

[Discover Packages](#) > [Standard library](#) > [log](#) > [syslog](#) 

syslog

package

standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: 8 |

Imported by: 6,196

Details

[Valid go.mod file](#)  [Redistributable license](#)  [Tagged version](#) [Stable version](#) [Learn more](#)

Repository

[cs.opensource.google/go/go](#)

Links

[Report a Vulnerability](#) Documentation 

<> Documentation

Rendered for [linux/amd64](#) 

Overview

Package syslog provides a simple interface to the system log service. It can send messages to the syslog daemon using UNIX domain sockets, UDP or TCP.

Only one call to Dial is necessary. On write failures, the syslog client will attempt to reconnect to the server and write again.

The syslog package is frozen and is not accepting new features. Some external packages provide more functionality. See:

<https://godoc.org/?q=syslog>

Index

[func NewLogger\(p Priority, logFlag int\) \(*log.Logger, error\)](#)[type Priority](#)[type Writer](#)[func Dial\(network, raddr string, priority Priority, tag string\) \(*Writer, error\)](#)[func New\(priority Priority, tag string\) \(*Writer, error\)](#)[func \(w *Writer\) Alert\(m string\) error](#)[func \(w *Writer\) Close\(\) error](#)[func \(w *Writer\) Crit\(m string\) error](#)[func \(w *Writer\) Debug\(m string\) error](#)[func \(w *Writer\) Emerg\(m string\) error](#)

```
func (w *Writer) Err(m string) error
func (w *Writer) Info(m string) error
func (w *Writer) Notice(m string) error
func (w *Writer) Warning(m string) error
func (w *Writer) Write(b []byte) (int, error)
```

Bugs

Examples

Dial

Constants

This section is empty.

Variables

This section is empty.

Functions

func NewLogger

```
func NewLogger(p Priority, logFlag int) (*log.Logger, error)
```

NewLogger creates a log.Logger whose output is written to the system log service with the specified priority, a combination of the syslog facility and severity. The logFlag argument is the flag set passed through to log.New to create the Logger.

Types

type Priority

```
type Priority int
```

The Priority is a combination of the syslog facility and severity. For example, LOG_ALERT | LOG_FTP sends an alert severity message from the FTP facility. The default severity is LOG_EMERG; the default facility is LOG_KERN.

```
const (

    // From /usr/include/sys/syslog.h.
    // These are the same on Linux, BSD, and OS X.
    LOG_EMERG Priority = iota
    LOG_ALERT
    LOG_CRIT
    LOG_ERR
    LOG_WARNING
    LOG_NOTICE
```

```

LOG_INFO
LOG_DEBUG
)

const (

    // From /usr/include/sys/syslog.h.
    // These are the same up to LOG_FTP on Linux, BSD, and OS X.
    LOG_KERN Priority = iota << 3
    LOG_USER
    LOG_MAIL
    LOG_DAEMON
    LOG_AUTH
    LOG_SYSLOG
    LOG_LPR
    LOG_NEWS
    LOG_UUCP
    LOG_CRON
    LOG_AUTHPRIV
    LOG_FTP

    LOG_LOCAL0
    LOG_LOCAL1
    LOG_LOCAL2
    LOG_LOCAL3
    LOG_LOCAL4
    LOG_LOCAL5
    LOG_LOCAL6
    LOG_LOCAL7
)

```

type **Writer**

```

type Writer struct {
    // contains filtered or unexported fields
}

```

A Writer is a connection to a syslog server.

func **Dial**

```

func Dial(network, raddr string, priority Priority, tag string) (*Writer, error)

```

Dial establishes a connection to a log daemon by connecting to address raddr on the specified network. Each write to the returned writer sends a log message with the facility and severity (from priority) and tag. If tag is empty, the os.Args[0] is used. If network is empty, Dial will connect to the local syslog server. Otherwise, see the documentation for net.Dial for valid values of network and raddr.

► **Example**

func New

```
func New(priority Priority, tag string) (*Writer, error)
```

New establishes a new connection to the system log daemon. Each write to the returned writer sends a log message with the given priority (a combination of the syslog facility and severity) and prefix tag. If tag is empty, the os.Args[0] is used.

func (*Writer) Alert

```
func (w *Writer) Alert(m string) error
```

Alert logs a message with severity LOG_ALERT, ignoring the severity passed to New.

func (*Writer) Close

```
func (w *Writer) Close() error
```

Close closes a connection to the syslog daemon.

func (*Writer) Crit

```
func (w *Writer) Crit(m string) error
```

Crit logs a message with severity LOG_CRIT, ignoring the severity passed to New.

func (*Writer) Debug

```
func (w *Writer) Debug(m string) error
```

Debug logs a message with severity LOG_DEBUG, ignoring the severity passed to New.

func (*Writer) Emerg

```
func (w *Writer) Emerg(m string) error
```

Emerg logs a message with severity LOG_EMERG, ignoring the severity passed to New.

func (*Writer) Err

```
func (w *Writer) Err(m string) error
```

Err logs a message with severity LOG_ERR, ignoring the severity passed to New.

func (*Writer) Info

```
func (w *Writer) Info(m string) error
```

Info logs a message with severity LOG_INFO, ignoring the severity passed to New.

func (*Writer) Notice

```
func (w *Writer) Notice(m string) error
```

Notice logs a message with severity LOG_NOTICE, ignoring the severity passed to New.

func (*Writer) Warning

```
func (w *Writer) Warning(m string) error
```

Warning logs a message with severity LOG_WARNING, ignoring the severity passed to New.

func (*Writer) Write

```
func (w *Writer) Write(b []byte) (int, error)
```

Write sends a log message to the syslog daemon.

Notes

Bugs

- This package is not implemented on Windows. As the syslog package is frozen, Windows users are encouraged to use a package outside of the standard library. For background, see <https://golang.org/issue/1108>.
- This package is not implemented on Plan 9.



Source Files

[View all](#)

[doc.go](#)

[syslog.go](#)

[syslog_unix.go](#)

Why Go

[Use Cases](#)

[Case Studies](#)

Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

Packages

[Standard Library](#)

[About Go Packages](#)

About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Golang Weekly](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google