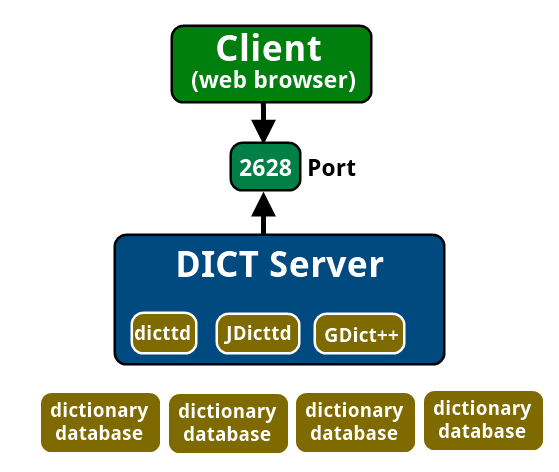
## DICT Dictionary Format

### Giao thức DICT (DICT Protocol)

DICT là một giao thức từ điển mạng (dictionary network protocol) được phát triển bởi DICT Development Group. Mục đích chính của giao thức này là tìm một giải pháp mới để thay thế cho giao thức Webster, cho phép người sử dụng có thể truy xuất nhiều từ điển trong một lần sử dụng. Dict server và client sử dụng port 2628 để giao tiếp với nhau.



Kiến trúc của DICT Protocol

Toàn bộ mô tả giao thức (specification) có thể đọc tại <http://tools.ietf.org/html/rfc2229>.

Mục đich chính của DICT là tạo ra một hệ thống từ điển trực tuyến, sử dụng thông qua Internet, trong đó lượng database miễn phí được chia sẻ từ khắp nơi.

Có rất nhiều cài đặt DICT server bằng nhiều ngôn ngữ khác nhau như:

* *Dicttd* (ứng dụng server chuẩn được phát triển bởi DICT Development Group).
* *GNU Dico*: một server được phát triển bởi GNU, hỗ trợ các định dạng từ điển khác ngoài DICT format. Mỗi một format được định nghĩa là một loadable module và có thể viết bằng C, Python hay Guile.
* *JDicttd*: được phát triển bởi Ho Ngoc Duc, viết hoàn toàn bằng Java. Hỗ trợ định dạng nén và dạng file chuẩn của DICT.
* *DictD++*: một tân binh server được viết bằng C++ (sử dụng STL, Boost và một vài thư viện khác), chạy trên FreeBSD, Linux Mandrake, Windows 2K/XP.

### Định dạng từ điển của DICT Protocol

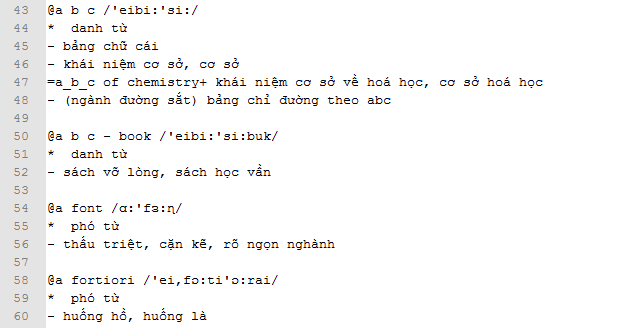
Cơ sở dữ liệu của DICT Protocol được phân phố dưới dạng riêng rẽ. Mỗi một cơ sở dữ liệu bao gồm hai file định dạng plain text: một được gọi là ***flat text*** (\*.dict) và ***index file*** (\*.index).

Cả 2 file đều là file thuần text và được thiết kể để có thể đọc được bằng bất cứ trình text editor nào (human-readable).

