

HI.CommunicationEngine
DNSクライアント
リファレンスマニュアル

ご注意

1. 本製品(ソフトウェア製品及びその関連ソフトウェア製品を含む。以下、同じ。)の使用に際しては、「外国為替及び外国貿易法」等、技術輸出に関する日本及び関連諸国の関係法規の遵守が必要となります。
2. 弊社は、本製品の使用に際しては、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に関し、別途、個別の契約書等(マニュアルの記載を含む。以下、同じ。)にて弊社による明示的な許諾がある場合を除き、その保証または実施権の許諾を行うものではありません。また本製品を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 本製品およびその仕様、またはマニュアルに記載されている事柄については、将来、事前の予告なしに変更することがありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書(マニュアルを含む)をご確認ください。
4. 本製品の使用(マニュアル記載事項に基づくものも含む)により直接または間接に生ずるいかなる損害についても、弊社は一切の責任を負いません。また、本製品の配布に使用される搭載機器や媒体が原因の損害に対しましても、弊社は一切の責任を負いません。
5. 本製品を、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。お客様の用途がこれに該当するかどうか疑問のある場合には、事前に弊社営業担当迄ご相談をお願い致します。
6. 本製品を使用してお客様のシステム製品を設計される際には、通常予測される故障発生率、故障モードをご考慮の上、本製品の動作が原因での事故、その他の拡大損害を生じないようにフェールセーフ等の十分なシステム上の対策を講じて頂きますようお願い致します。
7. 本製品およびマニュアルの著作権は弊社が所有しております。お客様は、弊社から提供された本製品を、別途、個別の契約書等にて定める場合を除き、いかなる場合においても全体的または部分的に複写・解析・改変することはできないものとします。
8. お客様は、別途、個別の契約書等にて定める場合を除き、本製品のマニュアルの一部または全部を無断で使用、複製することはできません。
9. 弊社は、本製品を1台のコンピュータで使用する権利をお客様に対してのみ許諾します。よって、本製品を第三者へ譲渡、貸与、賃借することは許諾しないものとします。但し、別途、個別の契約書等にて定められる場合はその条件に従います。
10. 本製品をはじめ弊社製品およびその関連製品についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

μ ITRON は、Micro Industrial TRON の略称です。TRON は、The Realtime Operating system Nucleus の略称です。

その他、本書で登場するシステム名、製品名は各社の登録商標または商標です。

はじめに

このマニュアルは、HI.CommunicationEngineのDNSクライアントについて説明します。

DNSクライアントは、TCP/IPマネージャと組み合わせることによってHI.CommunicationEngine上でDNSをサポートします。

HI.CommunicationEngineのDNSクライアント機能は、オンライン動作中に必要なときだけ利用することができます。

このリファレンスマニュアルではDNSクライアントのサービスコールとその使い方および関連事項を説明します。TCP/IPマネージャについては関連マニュアルを参照してください。

注) DNSクライアントを使用するためには、**基本セット (HI.CommunicationEngine-TCP V2.00以上)**が必要です。

【関連マニュアル】

- HI.CommunicationEngineTCP/IPマネージャリファレンスマニュアル
- 使用する μ ITRON のユーザーズマニュアル

目次

| | | |
|--------|---|----|
| 1. | 概要..... | 1 |
| 1.1 | 機能概要 | 1 |
| 1.2 | 関連するマネージャ | 1 |
| 1.3 | 関連するライブラリ | 1 |
| 1.4 | 構成 | 2 |
| 2. | DNSサービスコール | 3 |
| 2.1.1 | <i>DNS_init</i> DNSクライアント状態変数の初期化 | 5 |
| 2.1.2 | <i>DNS_start</i> ホストデータベースの初期化とDNSクライアントの開始 | 6 |
| 2.1.3 | <i>DNS_stop</i> DNSクライアントの停止 | 8 |
| 2.1.4 | <i>setdnssvpinfo</i> DNSサービスポイント情報の設定 | 9 |
| 2.1.5 | <i>deldnssvpinfo</i> DNSサービスポイント情報の削除 | 10 |
| 2.1.6 | <i>gethostbyname</i> 名前に対応するホストエントリの取得 | 11 |
| 2.1.7 | <i>gethostbyaddr</i> アドレスに対応するホストエントリの取得 | 13 |
| 2.1.8 | <i>addhostent</i> ホストエントリの登録 | 15 |
| 2.1.9 | <i>delhostent</i> ホストエントリの削除 | 17 |
| 2.1.10 | <i>requestdnupdate</i> DNS更新要求 | 18 |

図表目次

| | |
|-----------------------------|---|
| 図 1-1 DNSクライアント使用時の構成例..... | 2 |
| 表 2-1 DNSサービスコール | 3 |

1. 概要

1.1 機能概要

DNSクライアントは、ネームサーバと通信してドメイン名とIPアドレスの対応付けを行います。

DNSクライアントの基本機能は次のとおりです。

- (1) ドメイン名からIPアドレスを求めます。
- (2) IPアドレスからドメイン名を求めます。
- (3) IPアドレスの更新を要求します。

1.2 関連するマネージャ

DNSクライアントは、TCP/IPマネージャ上のUDPプロトコルを介してネットワーク上のネームサーバと通信します。DNSクライアントを使用するためには、TCP/IPマネージャが動作していることが必要です。

TCP/IPマネージャとのインタフェースにはITRON TCP/IP APIを使用し、UDPの通信端点を1つ使用します。TCP/IPマネージャを起動 (man_ip_startサービスコール) する場合には、最大UDP通信端点ID (udp_maxcepid) に1以上を指定してください。

尚、アプリケーション (ユーザプログラム) からはDNSクライアントが使用中の通信端点を同時に使うことはできません。DNSクライアントが使用中の通信端点をユーザプログラムで使用した場合、DNSクライアントの動作は保証できません。

1.3 関連するライブラリ

DNSクライアントは、HI.CommunicationEngineのオプション用共通タイマライブラリを使用します。共通タイマライブラリの組込みに関しては、DNSクライアント構築マニュアルを参照してください。

