

Default Streams: Threat or Menace?

The purpose of this report is to describe the actions needed in order to remove all existing default streams from Fedora. The concern is that default streams introduce complexity to the packager and user experience that should be backed out until such time as the design issues are addressed. The intention of this report is to identify how difficult it would be to do so in order to help FESCo reach a decision around proceeding with fixes or backing out the functionality.

General Problems

Upgrades with enabled streams - Stream is still available and still default

Upgrading from Fedora N with a module stream enabled as a default stream will attempt to continue with that stream into Fedora N+1. If that stream is available on Fedora N+1, the upgrade operation will succeed. At this time, the package manager cannot distinguish between packages installed because it was a default stream vs an explicitly-selected stream. For this document, I'll use the module stream `foo:1ts` as a representative of a module stream that exists in Fedora N and is still provided in Fedora N+1.

Possible Solutions

1. Do nothing. The users will upgrade to Fedora N+1 and have `foo:1ts` enabled. Users will need to manually run `dnf module reset foo` if they want to switch to the non-modular version. This will produce the least risk, but it's not automatic and therefore less user-friendly to those who want to get away from default modular content.
2. Automatically reset all module streams as part of the upgrade process. This carries with it the risk that an intentionally-enabled stream will be switched to one that will be incompatible with apps the user has installed. This is likely mitigated by the fact that RPM-level dependencies will likely prevent the upgrade process. It could still be an issue for any software the user installed outside of RPM.

Upgrades with enabled streams - Stream is still available but no longer default

Possible Solutions:

1. Users manually run `dnf module disable/reset` on any module. This would work today, but it requires documentation and user training.
2. Automatically reset all modules whose enabled stream matches the default stream on Fedora N. This provides an upgrade experience roughly consistent with old behavior for users who have never explicitly enabled the modular streams. This runs the risk of

breaking non-RPM apps that are dependent on user selected streams that just happen to be the default.

3. Automatically reset all modules. This runs the risk of breaking non-RPM apps that are dependent on that stream. This has the same issues as 2). It might affect more modules, so it should not be considered unless we discover there are technical reasons we cannot detect the default streams.

Upgrades with enabled streams - Stream no longer exists

Upgrading from Fedora N with a module stream enabled as a default stream will attempt to continue with that stream into Fedora N+1. If that stream is unavailable on Fedora N+1, the upgrade operation will fail. At this time, the package manager cannot distinguish between packages installed because it was a default stream vs an explicitly-selected stream. For this document, I'll use the module stream `foo:eo1stream` as a representative of a module stream that exists in Fedora N and is not provided in Fedora N+1.

Possible Solutions

1. Users manually run `dnf module disable/reset` on any module. This would work today, but it requires documentation and user training.
2. Automatically reset all modules whose enabled stream matches the default stream on Fedora N. This provides an upgrade experience roughly consistent with old behavior for users who have never explicitly enabled the modular streams. This runs the risk of breaking non-RPM apps that are dependent on user selected streams that just happen to be the default.
3. Automatically reset all modules whose enabled stream no longer exists in Fedora N+1.
4. Automatically reset all modules. This runs the risk of breaking non-RPM apps that are dependent on that stream. This has the same issues as 2). It might affect more modules, so it should not be considered unless we discover there are technical reasons we cannot detect the default streams.
5. Carry an empty (no produced artifacts) version of the stream for two Fedora releases. This would allow us to upgrade people cleanly to the non-modular version of the packages. However, this is currently impossible due to [BZ #1770285](#). This also has issues as 2) and 3), but would require no special reset commands in the upgrade process.

Current Default streams (in Rawhide as of 2019-11-08)

dwm:6.2

Analysis:

- Exists only as module stream packages.

- Contains a single source package (dwm) and produces two binary packages (dwm and dwm-user).

Solution:

- Build dwm 6.2 in non-modular Rawhide.
- Remove default from fedora-module-defaults.
- [Handle upgrades from existing systems.](#)

avocado:69lts

Analysis:

- Exists only as module stream packages.
- Contains two source packages: python-aexpect and python-avocado.

Solution:

- Build python-avocado and python-aexpect in non-modular Rawhide.
- Remove default from fedora-module-defaults.
- [Handle upgrades from existing systems.](#)

scala:2.10

Analysis:

- Contains five source packages: hawtjni, jansi, jansi-native, jline and scala
- Most of these packages are buildroot-only and are suppressed from the final output artifacts. Excluded:
 - ant-scala
 - hawtjni
 - hawtjni-javadoc
 - jansi-javadoc
 - jansi-native-javadoc
 - jline-javadoc
 - maven-hawtjni-plugin
- Most of these packages are built with special configuration flags to produce minimalistic versions. The scala build will need to be reworked to use the system default copies of these packages.
- This module buildrequires itself.
- The ant-scala subpackage that is suppressed may need to be investigated for inclusion or disabling at the build level.
- The scala package exists in non-modular Rawhide, version 2.10.6, whether it is actively maintained there is unclear. It [builds successfully](#).

Solution:

- Remove default from fedora-module-defaults.
- [Handle upgrades from existing systems.](#)
- Coordinate with the Stewardship SIG and the modular maintainers to make sure this is properly maintained.

ant:1.10

Analysis:

- Contains many buildroot-only packages
- The ant package exists in non-modular Rawhide, version 1.10.6. It is very actively maintained by the Stewardship SIG.

Solution:

- Remove default from fedora-module-defaults.
- [Handle upgrades from existing systems.](#)
- Figure out if the Stewardship SIG is OK to maintain this further until such time as the design issues of default streams are addressed.

gimp:2.10

Analysis:

- Exists as a module stream package but is also present in non-modular Fedora (same version, same maintainers)
- Contains a single source package gimp and produces 4 binary packages (gimp, gimp-devel, gimp-devel-tools and gimp-libs)

Solution:

- Remove default from fedora-module-defaults.
- [Handle upgrades from existing systems.](#)

maven:3.5 (with an open ticket to make maven:3.6 the default)

Analysis:

- Comprised of **many** source packages, most of which are filtered out.
- The maven package exists in non-modular Rawhide, version 3.6.1. It is very actively maintained by the Stewardship SIG.

Solution:

- Remove default from fedora-module-defaults
- [Handle upgrades from existing systems.](#)
- Figure out if the Stewardship SIG is OK to maintain this further until such time as the design issues of default streams are addressed.

eclipse:???

Analysis:

- There is no default stream, but the non-modular packages are orphaned, some of them are retired, most of them FTBFS. A [request](#) was made to make eclipse:latest the default, but there were some conflicts with other default modules and they are either not yet resolved or the maintainers are yet to request this again.
- The eclipse package in non-modular Rawhide exists, is not orphaned, but fails to install and build.

Solutions:

1. Do nothing. Eclipse remains accessible only via non-default module streams.
2. Revert all of the work done to move it into a module. The maintainer is not interested in this option, so it would either have to be the Stewardship SIG (and/or other volunteers) or option 1).