



[Growl](#) » [Documentation](#) » [Developer](#) » Network Protocol Format

## Network Protocol Format



### About

Growl currently supports two different network protocols:

- Distributed Objects over TCP (port 23052)
- GrowlTalk over UDP (default port: 9887)

This document describes the UDP protocol.

### Requirements

Networking support first appeared in Growl 0.7.

### Usage

Growl requires you to register the notifications your application sends (and set whether or not they're enabled on the GUI) before being able to actually send something to your Mac, so make sure you have "Allow application registration" enabled on Growl's preference pane. And, of course, make sure you set a password.

If remote application registration is not allowed, the Growl server will reject all incoming registration packets. Each individual notification should be sent as a single notification packet.

### Bonjour

Growl servers publish the Distributed Objects service via Bonjour as `_growl._tcp`. The UDP protocol is not advertised because its port number is configurable by the user.

### Protocol definition

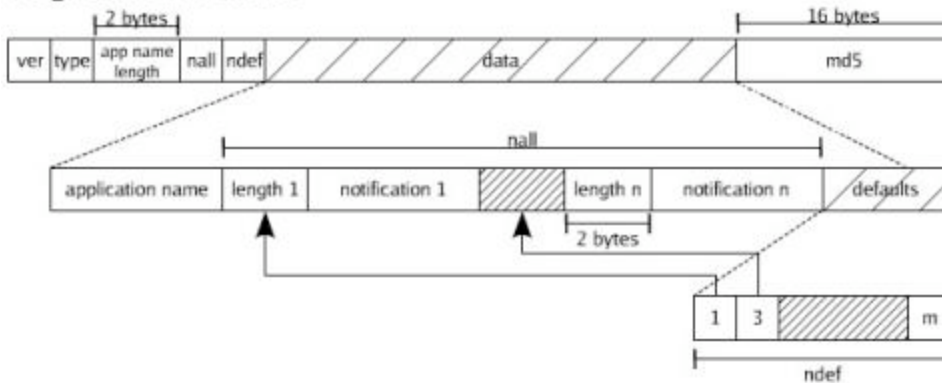
The protocol format is fairly straightforward, and the registration and notification packets look like this:

[Download](#)[About](#)[Applications](#)[Documentation](#)[Screenshots](#)[Forums](#)[Contact](#)[GrowlTrac](#)[Donate](#)

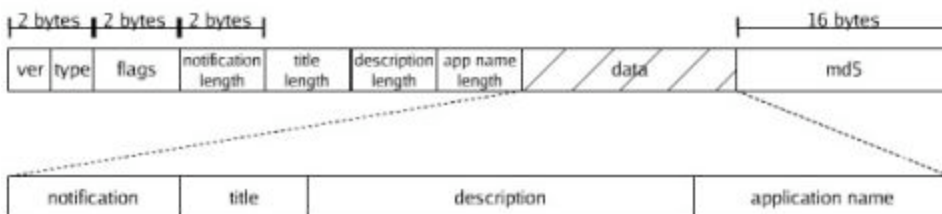
# Macworld



## Registration Packet



## Notification Packet



The following fields are common to all packet types:

### ver

The protocol version number. Currently GROWL\_PROTOCOL\_VERSION (1) or GROWL\_PROTOCOL\_VERSION\_AES128 (2). The value GROWL\_PROTOCOL\_VERSION\_AES128 indicates that the rest of the packet is encrypted using a 128-bit AES key. The AES key is derived from the password (PKCS12-style) and is used in Cipher Block Chaining mode with a fixed initialization vector and PKCS7 padding.

### type

This field identifies the type of the packet. Valid values are:

GROWL\_TYPE\_REGISTRATION (0)

The packet type of registration packets with MD5 authentication.

GROWL\_TYPE\_NOTIFICATION (1)

The packet type of notification packets with MD5 authentication.

GROWL\_TYPE\_REGISTRATION\_SHA256 (2)

The packet type of registration packets with SHA-256 authentication.

GROWL\_TYPE\_NOTIFICATION\_SHA256 (3)

The packet type of notification packets with SHA-256 authentication.

GROWL\_TYPE\_REGISTRATION\_NOAUTH (4)

The packet type of registration packets without authentication.

GROWL\_TYPE\_NOTIFICATION\_NOAUTH (5)

The packet type of notification packets without authentication.

### checksum

The size of this field depends on the packet type. It is either 16

bytes (MD5), 32 bytes (SHA-256) or 0 bytes (NOAUTH) wide. The checksum is used to verify the integrity of the packet as well as for authentication purposes. The checksum is computed over all fields of the packet preceding the checksum field and the UTF-8 encoded password.

The following fields occur only in registration packets:

app name length

The length of the UTF-8 encoded application name (in bytes).

nall

The number of notifications in the list.

ndef

The number of notifications in the list that are enabled by default.

application name

The UTF-8 encoded application name.

defaults

This field is ndef bytes long. Each byte contains an index into the notifications array.

The following fields occur only in notification packets:

flags

The flags field contains a signed 3-bit value (-2 to 2) and a sticky flag in the lowest (rightmost) nibble.

notification length

The length of the UTF-8 encoded notification name (in bytes).

title length

The length of the UTF-8 encoded notification tile (in bytes).

description length

The length of the UTF-8 encoded notification description (in bytes).

app name length

The length of the UTF-8 encoded application name (in bytes).

notification

The UTF-8 encoded notification name.

title

The UTF-8 encoded notification title.

description

The UTF-8 encoded notification description.

application name

The UTF-8 encoded application name.

## Examples

The reference client application which uses all features of the network protocol is [growlnotify](#). If your favorite programming language is not Objective-C, chances are that Rui Carmo has already written a GrowlTalk implementation for it:

- [netgrowl.py](#)
- [netgrowl.php](#)