



Learning to love release notes

Anne Edwards

Write the Docs 2018
anne@improbable.io



Who am I?

- technical writer for nearly four years
- three different companies - currently working at Improbable
- long-form docs, UI text, error messages... and, of course, release notes



What's this talk about?

- what release notes are, and my experience of them
- why I decided to write a style guide
- guidelines, templates, and techniques to make it easier to write useful release notes



What are release notes?

A personal perspective

Article [Talk](#) [Read](#) [Edit](#) [View history](#)

Release notes

From Wikipedia, the free encyclopedia



This article **does not cite any sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(December 2009)* ([Learn how and when to remove this template message](#))



What are release notes?

A personal perspective

- list of bullet points published alongside a software release
- bug fixes, new features, known issues
- not very glamorous, but necessary
- often an afterthought, frantically pulled together at the last minute



For example:

Fixes

- Fixed a bug where the app would quit when you clicked “Save”.

Features

- Added the option to customise the background.

Known issues

- In some cases, the app fails to load and displays an error.



SRP-1073

Perfectly normal
action no longer causes
everything to break



SRP-886

Finally added
a feature you requested
seven years ago



SRP-952

The behaviour is
no longer the opposite
of what you'd expect



When you strip away
the **technical details**,
you can spot the
patterns



Interviewer:

What **don't** you enjoy
about your current job?

Me, almost instantly:

Release notes.



- lots of context-switching
- lots of research
- not much space to work with
- generally not quite as fun as other stuff



Improbable:

- extremely technical subject matter
- developers writing the release notes
- not particularly user friendly
- “Fixed a bug where...”



#1: The one I don't even understand

Previous unresponded command responder will not be lost when a new command is received before it is sent.

Expanded out: “Previously... Now...”

Previously, a component command could fail when the component received a new command before sending the response to the previous command. **Now**, no unresponded commands will fail, no matter how many new commands are received before responding.

Simplified

Component commands no longer fail if the component receives a new command before it has sent the response to the previous command.



#2: The overly formal, distant one

It is now possible to directly use X , Y and Z as return types for commands. **It is not necessary any more to** wrap them in a user-defined type.

More direct

You can now directly use X , Y and Z as return types for commands. **You no longer need to** wrap them in a user-defined type.



#3: The stand-up update

Added the A and B fields to the C struct returned by D.

More user-focused

You can now see how many users are connected to a deployment (A), and the user capacity of the deployment (B), using D.



We've been giving a little feature you know and love some TLC behind the scenes, to tidy it up and make it easier to find. We'll let you know in-app once it's ready for the main stage.

Unread badges, which had been being a little inconsistent, have been given a stern talking to, and promise to be more reliable henceforth.



We've been giving a little feature you know and love some TLC behind the scenes, to tidy it up and make it easier to find. We'll let you know in-app once it's ready for the main stage.

Unread badges, which had been being a little inconsistent, have been given a stern talking to, and promise to be more reliable henceforth.

They're really wordy

- The main message is obscured
- More work for the reader



We've been giving a little feature you know and love some TLC behind the scenes, to tidy it up and make it easier to find. We'll let you know in-app once it's ready for the main stage.

Unread badges, which had been being a little inconsistent, have been given a stern talking to, and promise to be more reliable henceforth.

They use humour, idioms, and slang

- Likely to divide opinion
- Can be confusing for those less proficient at English
- Create extra barriers to understanding



We've been giving a little feature you know and love some TLC behind the scenes, to tidy it up and make it easier to find. We'll let you know in-app once it's ready for the main stage.

Unread badges, which had been being a little inconsistent, have been given a stern talking to, and promise to be more reliable henceforth.

They wouldn't reflect our voice

- Might work for certain companies, depending on personality
- Would be weird if we started writing like this!



Style guide: basic principles

Release notes should...

- be accurate, clear, and concise
(in that order)
- be user-focused
- give enough context



Style guide: basic principles

You **don't** need to...

- start bug fixes with “Fixed...”
- be formal and impersonal
- go into great detail



Style guide: writing tips and tricks

- Expand and simplify
As though you're solving an equation
- Imagine you're talking to someone
What do they care about? What do they need to know?
- Use a template
Less brain work for you *and* for your readers



Style guide: three useful templates

#1: You can now...

***You can now** de-register the same event callback of an obsolete Unity Reader or Writer more than once.*



Style guide: three useful templates

#2: X now/no longer does Y when Z.

*A GameObject with the same name as another asset type **no longer** clashes and causes problems **when** spawning new entities.*



Style guide: three useful templates

#3: X now/no longer does Y. This means you no longer/now need to do Z.

*The C++ SDK **now** automatically sends built-in metrics to SpatialOS. **This means you no longer need to** manually send these metrics when they are surfaced via `Dispatcher::OnMetrics`.*



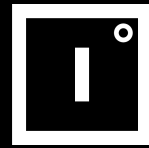
But remember...

These are guidelines, not rules.

If you've followed them and your release note sounds strange, *change it.*



A style guide can make
it easier for those **writing**
release notes, *and* for
those **reading** them



Thank you!

anne@improbable.io