

[White Paper]

**T-Engine Forum**  
Ubiquitous ID Center  
Specification  
**DRAFT**

910-S202-01.A0.01/UID-00005-01.A0.01

2008-10-23

---

簡易 ucode 解決プロトコル  
Simplified ucode Resolution Protocol

---

Number: 910-S202-01.A0.01/UID-00005-01.A0.01  
Title: 簡易 ucode 解決プロトコル  
Simplified ucode Resolution Protocol  
Status:  Working Draft,  Final Draft for Voting,  Standard  
Date: 2008/10/23

Copyright (C) 2008, T-Engine Forum, Ubiquitous ID Center, all rights reserved.

---

---

## 目次 (Table of Contents)

---

はじめに.....	5
規定範囲.....	5
本書の位置付け.....	5
参照規定.....	5
用語定義.....	6
1. 目的.....	7
1.1. 定義.....	7
1.2. 目的.....	7
2. ucode 解決サーバ.....	8
2.1. 概説.....	8
2.2. ucode 解決データベース.....	8
2.2.1. データモデル.....	8
2.3. ucode 解決.....	11
2.3.1. 基本動作.....	11
2.3.2. 検索アルゴリズム.....	12
2.3.3. カスケード検索.....	14
2.3.4. キャッシュ検索.....	15
2.3.5. プロキシサーバ.....	15
2.3.6. ucode 解決ゲートウェイ.....	16
2.4. コンテンツの取得.....	16
3. ucode 解決プロトコルのパケット構造及びコマンド.....	17
3.1. 概説.....	17
3.2. コマンド.....	17
3.3. パケット構造.....	17
3.4. コード定義.....	18
3.5. コマンド仕様.....	19
3.5.1. res_ucd.....	19

---

## 更新履歴

---

バージョン	更新日	更新内容
00.00.01	2006.10.17	初版発行.
01.A0.00	2008.10.17	<ul style="list-style-type: none"><li>・参照規定に RFC3986, RFC952, ucode 解決ゲートウェイを追加.</li><li>・コマンド名を res_ucd に変更.</li><li>・署名サーバの属性を追加.</li><li>・定数名 (マクロ) を変更.</li><li>・データタイプにホスト名を追加.</li><li>・ucode 解決サーバを指定するパラメタを廃止.</li><li>・コマンドパケットに Version を追加.</li><li>・コマンドパケットのリトルエンディアンを廃止して, 全てネットワークバイトオーダーに統一.</li><li>・検索アルゴリズムに attribute が一致しない場合の処理を明記.</li><li>・eTP 関連のステータスとエラーを削除.</li></ul>
01.A0.01	2008.10.23	<ul style="list-style-type: none"><li>・検索アルゴリズムを修正.</li></ul>

# 簡易 ucode 解決プロトコル

## Simplified ucode Resolution Protocol

---

### はじめに

---

#### 規定範囲

本仕様では、「モノ」や「場所」に割り当てられた ucode (ユビキタスコード) に基づいて、「モノ」や「場所」に関連付けられたコンテンツの位置情報 (ucode 解決情報) を取得するための、簡易 ucode 解決プロトコルについて規定する。また、ucode 解決情報を保持する ucode 解決サーバの内部構造について規定する。

#### 本書の位置付け

本書は簡易 ucode 解決プロトコルについての規定である。「ユビキタス ID アーキテクチャ」[1] に基づいた ucode 解決プロトコル (標準版) については [2] を参照すること。

#### 参照規定

- [1] T-Engine フォーラム, ユビキタス ID センター, 「ユビキタス ID アーキテクチャ」, 910-S002/UID-00002, 2006.
- [2] T-Engine フォーラム, ユビキタス ID センター, 「ucode 解決プロトコル (標準版)」, 910-S221/UID-00008, 2006.
- [3] T-Engine フォーラム, ユビキタス ID センター, 「ユビキタスコード: ucode」, 930-S101/UID-00010, 2006
- [4] T-Engine フォーラム, ユビキタス ID センター, 「ucode 解決ゲートウェイ」, 910-S204/UID-00007, 2006.
- [5] RFC3986, 「Uniform Resource Identifier (URI): Generic Syntax」, 2005.
- [6] RFC952, 「DOD INTERNET HOST TABLE SPECIFICATION」, 1985.

---

## 用語定義

---

- ucode  
現実世界の「モノ」や「場所」を識別することを目的とした定数名（ID）.
- ユビキタスコミュニケーター（UC）  
ucode タグとインタフェースをとり、また ucode を用いて情報サービスを受ける機能を提供するユーザ端末.

---

## 1. 目的

---

### 1.1. 定義

ucode 解決プロトコルは、現実世界の「モノ」や「場所」を識別するための ucode から、識別された「モノ」や「場所」に関連する情報の格納位置やアクセス方法を取得するためのデータベース検索プロトコルである。位置情報には、IPv4 アドレス、IPv6 アドレス、URL、ホスト名、e-mail アドレス、電話番号などが含まれる。

### 1.2. 目的

ユビキタス ID アーキテクチャでは、ucode タグに書き込まれた ucode により「モノ」や「場所」の識別が可能になるが、識別された「モノ」や「場所」に関する情報（例えば「モノの利用方法」や「その場所の地図」などのコンテンツ）を参照したいという要求がある。しかし、一般的に ucode タグは十分な量のメモリ容量を持たないため、コンテンツはタグの外部（例えば十分な容量を持ったネットワーク上のサーバ）に格納することが想定される。このため、ucode 解決サーバが、ucode とコンテンツの格納位置の対応付けデータベース（ucode 解決データベース）を管理し、ucode からコンテンツ位置を解決できる仕組みを提供する。このときの、ucode 解決サーバへのプロトコルとして、ucode 解決プロトコル（ucodeRP: ucode Resolution Protocol）を規定する。

