



**FREE eBook**

**LEARNING**

**dns**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#dns**

# Table of Contents

About.....	1
Chapter 1: Getting started with dns.....	2
Remarks.....	2
Examples.....	2
Performing name lookups in C.....	2
Credits.....	4

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [dns](#)

It is an unofficial and free dns ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official dns.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with dns

## Remarks

The Domain Name System (DNS) is a *hierarchical, distributed* global database, most commonly used for mapping hostnames to their respective IP addresses. It is defined in [RFC 1034](#) and [RFC 1035](#), and numerous updates thereof.

A domain name is a sequence of *labels* separated by the period character (.). Each label can have a maximum of 63 characters, and a domain name can have a maximum of 255 characters.

The DNS is often described as a tree structure, with the "root zone" at the top, the Top Level Domains (TLDs) e.g. `com`, `uk`, etc below that, etc. Within a domain name the labels are shown in a "little endian" order with the leaf node label appearing left-most, and the TLD appearing right-most. If a trailing period appears after the TLD then the name is said to be a *Fully Qualified Domain Name*, where the trailing period represents the root zone itself.

## Examples

### Performing name lookups in C

The `getaddrinfo()` function is the recommended POSIX function for interfacing with the system resolver. Depending on system configuration it will perform name lookups in the DNS, `/etc/hosts`, mDNS, etc.

It is preferred over the deprecated `gethostbyname()` family of functions because it supports both IPv4 and IPv6 addressing, and can also perform service name lookups at the same time (e.g. mapping `http` to port 80)

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>

...

struct addrinfo hints;
struct addrinfo *result;
int r;

memset(&hints, 0, sizeof(hints));
hints.ai_family = AF_UNSPEC;           // allow IPv4 or IPv6
hints.ai_socktype = SOCK_STREAM;     // make a stream (TCP) connection

r = getaddrinfo(hostname, "http", &hints, &result);
if (r != 0) {
    fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(r));
} else {
    // iterate over the linked list
    for (struct addrinfo *rp = result; rp != NULL; rp = rp->ai_next) {
        // use rp fields to create a socket and connect to it
    }
}
```

```
    }  
    freeaddrinfo(result);  
}
```

Read **Getting started with dns** online: <https://riptutorial.com/dns/topic/5356/getting-started-with-dns>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with dns	<a href="#">Alnitak</a> , <a href="#">Community</a>