

# Minnesota

- [Release Major Themes](#)
- [Additional Features/updates/project improvements](#)
- [Known Bugs](#)
- [Changelogs](#)

**Release Date:** May 31, 2023

**Code Freeze:** May 10, 2023

**Namers:** Brett Preston (Linux Foundation - former and original LF project director)

The Minnesota release is version 3.0 (the 12th overall release and the third major release behind the Edinburgh and Ireland releases). Due to API and other tech debt changes, Minnesota is NOT backward compatible with the EdgeX 1.x or EdgeX 2.x releases. The release is not an LTS release. [Jakarta](#) (2.1) remain an LTS release. It is anticipated that the community will create its second LTS on top of Minnesota in the fall or 2023 (the Napa 3.1 release),

## Release Major Themes

- Common and Simplified Configuration of the EdgeX services
  - This is the new ability to now configure key settings of EdgeX in a single configuration location rather than needing to apply settings across multiple microservice manually
  - Configuration settings here include logging, telemetry config, the host address, security credentials, database and message bus settings, etc
  - Previously each EdgeX microservice loaded its own configuration and therefore the user was required to change settings in multiple places for it to affect all services. This was time consuming and error prone
  - The new common configuration approach provides a layered strategy, where in one place, users can configure settings common to all services or those that should only apply to the application or device services
  - Configuration settings can be loaded via a new core-common-config-bootstrapper service or via a file referenced by a new command line option
  - See here for full details <https://docs.edgexfoundry.org/3.0/microservices/configuration/CommonConfiguration/>
- Simplify the file formats used for EdgeX configuration
  - In EdgeX v2, different services were configured by different means: app services via TOML or JSON, device services via TOML or JSON, device profiles via YAML or JSON, provision watchers by JSON.
  - To reduce confusion and the learning curve for the user, better to consolidate where possible. JSON can be derived from YAML, therefore EdgeX TSC decided to drop TOML
  - EdgeX V3 has replaced usage of TOML in app services and device services configuration with YAML, with TOML removed from EdgeX
- V3 APIs
  - This is a major release bump so the APIs are updated to v3. This includes minor changes: TODO
- Microservice authentication with Vault JSON Web Tokens (JWT)
  - When run in secure mode, the EdgeX microservices will now require an authentication token before they will respond to requests issued over the REST API
  - The EdgeX Secret Store (Vault) now provides the tokens for each service to invoke authenticated outgoing requests to the other EdgeX services
  - Previously authentication was provided by the API Gateway which meant that internal requests from the local microservices were unauthenticated
  - See here for full details: <https://docs.edgexfoundry.org/3.0/security/Ch-Authenticating/>
  - Another positive result of this work is the replacement of the Kong API Gateway with the lighter-weight NGINX API Gateway. Static and run-time memory is reduced as a result
- Removal of much technical debt from EdgeX V2 including complete removal of ZeroMQ, SecretPath to SecretName refactoring, removal of dead code in the API Gateway flows, and many more

## Additional Features/updates/project improvements

- Upgrade to Go 1.20
- Requirements (UCR) and design (ADR) now in place for configuration via URIs. Likely to be implemented in the next minor release. Had a dependency on the Common Configuration added in this release. <https://docs.edgexfoundry.org/3.0/design/ucr/URIs-for-Files/> and <https://docs.edgexfoundry.org/3.0/design/adr/0027-URIs-for-Files/>
- Requirements (UCR) in place for Bring Your Own Vault (BYOV) allowing for an environment where an existing Vault instance is not setup by EdgeX: <https://docs.edgexfoundry.org/3.0/design/ucr/Bring-Your-Own-Vault/>
- Requirements (UCR) in place for Core Data Retention - that is a configurable cache of data held by EdgeX: <https://docs.edgexfoundry.org/3.0/design/ucr/Core-Data-Retention/>
- Added a STRIDE security threat model for identifying and mitigating risks in EdgeX: <https://docs.edgexfoundry.org/3.0/threat-models/stride-model/EdgeX-STRIDE>
  - STRIDE stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
- EdgeX has moved its external facing chat to GitHub Discussions allowing for communication to be better kept, searched and tagged (compared to the EdgeX slack channels). We are seeing multiple community discussions started each day on GitHub
- DevOps team has now embedded the Swagger API docs into the main EdgeX docs tree, e.g., <https://docs.edgexfoundry.org/3.0/api/core/Ch-APICoreData/>