1.4 構成

DNS クライアント使用時の構成例を図 1-1 に示します。

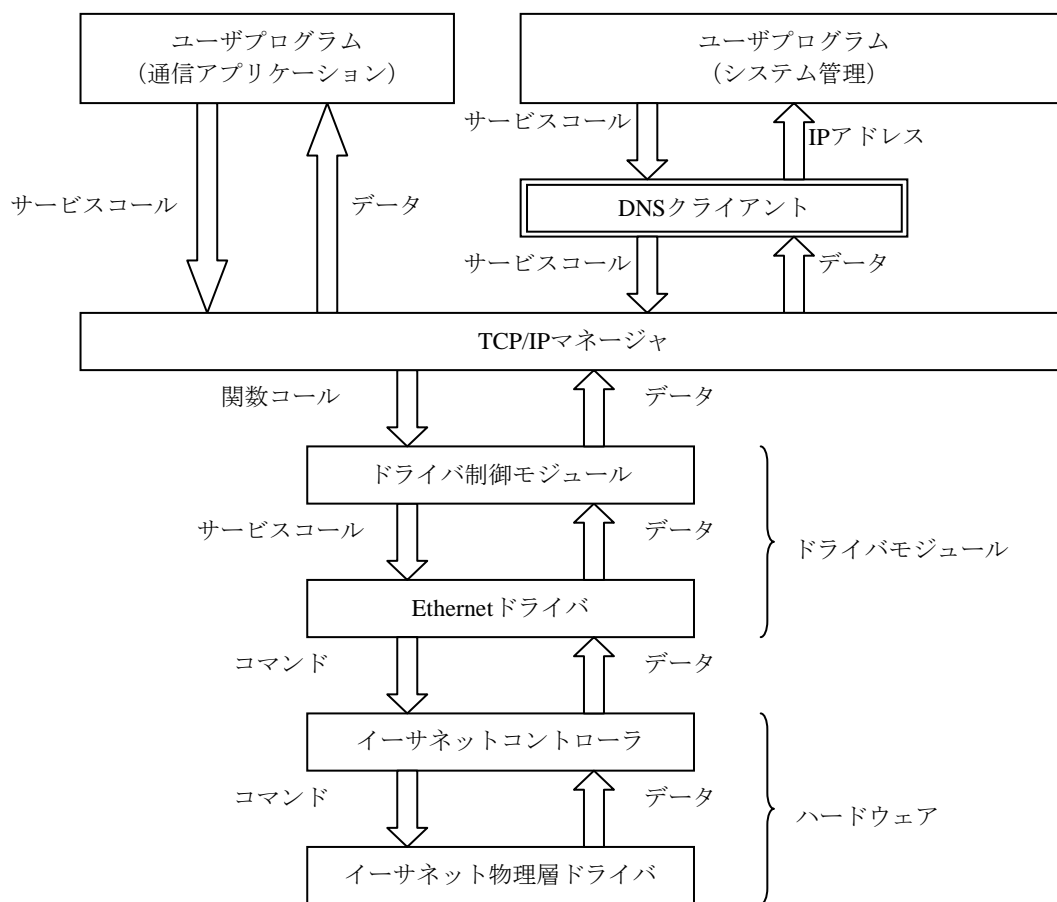


図 1-1 DNSクライアント使用時の構成例

2. DNS サービスコール

DNSサービスコールは、ユーザアプリケーションがDNSを利用する場合のインタフェースを提供します。

表 2-1 DNSサービスコール

| 項番 | サービスコール名称 | 機能 |
|----|-----------------|----------------------------|
| 1 | DNS_init | DNSクライアント状態変数の初期化 |
| 2 | DNS_start | ホストデータベースの初期化とDNSクライアントの開始 |
| 3 | DNS_stop | DNSクライアントの停止 |
| 4 | setdnssvpinfo | DNSサービスポイント情報の設定 |
| 5 | deldnssvpinfo | DNSサービスポイント情報の削除 |
| 6 | gethostbyname | 名前に対応するホストエントリの取得 |
| 7 | gethostbyaddr | ネットワークアドレスに対応するホストエントリの取得 |
| 8 | addhostent | ホストエントリの登録 |
| 9 | delhostent | ホストエントリの削除 |
| 10 | requestdnupdate | DNSサーバに対するDNS更新要求 |

本節では、DNSサービスコールについての詳細な説明を以下の形式で行っています。

| No. | サービスコール名 | 機能 | 【発行可能なシステム状態*1】 |
|-----|---------------|---------------------|-----------------|
| | C言語インタフェース | | |
| | サービスコール呼出し形式 | | |
| | パラメータ | | |
| | 型 | パラメータ | パラメータの意味 |
| | ・ | ・ | ・ |
| | ・ | ・ | ・ |
| | リターンパラメータ | | |
| | 型 | パラメータ | パラメータの意味 |
| | ・ | ・ | ・ |
| | ・ | ・ | ・ |
| | パケットの構造 | | |
| | リターン値/エラーコード | | |
| | リターン値またはニモニック | リターン値またはエラーコードの意味*2 | |
| | ・ | ・ | |
| | ・ | ・ | |
| | 解 説*3 | | |
| | | | |

*1 発行可能なシステム状態を以下のアルファベットで示します

- T：タスク実行状態
- D：ディスパッチ禁止状態
- L：CPUロック状態
- I：非タスク部実行状態

なお、各状態の詳細は各ITRONのユーザズマニュアルを参照してください。

発行可能なシステム状態以外の状態でサービスコールを発行した場合、システムの正常な動作は保証されません。

*2 エラーコードE_PARの理由として、アドレスが4の倍数以外、アドレスが奇数についてのエラーは、奇数アドレスからの16ビットや32ビットアクセスが可能なマイコン向けの製品では発生しません。

*3 オブジェクトの動的な生成(ID番号自動割付け含む)・削除機能を有さないOSを使用する場合は、該当しない説明が含まれます。

2.1.1 DNS_init

DNS クライアント状態変数の初期化

【T/D/L/I】

C言語インタフェース

```
void DNS_init ( void );
```

パラメータ

なし

リターンパラメータ

なし

解 説

DNSクライアントの状態変数を初期化します。

DNS_init は、DNSクライアント使用の前に1回だけ実行してください。

2.1.2 DNS_start ホストデータベースの初期化と DNS クライアントの開始

【T】

C 言語インタフェース

```
ER ercd = DNS_start(T_DNS_INI *par);
```

パラメータ

| | | |
|-----------|------|----------------------|
| T_DNS_INI | *par | DNS初期化情報を格納した領域のアドレス |
|-----------|------|----------------------|

リターンパラメータ

| | | |
|----|------|----------------|
| ER | ercd | リターン値またはエラーコード |
|----|------|----------------|

パラメータの構造

```
typedef struct{
    T_DNS_OSC  tskinf;      DNS初期化情報
    ID        maxsvpid;    DNSクライアントタスク情報
                        最大DNSサービスポイントID (1~4)
    H         entrycnt;    サービスポイント毎に登録可能なエントリの数
    H         namecnt;     エントリ毎に登録可能なホスト名の数
    H         addrcnt;     エントリ毎に登録可能なIPアドレスの数
    UW        *memadr;     使用するメモリの先頭アドレス
    W         memlen;     使用可能なメモリの長さ
    H         errNo;      発生したエラーの詳細を示す番号
} T_DNS_INI;

typedef struct{
    ID        tskid;       DNSクライアントタスクID番号
    PRI       tskpri;     DNSクライアントタスク優先度
    INT       stksz;      DNSクライアントタスクスタックサイズ
    ID        evfid;      DNSクライアントイベントフラグID番号
} T_DNS_OSC;
```

リターン値/エラーコード

| | |
|---------|--|
| E_OK | 正常終了 |
| E_PAR | パラメータエラー (parが奇数、maxsvpid ≤ 0、maxsvpid > 4、entrycnt < 0、namecnt < 0、addrcnt < 0、memadrが4の倍数以外または0) |
| E_NOMEM | メモリ不足 (必要なメモリが確保できない) |
| E_SYS | システムエラー (OSのリソース確保に失敗、タイマライブラリが動作していない) |
| E_ILUSE | サービスコール不正使用 (DNS_initが実行されていない) |

解説

ホストデータベースを初期化し、DNSクライアントを起動します。

maxsvpidにはDNSクライアントを使用するIPアドレス数の最大値を指定します。DNSサービスポイントIDはDNSを利用する自IPアドレスの数だけ必要となります。

DNSサービスポイントIDは複数の物理インタフェース上でDNSクライアントを利用する場合に、物理インタフェース毎にネームサーバとの通信を管理するために用います。

tskid、tskpri、stkszには、DNSクライアントタスクのタスクID番号、タスク優先度、タスクスタックサイズを指定します。

evfidには、DNSクライアントが使用するイベントフラグのID番号を指定します。

tskid、およびevfidに0を指定した場合は、OSのID番号自動割付け機能により、ID番号を割り付けます (オブジェクトの動的生成機能およびID自動割付け機能を有するOSの場合のみ)。

entrycntには、サービスポイント毎のデータベースに登録可能なホストエントリの数を指定します。

namecntには、ホストエントリ毎に登録可能なホスト名（ホストの公式名および別名）の目安となる数を指定します。ホスト名は最大でヌルを含み255文字まで許されていますが、1つのnamecntあたり20文字しか扱えません。そのため20文字を超えるホスト名の場合は、namecntを複数個分使用します。

addrcntには、ホストエントリ毎に登録可能なIPアドレステーブルの数を指定します。IPv4の場合、1つのaddrcntあたり8つのIPアドレスを扱えます。IPアドレスの数が8つを超える場合は、addrcntを複数個分使用します。

サービスポイント毎のデータベースに必要なメモリのバイト数は、

$$((\text{namecnt} + \text{addrcnt} + 1) \times 40) \times \text{entrycnt}$$

となります。

運用中に1つのホストエントリでnamecntを超えるホスト名が登録されたり、addrcntを超えるIPアドレステーブルが使用された場合は、登録できるホストエントリの数はentrycntで指定した数より小さくなります。

本サービスコールでは必要となるデータベースに必要なワークメモリをmemadrで指定されたメモリアドレスから切り出し、切り出したメモリの長さをmemlenに返します。

