

# AutoDispatch 2 - Core Configuration

12/21/2023 4:57 pm EST

## Configuration file

The AutoDispatch 2 uses a JSON file for configuration, as is standard practice for .NET Core applications. You'll need to know JSON syntax to edit this file. However, JSON syntax is very simple. The following site is a good starting point if you're not familiar with JSON syntax:

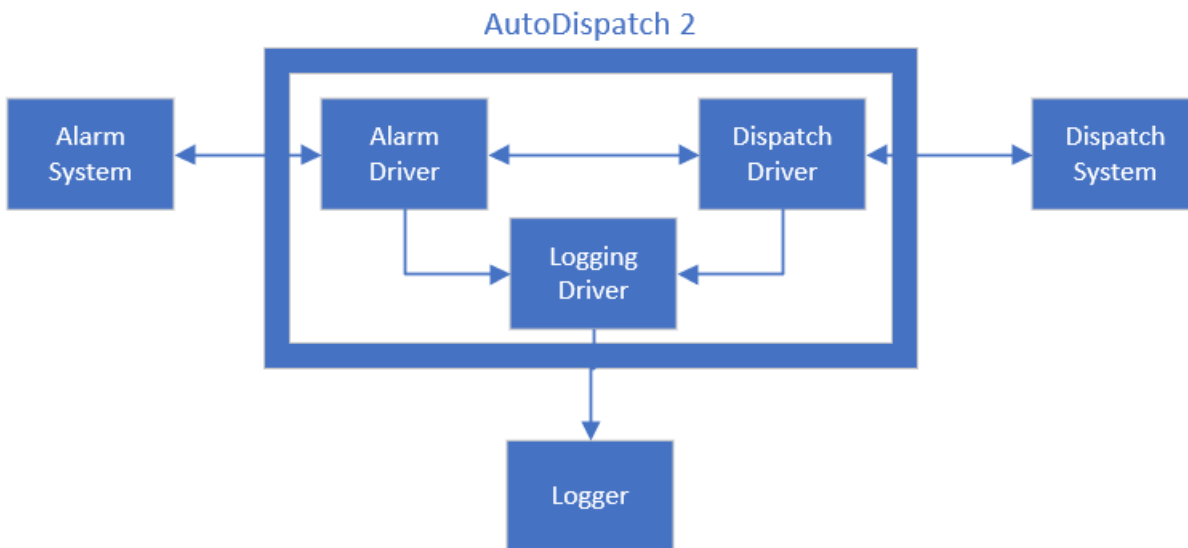
[https://www.w3schools.com/whatis/whatis\\_json.asp](https://www.w3schools.com/whatis/whatis_json.asp)

## Configuration sections

AutoDispatch 2 delegates all of the functionality of its operation to plugin driver files. There are three types of plugins:

- **Logging plugins**
  - Logging plugins are designed to write log information generated by the core product and the other plugins in a generic way for any logging platform.
  - Only a single logging plugin can be specified at a time
  - When AutoDispatch 2 is run as a command-line application, rather than as a service, it will always log to the console in addition to the specified logging driver.
  - AutoDispatch 2 has a hard dependency on the Manitou logger plugin and will log to both the console and the Manitou logger during startup before it has read the configuration and loaded the appropriate logging driver. It will then switch to the configured logging provider.
- **Alarm plugins**
  - Alarm plugins allow alarm monitoring software (like Manitou) to talk to AutoDispatch 2. These drivers perform the setup and communication necessary to talk to the specified alarm system.
  - Only one alarm plugin may be configured at a time.
  - As this driver type is how alarms ori
- **Dispatch plugins**
  - Dispatch plugins allow AutoDispatch 2 to send alarms to a CAD system and for the CAD system to send confirmation back to alarm.
  - Multiple dispatch plugins may be configured at once, but at least one **MUST** be configured.
  - If multiple dispatch plugins are specified, AutoDispatch 2 will use the PROVIDERID parameter of the alarm message to determine to which dispatch driver to route the message.
  - If no PROVIDERID parameter is present, AutoDispatch 2 will route to the first configured provider.
  - If only one dispatch plugin is configured, AutoDispatch 2 will always route requests to that driver.

## Conceptual diagram



The **alarm system** initiates an **alarm notification** and communicates directly to the **alarm driver**. The **AutoDispatch2** determines which **dispatch driver** should receive the message and forwards the message to that driver. The **Dispatch driver** contacts the **dispatch system**, delivers the alarm, and awaits confirmation. Confirmation is routed back through the system to the **alarm system**. Any logging information generated by **AutoDispatch 2**, or any driver, will be routed through the **logging driver** and forwarded to the **logger**.

## Structure of configuration file

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "Plugins": {
    "AlarmProvider": "ManitouAlarmProvider.dll",

    // Use this form when you want to only use one dispatch provider.
    "DispatchProvider": "AfaDispatchProvider.dll",

    // Use this form when you want to use more than one dispatch provider at a time.
    // "DispatchProviders": [ "Dc09DispatchProvider.dll", "GmsDispatchProvider.dll" ],

    "LoggingProvider": "ManitouLoggerProvider.dll"
  },

  // Driver specific configuration blocks go here
}
```

## Logging section

The logging section is the stock .NET Core logging configuration that all .NET Core applications share. It controls to which detail level log lines are emitted. In general, this can be left as-is, but if you need more or less detail from the logging components, this can be modified as needed.

## Plugins section

The plugins section specifies the alarm plugin, dispatch plugin, and logger plugin files.

If you want to use multiple dispatch providers, use the **DispatchProviders** array rather than the **DispatchProvider** string object. It is also possible to use **DispatchProviders** with just a single element in the array.