利用者は、ucodeRP を用いて ucode 解決サーバにアクセスし、タグの外部に置かれたコンテンツの位置情報やアクセス方法を取得した後に、得られた情報に基づいて実際のコンテンツにアクセスできる。

## 2. ucode 解決サーバ

### 2.1. 概説

ucode 解決サーバは、それぞれの ucode とコンテンツ位置の対応付け情報を管理する、ユビキタス ID アーキテクチャの基盤ノードの 1 つである。あらゆる「モノ」や「場所」を識別するという ucode の特性から、ucode 解決サーバは膨大な数の ucode を管理する必要がある。従って、ucode 解決サーバは、複数のサーバが広範囲に分散して ucode を管理することで、ucode の爆発的な増加にも対応できる仕組みとなっている。ucode 解決サーバはツリー型に多段に階層化され、上位のサーバは国や国際標準化団体が、下位のサーバは企業や個人が運用することを想定している。

なお、本仕様書では、ucode とコンテンツ位置の対応付け情報を、ucode 解決情報と呼ぶ。ucode 解決情報は ucode 解決データベースに格納される。

### 2.2. ucode 解決データベース

#### 2.2.1. データモデル

ucode 解決情報を管理する ucode 解決データベースは、「データファイル」と「データエントリ」の 2 つの要素から構成される。

分散配置された ucode 解決サーバには、それぞれ管理すべき ucode 領域が割り当てられる。各 ucode 解決サーバは、管理すべき ucode 領域を示す「データファイル」を持っており、具体的な ucode 解決情報はデータファイル内の「データエントリ」と呼ばれるレコードに登録する(図 1)。データファイルには 1 つ以上のデータエントリが登録され、各データエントリの ucode 解決情報は必ず所属データファイルの ucode 領域のものでなくてはならない。

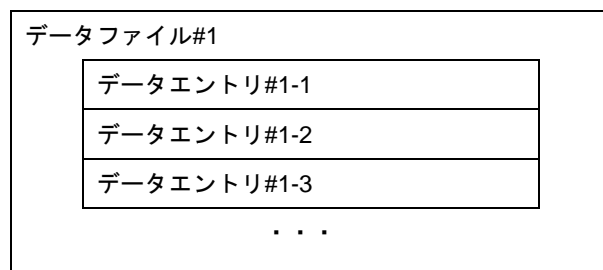


図 1: データファイルとデータエントリ間の関係

ucode 解決データベースでは、広大な ucode 空間を、「データファイル」をツリー型に階層化していくことで分散管理する(図 2)。例えば、上位階層には Top Level Domain(TLD)などのより広い ucode 領域を管理するデータファイルが、下位階層には Class Code(CC), Second Level Domain(SLD), Identification Code(IC)などのより絞られた ucode 領域を管理するデータファイルが配置される。各 ucode 解決サーバは 1 つ以上のデータファイルを管理することが可能である。

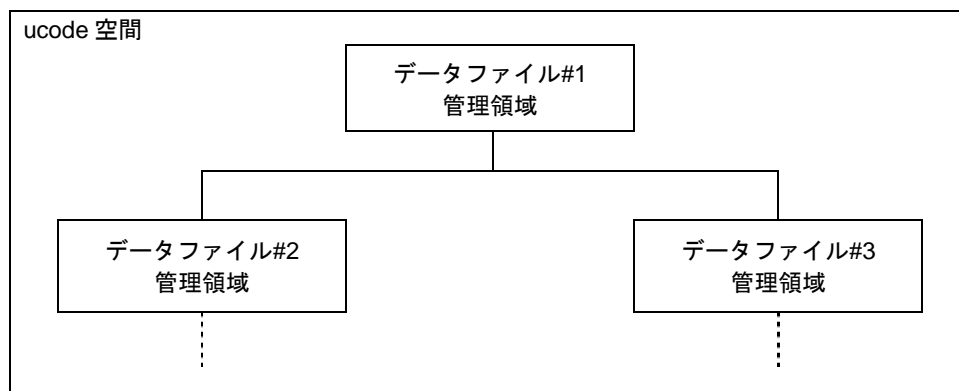


図 2 : データファイルの階層化

表 1 に、ucode 解決サーバが管理するデータファイルの例を示す。この例では、2 つのデータファイルを管理している。また、表 1-(2)のデータファイル内に登録されたデータエントリを表 2 に示す。この例では、3 つのデータエントリが登録されている。

表 1 : データファイルの例

dbucode	dbmask	cascade_mode
(1) 0efffec000000000 000000000040000	fffffffffffffffff ffffffffffffff0000	UIDC_NOGSC
(2) 0efffec000000000 000000000050000	fffffffffffffffff ffffffffffffff0000	UIDC_CSC

※ 1 レコードが 1 データファイルを示す。

表 2 : データエントリの例 (表 1-(2)のデータファイルに登録されたエントリ)

ucode	ucodemask	attribute	ver	ttd	datatype	data
(1) 0efffec000000000 000000000050100	fffffffffffffffff ffffffffffffff00	UIDC_ATTR _IS	1	0	UIDC_DATA TYPE_URL	http://www.ui dcenter.org/g etinfo.cgi
(2) 0efffec000000000 000000000050200	fffffffffffffffff ffffffffffffff00	UIDC_ATTR _RS	1	3600	UIDC_DATA TYPE_IPV4	192.168.10.1
(3) 0efffec000000000 000000000050300	fffffffffffffffff ffffffffffffff00	UIDC_ATTR _RS	2	0	UIDC_DATA TYPE_IPV4	192.168.10.2