ワークメモリとして「データベース」の他に通信用バッファと変換用バッファとして、2244バイト必要です。

memlenには、 $\underline{(((\text{namecnt} + \text{addrcnt} + 1) \times 40) \times \text{entrycnt}) + 2244} \times \text{maxsvpid}$ を設定してください。

指定したmemlenが必要なメモリサイズに満たない場合は、memlenに必要な長さを設定し、エラーコードとしてE_NOMEMを返します。

リターン値がE_PAR、またはE_SYSの場合、発生したエラーの詳細を示す番号をerrNoに格納します。errNoに格納される番号の種類を以下に示します。ただし、parが不正なアドレスの場合、errNoには格納しません。

リターン値がE_PARの場合：

| errNo | エラー内容 |
|-------|--------------|
| 1 | tskidが不正 |
| 2 | tskpriが不正 |
| 3 | stkszが不正 |
| 4 | evfidが不正 |
| 5 | maxsvpidが不正 |
| 6 | entrycntが不正 |
| 7 | namecntが不正 |
| 8 | addrcntが不正 |
| 9 | memadrが不正 |
| 99 | その他のパラメータエラー |

リターン値がE_SYSの場合：

| errNo | エラー内容 |
|-------|--------------------|
| 1 | イベントフラグの生成に失敗 |
| 2 | DNSクライアントタスクの生成に失敗 |
| 3 | DNSクライアントタスクの起動に失敗 |
| 4 | タイマライブラリが動作していない |
| 99 | その他のシステムエラー |

2.1.3 DNS_stop DNS クライアントの停止

【T】

C 言語インタフェース

```
ER ercd = DNS_stop(void);
```

パラメータ

なし

リターンパラメータ

| | | |
|----|------|----------------|
| ER | ercd | リターン値またはエラーコード |
|----|------|----------------|

リターン値/エラーコード

| | |
|------|------|
| E_OK | 正常終了 |
|------|------|

| | |
|---------|--------------------------------|
| E_ILUSE | サービスコール不正使用 (DNSクライアントが停止している) |
|---------|--------------------------------|

解説

DNSクライアントを停止します。

DNS_startで切り出したワークメモリを解放し、DNSクライアントタスク、およびDNSクライアントのイベントフラグを削除します (オブジェクトの動的生成機能を有するOSの場合のみ)。

2.1.4 setdnssvpinfo DNS サービスポイント情報の設定

【T】

C 言語インタフェース

```
ER ercd = setdnssvpinfo(ID svpid, T_DNS_SVP *par );
```

パラメータ

| | | |
|-----------|-------|---------------------------|
| ID | svpid | DNSサービスポイントID |
| T_DNS_SVP | *par | DNSサービスポイント情報を格納した領域のアドレス |

リターンパラメータ

| | | |
|----|------|----------------|
| ER | ercd | リターン値またはエラーコード |
|----|------|----------------|

パラメータの構造

```
typedef struct{
    DNSサービスポイント情報
    UW    myipaddr;    DNSクライアントのIPアドレス
    UH    myportmin;   DNSクライアントのポート番号最小値
    UH    myportmax;   DNSクライアントのポート番号最大値
    UW    svipaddr[MAXDNSSVRCNT];  DNSサーバのIPアドレス (MAXDNSSVRCNT=4)
    ID    udpcepid;    UDP通信端点ID
    H     respwaittime;  DNSサーバからの応答待ち時間 (秒)
} T_DNS_SVP;
```

リターン値/エラーコード

| | |
|---------|--|
| E_OK | 正常終了 |
| E_ID | 不正ID番号 (svpid ≤ 0、svpid > maxsvpid、udpcepid < 0、udpcepid > udp_maxcepid) |
| E_OBJ | オブジェクト状態不正 (すでに登録されている) |
| E_PAR | パラメータエラー (parが4の倍数以外、myipaddrが未登録、myportmin > myportmax、myportmin = 0、myportmax = 0、reswaittime ≤ 0) |
| E_ILUSE | サービスコール不正使用 (DNSクライアントが停止している) |

解説

DNSサービスポイント情報を設定します。

svpidには、DNSサービスポイント情報を設定するDNSサービスポイントのIDを指定します。

myipaddrには、指定したDNSサービスポイントが使用するクライアント側のIPアドレスを指定します。
myportminには、指定したDNSサービスポイントが使用するクライアント側のポート番号の最小値を指定します。

myportmaxには、指定したDNSサービスポイントが使用するクライアント側のポート番号の最大値を指定します。

DNSサービスポイント毎に、指定したIPアドレスとポート番号を使用してネームサーバに問合せを行います。問合せに用いるポート番号は問合せ毎にmyportminからmyportmaxの範囲で自動更新して使用します。

svipaddr[n]には、DNSサーバのIPアドレスを設定します。MAXDNSSVRCNTの数だけDNSサーバのIPアドレスを登録できます。n=0の位置から順に登録し、未登録のsvipaddr[n]には0を設定してください。

udpcepidには、DNSサーバとの通信に用いるTCP/IPマネージャ上のUDP通信端点のIDを指定します。udpcepidに0を指定した場合は、空いているUDP通信端点を探して使用します。

reswaittimeには、DNSサーバ1箇所あたりの応答待ち時間 (秒) を設定します。reswaittimeに60を超える値を指定した場合は、応答待ち時間として60秒が使用されます。

登録されているDNSサービスポイント情報を書き換えることはできません。変更する場合は、一度DNSサービスポイント情報を削除 (deldnssvpinfo) してから、再度設定 (setdnssvpinfo) してください。

2.1.5 deldnssvpinfo DNS サービスポイント情報の削除

【T】

C 言語インタフェース

```
ER ercd = deldnssvpinfo(ID svpid );
```

パラメータ

| ID | svpid | DNSサービスポイントID |
|----|-------|---------------|
|----|-------|---------------|

リターンパラメータ

| ER | ercd | リターン値またはエラーコード |
|----|------|----------------|
|----|------|----------------|

リターン値/エラーコード

| | |
|---------|--------------------------------------|
| E_OK | 正常終了 |
| E_ID | 不正ID番号 (svpid ≤ 0, svpid > maxsvpid) |
| E_OBJ | オブジェクト状態不正 (指定したDNSサービスポイントIDは使用中) |
| E_ILUSE | サービスコール不正使用 (DNSクライアントが停止している) |

解説

DNSサービスポイント情報を削除します。

svpidには、DNSサービスポイント情報を削除するDNSサービスポイントのIDを指定します。

DNSクライアントが指定したDNSサービスポイントを使用して通信を行っている場合は、エラーコードとしてE_OBJを返します。

deldnssvpinfoでは、サービスポイント上に構成済みのホストエントリのキャッシュを削除します。

2.1.6 gethostbyname 名前に対応するホストエントリの取得

【T】

C言語インタフェース

```
T_HOSTENT *retcode = gethostbyname(ID svpid, char *name, T_DNS_PAR *par);
```

パラメータ

| | | |
|-----------|-------|-------------------------|
| ID | svpid | DNSサービスポイントID |
| char | *name | ホスト名の文字列 |
| T_DNS_PAR | *par | 戻り情報作成領域の情報を格納した領域のアドレス |

リターンパラメータ

| | | |
|----------------|--------------|-----------------------------|
| struct hostent | *retcode | リターン値 (0またはホストのエントリ情報のアドレス) |
| int | par->errcode | エラーコード |

パラメータの構造

```
typedef struct {
    戻り情報作成領域の情報
    char    *entbuf;        ホストのエントリ情報を作成する領域のアドレス
    int     entbuflen;      ホストのエントリ情報を作成する領域の長さ
    char    *aliasbuf;     ホストの別名情報を作成する領域のアドレス
    int     aliasbuflen;   ホストの別名情報を作成する領域の長さ
    char    *addrbuf;      ホストのネットワークアドレス情報を作成する領域のアドレス
    int     addrbuflen;    ホストのネットワークアドレス情報を作成する領域の長さ
    int     errcode;       エラーコード
} T_DNS_PAR;

typedef struct{
    ホストのエントリ情報
    char    *h_name;       ホストの公式名
    char    **h_aliases;   ホストの別名の配列 (NULLで終了する) の先頭アドレス
    int     h_addrtype;    返されるアドレスのファミリ型 (常にAF_INET (2) )
    int     h_length;      バイト単位でのアドレスの長さ (常に4)
    char    **h_addr_list; ホスト用のネットワークアドレス (ネットワークバイトオーダ)
                          の配列の先頭アドレス
} T_HOSTENT;
```

