

replace-ids



- [replaceIDs\(\)](#)
 - [Why is it needed? {#explanation}](#)
 - [Usage](#)
 - [Example](#)
 - [Conflicts](#)
-

```
title: replaceIDs() in Iconify Utils
functions:
  iconToSVG: './icon-to-svg.md'
```

replaceIDs()

This function is part of [Iconify Utils package](#).

Function `[func]replaceIDs()` is used for replacing IDs for icon content to avoid conflicts.

All icon components use this function for each rendered icon.

Why is it needed? {#explanation}

IDs are used in icon elements, such as reusable paths, masks, animations. They are supposed to be unique.

When icons are linked as external resources, this is not an issue. However, when icons are embedded in document, having multiple elements with the same ID from different icons can and will cause chaos. Even using the same icon in multiple places in document without changing IDs can cause errors.

This is why when embedding icons, it is important to never repeat the same IDs.

Usage

Function has the following parameter:

- `[prop]content`, `[type]string`. Icon content.

Function returns content with replaced IDs.

In first parameter you can pass `[prop]body` property of data generated by `[func]iconToSVG()` or you can pass an entire `[tag]svg`.

Example

```
src: libraries/utils/replace-ids.ts
title: 'example.ts'
extra:
  - src: libraries/utils/replace-ids.svg
    title: 'Result (first console.log):'
  - src: libraries/utils/replace-ids2.svg
    title: 'Result (second console.log):'
```

Conflicts

Replacement function will use original IDs to generate new IDs by adding numbers after original ID.

That means if icon has IDs `[str]a`, `[str]b` (for example, for clip paths) and so on, first icon will have IDs unchanged `[str]a`, `[str]b`, second icon will have IDs `[str]a1`, `[str]b1` and so on.

In Iconify icon sets, all icons use unique IDs based on hash of their name and prefix, for example, `[str]SVGqdEcMdXs`. This ID is used only in one icon.

Because original ID is used to generate new ID, the first time icon is rendered, it will have ID `[str]SVGqdEcMdXs` (no changes), second time `[str]SVGqdEcMdXs1` (`[str]1` appended), third time `[str]SVGqdEcMdXs2` (`[str]2` appended) and so on.

This results in unique, but predictable IDs for each rendered icon.

```
👤 #2
👤 👤 👤 2026-10-31 14:29:26 UTC
👤 👤 👤 2026-10-31 14:29:27 UTC
```