※ 1 レコードが 1 データエントリを示す。

データファイルでは、dbucode と dbmask の 2 つの値により管理すべき ucode 範囲を指定する。dbucode & dbmask(ビット積)が ucode の範囲を示す値となる。例えば、表 1-(1)のデータファイルは、0efffec00…0040000～0efffec00…004ffff の範囲の ucode を管理することを示している。各データファイルは、dbucode & dbmask をキーにして管理される。

データエントリでは、同様に、ucode と ucodemask の 2 つの値により、エントリが指し示す ucode 範囲を指定する。ucode & ucodemask が、ucode の範囲を示す値となる。例えば、表 2-(2)のデータエントリは、0efffec00…0050200～0efffec00…00502ff の範囲の ucode を指し示すことを表わしている。各データエントリは、ucode & ucodemask をキーにして管理される。また、attribute～data までの各属性には、キーが示す ucode 範囲に対する ucode 解決情報が登録されている。このように、データエントリでは、1 つ 1 つの ucode ではなく、ucode 範囲に対して ucode 解決情報を登録する。

表 3 と表 4 にデータファイルの属性と属性値の定義を、表 5 と表 6 にデータエントリの属性と属性値の定義を示す。

表 3：データファイル属性一覧

データファイル属性	意味
dbucode	データファイルが管理する ucode
dbmask	dbucode のうち有効ビットを示すビットマスク
cascade_mode	サポートするカスケード接続のモード

表 4：データファイル属性値定義

データファイル属性	属性値	定数名	意味
cascade_mode	0x01	UIDC_NOGSC	カスケード検索機能を提供しない
	0x02	UIDC_CSC	ucode 解決サーバへのカスケード検索機能を提供する

表 5：データエントリ属性一覧

データエントリ属性	意味
ucode	データエントリが示す ucode
ucodemask	ucode のうち有効ビットを示すビットマスク
attribute	data が指し示すノードの種別
version	データエントリのバージョン
ttd	データエントリのキャッシュ有効期限 (秒)
datatype	data の表現形式
data	解決要求 ucode に対応する位置情報

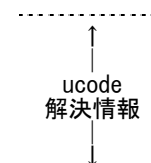


表 6：データエントリ属性値定義

データエントリ属性	属性値	定数名	意味
attribute	0x0001	UIDC_ATTR_RS	ucode 解決サーバ
	0x0002	UIDC_ATTR_IS	情報サーバ
	0x0003	UIDC_ATTR_SIGS	署名サーバ
	0x00ff	UIDC_ATTR_USER	ユーザ定義
datatype	0x0001	UIDC_DATATYPE_UCODE_128	128bit 型 ucode
	0x0002	UIDC_DATATYPE_UCODE_256	256bit 型 ucode
	0x0003	UIDC_DATATYPE_UCODE_384	384bit 型 ucode
	0x0004	UIDC_DATATYPE_UCODE_512	512bit 型 ucode
	0x0011	UIDC_DATATYPE_IPV4	IPv4 アドレス
	0x0012	UIDC_DATATYPE_IPV6	IPv6 アドレス
	0x0013	UIDC_DATATYPE_URL	URL
	0x0014	UIDC_DATATYPE_HOST	HOST
	0x0021	UIDC_DATATYPE_EMAIL	e-mail アドレス
	0x0031	UIDC_DATATYPE_PHONE	電話番号
	0x00fe	UIDC_DATATYPE_TXT	テキスト情報
	0x00ff	UIDC_DATATYPE_USER	ユーザ定義型

データエントリに登録された ucode 解決情報は、ucode とコンテンツ位置を対応付ける情報であり、典型的にはコンテンツが格納されているサーバのアドレスが登録される。本仕様では、このサーバを情報サーバと呼ぶ。

データファイルは多段階に階層化されているため、データエントリに直接情報サーバアドレスが登録されるとは限らない。上位層のデータファイルには、次に検索すべき下位階層のデータファイルを管理している ucode 解決サーバアドレスが登録される。

このように、ucode 解決データベースは、ビットマスクを利用することで任意のビット境界による多段階の階層化が可能で、特定のビットに固定的な情報を割り当てない動的な構造となっている。また、ucode の割り当て構造とも論理的に独立している。

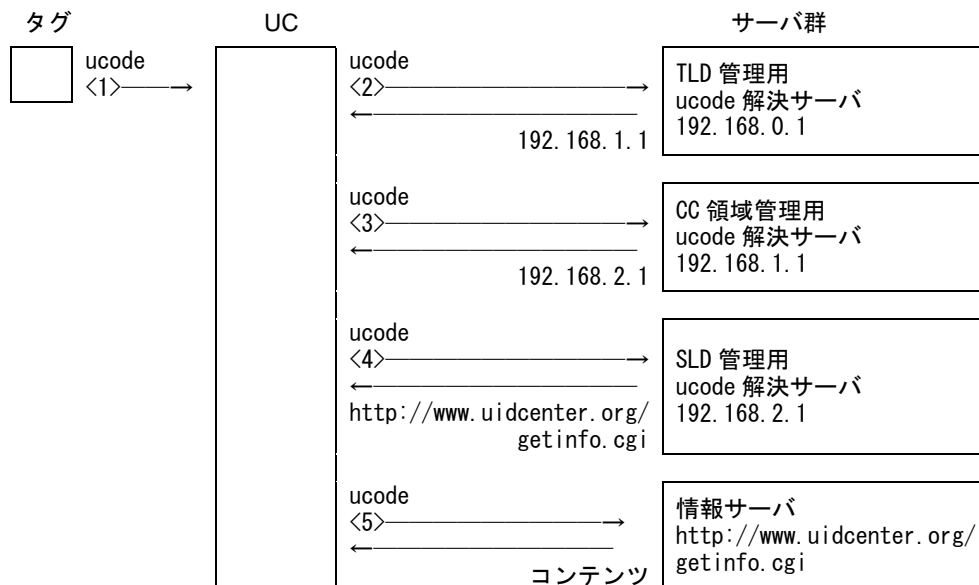
## 2.3. ucode 解決

### 2.3.1. 基本動作

ユビキタス ID アーキテクチャでは、ucode を取得したユビキタスコミュニケーター (UC) が、ucode 解決データベースを検索して情報サーバアドレスを取得し、コンテンツへのアクセスを行う。従って、ネットワークを介して ucode 解決データベースを検索できる機能が必要となる。本節では、ucode をキーにした ucode 解決データベースの検索メカニズムについて述べる。