リターン値

| | |
|----------|------------------------|
| 0以外 | 正常終了 (ホストのエントリ情報のアドレス) |
| 0 (NULL) | エラー |

エラーコード

| | | |
|---------------|-----|--|
| DNSERR_OK | 0 | 正常終了 |
| DNSERR_SVPID | 1 | サービスポイントIDが不正 |
| DNSERR_ENTPAR | 2 | エントリ情報領域のパラメータ不正 |
| DNSERR_ALIPAR | 3 | 別名情報領域のパラメータ不正 |
| DNSERR_ADRPAR | 4 | ネットワークアドレス情報領域のパラメータ不正 |
| DNSERR_NAME | 5 | ホスト名パラメータ不正 |
| DNSERR_ADROBJ | 101 | DNSクライアントのアドレス未設定 |
| DNSERR_CEPOBJ | 102 | 通信端点生成エラー |
| DNSERR_SNDOBJ | 103 | パケット送信失敗 |
| DNSERR_BUFOBJ | 104 | ホストエントリ用のキャッシュ不足 |
| DNSERR_SYSOBJ | 108 | システム状態エラー |
| DNSERR_OBJ | 109 | その他の状態エラー |
| DNSERR_NOFD | 210 | 指定したホストの公式名が見つからない(HOST_NOT_FOUND) |
| DNSERR_TMPO | 220 | ネームサーバで一時的なエラーが起こった (TRY_AGAIN) |
| DNSERR_NOREC | 230 | ネームサーバの応答がない。または、回復不可エラー (NO_RECOVERY) |
| DNSERR_NOADR | 240 | 名前は有効であるが、IPアドレスがない (NO_ADDRESSまたはNO_DATA) |
| DNSERR_RLWAI | 901 | 待ち状態が強制解除された |

解説

ホスト名に対するホストエントリを取得します。
指定された名前から対応するhostent型のエントリ情報を作成し、その情報の先頭アドレスを返します。

svpidにはDNSサービスポイントIDを指定します。

nameにはヌルを終端とする文字列によりホスト名またはドット区切りのIPアドレスを指定します。ヌルを含む長さは255バイト以下である必要があります。ドットで区切られた区間の文字列の長さは63バイト以下である必要があります。長さが255バイトを超える場合やドットで区切られた長さが63バイトを超える場合には、エラーとしてerrcodeにDNSERR_NAMEを設定し、リターン値として0（エラー）を返します。

本サービスコールはホストのエントリ情報をユーザがparに指定した領域に作成します。parが4の倍数でない場合やNULLの場合には、errcodeに何も設定を行わずにリターン値として0を返します。

entbufにはホストのエントリ情報を作成する領域の先頭アドレスを指定します。

entbuflenにはホストのエントリ情報を作成する領域の長さをバイト単位で指定します。

必要な長さは $(\text{ホストのエントリ情報構造体の長さ}) + (\text{ヌルを含むホストの公式名用領域の長さ})$ です (ホストのエントリ情報構造体の長さ = sizeof(T_HOSTENT))。

entbufが4の倍数以外またはentbuflenがsizeof(T_HOSTENT)より小さい場合は、エラーとしてerrcodeにDNSERR_ENTPARを設定し、リターン値として0を返します。

entbuflenからsizeof(T_HOSTENT)を引いた長さが、ヌルを含むホストの公式名の長さより短い場合は、h_nameにNULL(0)を設定します。(確実にホストの公式名を取得するにはsizeof(T_HOSTENT)+255バイトの領域が必要になります。)

aliasbufにはホストの別名情報を作成する領域の先頭アドレスを指定します。

aliasbuflenにはホストの別名情報を作成する領域の長さをバイト単位で指定します。

必要な長さは $(\text{ヌルを含む別名の文字数の総和}) + ((\text{別名の数}) + 1) \times 4$ です。

aliasbufが4の倍数以外またはaliasbuflenが4より小さい場合は、エラーとしてerrcodeにDNSERR_ALIPARを設定し、リターン値として0を返します。

ホストの別名情報はaliasbuflenで指定した長さと格納できる分だけしか作成しません。

ホストの別名がない場合は、h_aliasesにNULL(0)を設定します。

addrbufにはホストのネットワークアドレス情報を作成する領域の先頭アドレスを指定します。

addrbuflenにはホストのネットワークアドレス情報を作成する領域の長さをバイト単位で指定します。

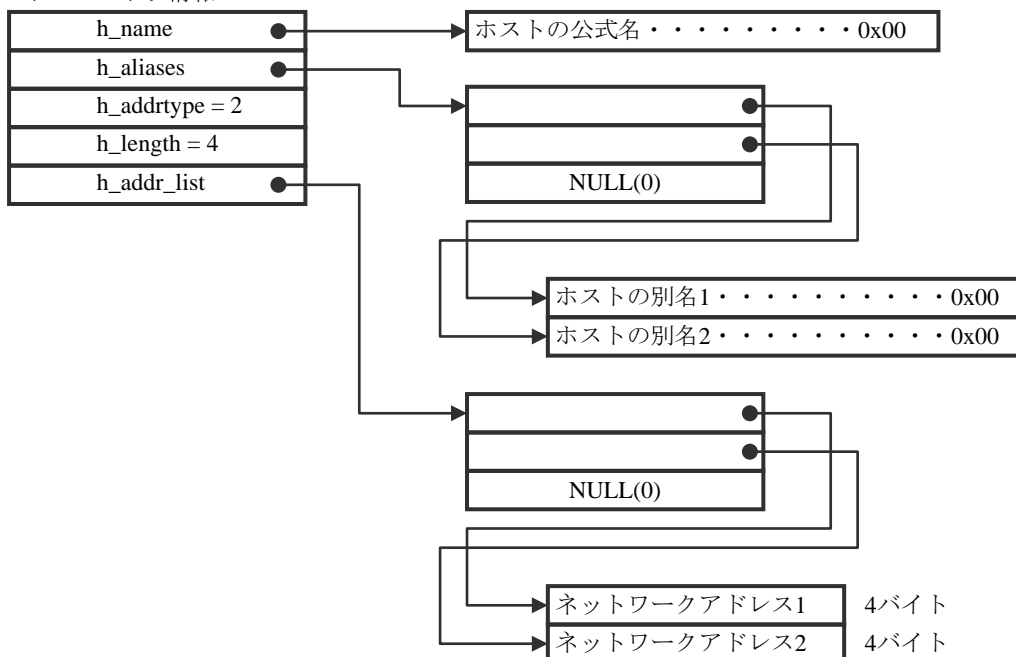
必要な長さは $(\text{ネットワークアドレスの数}) \times 2 + 1) \times 4$ です。

addrbufが4の倍数以外またはaddrbuflenが4より小さい場合は、エラーとしてerrcodeにDNSERR_ADRPARを設定し、リターン値として0を返します。

ホストのネットワークアドレス情報はaddrbuflenで指定した長さと格納できる分だけしか作成しません。

ホストのネットワークアドレスがない場合は、h_addr_listにNULL(0)を設定します。

ホストのエントリ情報



2.1.7 gethostbyaddr アドレスに対応するホストエントリの取得

【T】

C言語インタフェース

```
T_HOSTENT *retcode = gethostbyaddr(ID svpid, char *addr, int len, int type, T_DNS_PAR *par);
```

パラメータ

| | | |
|-----------|-------|-----------------------------|
| ID | svpid | DNSサービスポイントID |
| char | *addr | IPアドレスを格納した領域のアドレス |
| int | len | IPアドレスの長さ (4 固定 (IPv4)) |
| int | type | IPアドレスタイプ (2 固定 (AF_INET)) |
| T_DNS_PAR | *par | 戻り情報作成領域の情報を格納した領域のアドレス |

リターンパラメータ

| | | |
|----------------|--------------|-----------------------------|
| struct hostent | *retcode | リターン値 (0またはホストのエントリ情報のアドレス) |
| int | par->errcode | エラーコード |

