having Jurisdiction (AHJs) maintain address records within their Computer-Aided Dispatch (CAD) systems that differ slightly from the physical address records maintained in Manitou.

These differences can cause addresses to fail to be validated during the ASAP-to-PSAP Address Validation process. This prevents alarms from being able to be dispatched electronically. To ensure seamless interoperability, this project introduces an address override system within the AutoDispatch Gateway service, allowing site address fields to be transparently substituted or normalized according to AHJ-specific mappings defined in a JSON configuration file.

Specifications

Only the following fields are available to transform:

- streetNumber
- streetPreDirection
- streetName
- streetType
- streetPostDirection
- unit
- city
- county
- state
- zipCode

Case-insensitive replacement (for example, 'st.' becomes 'STREET')

Wildcard prefix/suffix match (*Main, Main*)

Null transform clears field value

@@APPEND: token appends content (Suite 100 becomes Suite 100 (Building A))

Setup

First identify where the AutoDispatchGateway service is running from (where the exe is located), which is typically from C:\Program Files (x86)\Bold Technologies\Manitou\AutoDispatch2.

In that location find the appsettings.json file and open it in Notepad. Scroll down toward the middle of the file and find the section for TMAASAP dispatch driver. After updating to patch 48, at the bottom of that section, there is a new variable to set.

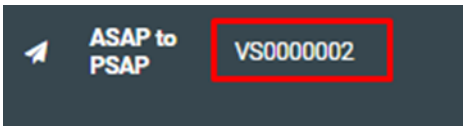
Note: This should be the filename only (no path) and the file should exist in the same directory as the executable.

```
"addressOverrideFile": "TmaAsapAddressOverrides.json"
```

The addressOverridFile is the JSON file where any transformations are listed. The file name can be anything, but the file type must be JSON. The file name in the appsettings.json needs to match the JSON file you create in the same directory.

Next, create the "TmaAsapAddressOverrides.json" file (or whatever the preferred name is).

The override JSON file will be grouped by dispatchAgencyID. The dispatchAgencyID maps to the ASAP to PSAP number set up under an Authority.



There is no limit to the number of dispatchAgencies that can be listed, but each will have its own section for its own individual transformations.

Please see the TmaAsapAddressOverrides.json attached to Address Override Mechanism for TMA ASAP-to-PSAP 2 as an example.

Scroll to the middle of the example to dispatchAgencyID VS0000002, which shows examples of all the different possible transformations.

Usage

When sending out to ASAP/PSAP the address might look like this:

```
"ADDR": "421 S Windchime Pl NE Suite #205 Colorado Springs CO 80919-1984"
```

Broken down to the individual components, it gets sent like this:

```
    "ADNO": "421",
    "ADPR": "S",
```

```
"ADST": "Windchime",
"ADTY": "PL",
"ADPO": "NE",
"ADUN": "STE 205",
"CITY": "Colorado Springs",
"REGN": "CO",
"POST": "80919-1984",
```

Based on the example TmaAsapAddressOverrides.json, these are how the fields will be transformed when they are sent to TMAASAP dispatch. This is also how it will show in the logger from the Auto Dispatch Gateway:

```
Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetNumber ), value( 421 ), transformedValue( 418 )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetPreDirection ), value( S ), transformedValue( Southernly2 )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetName ), value( Windchime ), transformedValue(  )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetType ), value( PL ), transformedValue( PLACE )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetPostDirection ), value( NE ), transformedValue( NorthEast )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( unit ), value( STE 205 ), transformedValue( STE 205  (Building A) )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( city ), value( Colorado Springs ), transformedValue( COLORADO SPRINGS )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( state ), value( CO ), transformedValue( COLORADO )

Value transformed: dispatchAgencyId( VS0000002 ), fieldName( zipCode ), value( 80919-1984 ), transformedValue( 80920 )
```

It shows the initial value, then what the transformed value is.

NULL Value:

This is ann example of choosing the transformed value in the JSON file to be null for the Streetname.

```
{
                    "fieldName": "streetName",
                    "sourceValue": "*Wind*",
                    "transformedValue": null
                }
```

The value being sent from Manitou is “Windchime”. The JSON entry is looking for anything before and anything after “wind” as defined by using the * before and after as a wildcard. You can see in the logger line, the transformed value, what gets sent to TMAASAP dispatch, is blank ( ).

```
Value transformed: dispatchAgencyId( VS0000002 ), fieldName( streetName ), value( Windchime ), transformedValue(  )
```

@@APPEND Value:

This is an example of choosing the transformed value in the JSON file to include an appended value for the Unit.

```
{
        "fieldName": "unit",
        "sourceValue": "STE 205",
        "transformedValue": "@@APPEND: (Building A)"
}
```

The value being sent from Manitou is “STE 205”. With that being the source value, the JSON file says to append whatever comes after the : to the source value to TMAASAP dispatch.

```
Value transformed: dispatchAgencyId( VS0000002 ), fieldName( unit ), value( STE 205 ), transformedValue( STE 205  (Building A)
```

When setting the transformed value, it will only send what the transformed value is specified as. For example, if you update the Streetname:

```
{
        "fieldName": "streetName",
        "sourceValue": "Wind*",
        "transformedValue": “WIND”
}
```

Notice there is a wildcard after “Wind” so it will find any streetname beginning with “wind” and whatever comes after it. This is how it will be transformed and sent to TMAASAP dispatch:

```
Value transformed: dispatchAgencyId( VS0000002 ),  fieldname( streetName ), value( Windchime ), transformedValue( WIND )
```

Notice that it changes the Streetname to the complete transformed value. It does not change it to WINDchime, appending the changed portion to the rest of the name. What is listed in the transformed value is what will be sent.

Making changes to the TmaAsapAddressOverrides.json does not require a restart of Auto Dispatch Gateway. Each time the file is changed and saved, it will automatically reload and show this in the logger:

```
Auto Dispatch Gateway Successfully loaded TMAASAP Address Overrides file. 3 Dispatch Agency sections found. 42 total address overrides found.
```

It shows how many dispatchAgencyIDs are found and the total address overrides found.