NAME

resolver, res_query, res_search, res_mkquery, res_send, res_init, dn_comp, dn_expand − resolver routines

SYNOPSIS

#include <sys/types.h>
#include <net/gen/in.h>
#include <net/gen/nameser.h>
#include <net/gen/resolv.h>

res_query(dname, class, type, answer, anslen)
char *dname;
int class, type;
u_char *answer;
int anslen;

res_search(dname, class, type, answer, anslen)
char *dname;
int class, type;
u_char *answer;
int anslen;

res_mkquery(op, dname, class, type, data, datalen, newrr, buf, buflen)
int op;
char *dname;
int class, type;
char *data;
int datalen;
struct rrec *newrr;
char *buf;
int buflen;

res_send(msg, msglen, answer, anslen)
char *msg;
int msglen;
char *answer;
int anslen;

res_init()

dn_comp(exp_dn, comp_dn, length, dnptrs, lastdnptr)
char *exp_dn, *comp_dn;
int length;
char **dnptrs, **lastdnptr;

dn_expand(msg, eomorig, comp_dn, exp_dn, length)
char *msg, *eomorig, *comp_dn, exp_dn;
int length;

DESCRIPTION

These routines are used for making, sending and interpreting query and reply messages with Internet domain name servers.

Global configuration and state information that is used by the resolver routines is kept in the structure *_res*. Most of the values have reasonable defaults and can be ignored. Options stored in *_res.options* are defined in *resolv.h* and are as follows. Options are stored as a simple bit mask containing the bitwise ‘‘or’’ of the options enabled.

RES_INIT
: True if the initial name server address and default domain name are initialized (i.e., *res_init* has been called).

RES_DEBUG
: Print debugging messages.

RES_AAONLY
: Accept authoritative answers only. With this option, *res_send* should continue until it finds an authoritative answer or finds an error. Currently this is not implemented.

RES_USEVC
: Use TCP connections for queries instead of UDP datagrams.

RES_STAYOPEN
: Used with RES_USEVC to keep the TCP connection open between queries. This is useful only in programs that regularly do many queries. UDP should be the normal mode used.

RES_IGNTC
: Unused currently (ignore truncation errors, i.e., don't retry with TCP).

RES_RECURSE
: Set the recursion-desired bit in queries. This is the default. (*res_send* does not do iterative queries and expects the name server to handle recursion.)

RES_DEFNAMES
: If set, *res_search* will append the default domain name to single-component names (those that do not contain a dot). This option is enabled by default.

RES_DNSRCH
: If this option is set, *res_search* will search for host names in the current domain and in parent domains; see *hostname*(7). This is used by the standard host lookup routine *gethostbyname*(3). This option is enabled by default.

The *res_init* routine reads the configuration file (if any; see *resolver*(5)) to get the default domain name, search list and the Internet address of the local name server(s). If no server is configured, the host running the resolver is tried. The current domain name is defined by the hostname if not specified in the configuration file; it can be overridden by the environment variable LOCALDOMAIN. Initialization normally occurs on the first call to one of the following routines.

The *res_query* function provides an interface to the server query mechanism. It constructs a query, sends it to the local server, awaits a response, and makes preliminary checks on the reply. The query requests information of the specified *type* and *class* for the specified fully-qualified domain name *dname* . The reply message is left in the *answer* buffer with length *anslen* supplied by the caller.

The *res_search* routine makes a query and awaits a response like *res_query*, but in addition, it implements the default and search rules controlled by the RES_DEFNAMES and RES_DNSRCH options. It returns the first successful reply.

The remaining routines are lower-level routines used by *res_query*. The *res_mkquery* function constructs a standard query message and places it in *buf*. It returns the size of the query, or −1 if the query is larger than *buflen*. The query type *op* is usually QUERY, but can be any of the query types defined in <*arpa/nameser.h*> . The domain name for the query is given by *dname*. *Newrr* is currently unused but is intended for making update messages.

The *res_send* routine sends a pre-formatted query and returns an answer. It will call *res_init* if RES_INIT is not set, send the query to the local name server, and handle timeouts and retries. The length of the reply message is returned, or −1 if there were errors.

The *dn_comp* function compresses the domain name *exp_dn* and stores it in *comp_dn*. The size of the compressed name is returned or −1 if there were errors. The size of the array pointed to by *comp_dn* is given by *length*. The compression uses an array of pointers *dnptrs* to previously-compressed names in the current message. The first pointer points to to the beginning of the message and the list ends with NULL. The limit to the array is specified by *lastdnptr*. A side effect of *dn_comp* is to update the list of pointers for labels inserted into the message as the name is compressed. If *dnptr* is NULL, names are not compressed. If *lastdnptr* is NULL, the list of labels is not updated.

The *dn_expand* entry expands the compressed domain name *comp_dn* to a full domain name The compressed name is contained in a query or reply message; *msg* is a pointer to the beginning of the message. The uncompressed name is placed in the buffer indicated by *exp_dn* which is of size *length*. The size of compressed name is returned or −1 if there was an error.

FILES

/etc/resolv.conf see resolver(5)

SEE ALSO

gethostbyname(3), named(8), resolver(5), hostname(7),
RFC1032, RFC1033, RFC1034, RFC1035, RFC974,
SMM:11 Name Server Operations Guide for BIND