パラメータの構造

```
typedef struct {
    char *entbuf;          ホストのエントリ情報を作成する領域のアドレス
    int entbuflen;        ホストのエントリ情報を作成する領域の長さ
    char *aliasbuf;       ホストの別情報を作成する領域のアドレス
    int aliasbuflen;      ホストの別情報を作成する領域の長さ
    char *addrbuf;        ホストのネットワークアドレス情報を作成する領域のアドレス
    int addrbuflen;       ホストのネットワークアドレス情報を作成する領域の長さ
    int errcode;          エラーコード
} T_DNS_PAR;

typedef struct {
    char *h_name;         ホストの公式名
    char **h_aliases;     ホストの別名の配列 (NULLで終了する) の先頭アドレス
    int h_addrtype;       返されるアドレスのファミリー型 (常にAF_INET (2) )
    int h_length;         バイト単位でのアドレスの長さ (常に4)
    char **h_addr_list;   ホスト用のネットワークアドレス (ネットワークバイトオーダ)
                        の配列の先頭アドレス
} T_HOSTENT;
```

リターン値

| | |
|----------|------------------------|
| 0以外 | 正常終了 (ホストのエントリ情報のアドレス) |
| 0 (NULL) | エラー |

詳細エラーコード

| | | |
|----------------|-----|------------------------|
| DNSERR_OK | 0 | 正常終了 |
| DNSERR_SVPID | 1 | サービスポイントIDが不正 |
| DNSERR_ENTPAR | 2 | エントリ情報領域のパラメータ不正 |
| DNSERR_ALIPAR | 3 | 別情報領域のパラメータ不正 |
| DNSERR_ADRPAR | 4 | ネットワークアドレス情報領域のパラメータ不正 |
| DNSERR_ADR | 6 | IPアドレスパラメータ不正 |
| DNSERR_ADRLEN | 7 | IPアドレス長不正 |
| DNSERR_ADRTYPE | 8 | IPアドレスタイプ不正 |
| DNSERR_ADROBJ | 101 | DNSクライアントのアドレス未設定 |
| DNSERR_CEPOBJ | 102 | 通信端点生成エラー |
| DNSERR_SNDOBJ | 103 | パケット送信失敗 |
| DNSERR_BUFOBJ | 104 | ホストエントリ用のキャッシュ不足 |
| DNSERR_SYSOBJ | 108 | システム状態エラー |
| DNSERR_OBJ | 109 | その他の状態エラー |

| | | |
|--------------|-----|---|
| DNSERR_NOFD | 210 | 指定したホストの公式名が見つからない (HOST_NOT_FOUND) |
| DNSERR_TMPO | 220 | ネームサーバで一時的なエラーが起こった (TRY_AGAIN) |
| DNSERR_NOREC | 230 | ネームサーバの応答がない。または、回復不可エラー (NO_RECOVERY) |
| DNSERR_NOADR | 240 | 名前は有効であるが、IPアドレスがない(NO_ADDRESSまたはNO_DATA) |
| DNSERR_RLWAI | 901 | 待ち状態が強制解除された |

解説

アドレスに対するホストエントリを取得します。
指定されたアドレスから対応するhostent型のエントリ情報を作成し、その情報の先頭アドレスを返します。

svpidには、DNSサービスポイントIDを指定します。

addrには、バイナリ形式のネットワークバイトオーダー (ビッグエンディアン) でIPアドレスを格納した領域の先頭アドレスを指定します。

lenには、IPアドレスの長さを指定します。本製品ではIPv4にのみ対応しているため、lenに4以外の値を指定した場合は、エラーとしてerrcodeにDNSERR_ADRLLENを設定しリターン値として0を返します。

typeには、IPアドレスの種別を指定します。本製品ではAF_INET(2)にのみ対応しているため、typeに2以外の値を指定した場合は、エラーとしてerrcodeにDNSERR_ADRTYPEを設定しリターン値として0を返します。

本サービスコールはホストのエントリ情報をユーザがparに指定した領域に作成します。

parが4の倍数でない場合やNULLの場合には、errcodeに何も設定を行わずにリターン値として0を返します。

entbufにはホストのエントリ情報を作成する領域の先頭アドレスを指定します。

entbuflenにはホストのエントリ情報を作成する領域の長さをバイト単位で指定します。

必要な長さは $(\text{ホストのエントリ情報構造体の長さ}) + (\text{ヌルを含むホストの公式名用領域の長さ})$ です (ホストのエントリ情報構造体の長さ = sizeof(T_HOSTENT))。

entbufが4の倍数以外またはentbuflenがsizeof(T_HOSTENT)より小さい場合は、エラーとしてerrcodeにDNSERR_ENTPARを設定し、リターン値として0を返します。

entbuflenからsizeof(T_HOSTENT)を引いた長さが、ヌルを含むホストの公式名の長さより短い場合は、h_nameにNULL(0)を設定します。(確実にホストの公式名を取得するにはsizeof(T_HOSTENT)+255バイトの領域が必要になります。)

aliasbufにはホストの別名情報を作成する領域の先頭アドレスを指定します。

aliasbuflenにはホストの別名情報を作成する領域の長さをバイト単位で指定します。

必要な長さは $(\text{ヌルを含む別名の文字数の総和}) + ((\text{別名の数}) + 1) \times 4$ です。

aliasbufが4の倍数以外またはaliasbuflenが4より小さい場合は、エラーとしてerrcodeにDNSERR_ALIPARを設定し、リターン値として0を返します。

ホストの別名情報はaliasbuflenで指定した長さと格納できる分だけしか作成しません。

ホストの別名がない場合は、h_aliasesにNULL(0)を設定します。

addrbufにはホストのネットワークアドレス情報を作成する領域の先頭アドレスを指定します。

addrbuflenにはホストのネットワークアドレス情報を作成する領域の長さをバイト単位で指定します。

必要な長さは $(\text{(ネットワークアドレスの数)} \times 2 + 1) \times 4$ です。

addrbufが4の倍数以外またはaddrbuflenが4より小さい場合は、エラーとしてerrcodeにDNSERR_ADRPARを設定し、リターン値として0を返します。

ホストのネットワークアドレス情報はaddrbuflenで指定した長さと格納できる分だけしか作成しません。

ホストのネットワークアドレスがない場合は、h_addr_listにNULL(0)を設定します。

2.1.8 addhostent

ホストエントリの登録

【T】

C言語インタフェース

```
ER ercd = addhostent(ID svpid, T_HOSTENT *host, unsigned long lifetime, int *errcode);
```

パラメータ

| | | |
|---------------|----------|------------------------|
| ID | svpid | DNSサービスポイントID |
| T_HOSTENT | *host | ホストのエントリ情報を格納した領域のアドレス |
| unsigned long | lifetime | ホストエントリの生存時間 (秒) |
| int | *errcode | 詳細エラーコードを格納する領域を示すアドレス |

リターンパラメータ

| | | |
|-----|---------|----------------|
| ER | ercd | リターン値またはエラーコード |
| int | errcode | 詳細エラーコード |

パラメータの構造

```
typedef struct{
    char      *h_name;      ホストの公式名
    char      **h_aliases;  ホストの別名の配列 (NULLで終了する) の先頭アドレス
    int       h_addrtype;   アドレスのファミリ型 (常にAF_INET (2) )
    int       h_length;     バイト単位のアドレスの長さ (常に4)
    char      **h_addr_list; ホスト用のネットワークアドレス (ネットワークバイトオーダ)
                          の配列の先頭アドレス
} T_HOSTENT;
```

リターン値/エラーコード

| | |
|---------|---|
| E_OK | 正常終了 |
| E_ID | 不正ID番号 (svpid ≤ 0, svpid > maxsvpid) |
| E_PAR | パラメータエラー (hostが4の倍数以外、ホストエントリ情報が不正、errcodeが4の倍数以外、またはNULL(0)) |
| E_OBJ | オブジェクト状態不正 (指定したDNSサービスポイントIDは使用中) |
| E_BOVR | バッファオーバーフロー (ホストエントリを登録するキャッシュが足りない) |
| E_ILUSE | サービスコール不正使用 (DNSクライアントが停止している) |