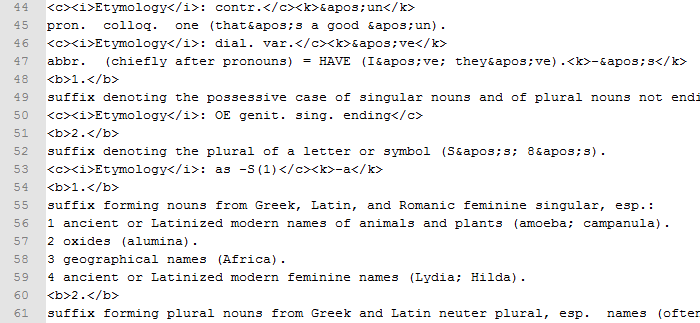
**Flat text (Dictionary file)**

* Đây là tập tin chứa nội dung chính định nghĩa của các từ mà từ điển hỗ trợ. Nội dung có thể là thuần text hay chứa HTML tag.
* Nội dung của file được tạo ra bởi một công cụ được cung cấp sẵn bởi DICT Development Group được gọi là dictfmt. Cấu trúc của flat file được xây dựng sao cho client có thể truy xuất ngẫu nhiên (random access) nhằm giảm thời gian load file và tối ưu khả năng truy xuất đa luồng từ server.

Một số ví dụ:



Nội dung thuần text của flat file



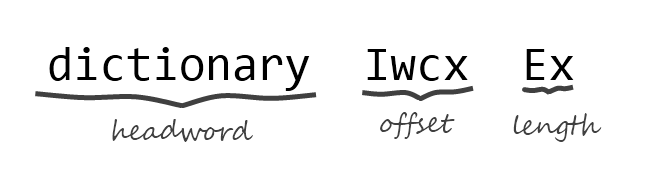
Nội dung file flat text chứa các tag đặc biệt

**Index File**

* Index file là file chứa tất cả các từ khóa mà từ điển hỗ trợ. DICT Server sẽ tìm kiếm từ khóa trên file index, nếu có, nội dung được mã hóa trong file index sẽ được chuyển đổi về đúng định dạng và tiến hành lấy dữ liệu từ flat text.
* Một file index bao gồm các dòng, mỗi dòng là một bộ định nghĩa (record) của từ điển được phân cách bởi ký tự TAB:

1. Headword: từ khóa.
2. Offset: vị trí bắt đầu của dữ liệu trong flat text.
3. Length: độ dài của dữ liệu bắt đầu từ offset.

Ví dụ:



Nội dung một record trong index file

* Nội dung của Offset và Length là số và được mã hóa bởi giải thuật Base64 Encoding. Mục đích nhắm làm giảm số lượng ký tự sử dụng trong việc thể hiện số.

**Mã hóa Base64 trong DICT Protocol format.**

Mã hóa Base64 là một dạng mã hóa sử dụng 64 ký tự ASCII để chuyển binary data (số, byte,..) sang dạng có thể biểu diễn trên text editor (sử dụng ASCII),

Dưới đây là bảng các ký tự ASCII được sử dụng trong Base64 encoding và giá trị của nó so với binary data.



Bảng các ký tự được sử dụng trong mã hóa Base64

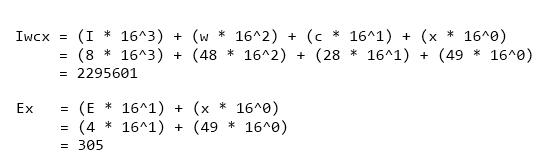
Trong DICT Protocol, các giá trị offset và length của index record được mã hóa bởi Base64 nhằm tiết kiệm số lượng ký tự dùng để thể hiện số. Để có thể chuyện ngược dữ liệu đã được mã hóa, tao chỉ cần thực hiện giải thuật tương tự như chuyển Base10 sang Base2.

Ví dụ:

Nếu record có nội dung

dictionary Iwcx Ex

thì:



Có nghĩa là từ nội dung của từ dictionary sẽ bắt đầu từ vị trí 2295601 và có độ dài (tính theo byte) 305.

### Sử dụng trực tiếp DICT Protocol database

Do tính chất của DICT Protocol chủ yếu phục vụ cho ứng dụng web client, cho nên sử dụng DICT Protocol trong đề án thực sự không hiệu quả. Nhóm thực hiện đề án quyết định tự implement cách đọc riêng cho megadict.

Nghĩa là vẫn sử dụng định dạng từ điển của DICT Protocol, nhưng phương pháp đọc và xử lý tìm kiếm và đọc dữ liệu được implement mới hoàn toàn và không giống với phương pháp được sử dụng ở DICT Protocol.