ucode 解決の基本的動作を図 3 に示す。ucode 解決サーバが多段階に階層化されている場合には、UC は ucode 解決サーバ以外のアドレスが見つかるまで再帰的

に ucode 解決サーバへの問い合わせを行う。図 3 には、ucode 領域が TLD, CC 領域, SLD の三段に階層化されているときの動作が示されている。



- <1> タグから ucode を読み出す。
- <2> TLD 管理用 ucode 解決サーバへ ucode を送信し、CC 領域管理用 ucode 解決サーバの位置情報(192.168.1.1)を取得する。
- <3> CC 領域管理用 ucode 解決サーバへ ucode を送信し、SLD 管理用 ucode 解決サーバの位置情報(192.168.2.1)を取得する。
- <4> SLD 管理用 ucode 解決サーバへ ucode を送信し、情報サーバの位置情報 (<http://www.uidcenter.org/getinfo.cgi>)を取得する。
- <5> 情報サーバへ ucode を送信し、コンテンツを取得する。

図 3 : ucode 解決の基本動作例。<2>から<4>までが ucodeRP 仕様として定義される。

ucode 解決サーバは、受信した ucode と保持しているデータファイルを照合する。データファイルが管理する ucode 範囲に含まれていた場合には続いてデータエントリの検索を行い、検索結果を UC に返信する。UC は、返信された ucode 解決情報が ucode 解決サーバアドレスの場合には、次の ucode 解決サーバへ ucode 解決要求を送信する(<3>, <4>)。また、情報サーバアドレスの場合には、情報サーバからコンテンツを取得する(<5>)。このうち、<2>~<4>のアクセスプロトコルが、ucodeRP 仕様として定義される。

### 2.3.2. 検索アルゴリズム

ucode 解決サーバが UC から ucode 解決要求を受信したときの、データファイル及びデータエントリの検索アルゴリズムについて述べる。

UC は、表 7 に示す 3 つのパラメタを ucode 解決要求として ucode 解決サーバに

送信する。データファイルやデータエントリと同様に、検索要求も ucode とマスクビットを指定する。

表 7 : ucode 解決要求パラメタ

パラメタ名	意味
queryucode	解決対象の ucode
querymask	queryucode のうち有効ビットを示すビットマスク
queryattribute	獲得したい ucode 解決情報の属性

ucode 解決サーバは、ucode 解決要求を受信すると、まず対象となるデータファイルを検索し、次に検索されたデータファイル内のデータエントリの検索を行う。このときの検索条件を以下に示す。

1. データファイルの検索条件
  - ・ (queryucode & querymask & dbmask) が (dbucode & querymask & dbmask) と等しくなるデータファイル。
2. データエントリの検索条件（以下の条件を全て満たすこと）
  - ・ (queryucode & querymask & ucodemask) が (ucode & querymask & ucodemask) と等しくなり、かつ (querymask & ucodemask) が ucodemask と等しくなるデータエントリ。
  - ・ queryattribute が attribute と等しいデータエントリ。queryattribute が attribute と等しいエントリがない場合は、attribute が UIDC\_ATTR\_RS と等しいエントリ。ただし、queryattribute が UIDC\_ATTR\_ANONYMOUS である場合にはこの条件を満たす必要はない。

データファイルの検索に失敗した場合には、その時点でサーバは検索失敗を返信する。データファイルの検索に成功したが、データファイル内のデータエントリの検索に失敗した場合にも、検索失敗となる。

データエントリの検索に成功すると、ucode 解決サーバは表 8 に示すパラメタを含む ucode 解決情報をクライアントに返信する。

表 8 : ucode 解決情報のパラメタ

パラメタ名	意味
dataattribute	検索された位置情報の属性
dataversion	ucode 解決情報のバージョン
ttl	ucode 解決情報の有効期間（秒）
datatype	検索された位置情報の種類と形式
data	検索された位置情報
returnmask	解決に使用したビットを示すビットマスク値
resolvemode	実施された ucode 解決データベースの検索モード

ucode 解決情報には、データエントリが持つ ucode 解決情報に加えて、解決に使用したビットを示す returnmask と、実施された検索モードを示す resolvemode が含