詳細エラーコード (E_PAR発生時)

| | | |
|----------------|----|---------------------|
| DNSERR_HOSTENT | 11 | ホストエントリ情報領域のパラメータ不正 |
| DNSERR_HNAME | 12 | ホストの公式名情報の不正 |
| DNSERR_HALI | 13 | ホストの別名情報の不正 |
| DNSERR_HADRTYP | 14 | アドレスのファミリ型が不正 |
| DNSERR_HLEN | 15 | バイト単位のアドレスの長さが不正 |
| DNSERR_HADRLI | 16 | ネットワークアドレス配列の情報不正 |

解説

指定したサービスポイントのホストエントリキャッシュ領域にホストエントリ情報を登録します。

svpidには、ホストエントリ情報を登録するDNSサービスポイントのIDを指定します。

hostには、ホストエントリ情報を格納した領域のアドレスを指定します。

hostが4の倍数以外、またはホストエントリ情報が不正の場合は、errcodeに対応する詳細エラーコードを設定しリターン値としてE_PARを返します。

h_aliases (ホストの別名の配列の先頭アドレス) やh_addr_list (ホスト用のネットワークアドレスの配列の先頭アドレス) が4の倍数以外の場合は、errcodeに対応する詳細エラーコードを設定しリターン値としてE_PARを返します。

ホストエントリ情報内のポインタ (h_name、h_aliases、h_addr_listやh_aliasesの先のポインタ配列、h_addr_listの先のポインタ配列) が正しい領域を指していない場合の動作は保証されません。

`lifetime`には、登録するホストエントリ情報の生存時間を秒単位で指定します。`lifetime`に0を指定した場合は、生存時間0として扱うため登録は行いません。

`errcode`には、詳細エラーコードを格納する領域を示すアドレスを指定します。`errcode`が4の倍数でない場合やNULLの場合には、`errcode`に何も設定を行わずにリターン値としてE_PARを返します。

2.1.9 delhostent

ホストエントリの削除

【T】

C言語インタフェース

```
ER ercd = delhostent(ID svpid);
```

パラメータ

| | | |
|----|-------|---------------|
| ID | svpid | DNSサービスポイントID |
|----|-------|---------------|

リターンパラメータ

| | | |
|----|------|----------------|
| ER | ercd | リターン値またはエラーコード |
|----|------|----------------|

リターン値/エラーコード

| | |
|---------|--------------------------------------|
| E_OK | 正常終了 |
| E_ID | 不正ID番号 (svpid ≤ 0, svpid > maxsvpid) |
| E_OBJ | オブジェクト状態不正 (指定したDNSサービスポイントIDは使用中) |
| E_ILUSE | サービスコール不正使用 (DNSクライアントが停止している) |

解説

指定したサービスポイントのホストエントリキャッシュを全削除します。

svpidには、ホストエントリ情報を削除するDNSサービスポイントのIDを指定します。

DNSクライアントが指定したDNSサービスポイントを使用して通信を行っている場合は、エラーコードとしてE_OBJを返します。

2.1.10 requestdnsupdate

DNS 更新要求

【T】

C 言語インタフェース

int retcode = requestdnsupdate (ID svpid, T_DDNSPAR *par);

パラメータ

| | | |
|-----------|--------|--------------------|
| ID | svpid; | DNSサービスポイントID |
| T_DDNSPAR | *par; | 更新要求情報を示す領域の先頭アドレス |

リターンパラメータ

| | | |
|-----|--------------|--------|
| int | retcode | リターン値 |
| int | par->errcode | エラーコード |

パラメータの構造

```

typedef struct{
    char *zname;          更新情報
                        更新するゾーンの名称
    unsigned short zclass; 更新するゾーンのクラス
    short prereqcnt;     更新対象の条件を示す配列の数
    T_DNSPREREQ *prereq; 更新対象の条件を示す配列の先頭アドレス
    short upcnt;         更新内容の配列数
    short adddatcnt     更新要求の追加データの長さ。
    T_DNSUP *update;     更新内容を示す配列の先頭アドレス
    unsigned char *adddat 更新要求の追加データの先頭アドレス
    int errcode;         エラーコード
} T_DDNSPAR;

typedef struct{
    unsigned short pretype; 更新対象条件
                        更新条件種別
    union{
        T_HOSTENT *hent;    更新条件データを示す領域のアドレス
        T_DNSRRRTYP *rrtyp;
        char *alldata;
    }predata;
} T_DNSPREREQ;

typedef struct{
    unsigned short uptype; 更新内容
                        更新内容種別
    unsigned long lifetime; 更新内容の生存時間
    union{
        T_HOSTENT *hent;    更新データを示す領域のアドレス
        T_DNSRRRTYP *rrtyp;
    }update;
} T_DNSUP;

typedef struct{
    char *h_name;          ホストのエントリ情報
                        ホストの公式名
    char **h_aliases;     ホストの別名の配列 (NULLで終了する) の
                        先頭アドレス
    int h_addrtype;       返されるアドレスのファミリ型 (常にAF_INET (2) )
    int h_length;         バイト単位のアドレスの長さ (常に4)
    char **h_addr_list;   ホスト用のネットワークアドレス
                        (ネットワークバイトオーダ)の配列の先頭アドレス
} T_HOSTENT;

```



```

typedef struct{
    char *hostname;          リソースレコード情報
                           ホストの公式名
    unsigned short type;    リソースレコードタイプ (アドレス情報タイプ: 1、
                           別名情報タイプ: 5)
} T_DNSRRYP;

```

リターン値

| | |
|----|------|
| 0 | 正常終了 |
| -1 | エラー |

詳細エラーコード

| | | |
|-----------------|-----|---|
| DNSERR_OK | 0 | 正常終了 |
| DNSERR_SVPID | 1 | サービスポイントIDが不正 |
| DNSERR_ZNAME | 21 | ゾーン名パラメータ不正 |
| DNSERR_ZCLASS | 22 | ゾーンクラスパラメータ不正 |
| DNSERR_PRECNT | 23 | 更新条件数パラメータ不正 |
| DNSERR_PREPAR | 24 | 更新条件パラメータ不正 |
| DNSERR_PRETYPE | 25 | 更新条件種別パラメータ不正 |
| DNSERR_PREDATA | 26 | 更新条件データパラメータ不正 |
| DNSERR_UPCNT | 27 | 更新内容の配列数パラメータ不正 |
| DNSERR_UPPAR | 28 | 更新内容パラメータ不正 |
| DNSERR_UPTYPE | 29 | 更新内容種別パラメータ不正 |
| DNSERR_UPDATA | 30 | 更新内容データパラメータ不正 |
| DNSERR_ADDPAR | 31 | 追加データパラメータ不正 |
| DNSERR_ADDDATA | 32 | 追加データパラメータ不正 |
| DNSERR_ADDCNT | 33 | 追加データの長さのパラメータ不正 |
| DNSERR_MSGOVR | 34 | メッセージ最大長(512byte)オーバー |
| DNSERR_ADROBJ | 101 | DNSクライアントのアドレス未設定 |
| DNSERR_CEPOBJ | 102 | 通信端点生成エラー |
| DNSERR_SNDOBJ | 103 | パケット送信失敗 |
| DNSERR_SYSOBJ | 108 | システム状態エラー |
| DNSERR_OBJ | 109 | その他の状態エラー |
| DNSERR_NOFD | 210 | 指定したホストの公式名が見つからない (HOST_NOT_FOUND) |
| DNSERR_TMPO | 220 | ネームサーバで一時的なエラーが起こった (TRY_AGAIN) |
| DNSERR_NOREC | 230 | ネームサーバの応答がない。または、回復不可エラー (NO_RECOVERY) |
| DNSERR_NOADR | 240 | 名前は有効であるが、IPアドレスがない(NO_ADDRESSまたはNO_DATA) |
| DNSERR_NXDOMAIN | 301 | 指定のゾーンに存在すべきホスト名がない(NXDOMAIN) |
| DNSERR_YXDOMAIN | 302 | 指定のゾーンに存在すべきでないホスト名が存在する(YXDOMAIN) |
| DNSERR_NXRRSET | 303 | 指定のゾーンに存在すべき情報 (ホストの別名情報やホストのネットワークアドレスの情報) がない(NXRRSET) |
| DNSERR_YXRRSET | 304 | 指定のゾーンに存在すべきでない情報 (ホスト名やホストのネットワークアドレスの情報) が存在する(YXRRSET) |
| DNSERR_SERVFAIL | 305 | ダイナミック更新を実行できる状態のDNSサーバが登録されていない。 |
| DNSERR_REFUSED | 306 | DNSサーバがポリシーやセキュリティ上の理由で更新を拒否した。 |
| DNSERR_NOTZONE | 307 | ゾーンがない。または、DNSサーバが指定したゾーンの権威でない。 |
| DNSERR_RLWAI | 901 | 待ち状態が強制解除された |

