

# Package ‘dauphin’

May 8, 2026

**Title** Compact Standard for Australian Phone Numbers

**Version** 0.3.2

**Description** Phone numbers are often represented as strings because there is no obvious and suitable native representation for them. This leads to high memory use and a lack of standard representation. The package provides integer representation of Australian phone numbers with optional raw vector calling code. The package name is an extension of 'au' and 'ph'.

**License** GPL-2

**Encoding** UTF-8

**Suggests** tinytest

**RoxygenNote** 7.2.0

**Imports** utils

**NeedsCompilation** yes

**Author** Hugh Parsonage [aut, cre]

**Maintainer** Hugh Parsonage <hugh.parsonage@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-07-24 06:20:02 UTC

## Contents

dauphin_mobile . . . . .	1
<b>Index</b>	<b>4</b>

---

dauphin_mobile	<i>Standardize Australian phone numbers</i>
----------------	---

---

## Description

Standardize Australian phone numbers

**Usage**

```

dauphin_mobile(mob, ignore_calling_code = NA)

dauphin_landline(
  landline,
  default_area_code = getOption("dauphin.default_area_code", 1L)
)

dauphin_mobile_landline(
  mob,
  landline,
  default_area_code = getOption("dauphin.default_area_code", 1L)
)

## S3 method for class 'dauphin_mobile'
print(x, ...)

```

**Arguments**

`mob, landline` Character vectors in which mobile numbers and landline numbers are expected.

`ignore_calling_code` `logical(1)` Whether to ignore the calling code +61 in the result.

`default_area_code` An integer between 1 and 10 giving, for landline numbers with 8 digits, what area code should be set. By default, it is 1L, but users should set the option "dauphin.default\_area\_code" so to correctly standardize non-mobile numbers.

`x` A vector, generally a character vector, in which phone numbers are expected.

`...` Arguments passed to other methods.

**Value**

Mobile phone numbers or landline numbers are represented as integer vectors. International calling prefixes extend the number beyond the representation of signed integers. We use raw vectors for the international prefix, if required.

If `ignore_calling_code = TRUE`, the integer vector is returned. Elements of `x` for which the mobile phone number could not be extracted map to `NA_integer_` in the result.

If `ignore_calling_code = FALSE`, then a list is returned. The second element of the list is the calling prefix.

If `ignore_calling_code = NA` then it is set to `TRUE` if `x` appears to have international prefixes already.

`dauphin_mobile` An integer vector, the integer representation of the mobile phone. If the calling code is required or requested, a list of two vectors is returned, with the second element a raw vector with a representation of the corresponding number.

`dauphin_landline` An integer vector, the integer representation of the landline.

`dauphin_mobile_landline` A list of two vectors with the mobile and landline vectors respectively, even if the character vectors passed are in the wrong order (or partially in the wrong order). Useful if some of the entries are in the wrong place.

**Examples**

```
dauphin_mobile("0400 123 456")
dauphin_mobile("+61400123456", ignore_calling_code = FALSE)

dauphin_mobile_landline("0424 123 456", "03 1234 5678")
dauphin_mobile_landline(c("0424 123 456", "03 1234 5678"),
                        c(NA, "0424 123 456"))
```

# Index

dauphin\_landline (dauphin\_mobile), 1  
dauphin\_mobile, 1  
dauphin\_mobile\_landline  
    (dauphin\_mobile), 1  
  
print.dauphin\_mobile (dauphin\_mobile), 1