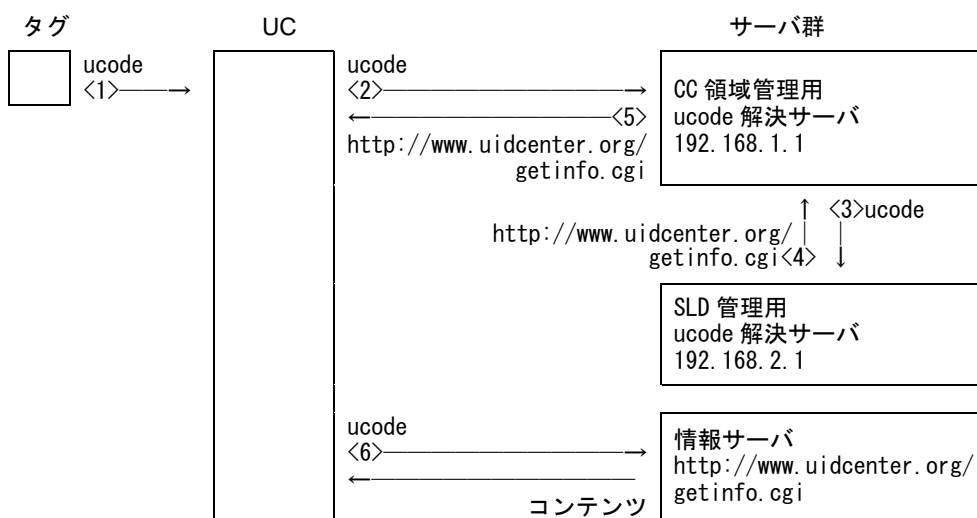
まれる。

また、検索に失敗した場合の挙動として、サーバは検索が成功するであろうサーバ(上位サーバ等)のアドレスを検索結果として返信してもよい。複数のデータエントリが検索された場合の選択基準については、本仕様では規定しない。

### 2.3.3. カスケード検索

基本動作に加えて、ucode 解決サーバはカスケード検索機能をサポートしてもよい。カスケード検索は、ucode 解決サーバが数珠つなぎに ucode 解決データベースの検索を行う方式で、処理速度やネットワークが貧弱なクライアントに対する負荷を軽減することが可能である。本節では、ucode 解決サーバへのカスケード検索について規定する。

CC 領域管理用 ucode 解決サーバがカスケード検索する場合の動作例を図 4 に示す。カスケード検索では、UC は一回の ucode 解決要求で、情報サーバアドレスの獲得が可能である。



- <1> タグから ucode を読み出す。
- <2> CC 領域管理用 ucode 解決サーバへ ucode を送信する。CC 領域管理用 ucode 解決サーバは受信 ucode を検索した結果、SLD 管理用 ucode 解決サーバアドレス(192.168.2.1)を取得する。
- <3> CC 領域管理用 ucode 解決サーバは、SLD 管理用 ucode 解決サーバへ ucode を送信する。SLD 管理用 ucode 解決サーバは受信した ucode を検索した結果、情報サーバの位置情報 (http://www.uidcenter.org/getinfo.cgi) を取得する。
- <4> SLD 管理用 ucode 解決サーバは、検索結果 (http://www.uidcenter.org/getinfo.cgi) を CC 領域管理用 ucode 解決サーバに返信する。
- <5> CC 領域管理用 ucode 解決サーバは、検索結果 (http://www.uidcenter.org/getinfo.cgi) を UC に返信する。
- <6> 情報サーバへ ucode を送信し、コンテンツを取得する。

図 4 : カスケード検索例。<2>から<5>までが ucodeRP として定義される。

CC 領域管理用 ucode 解決サーバは、次に問い合わせるべき SLD 管理用 ucode 解決サーバに検索要求を直接送信する(<3>)。このとき、CC 領域管理用 ucode 解決サーバは、SLD 管理用 ucode 解決サーバに対しては単なる解決クライアントとして振舞う。なお、図 6 は、ucode 解決サーバが2段にカスケードされた場合の構成例であるが、一般的にはn段(nは1以上の自然数)のカスケードが可能である。

ucode 解決サーバは、以下の条件を全て満たした場合にカスケード検索を実行する。

1. 検索要求でカスケード接続が要求されている。
2. 検索対象となったデータファイルがカスケード接続機能をサポートしている。すなわち、`cascade_mode` の値が `UIDC_GSC` である。
3. 検索結果として得られたアドレスが、ucode 解決サーバのアドレスである。

カスケード検索先からエラー応答が返信された場合は、エラー応答を UC にそのまま転送してもよいし、カスケード検索をしなかったものとして通常の結果を返信してもよい。

なお、情報サーバへのカスケード接続については、本仕様では規定しない。

#### 2.3.4. キャッシュ検索

ucode 解決サーバや UC は、検索の効率化のために検索結果として得られた ucode 解決情報をキャッシュしてもよい。また、キャッシュサーバ等のキャッシュ専用ノードを設置してもよい。

ただし、キャッシュにより応答を行う場合には、キャッシュ応答であることを明示しなくてはならない。また、各データエントリの `ttl` フィールドは、エントリのキャッシュ許可期間を示しており、この期間を過ぎたキャッシュを利用してはならない。なお、`ttl` が 0 のエントリはキャッシュ不可であることを示す。

クライアントはキャッシュによる応答を許可するかどうかを指定できる。ucode 解決サーバはキャッシュ応答不可の検索要求に対してキャッシュ応答を行ってはならない。

#### 2.3.5. プロキシサーバ

UC のネットワークアドレス等の秘匿のために、プロキシサーバを経由して ucode 解決サーバへアクセスしてもよい。プロキシサーバについては、本仕様では規定しない。

### 2.3.6. ucode 解決ゲートウェイ

処理速度やネットワークが貧弱なクライアントに対する負荷軽減のために、ucode 解決ゲートウェイを設置してもよい。ucode 解決ゲートウェイは、クライアントから解決要求を受け取り、ucode 解決サーバや情報サーバに対して一連の検索処理を行った後に、最終検索結果をクライアントに返す。ucode 解決ゲートウェイの機能やアクセスプロトコルについては[4]で規定する。

## 2.4. コンテンツの取得

UC は、ucode 解決により取得した情報サーバアドレスに基づいて、コンテンツへのアクセスを行う。コンテンツへのアクセス方式については本仕様では規定しない。ただし、UC は ucode 解決情報からアクセスプロトコルを判別できる場合がある。例えば、URL であればスキーム情報("http://"など)により判別が可能である。

多くの場合、コンテンツは ucode に応じて変化すると予想されることから、CGI の引き数などで ucode を渡すことが考えられる。例えば、UC は以下のような HTTP 要求を情報サーバに送信する。

```
http://www.uidcenter.org/getinfo.cgi?ucode=0efffec0000000000000000000000050123
```

### 3. ucode 解決プロトコルのパケット構造及びコマンド

#### 3.1. 概説

ucodeRP は, ucode 解決データベースへのアクセス機能を提供するプロトコルである. UC は, ucodeRP を利用して ucode 解決サーバへのアクセスを行い, ucode 解決情報を取得することが可能である. ucodeRP はアプリケーションレイヤのプロトコルであり, 下位層のプロトコルに依存しない.

