

# Package ‘pds3’

June 6, 2018

**Version** 0.5.0

**Title** Manipulate PDS3 ODL Files

**Description** Reads Planetary Data Systems v3 (PDS3 - <<https://pds.jpl.nasa.gov/>>) Object Description Language (ODL) formatted files, typically used in NASA missions for storing metadata about observations.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**ByteCompile** true

**Depends** R (>= 3.5.0)

**Suggests** testthat, covr, lintr, knitr, curl, rmarkdown

**Imports** rly

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Micah Waldstein [aut, cre]

**Maintainer** Micah Waldstein <[micah@waldste.in](mailto:micah@waldste.in)>

**Repository** CRAN

**Date/Publication** 2018-06-06 09:02:55 UTC

## R topics documented:

pds3_lexer_parser . . . . .	2
pds3_read . . . . .	2

<b>Index</b>	<b>4</b>
--------------	----------

---

pds3\_lexer\_parser      *Access PDS3 Lexer & Parser*

---

### Description

Provides access to the underlying lexer & parser to process PDS3 files

### Usage

```
pds3_lexer_parser(debug = FALSE)
```

### Arguments

debug                    (default: FALSE) Sets lexer/parser to debug mode for testing.

### Details

Typically you don't need to access this directly, but if you are processing a large number of files you may want to cache this and pass it in to 'read\_pds3' manually to prevent them from being re-created on each run.

### Value

A list with the following components:

**lexer** The rly lexer

**parser** The rly parser

### Examples

```
util <- pds3_lexer_parser()
```

---

pds3\_read                    *Parse PDS3*

---

### Description

Parses a PDS3 file to extract meta information

### Usage

```
pds3_read(x, util = pds3_lexer_parser(), assume_complete = TRUE)
```

**Arguments**

- x** A String of the PDS3 to parse, containing one of the following:  
**Content** String you want to parse  
**Filename** Path to a file containing a PDS3
- util** A list containing a PDS3 lexer & parser as returned by 'getLexerParser()'. Typically not required, but improves performance if parsing many files.
- assume\_complete**  
(default: TRUE) Assume that there is no content after 'END'. If set to false, will pre-parse the file to find END. If you start getting parsing errors, you may need to set this flag to false.

**Value**

A list with the following components:

**label** Raw label content. If there is extra\_data, it will be excluded

**extra\_data** Content following 'END'. If assume\_complete is F or there is no content after the label, this will be an empty string

**odl** Parsed label content.

**Examples**

```
pds3_read('PDS_VERSION_ID = PDS3
PRODUCT_CREATION_TIME      = 2017-05-31T18:42:49
END')
```

# Index

pds3\_lexer\_parser, [2](#)  
pds3\_read, [2](#)