Form 11 Shortcodes

> The Form 11 Shortcodes feature allows administrators to dynamically auto-fill specific violation details into system templates. Its primary purpose is to automatically pull nested data directly from a violation record—such as ticket numbers, user profiles, or vehicle details—and populate the issued Form 11 accurately without manual data entry. This article is intended for OperationsCommander (OPSCOM) administrators responsible for configuring system templates and violation notices.

Setup and Configuration

This feature involves adding specific formatting strings directly to your system templates.

Admin Side: Administrators must have the appropriate system role permissions to access and edit system templates.

1. Navigate to the specific system template you wish to modify.
2. Insert the required shortcode syntax into the text editor where you want the violation data to appear.
3. Save the template.

User Side: This is strictly a backend configuration tool. End-users will simply see the fully rendered Form 11 document with their specific violation details already auto-filled.

Using this Feature

Administrators can use the following syntax rules and data packets to construct dynamic shortcodes for their Form 11 templates.

Understanding the Base Shortcode

The base shortcode for pulling Form 11 violation data is formatted as follows:

`[values show=violation._____]`

The blank space must be filled with a specific variable name to pull from the violation information packet. The packet itself is always called `violation`, and this identifier must be placed at the very start of any value call.

Navigating Nested Values

For each nested value within the system's data packet, the parent packet name must be added as part of the variable string. Each part of the path is separated by a period (`.`), with the specific variable you want to output placed at the end.

Example Shortcodes

The following table provides common examples of Form 11 shortcodes. To return a specific value in the same section, simply replace the provided variable name with the sibling value you want to output.

Example Shortcode	Description
`[values show=violation.Ticket]`	Returns the violation ticket number.
`[values show=violation.user_info.username]`	Returns the username of the user the violation belongs to.

Example Shortcode	Description
`[values show=violation.writer.DisplayName]`	Returns the display name of the admin who assigned the violation to the user.
`[values show=violation.vehicle.Plate]`	Returns the plate of the vehicle the violation was assigned to, if applicable.
`[values show=violation.vehicle.province.ProvName]`	Returns the province or state of the vehicle plate.
`[values show=violation.vehicle.colour.ColourName]`	Returns the registered colour of the vehicle.
`[values show=violation.vehicle.plate_type.TypeName]`	Returns the plate type of the vehicle.
`[values show=violation.vehicle.vehicle_type.TypeName]`	Returns the type of the vehicle.
`[values show=violation.vehicle.make.MakeName]`	Returns the make of the vehicle.
`[values show=violation.location.LocationName]`	Returns the name of the location the infraction happened.
`[values show=violation.details.0.Ticket]`	Returns the violation ticket number (alternate value location).
`[values show=violation.details.0.type.ViolationDescr]`	Returns the violation type description.

> If a shortcode is entered incorrectly, or if the targeted data field is empty (null) for that specific user or vehicle, the system will simply leave that space blank on the generated Form 11.

Violation Packet Reference

The violation info packet is formatted like the JSON example below. The values on the left are the exact variable names you must use in your shortcode path, while the values on the right are example outputs returned by this packet.


```
{
    "ViolationID": 237,
    "SemPermitID": null,
    "Ticket": "TT-19026",
    "VehicleID": 5,
    "Spoiled": 0,
    "TicketType": 1,
    "Issued": "2024-07-17 16:11:00",
    "convNotice": null,
    "Due": "2024-07-27 16:11:00",
    "ViolationTypeID": -1,
    "Fine": "50.0000",
    "AdjustedFine": "20.0000",
    "Towing": "0.0000",
    "taxAmount": "0.0000",
    "Writer": 7,
    "LocationID": 10,
    "Comment": null,
    "TicketAppeal": null,
    "appealType": 0,
    "appealFormat": null,
    "AccessAdminID": 50,
    "ActionedPer": null,
    "Created": "2024-07-17 20:12:08",
    "AppealUserID": 0,
    "AppealProcessDate": null,
```

```
"AppealAdminID": 0,
"AppealComment": null,
"AppealAdminComment": null,
"AutoNotice": 0,
"ProcessedByCollection": null,
"SentToCollections": null,
"PrivateComments": "",
"DriveAway": 0,
"UUID": null,
"VioNotice": null,
"latitude": null,
"longitude": null,
"Warning": null,
"userid": null,
"incidentID": null,
"failToIdentify": null,
"pin": "YL",
"duplicate": null,
"AdjustmentReason": null,
"notices_count": 1,

"user_info": {
        "UserID": 5,
        "studentNo": null,
        "employNo": "00800147",
        "username": "mashbury",
        "firstName": "Michael",
        "preferredname": null,
        "lastName": "Ashbury",
        "street": "688 Zaproxy Ridge",
        "city": "East Romaineburgh",
        "prov": 9,
        "postal": "K2G5B1",
        "email": "mash-bury@tomahawk.ca",
        "display_name": "Michael Ashbury",
        "account_no": "00800147",
        "prov_name": "Ontario",
},

"writer": {
```

```
        "AdminAccessID": 7,
        "Username": "admin",
        "DisplayName": "Administrator"
},

"vehicle": {
        "VehicleID": 5,
        "Active": 1,
        "Plate": "MASH",
        "Year": 1998,
        "TotalVio": 12,
        "TotalUnpaid": 2,
        "TotalWarning": 0,
        "vehicleAlert": true,
        "vin": "JNAMA43H9XGE50339",
        "electric_vehicle": false,

        "province": {
                "ProvName": "West Virginia",
                "ProvCode": "WV",
        },

        "make": {
              "MakeName": "Toyota"
        },

        "colour": {
              "ColourName": "Black"
        },

        "plate_type": {
              "TypeName": " Passenger"
        }

        "vehicle_type": {
              "TypeName": " Van"
        },

},
```

```
    "location": {
            "LocationID": 10,
            "LocationName": "Hospital & Administration Building",
    },

    "details": {
            "ViolationsDetailID": 359,
            "ViolationID": 237,
            "Ticket": "TT-19026",
            "ViolationTypeID": 16,
            "LocationID": 10,
            "offenceFine": 50,
            "discountFlag": 1,
            "discountAmount": 20,
            "discountHours": 240,

            "type": {
                    "ViolationTypeID": 16,
                    "ViolationDescr": "No Parking Snow Route - Mandatory Towing",
                    "DefaultCost": 50,
                    "discountFlag": true,
                    "discountAmount": 20,
                    "discountHours": 240,
            }
    }
}
```


Best Practices and Considerations

- **Maintain exact syntax: Variable names are case-sensitive and must exactly match the data packet format.** For example, using `provname` instead of `ProvName` will cause the shortcode to fail.
- **Understand nested logic: When targeting nested arrays (like user_info or vehicle), ensure you include all parent variables separated by periods in your shortcode path.** You cannot skip levels in the hierarchy.

- **Test templates thoroughly: Always test new Form 11 templates by generating a sample notice for a dummy violation.** This verifies that all shortcodes are rendering correctly and formatting as expected before they are sent to live users.

Revision #9
Created 26 July 2024 11:29:53
Updated 13 July 2026 13:14:29