#### 3.2. コマンド

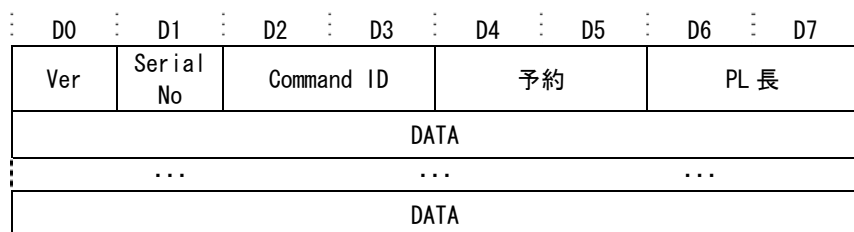
本仕様書で規定する ucodeRP コマンド一覧を表9に示す.

表 9 : ucodeRP コマンド一覧

コマンド名	機能
res_ucd	ucode 解決データベースの検索を行う

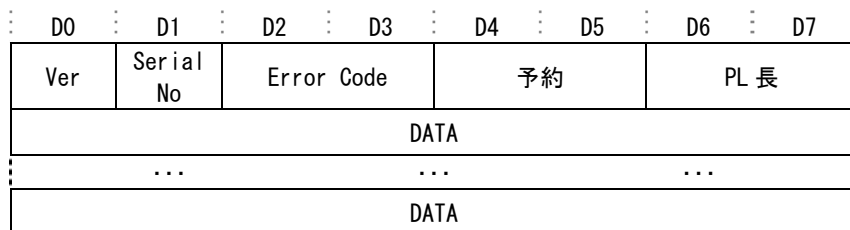
#### 3.3. パケット構造

ucodeRP の送りパケット構造定義を図5に, 戻りパケット構造定義を図6に示す. すべてのパラメタはネットワークバイトオーダで格納される.



フィールド	意味
Ver	ucodeRP プロトコルのバージョン. 1 固定.
Serial No	ucodeRP 要求の識別番号. 送信側が自由に設定できる.
Command ID	ucodeRP コマンドコード.
PL 長	ucodeRP パケット全体の長さ(8 オクテットブロック数). 例えば, 48 オクテットであれば, PL 長=0x0006 となる.
DATA	コマンド毎に規定されるコマンドパラメタ.
予約	0 固定とする.

図 5 : 送り ucodeRP パケット



フィールド	意味
Ver	ucodeRP プロトコルのバージョン. 1 固定.
Serial No	ucodeRP 応答の識別番号. 対応する ucodeRP 要求の Serial No + 1 の値としなくてはならない.
Error Code	ucodeRP コマンドの終了ステータス.
PL 長	ucodeRP パケット全体の長さ (8 オクテットブロック数). 例えば, 48 オクテットであれば, PL 長=0x0006 となる.
DATA	コマンド毎に規定されるレスポンスパラメタ.
予約	0 固定とする.

図 6 : 戻り ucodeRP パケット

### 3.4. コード定義

#### (1) コマンドコード

ucodeRP コマンドコードの一覧を表 10 に示す.

表 10 : ucodeRP コマンドコード一覧

コマンドコード	コマンド名
0x0001	res_ucd

#### (2) エラーコード

ucodeRP エラーコード一覧を表 11 に示す.

表 11 : エラーコード一覧

状態	値	定数名	意味
正常終了	0x0000	E_UIDC_OK	解決成功
異常終了	0xffef	E_UIDC_NOSPT	未サポート機能
	0xffdf	E_UIDC_PAR	パラメタエラー
	0xffcc	E_UIDC_NOEXS	エントリが存在しない

※ resolvemode==UIDC\_RSMODE\_CASCADE の場合には, カスケード先が返したエラーコードとなる.

### 3.5. コマンド仕様

#### 3.5.1. res\_ucd

##### (1) 機能概要

指定された ucode をキーに ucode 解決データベースを検索し、ucode 解決情報を提供する。

##### (2) 送りパラメタ

res\_ucd コマンドの送りパケット構造定義を図 7 に、パラメタ一覧を表 12 に、パラメタ値定義を表 13 に示す。

D0	D1	D2	D3	D4	D5	D6	D7
Ver	Serial No	Command ID		予約		PL 長	
t				予約			
querymode		queryattribute		ucodetype		ucodelength	
queryuicode (0)							
queryuicode (1)							
...		...		...		...	
querymask (0)							
querymask (1)							
...		...		...		...	

