

Listed and Unlisted Versions on AMO (PRD)

Version 2016

Product Manager: Jorge Villalobos

Key Dates

Date	Objective
2016-08-31	Spec stage begins
2016-09-26	PM team meeting
2016-09-30	PRD finalized, issues filed and assigned.

Overview

The introduction of signing has added unnecessary complexity and restrictions to the distribution of add-ons. While it used to require little effort for an add-on to have listed versions on [AMO](#) and unlisted versions distributed in other forms, the current system forces developers to choose one or the other, or use different unique IDs for both distributions (which leads to hacky solutions to avoid two distributions being installed at the same time).

Goals/Objectives

1. Enable listed and unlisted versions to be submitted for an add-on, so top developers like LastPass can continue having multiple distribution channels without hacky workarounds.
2. Simplify the submission process by making the listed and unlisted flows clearer and reducing the amount of steps required to submit an add-on.

Problem Statement

In order to have a listed channel on AMO and different unlisted versions distributed elsewhere, developers are forced to use different add-on IDs or hacks (beta versions on AMO). The current system limits developers to either go listed or unlisted, and switching from one to the other

requires deleting all previous submission history. Additionally, the visibility options (Listed / Unlisted / Hidden) are a common source of confusion.

Stakeholders

- Kev Needham, Add-ons Product Manager.
- Jorge Villalobos, AMO Product Manager.
- Stuart Colville, AMO Engineering Manager.
- Bram Pitoyo, UX.
- Krupa Raj, QA.

Product Features

1. When a developer uploads a new version of an add-on, they will have the option to choose if the version should be listed or unlisted.
2. Developers will only be required to enter the necessary information to create a listed or unlisted add-on.
3. If an add-on only has unlisted versions, all irrelevant listing management options will be hidden.

User Stories

Title	Create add-on
Description	1. As a developer, I want to upload my add-on to AMO so it can be used in Firefox. 2. As a developer, I want to do the least amount of data input to have my add-on submission accepted on AMO. 3. As a developer, I want to understand the different ways in which my add-on can be distributed and updated.

Title	Edit add-on
Description	1. As a developer, I want to edit my add-on metadata. 2. As a developer, I want to disable my add-on listing. 3. As a developer, I want to re-enable my add-on

	listing. 4. As a developer, I want to disable any versions in my add-on listing.
--	---

Title	Version channels
Description	1. As a developer, I want to be able to distribute my add-on on AMO and/or on my website under the same GUID. 2. As a developer, I want to be able to change my distribution channel as needed. 3. As a developer, I want to manage my listed and unlisted versions in the same place. 4. As a developer, I want to manage a beta update channel for my add-ons so power users can get updated and signed code early on for testing.

For all AMO user stories and features, see [overhaul spec and breakdowns](#).

Sketches and User Experience Specification

Current mockups

- [New unlisted add-on](#).
- [New listed add-on](#).
- [New unlisted version](#).
- [New listed version](#).
- [Status and Versions page showing listed and unlisted versions](#).
 - [Sidebar](#).
- [Unlisted add-on management](#).

MVP

- A developer should be able to submit and manage listed and unlisted versions for an add-on.
 - Listed / unlisted channel selection during new version submission.
<https://github.com/mozilla/addons/issues/192>
 - No need for a concept of “switching”. Just upload to the right channel.
 - New version submission: if an add-on only has unlisted versions and a listed version is submitted for the first time, show the “Edit listing” step from the first submission flow.

- Show channel for each version in Status & Versions page.
<https://github.com/mozilla/addons/issues/193>
 - Two columns vs. tags attached to each version (prefer tags)
- Replace “add-on visibility” (Listed / Unlisted / Hidden) selection.
<https://github.com/mozilla/addons/issues/194>
 - Hide / unhide listing option would only be needed if the add-on has listed versions.

Additional

- Clarify / simplify submission flow.
 - Don't list submission steps?
<https://github.com/mozilla/addons/issues/198>
 - For listed add-ons, only ask for required fields.
<https://github.com/mozilla/addons/issues/197>
 - Remove image selection (apply green brick icon by default).
 - Merge license selection into describe step. (easily merged because it's a bunch of radio buttons)
 - Upload step is overloaded. Channel selection, file selection, validation, platform selection, source code upload
<https://github.com/mozilla/addons/issues/199>
 - Channel selection needs to come first, because validation for listed/unlisted is different (“sign” vs. “publish”)
 - Select channel -> File upload -> Validate
 - We look at the manifest to determine Android/non-Android
 - Step 3: "The detail page will be:" -> bad wording for unlisted because there will be no public page.
 - Where should we put “select a slug”? Okay to have it before you download the add-on (last step)
 - When is your submission considered “complete” – ie. If you close your tab or return later to submission?
 - PS. For listed add-ons, they won't be handed over to a reviewer until “Edit Listing” step has been completed
 - Add Notes to Reviewer / Version Notes.
<https://github.com/mozilla/addons/issues/200>
 - Potentially add this to the “Edit Listing” step, below the “Description” step
 - When you submit a new version, after you pass validation, you get two boxes: notes to reviewer (reviewer-facing) and version notes (user-facing). Maybe label it differently: “sent to Mozilla” vs. “visible to users”
 - Is version notes helpful for initial version? Yes, even for initial version.
- Hide unnecessary management options when add-on is only unlisted:

- Hide "Manage Developer Profile", "Manage Payments", "View Statistics Dashboard", "Compatibility Reports"
<https://github.com/mozilla/addons/issues/195>
- On Edit Listing, hide: Experimental, Categories, Tags, Images, Support information, Technical details (except UUID and whiteboard)
<https://github.com/mozilla/addons/issues/196>
 - Why not hide the entire “Edit Listing” page? Because you can still edit the slug, name and summary.
 - Call it something other than the word “Listing”, because the add-on is already unlisted? Think of a good name for this. E.g. “Reviewer-facing information”, etc.

Risks/Issues

- Managing listed and unlisted versions isn’t a trivial task. Developers need to understand what the presence or absence of an updateURL may mean for updates, and avoid traps where users are stuck in the wrong channel.
 - To mitigate this, we need to make sure the copy around version channel selection is very clear about its meaning and impact, especially the first time the “other” channel is chosen.
- Having listed and unlisted versions for a single ID introduce complexity for tracking statistics and blocklisting add-ons.
 - One possible mitigation is to require or strongly recommend unlisted add-ons to include an updateURL, even if invalid, to make a clear distinction between listed and unlisted versions.

Dependencies

AMO supports a limited visibility channel with the beta versions feature. Beta versions are only available to add-ons that are listed and have passed review. It would make technical sense to re-implement beta versions as a channel. However, having an unlisted channel means the beta feature is not as needed as before.

Developers with beta versions were surveyed to assess the impact of dropping this feature in favor of an unlisted channel, and a decision should be made about dropping the feature or adapting it to this model. This is not a blocker, as these changes don’t break the beta version flow.

Milestones

Date	Objective
------	-----------

2016-10-03	Engineering work begins.
2016-10-31?	Progress check-in
2016-12-05 -> 09	Hawaii Work Week.

Next steps

-