Release Process

MidPoint is an open-source project that honors almost all the experience and best practice learned on other open-source projects. Most importantly we follow the *release early, release often* principle. MidPoint releases are quite frequent. The midPoint release process is also considerably agile but it is still quite predictable.

Release Types

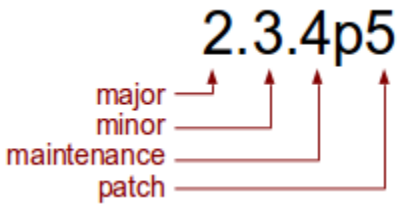
MidPoint is delivered in releases that follow quite a regular pattern. There are four types of releases:

Release type	What it delivers	How often it happens
Major	New project generation. Major releases deliver substantial changes. They may change even some of the features that were delivered and stabilized in minor releases. The change may not be backward compatible but it brings major improvements.	Once every few years
Minor	New features. This is the most common type of release. It is a gradual change that is almost always backward-compatible.	Twice per year
Maintenance	Bugfixes, stabilization, rarely even some minor features. The focus is on improving product quality and stabilization.	As needed.
Patch	Bug "hotfixes" that needs to be delivered immediately. Only fix critical or security bugs or bugs that affect customers with special-purpose support contracts.	As needed

Major, minor and maintenance releases are public. They will be announced, the documentation updated, etc. The patch releases are low-overhead releases. These will not be publicly announced (unless they are security fixes), there will be no public release notes or documentation updates. We need this to keep the development focused and the cost reasonable. Patch releases are usually done with a purpose to fix specific bug or address a specific situation and strongly relate to customers covered by support agreements. See below for more details. If several patch releases accumulate and the minor release is still quite far away the patch releases are summarized into a maintenance release.

Version Number

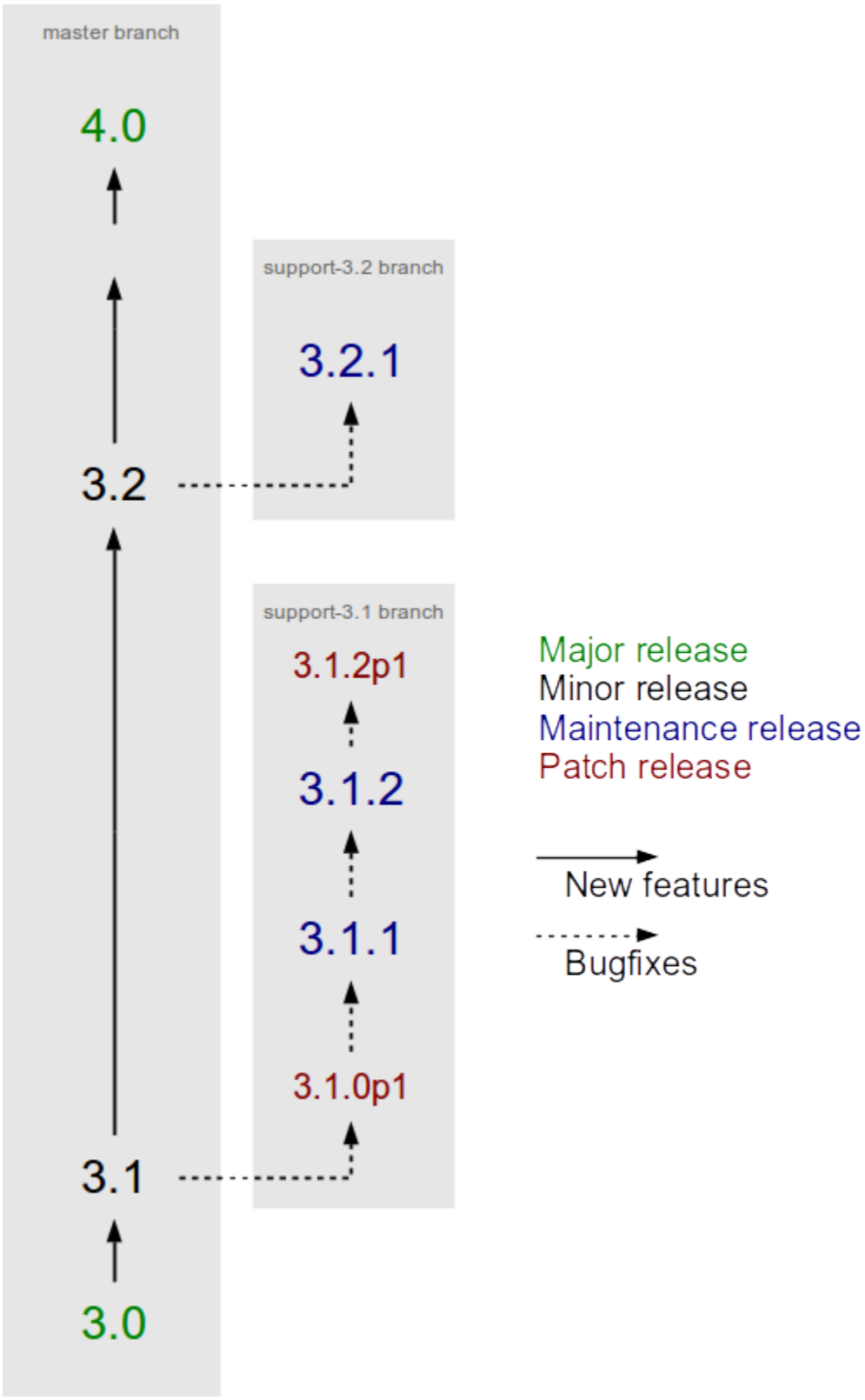
MidPoint version number reflects the type of a release that produced it:



