

# Long Term Support

*This policy was updated in the summer of 2021, and approved by TSC vote on September 10, 2021 (in advance of Jakarta - the first anticipated EdgeX LTS)*

- [Release Cadence](#)
- [Definition of Support](#)
- [STS](#)
- [LTS](#)
- [Changes to an LTS](#)
- [Support of an LTS Release](#)
- [Recommended Use of Releases](#)
- [What's covered in an LTS Release?](#)
- [LTS Rebuilds](#)
- [Go Based Projects](#)
  - [Archiving Go Dependencies](#)
- [C Based Projects](#)
- [Rebuilding Docker Images](#)

STS is an abbreviation for “Short Term Support”.

LTS is an abbreviation for “Long Term Support”.

## Release Cadence

EdgeX Foundry plans to produce and release a new named version (like Delhi, Edinburgh, Fuji, etc.) of EdgeX roughly every six (6) months. Named versions will also have a version number in the form of X.Y (X is the major release number and Y is the minor release number; example 1.0 or 2.1). Depending on the features and compatibility with the prior release, the named release may be a major or minor release (see [releases, versions and patches](#)). The EdgeX TSC decides the nature (major or minor release) and final contents of each release approximately six months in advance of the release.

Note, EdgeX named releases incorporate core services, supporting services, and security services. The SDKs, app services and device services release independently, they must stay in sync on major versions and typically release in close proximity to the named released services.

## Definition of Support

Support, by the EdgeX community, is defined as:

- Community members best effort and development priority to fix major flaws (as determined by the project's appropriate working group chairperson (s) in consultation with the release manager) as soon as possible. Major flaws are critical or high bugs as defined [below](#).
- Availability of the source code, test code and artifacts, and program artifacts such as micro service executables, shared libraries, container images (Docker), packages (snaps), etc. through the support period that were created and made available at the time of the release (see what is included in support below)
- Availability of the platform documentation through the support period that were created to explain and operate the release and all associated artifacts that were made available at the time of the release (see what is included in support below).

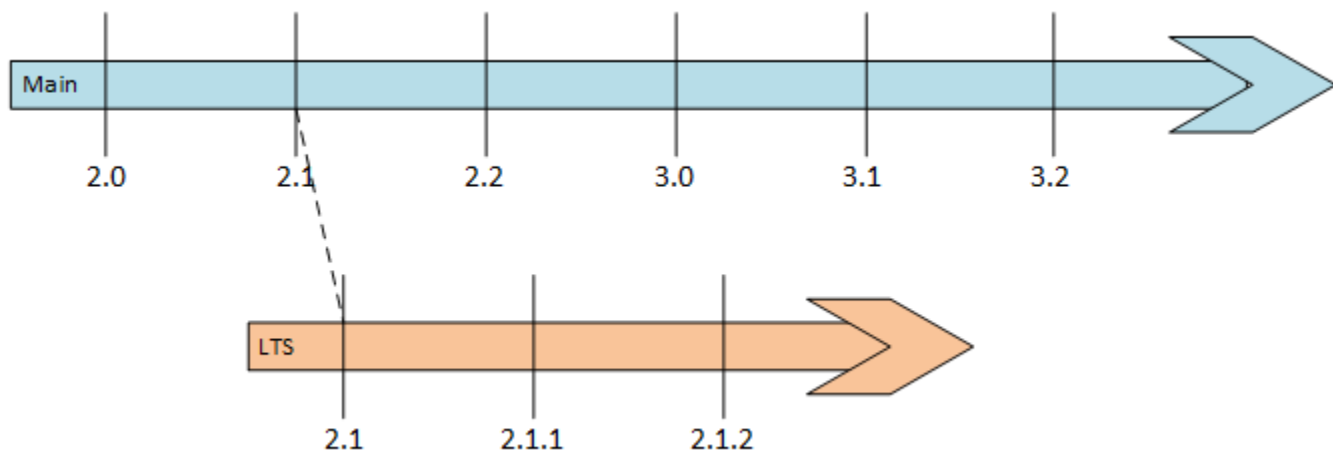
## STS

STS releases are supported for the duration of the next development cycle (~6 months). Support for an STS release ends once the next release occurs. Working group chairs are responsible for escalating issues that might trigger a Core patch release to the release czar and TSC. Timing of patch releases to resolve such issues will be coordinated by the release czar and the TSC. Patch releases of application/device SDKs and services are the responsibility of the working group chairs and the release czar (i.e. they don't require TSC approval).

## LTS

The TSC officially designates a release for LTS based on the needs of the community and the stability of the release. At the point of an LTS designated release, the TSC will define the support period for that release, which is 24 months by default. Support may be extended at TSC discretion.

When an LTS is released, new branches are created for the LTS release code allowing its independent maintenance and allowing development of EdgeX to continue on the projects' main branches.



## Changes to an LTS

During the development cycle of an LTS release, no new major functionality (at the discretion of the TSC) will be added. After the release happens, only significant security and bug fixes will be addressed. Significant is defined as high or critical issues per the following EdgeX definitions of issues and bugs:

- Critical - bug causes the system or service to crash and there is no work around for the function; or the issue is deemed a critical level CVE (per [C VSS](#))
- High - bug for a feature/function does not work and there is no work around for the function; or the issue is deemed a high-level CVE
- Medium - bug makes it more difficult to complete a function/task but there is a work around for the issue; or the issue is deemed a medium level CVE
- Low - bug is cosmetic in nature (examples: a misspelling in a UI or sort order error) and does not impede any function or task; or the issue is deemed a low-level CVE.