解説

DNSサーバに対し、指定のゾーンにリソースレコードの追加、削除を要求します。
指定された引数の値に対応する更新要求をDNSサーバに発行し、その結果を返します。

本サービスコールによってDNSサーバのホストエントリ情報を更新した場合でも、サービスポイントのホストエントリキャッシュは更新しません。本サービスコールを使用する場合は、サービスポイントのホストエントリキャッシュを一度、全削除するようにしてください。

更新要求は、setdnssvpinfo()において登録されたDNSサーバのIPアドレスに対し発行します。最初のIPア

ドレスから発行しその結果が、DNSサーバからの返答がない場合やDNSサーバがDNSのダイナミック更新（DDNS）に対応していない場合やDNSサーバが更新できない状態の場合は、次のIPアドレスに対して再度更新要求を発行します。

再度発行を繰り返した結果、再度発行するためのIPアドレスが無くなった場合にはエラーとしてerrcodeにDNSERR_NORECを設定しリターン値として-1（エラー）を返します。

svpidには、DNSサービスポイントIDを指定します。

parには、更新情報を示す領域のアドレスを指定します。parが4の倍数でない場合やNULLの場合には、errcodeに何も設定を行わずにリターン値として-1（エラー）を返します。

znameとzclassにより更新対象のゾーンを指定します。

znameには更新するゾーンの名前としてヌルを終端とする文字列を指定します。指定する文字列はヌルを含む長さが255バイト以下である必要があります。長さが255バイトを超える文字列を指定した場合は、エラーとしてerrcodeにDNSERR_ZNAMEを設定し、リターン値として-1（エラー）を返します。

zclassはゾーンのクラスを指定します。通常はインターネットの場合の1となります。

要求を発行したDNSサーバにおいて、znameとzclassにより指定したゾーンの権威でない場合はエラーとしてerrcodeにDNSERR_NOTZONEを設定し、リターン値として-1（エラー）を返します。

prereqcntとprereqにより更新対象のゾーンが満たしているべきの前提条件を指定します。

prereqには前提条件の内容を示す配列の先頭アドレスを指定します。prereqcntにはprereqの示す配列の数を指定します。特に条件が無い場合には、prereqcntに0を指定しprereqにNULLを指定します。

pretypeには、前提条件の種別を指定します。predataには、前提条件のデータを指定します。predataのデータ型はpretypeに指定した値によって異なります。

前提条件種別ごとにpretypeに指定する値とそのときのpredataの型を以下の表に示します。

| 前提条件種別 | pretypeに指定する値 | predataの型 | 条件の内容 |
|------------------------|----------------------|----------------------------|--|
| 指定の名前を使用している | DNS_NAMEUSE (1) | T_HOSTENT 型 (hent) | 指定のゾーンにh_nameで指定したホスト公式名の情報が存在すること。 |
| 指定の名前を使用していない | DNS_NAMENOT (2) | | 指定のゾーンにh_nameで指定したホスト公式名の情報が存在しないこと。 |
| 指定のリソースレコードが指定した値で存在する | DNS_RRVALDEP (3) | | 指定したゾーンにpredataで指定した情報が指定した値で全て存在すること。 |
| 指定リソースレコードが存在する | DNS_RRVALIN (4) | T_DNSRRTYP 型 (rrtyp) | 指定したゾーンにpredataで指定したリソースレコードが存在すること。ドメイン名前とタイプが等しい必要がある。値は異なってもよい。 |
| 指定リソースレコード集合は存在していない | DNS_RRNOT (5) | | 指定したゾーンにpredataで指定したリソースレコードが存在しないこと。ドメイン名前とタイプが同じか等しいリソースレコードが存在してはいけない。 |
| 送信データ指定 | DNS_ALLDATA (255) | UB *型 (alldata) | pretypeがDNS_ALLDATAの場合は条件でなく、更新要求の条件セクション（prerequisite section）で送信するデータをpredataに指定します。 |

pretypeに表以外の値を指定した場合は、エラーとしてerrcodeにDNSERR_PRETYPEを設定しリターン値として-1（エラー）を返します。

指定したゾーンがprereqの配列で指定した条件をどれか1つでも満たしていない場合、DNSサーバにおいてゾーンの内容は更新されません。対応するエラーコードをerrcodeに設定し、リターン値として-1（エラ

一) を返します。

前提条件の種別ごとの概要とエラーコードについて以下に示します。

(1) `pretype=DNS_NAMEUSE` (指定の名前を使用している)

指定ゾーンの条件として「`h_name`で指定したホスト公式名の情報が何か存在すること」を指定します。`predata`のパラメータのうち、`h_name`には存在している必要があるホスト名を指定します。`h_name`以外のパラメータは指定する必要はありません。`h_name`以外のパラメータは無視します。

`zname`と`zclass`により指定したゾーンに指定したホスト公式名の情報が無い場合、DNSサーバにおいてゾーンの内容は更新されません。エラーとして`errcode`に`DNSERR_NXDOMAIN`を設定し、リターン値として-1 (エラー) を返します。

(2) `pretype=DNS_NAMENOT` (指定の名前を使用していない)

指定ゾーンの条件として「`h_name`で指定したホスト公式名の情報が何も存在していないこと」を指定します。`predata`のパラメータのうち、`h_name`には存在してはいけないホスト名を指定します。`h_name`以外のパラメータは指定する必要はありません。`h_name`以外のパラメータは無視します。

`zname`と`zclass`により指定したゾーンに指定したホスト公式名の情報が存在する場合、DNSサーバにおいてゾーンの内容は更新されません。エラーとして`errcode`に`DNSERR_YXDOMAIN`を設定し、リターン値として-1 (エラー) を返します。

(3) `pretype=DNS_PRVALDEP` (指定のリソースレコードが指定した値で存在する)

指定ゾーンの条件として「`h_name`で指定したホスト公式名の情報に`predata`で指定した情報が全て存在していること」を指定します。`predata`のパラメータのうち、`h_name`には存在している必要があるホスト名を指定します。`h_aliases`には存在している必要があるホストの別名を指定します。`h_addr_list`には存在している必要があるネットワークアドレスを指定します。

`h_aliases`に`NULL(0)`以外を指定した場合は、ホスト別名についての情報が指定ゾーンに過不足無しで存在することが条件になります。`h_aliases`に`NULL(0)`を指定した場合は、ホストの別名の情報を条件としません。他の条件を満たしているならば、ホストの別名があってもなくてもDNSサーバにおいてゾーンの内容は更新されます。

`h_addr_list`に`NULL(0)`以外を指定した場合は、ネットワークアドレスの配列にある情報が指定ゾーンに過不足無しで存在することが条件になります。`h_addr_list`に`NULL(0)`を指定した場合は、ネットワークアドレスの情報を条件としません。他の条件を満たしているならば、ネットワークアドレスの情報があってもなくてもDNSサーバにおいてゾーンの内容は更新されます。

`zname`と`zclass`により指定したゾーンが上記に示した条件を満たしていない場合、DNSサーバにおいてゾーンの内容は更新されません。指定した条件の情報が無い場合には、エラーとして`errcode`に`DNSERR_NXRRSET`を設定し、リターン値として-1 (エラー) を返します。

指定した条件以外の情報が存在する場合には、エラーとして`errcode`に`DNSERR_YXRRSET`を設定し、リターン値として-1 (エラー) を返します。

`zname`と`zclass`により指定したゾーンに指定したホスト公式名の情報が無い場合、DNSサーバにおいてゾーンの内容は更新されません。エラーとして`errcode`に`DNSERR_NXDOMAIN`を設定し、リターン値として-1 (エラー) を返します。