Development process

We try to make the development process very efficient. That means we try to avoid duplication of work and overhead as much as possible. We do that by keeping the process very linear. It means that any new development goes to the development master branch. This also applies to bugfixes. Any reported bug is first fixed in the master branch (if possible). Then it might be back-ported to the maintenance branch under some circumstances. But we avoid branching and splitting the development as much as we can. Developing in a single master branch makes the development simple, avoids confusion and forces the developers to integrate their code immediately. This reduces the need for explicit integration phases and it keeps the project cost reasonable.

An example of the flow of midPoint releases is provided in the next diagram.



The development process is guided by the regular pace of minor deployments. We usually have two minor releases in each year and we plan to keep that even for the future. These are usually spring release (in April-May time frame) and autumn release (in October-November time frame). These are the releases that bring new features.

Maintenance and patch releases only fix bugs. Source code for such releases is split from the *master branch* into a *support branch* at the point where a minor release is finished. The goal of maintenance and patch releases are only to fix bugs and severe issues. As stated above the fix for such issues are implemented in master branch and then moved to the support branch and released. It means that all the fixes are already in master branch, integrated and tested continually. Therefore the support branch code may be discarded when a new minor release is ready. However it means that vast majority of the work on maintenance and patch releases is essentially wasted. Therefore we usually try to avoid these releases and just stick to minor releases. See the next section for more information about the releases.

Bug reports and Enhancement Requests

We want to make midPoint an excellent product. Therefore we appreciate any information or request that helps improve the product. We appreciate any such request no matter if it comes from existing customer, partner, potential user or any other interested party. There is no need to be covered by support agreement become a partner or have other formal relationship with Evolveum. We consider every idea or feature request.

Yet another difference is the way how the requests are delivered in the releases. Major and minor releases will contain all the bugfixes and features done to the date. However these is a difference when it comes to maintenance and patch releases. Such releases will be published only if:

- There are security issues or critical issues that impact large user population. Something that has a major impact such as a security vulnerability.
- There is an non-trivial issue raised by midPoint subscriber that explicitly requests that the issue has to be addressed in the maintenance or patch release.

The reason that we limit the "support privilege" only to subscribers is simple. Each maintenance/patch release has an overhead and especially back-porting of the bugfixes is not easy. This takes the time and energy from the main development on master branch. In addition to this all work done on patch releases is essentially wasted as the code is dropped when a new minor release comes. Therefore we need to make sure someone pays for this extra work. Otherwise all the midPoint users will suffer in a form of slower midPoint development.

When it comes to the feature requests and improvement ideas we obviously prioritize them as well. Simply speaking, feature requests are exclusive privilege of those that have active platform subscription. But if you have really interesting idea about the future of midPoint then please feel free to make a feature request anyway. Some great ideas are incorporated into the development plan. However, feature requests from platform subscribers are absolute priority, therefore community feature requests will be always moved out to make space for subscriber features.

See Also

- midPoint Releases
- Long-Term Support
- Using Support Branch

External links

- What is midPoint Open Source Identity & Access Management
- Evolveum - Team of IAM professionals who developed midPoint