Work group chairpersons, in consultation with the release czar, will coordinate work on LTS bug fixes and make determinations on new LTS patch releases. Disagreements on whether a bug fix should be made on an LTS and/or the need for a new LTS patch release will be adjudicated by the EdgeX TSC. While a patch update to the LTS release is typically reserved for critical or high issues/bugs, the TSC reserves the right to make a patch release on-demand for any bug/security issue discovered.

The next major/minor release that follows after an LTS release is by definition an STS release. The EdgeX TSC can designate any release LTS, but a release is not LTS by default extension of the prior release. So, for example, if version 2.1 is designated an LTS release, version 2.2 created as part of the normal cadence six months after 2.1 is not automatically an LTS by extension.

## Support of an LTS Release

An LTS release is meant to be a stable release for which organizations typically base products and for which the EdgeX community offers “support” as defined [above](#).

Additional professional support (LTS or STS) including guaranteed service-level agreements and commercial support coverage may be provided by other members of the EdgeX Foundry community. The EdgeX Foundry project and Linux Foundation can provide connections to providers as necessary.

## Recommended Use of Releases

The community (developers) is encouraged to use the latest EdgeX development releases when creating new products and services. Businesses and organizations (adopters) are encouraged to use the latest LTS release when releasing (for production settings) their own products and services that are built on top of or including EdgeX. The EdgeX Foundry community makes announcements on its website about upcoming release end of life LTS events. Users are encouraged to watch the EdgeX Foundry web site and Wiki for support notifications. The [Releases](#) page of the EdgeX Wiki will always provide the latest status and news regarding EdgeX LTS releases.

## What's covered in an LTS Release?

The following EdgeX components are covered by this LTS policy

- The EdgeX core and supporting services.
- The latest compatible major/minor releases of EdgeX SDKs (application and device service SDKs at the time of this writing)
- The latest compatible major/minor releases of all reference application services
- The latest compatible major/minor releases of all reference device services
- The documentation (if there is a need to clarify a critical/high flaw)
- The release Docker Compose files (and associated Compose Builder tool)
- Deployment artifacts to include the Docker images, Snaps

Work such as examples (in the [edgex-examples](#) repository) will typically *not* be covered under LTS.

The items covered under LTS will be itemized in the release notes by the TSC when an LTS is designated. The TSC reserves the right to exclude some elements from LTS when designating an LTS. As mentioned previously in this document, the support period for the LTS and all of its elements will be stipulated at the time of LTS designation and will also be mentioned in the release notes for the release.

## LTS Rebuilds

## Go Based Projects

### Archiving Go Dependencies

- At the time of the release, Go module dependencies will be exported and committed to the repository using the `go mod vendor` command. This will allow the Go dependencies to be updated and the Go binary to be rebuilt.
- After the initial commit of go dependencies into the repository, if there are any changes to dependencies, developers should commit those back into the repository

## C Based Projects

- At this time, EdgeX builds all C-based dependencies from source code, so we just archive the required sources.

## Rebuilding Docker Images

All EdgeX docker images take advantage of a parameterized `FROM`. This allows us to easily change the base image when we build docker images. We can take advantage of this to build base images containing all build dependencies needed for rebuilding.

- All base images used in the build pipelines should be archived in the Nexus3 release repository and the Jenkins pipeline build scripts should be updated to use said Nexus3 repository.

Example Base Images:

- [nexus3.edgexfoundry.org:10003/edgex-devops/edgex-golang-base:golang-1.16](https://nexus3.edgexfoundry.org:10003/edgex-devops/edgex-golang-base:golang-1.16) -> [nexus3.edgexfoundry.org:10002/release/edgex-golang-base:golang-1.16](https://nexus3.edgexfoundry.org:10002/release/edgex-golang-base:golang-1.16)
- [nexus3.edgexfoundry.org:10003/edgex-devops/edgex-golang-base-arm64:golang-1.16](https://nexus3.edgexfoundry.org:10003/edgex-devops/edgex-golang-base-arm64:golang-1.16) -> [nexus3.edgexfoundry.org:10002/release/edgex-golang-base-arm64:golang-1.16](https://nexus3.edgexfoundry.org:10002/release/edgex-golang-base-arm64:golang-1.16)
- [nexus3.edgexfoundry.org:10003/edgex-devops/edgex-gcc-base:1.0.0](https://nexus3.edgexfoundry.org:10003/edgex-devops/edgex-gcc-base:1.0.0) -> [nexus3.edgexfoundry.org:10002/release/edgex-gcc-base:1.0.0](https://nexus3.edgexfoundry.org:10002/release/edgex-gcc-base:1.0.0)
- [nexus3.edgexfoundry.org:10003/edgex-devops/edgex-gcc-base-arm64:1.0.0](https://nexus3.edgexfoundry.org:10003/edgex-devops/edgex-gcc-base-arm64:1.0.0) -> [nexus3.edgexfoundry.org:10002/release/edgex-gcc-base-arm64:1.0.0](https://nexus3.edgexfoundry.org:10002/release/edgex-gcc-base-arm64:1.0.0)