Publishing an update

So you're ready to release a new version of your app. How do you go about doing that?

Archive your app

Put a copy of your .app (with the same name as the version it's replacing) in a .zip, .tar.gz, or .tar.bz2. If you distribute your .app in a .dmg, do *not* zip up the .dmg.

Make sure symlinks are preserved when you create the archive. macOS frameworks use symlinks, and their code signature will be broken if your archival tool follows symlinks instead of archiving them. For creating zip archives, `ditto` can be used (behaves the same as Finder's Compress option):

```
ditto -c -k --sequesterRsrc --keepParent <src_path_to_app> <zip_dest>
```

Alternatively, create an Installer .pkg with the same name as your app and put that .pkg in one of the aforementioned archive formats. By default Sparkle launches Installer with a GUI. If instead of .pkg extension you use .sparkle_guided.pkg, then installation will run in the background without asking the user to confirm each step.

Secure your update

In order to prevent corruption and man-in-the-middle attacks against your users, you must cryptographically sign your updates.

To cryptographically sign your updates, Sparkle includes a script to help you make a DSA signature of the archive. From the Sparkle distribution:

```
./bin/sign_update path_to_your_update.zip path_to_your_dsa_priv.pem
```

The output string is your update's DSA signature; you'll add this as an attribute to your enclosure in the next step. You can remove any newlines in this string.

Additionally, in macOS 10.11 Apple has added App Transport Security policy which blocks macOS apps from using insecure HTTP connections. This restriction applies to Sparkle as well, so you will need to serve your appcast and the update files over HTTPS.

Update your appcast

You need to create an `<item>` for your update in your appcast. See the sample appcast for an example. Here's a template you might use:

```
<item>
    <title>Version 2.0 (2 bugs fixed; 3 new features)</title>
    <sparkle:releaseNotesLink>
        https://example.com/release_notes/app_2.0.html
    </sparkle:releaseNotesLink>
    <pubDate>Mon, 05 Oct 2015 19:20:11 +0000</pubDate>
    <enclosure url="https://example.com/downloads/app.zip.or.dmg.or.tar.etc"
               sparkle:version="2.0"
               sparkle:dsaSignature="MC0CFBfeCa1JyW30nbkBwainOzrN6EQuAh="
               length="1623481"
               type="application/octet-stream" />
</item>
```

Test your update, and you're done!

Downloading from a web site

If you want to provide a download link, instead of having Sparkle download and install the update itself, you can omit the `<enclosure>` tag and add `<sparkle:version>` and `<link>` tags. For example:

```
<item>
  <title>Version 1.2.4</title>
  <sparkle:releaseNotesLink>https://example.com/release_notes_test.html</sparkle:releaseNotesLink>
  <pubDate>Mon, 28 Jan 2013 14:30:00 +0500</pubDate>
  <sparkle:version>1.2.4</sparkle:version>
  <link>https://example.com/manual_update_info.html</link>
</item>
```

Delta updates

If your app is large, or if you're updating primarily only a small part of it, you may find delta updates useful: they allow your users to only download the bits that have changed.

Internal build numbers

If you use internal build numbers for your `CFBundleVersion` key (like an SVN revision number) and a human-readable `CFBundleShortVersionString`, you can make Sparkle hide the internal version from your users.

Set the `sparkle:version` attribute on your enclosure to the internal, machine-readable version (ie: "1248"). Then set a `sparkle:shortVersionString` attribute on the enclosure to the human-readable version (ie: "12.X Sea Lion").

Remember that the internal version number (`CFBundleVersion` and `sparkle:version`) is intended to be machine-readable and is not generally suitable for formatted text.

Minimum system version requirements

If an update to your application raises the required version of macOS, you can only offer that update to qualified users.

Add a `sparkle:minimumSystemVersion` child to the `<item>` in question specifying the required system version, such as "10.8.4":

```
<item>
    <title>Version 2.0 (2 bugs fixed; 3 new features)</title>
    <sparkle:minimumSystemVersion>10.8.4</sparkle:minimumSystemVersion>
</item>
```

Note that Sparkle only works with macOS 10.7 or later, so that's the minimum version you can use.

Embedded release notes

Instead of linking external release notes using the `<sparkle:releaseNotesLink>` element, you can also embed the release notes directly in the appcast item, inside a `<description>` element. If you wrap it in `<![CDATA[ ... ]]>`, you can use unescaped HTML.

```
<item>
    <title>Version 2.0 (2 bugs fixed; 3 new features)</title>
    <description><![CDATA[
        <h2>New Features</h2>
        ...
    ]]>
    </description>
    ...
</item>
```

You can embed just marked up text (it'll be displayed using standard system font), or a full document with `<!DOCTYPE html>` `<style>`, etc.

Localization

You can provide additional release notes for localization purposes. For instance:

```
<sparkle:releaseNotesLink>https://example.com/app/2.0.html</sparkle:releaseNotesLink>
<sparkle:releaseNotesLink xml:lang="de">https://example.com/app/2.0_German.html</sparkle:releaseNotesLink>
```

Use the `xml:lang` attribute with the appropriate two-letter country code for each localization. You can also use this attribute with the `<description>` tag.

Alternate download locations for other operating systems

Sparkle is available for Windows.

To keep the appcast file compatible with the standard Sparkle implementation, a new tag has to be used for cross platform support. It is suggested to use the following to specify downloads for non macOS systems:

```
<sparkle:enclosure sparkle:os="os_name" ...>
```

Replace *os_name* with either “windows” or “linux”, respectively (mind the lower case!). Feel free to add other OS names as needed.

Xcode integration

There’s an old description of how to automate the archiving and signing process with a script build phase in Xcode. Find other guides.

© 2017 Sparkle Project. All Rights Reserved.

This website is open source.