## Known Bugs

- An error occurs when using an external MQTT connection with SecretName (vs SecretPath): <https://github.com/edgexfoundry/go-mod-bootstrap/issues/550>.
  - Workaround is here: <https://docs.edgexfoundry.org/3.0/microservices/core/command/Ch-Command/#configuring-for-secure-mqtt-connection>
  - **This issue is fixed in v3.1.**
- Insecure secrets callback does work for removed secrets: <https://github.com/edgexfoundry/go-mod-bootstrap/issues/534>. This applies to any map in Writable section (ASC pipeline functions, etc).
  - The workaround is to restarting service after removing entry.
  - This issue will be fixed in future v3.x release
- App Service template not using released version of App SDK: <https://github.com/edgexfoundry/app-functions-sdk-go/issues/1427>.
  - The work around is to set the App SDK version in your new app service to v3.0.0 when using the App Service template.
  - **This issue is fixed in v3.1.**
- The start time and end time fields of the Interval object in the Support Scheduler service always use UTC to run scheduling: <https://github.com/edgexfoundry/edgex-go/issues/4594>, but they should use the local timezone of the host system.
  - **This issue is fixed in v3.1.**
- Device USB Camera service will panic when adding device manually if CardName or SerialNumber data are missing: <https://github.com/edgexfoundry/device-usb-camera/issues/263>.
  - The work around is to follow the [documentation here](#) when adding devices manually.
  - **This issue is fixed in v3.1.**
- Unable to update existing devices using the Device C SDK function `edgex_metadata_client_add_or_modify_device()`. <https://github.com/edgexfoundry/device-sdk-c/issues/452>.
  - **This issue is fixed in v3.1.**
- App Service's standard routes (/version, /config, /secret) are not secured when running in Secure mode. <https://github.com/edgexfoundry/app-functions-sdk-go/issues/1435>
  - **This has been fix in 3.0.1 patch releases** of the App Functions SDK, App Service Configurable and App RFID LLRP Inventory. <https://github.com/edgexfoundry/app-functions-sdk-go/pull/1447>
    - <https://github.com/edgexfoundry/app-functions-sdk-go/releases/tag/v3.0.1>
    - <https://github.com/edgexfoundry/app-service-configurable/releases/tag/v3.0.1>
    - <https://github.com/edgexfoundry/app-rfid-llrp-inventory/releases/tag/v3.0.1>
- App Service Configurable's **external-mqtt-trigger** profile is missing **InsecureSecrets** section for MQTT. <https://github.com/edgexfoundry/app-service-configurable/issues/578>
  - Work around it to add the **InsecureSecrets** section manually and set **SecretName** properly as is done in this fix for Napa. <https://github.com/edgexfoundry/app-service-configurable/pull/579/files>
    - This can be done in the configuration.yaml and rebuild docker image or in Consul once service has started.
  - **This issue is fixed in v3.1.**
- The Device C SDK is unable to correctly parse the Unicode replacement rune in the JSON payload received from the message bus. <https://github.com/edgexfoundry/device-sdk-c/issues/465>.
  - Workaround is here: <https://github.com/edgexfoundry/device-sdk-c/issues/465#issuecomment-1705827784>
  - **This issue is fixed in v3.1.**
- App SDK attempts to connect to MessageBus when already connected. Results in an error when using NATS. <https://github.com/edgexfoundry/app-functions-sdk-go/issues/1497>
  - **This issue is fixed in v3.1.**

## Changelogs

[Core, Supporting and Security Services Changelog](#)

[Go Device Service SDK Changelog](#)

[C Device Service SDK Changelog](#)

[App Functions SDK Changelog](#)

[App Service Configurable Changelog](#)

[CoAP Device Service Changelog](#)

[Modbus Device Service Changelog](#)

[MQTT Device Service Changelog](#)

[REST Device Service Changelog](#)

[SNMP Device Service Changelog](#)

[Virtual Device Service Changelog](#)

[EdgeX User Interface Changelog](#)