図 7 : res\_ucd コマンド送りパケット

表 12 : 送りパラメタ一覧

フィールド	意味
t	コマンドの送信時刻 (2000 年 1 月 1 日 AM 0:00 からの積算秒数 (グリニッジ標準時))
querymode	ucode 解決データベースの検索モード
queryattribute	検索したいデータの属性
ucodetype	検索する ucode の型
ucodelength	queryuicode/querymask の長さ合計 (byte)
queryuicode	検索する ucode
querymask	ucode のうち有効ビットを示すビットマスク値
予約	0 固定

表 13 : 送りパラメタ値定義

パラメタ	値	定数名	意味
querymode (論理和で指定)	0x0000	UIDC_RSMODE_RESOLUTION	ucode 解決を指示
	0x0001	UIDC_RSMODE_CACHE	キャッシュ検索を許可
	0x0002	UIDC_RSMODE_CASCADE	カスケード検索を許可
queryattribute	0x0000	UIDC_ATTR_ANONYMOUS	属性を指定しない
	0x0001	UIDC_ATTR_RS	ucode 解決サーバ
	0x0002	UIDC_ATTR_IS	情報サーバ
	0x0003	UIDC_ATTR_SIGS	署名サーバ
	0x00ff	UIDC_ATTR_USER	ユーザ定義
ucodetype	0x0001	UIDC_DATATYPE_UCODE_128	128bit 型 ucode
	0x0002	UIDC_DATATYPE_UCODE_256	256bit 型 ucode
	0x0003	UIDC_DATATYPE_UCODE_384	384bit 型 ucode
	0x0004	UIDC_DATATYPE_UCODE_512	512bit 型 ucode
queryucode	<ucode binary>	-	ucode のバイナリ値
querymask	<ucode binary>	-	ビットマスクのバイナリ値

<ucode binary> ::= 16(%x00-ff)<sup>※1※2</sup>

※1 : n<element>は, n 個の<element>が並ぶことを示す.

※2 : %x は, 16 進の数値を示す.

### (3) 戻りパラメタ

res\_ucd コマンドの戻りパケット構造定義を図 8 と図 9 に, パラメタ一覧を表 14 に, パラメタ値定義を表 15 に示す.

D0	D1	D2	D3	D4	D5	D6	D7
Ver	Serial No	Error Code		予約		PL 長	
ttl				予約		dataversion	
resolvemode		dataattribute		datatype		datalength	
data (0)							
...				...			
data (N)						0 padding	
予約						masklength	
returnmask (0)							
returnmask (1)							
...				...			

図 8 : res\_ucd コマンド戻りパケット (正常時)

D0	D1	D2	D3	D4	D5	D6	D7
Ver	Serial No	Error Code	予約			PL 長	

図 9 : res\_ucd コマンド戻りパケット (エラー時)

表 14 : 戻りパラメータ一覧

フィールド	意味
ttl	取り出されたデータの有効期間 (秒)
dataversion	検索データのバージョン
resolvemode	実施されたデータベースの検索モード
dataattribute	検索されたデータの属性
datatype	検索されたデータの種類と形式
datalength	検索されたデータのデータ長
data	検索されたデータ
masklength	returnmask のバイト数
returnmask	解決されたデータのビットマスク値
予約	0 固定

表 15 : 戻りパラメータ値定義

パラメータ	値	定数名	意味
resolvemode (論理和で指定)	0x0000	UIDC_RSMODE_RESOLUTION	通常の ucode 解決応答
	0x0001	UIDC_RSMODE_CACHE	キャッシュ検索による応答
	0x0002	UIDC_RSMODE_CASCADE	カスケード検索による応答
dataattribute	0x0001	UIDC_ATTR_RS	ucode 解決サーバアドレス
	0x0002	UIDC_ATTR_IS	情報サーバ
	0x0003	UIDC_ATTR_SIGS	署名サーバ
	0x00ff	UIDC_ATTR_USER	ユーザ定義
datatype	0x0001	UIDC_DATATYPE_UCODE_128	128bit 型 ucode
	0x0002	UIDC_DATATYPE_UCODE_256	256bit 型 ucode
	0x0003	UIDC_DATATYPE_UCODE_384	384bit 型 ucode
	0x0004	UIDC_DATATYPE_UCODE_512	512bit 型 ucode
	0x0011	UIDC_DATATYPE_IPV4	IPv4 アドレス
	0x0012	UIDC_DATATYPE_IPV6	IPv6 アドレス
	0x0013	UIDC_DATATYPE_URL	URL
	0x0014	UIDC_DATATYPE_HOST	ホスト名
	0x0021	UIDC_DATATYPE_EMAIL	e-mail アドレス
	0x0031	UIDC_DATATYPE_PHONE	電話番号
	0x00fe	UIDC_DATATYPE_TXT	テキスト情報
0x00ff	UIDC_DATATYPE_USER	ユーザ定義型	
data	<data value>	-	検索されたデータの実体

```
<data value> ::= <ucode binary> | ; for UIDC_DATATYPE_UCODE_*
                <IPv4 address binary> | ; for UIDC_DATATYPE_IPV4
                <IPv6 address binary> | ; for UIDC_DATATYPE_IPV6
                <URL の ASCII 表記※1> | ; for UIDC_DATATYPE_URL
                <host の ASCII 表記※2> | ; for UIDC_DATATYPE_HOST
                *CHAR ; for other types
<IPv6 address binary> ::= 16(%x00-ff)
CHAR = <US-ASCII 文字>

※1 : [5]を参照
※2 : [6]を参照
```

## 付録 A. ucode 解決サーバの構築例及び動作例

### A.1. ucode 解決サーバの構築例

ucode を, TLD(上位 20bit), Class Code(CC)(上位 24bit), SLD(上位 112bit)の 3 階層のデータファイルで管理し, 各データファイルを別々の ucode 解決サーバで管理したときの構成例を示す. サーバ群の階層構造は以下ようになる.