(4) `pretype=DNS_PRVALIN` (指定のリソースレコードが存在する)

指定ゾーンの条件として「`hostname`で指定したホスト公式名の情報に`type`で指定したタイプの情報が何か存在していること」を指定します。

`predata`のパラメータのうち、`hostname`には存在している必要があるホスト名を指定します。`type`には存在している必要があるリソースレコードのタイプを指定します。

`zname`と`zclass`により指定したゾーンにある指定したホスト公式名の情報に指定したタイプの情報が何も無い場合、DNSサーバにおいてゾーンの内容は更新されません。エラーとして`errcode`に`DNSERR_NXRRSET`を設定し、リターン値として-1 (エラー) を返します。

(5) `pretype=DNS_PRNOT` (指定のリソースレコードが存在していない)

指定ゾーンの条件として「`hostname`で指定したホスト公式名の情報に`type`で指定したタイプの情報が何も存在していないこと」を指定します。

`predata`のパラメータのうち、`hostname`には存在している必要があるホスト名を指定します。`type`には存在してはいけないリソースレコードのタイプを指定します。

指定したゾーンの情報の中に指定したホスト公式名の指定したタイプの情報が存在している場合、DNS

サーバにおいてゾーンの内容は更新されません。エラーとしてerrcodeにDNSERR_YXRRSETを設定し、リターン値として-1（エラー）を返します。

(6) pretype=DNS_ALLDATA（送信データ指定）

pretypeがDNS_ALLDATAの場合は条件でなく、条件セクション（prerequisite section）で送信するデータをalldataに直接指定します。alldataに指定したデータがリソースレコードのフォーマットになっていない場合は、errcodeにDNSERR_PREDATAを設定し、リターン値として-1（エラー）を返します。

update、upcntにより更新内容についての情報を指定します。

updateには更新内容を示す配列の先頭アドレスを指定します。upcntにはupdateの示す配列の数を指定します。

uptypeには更新内容の種類を示します。lifetimeには、更新する情報の生存時間を秒単位で指定します。

updataには更新する内容を示します。updataのデータ型はuptypeに指定した値によって異なります。

更新内容の種類ごとにuptypeに指定する値とそのときのupdataの型を次の表に示します。

| 更新内容の種類 | uptypeに指定する値 | dataの型 | 内容 |
|------------------------|------------------|-------------|---|
| 情報を追加する | DNS_UPADD(1) | T_HOSTENT型 | 指定のゾーンにupdataで指定した情報を追加する。 |
| 指定ホスト名の情報を全て削除する | DNS_UPDELNAME(2) | | 指定のゾーンからh_nameで指定したホスト公式名の情報を全て削除する。 |
| 指定ホスト名の情報から指定の情報を削除する | DNS_UPDELRR(3) | | 指定したゾーンからh_nameで指定したホスト公式名の情報のうち、h_aliases、h_addr_listに指定した情報を全て削除する。 |
| 指定ホスト名の指定タイプの情報を全て削除する | DNS_UPDELTYP(4) | T_DNSRRTYP型 | 指定したゾーンにtypeで指定したリソースレコードが存在すること。ドメイン名前とタイプが等しい必要がある。 |

更新内容の種類ごとの概要について以下に示します。

(1) uptype=DNS_UPADD（情報を追加する）

指定ゾーンにupdataの情報を追加します。lifetimeに指定した時間がこの情報の生存時間となります。

updataのパラメータのうち、h_nameには追加する情報のホスト名を指定します。h_aliasesには追加するホストの別名を指定します。h_addr_listには追加するネットワークアドレスを指定します。

h_aliasesにNULL(0)以外を指定した場合は、h_aliasesにより指定したホスト別名についての情報を指定ゾーンに全て追加します。h_addr_listにNULL(0)以外を指定した場合は、h_addr_listにより指定したネットワークアドレスの配列にある情報を全て指定ゾーンに追加します。

すでに指定ゾーンにupdataの情報が存在する場合、DNSサーバはその分の情報を無視します。

(2) uptype=DNS_UPDELNAME（指定ホスト名の情報を全て削除する）

指定ゾーンからh_nameで指定したホスト公式名の情報を全て削除します。updataのパラメータのうち、h_nameには削除する情報のホスト名を指定します。h_name以外のパラメータは指定する必要はありません。

(3) uptype=DNS_UPDELRR（指定ホスト名の情報から指定の情報を削除する）

指定ゾーンからh_nameで指定したホスト公式名の情報のうち、h_aliasesとh_addr_listにより指定した情報を削除します。

updataのパラメータのうち、h_nameには削除対象の情報のホスト名を指定します。h_aliasesには削除するホストの別名を指定します。h_addr_listには削除するネットワークアドレスを指定します。

h_aliasesにより指定したホスト別名についての情報が指定ゾーンに存在する場合は、指定ゾーンから指定

したホスト別名の情報を全て削除します。存在しない場合は何もしません。

`h_addr_list`により指定したネットワークアドレスの配列にある情報が指定ゾーンに存在する場合は、指定ゾーンから削除します。存在しない場合は何もしません。

(4) `uptype=DNS_UPDELTYP` (指定ホスト名の指定タイプの情報を全て削除)

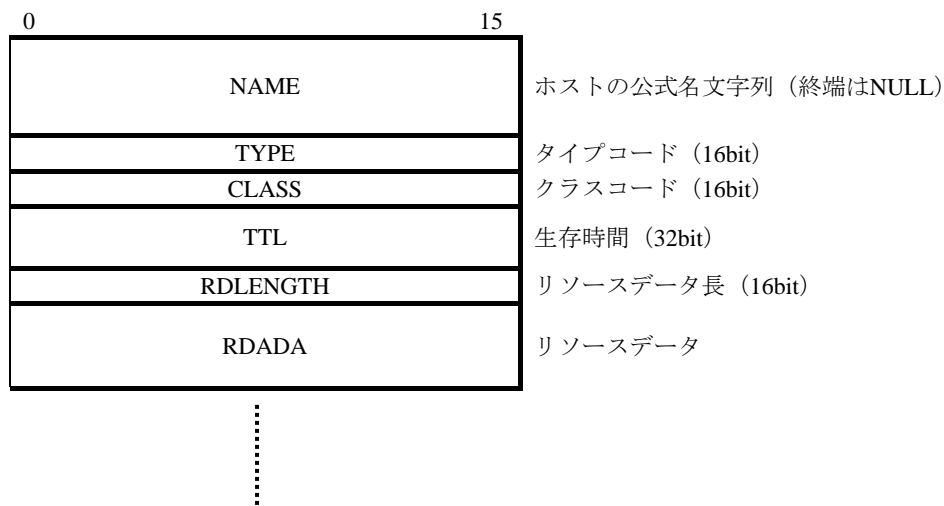
指定ゾーンから`hostname`で指定したホスト公式名の情報のうち、`rrtyp`により指定したリソースレコードタイプの情報を全て削除します。

`update`のパラメータのうち、`hostname`には削除対象の情報のホスト名を指定します。`type`には削除するリソースレコードのタイプを指定します。IPアドレスの情報を削除する場合には1を指定します。別名情報を削除する場合には5を指定します。

指定したゾーンに対象となる情報が無い場合には何もしません。

`addat`には更新要求の追加データの先頭アドレスを指定します。`addatcnt`には`addat`の示すデータの長さを指定します。`addat`に指定したデータはDNSサーバへの更新要求の追加データセクションに設定されます。`addat`に指定したデータがリソースレコードのフォーマットになっていない場合は、`errcode`に `DNSERR_ADDDATA`を設定し、リターン値として-1 (エラー) を返します。

`alldata`、および、`addat`に指定することができるリソースレコードフォーマットを以下に示します。



HI.CommunicationEngine
DNSクライアント リファレンスマニュアル
CM7000DNS01J-6

| | | | |
|------|----------------------------------|----|-----|
| 発行年月 | 2014年 | 7月 | 第6版 |
| 発行 | ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 | | |
| 編集 | ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 | | |

©ルネサスセミコンダクタパッケージ&テストソリューションズ株式会社 2014