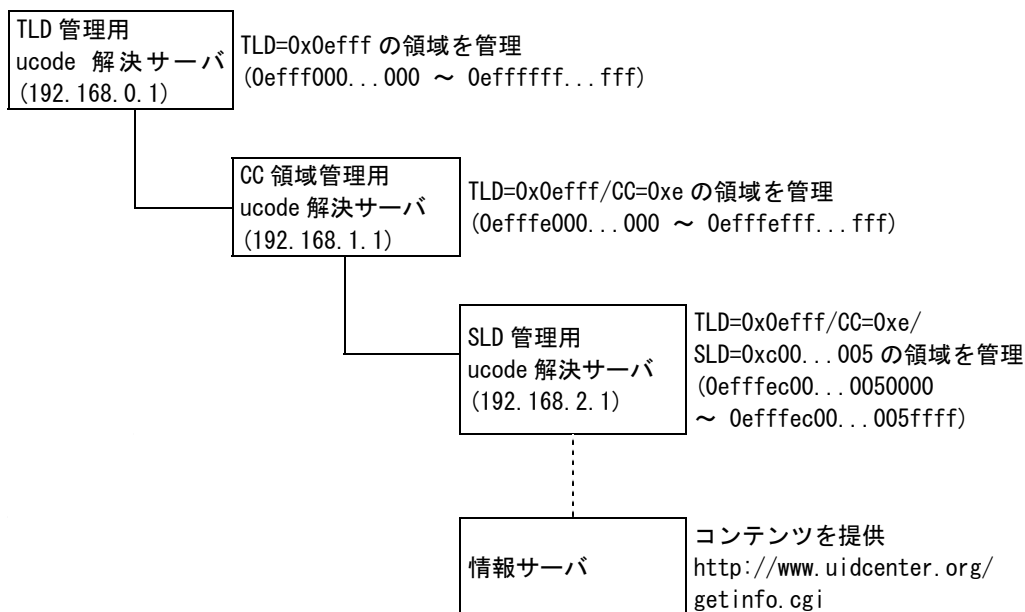


図 10 : サーバ群の階層構造

#### (1) TLD 管理用 ucode 解決サーバ

TLD 管理用 ucode 解決サーバが管理するデータファイルと, それに含まれるデータエントリの構築例を以下に示す. このデータファイルは, 0efff000...000 ~ 0effffff...fff の ucode 範囲を管理している. また, データエントリでは, 0efffe000...000 ~ 0efffefff...fff の ucode 範囲に対して, 下位階層の ucode 解決サーバを指し示す IPv4 アドレス "192.168.1.1" を対応付けている.

データファイル

```

dbucode      = 0efff0000000000000000000000000000
dbmask       = fffff00000000000000000000000000000
cascade_mode = UIDC_NOCSG
    
```

データエントリ

ucode	=	0efffe00000000000000000000000000
ucodemask	=	ffffff00000000000000000000000000
attribute	=	0x0001 (UIDC_ATTR_RS)
version	=	0x0001
ttl	=	0x00015180
datatype	=	0x0011 (UIDC_DATATYPE_IPV4)
data	=	0xC0A80101 // "192.168.1.1"の Binary 表記

(2) CC 領域管理用 ucode 解決サーバ

CC 領域管理用 ucode 解決サーバが管理するデータファイルと、それに含まれるデータエントリの構築例を以下に示す。このデータファイルは、0efffe000…0000～0efffefff…ffff の ucode 範囲を管理している。また、データエントリでは、0efffec00…0050000～0efffec00…005ffff の ucode 範囲に対して、下位階層の ucode 解決サーバを指し示す IPv4 アドレス“192.168.2.1”を対応付けている。

データファイル

dbucode	=	0efffe00000000000000000000000000
dbmask	=	ffffff00000000000000000000000000
cascade_mode	=	UIDC_NOCS

データエントリ

ucode	=	0efffec00000000000000000000050000
ucodemask	=	ffffffffffffffffffffffffffff0000
attribute	=	0x0001 (UIDC_ATTR_RS)
version	=	0x0001
ttl	=	0x00015180
datatype	=	0x0011 (UIDC_DATATYPE_IPV4)
data	=	0xC0A80201 // "192.168.2.1"の Binary 表記

(3) SLD 管理用 ucode 解決サーバ

SLD 管理用 ucode 解決サーバが管理するデータファイルと、それに含まれるデータエントリの構築例を以下に示す。このデータファイルは、0efffec00…0050000～0efffec00…005ffff の ucode 範囲を管理している。また、データエントリでは、0efffec00…0050100～0efffec00…00501ff の ucode 範囲に対して、情報サーバを指し示す URL “http://www.uidcenter.org/getinfo.cgi”を対応付けている。

データファイル

dbucode	=	0efffec00000000000000000000050000
dbmask	=	ffffffffffffffffffffffffffff0000
cascade	=	UIDC_NOCS





検索結果

attribute	=	0x0002 (UIDC_ATTR_IS)
version	=	0x0001
ttd	=	0x00015180
datatype	=	0x0013 (UIDC_DATATYPE_URL)
data	=	"http://www.uidcenter.org/getinfo.cgi"
returnmask	=	ffffffffffffffffffffffffffff00
resolvmode	=	0x0000 (UIDC_RSMODE_RESOLUTION)

上位 120bit を使用して ucode 解決が行われ、情報サーバの URL  
"http://www.uidcenter.org/getinfo.cgi" が検索結果として返信される。

---

## 索引

---

<b>R</b>	<b>き</b>
res_ucd.....18	キャッシュ検索 .....15
<b>U</b>	<b>て</b>
ucodeRP.....7	データエントリ .....8
ucode 解決ゲートウェイ .....16	データファイル .....8
ucode 解決情報 .....8	
ucode 解決プロトコル.....7	<b>ふ</b>
ucode 解決サーバ.....8	プロキシサーバ .....15
ucode 解決データベース .....8	
<b>か</b>	
カスケード検索 .....14	

