

[White Paper]

T-Engine Forum
Ubiquitous ID Center
Document
DRAFT

910-S202-01.A0.01/UID-00005-1.A0.01

2008-10-23

Simplified ucode Resolution Protocol

Number: 910-S202-01.A0.01/UID-00005-1.A0.010
Title: 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.

Table of Content

Introduction.....	4
Scope.....	4
Position of this Document	4
Reference Documents	4
Definition of Terms.....	4
1. Objective	6
1.1. Definition.....	6
1.2. Objective.....	6
2. ucode Resolution Server	7
2.1. ucode Resolution Server	7
2.2. ucode Resolution Database.....	7
2.2.1. Data Model.....	7
2.3. ucode Resolution.....	11
2.3.1. Basic Behavior	11
2.3.2. Search Algorithm.....	12
2.3.3. Cascade Search.....	13
2.3.4. Cache Search	15
2.3.5. Proxy Server	15
2.3.6. ucode Resolution Gateway.....	15
2.4. Retrieval Content	16
3. Packet Structure and Commands of ucode Resolution Protocol.....	17
3.1. Overview	17
3.2. Command.....	17
3.3. Packet Structure.....	17
3.4. Code Definition.....	18
3.5. Command Specification	18
3.5.1. res_ucd.....	18

Simplified ucode Resolution Protocol

Introduction

Scope

This document specifies a simplified ucode resolution protocol to retrieve the storage location or access method of information associated with “objects” or “locations” (called “ucode resolution information”) according to ucodes (ubiquitous codes) assigned to “objects” and “locations.”

Position of this Document

This document specifies a simplified ucode resolution protocol. Please refer to [2] on the “ucode resolution protocol (standard version)” based on “ubiquitous ID architecture” [1].

Reference Documents

- [1] T-Engine Forum, Ubiquitous ID Center, “Ubiquitous ID Architecture,” 910-S002/UID-00002, 2006.
- [2] T-Engine Forum, Ubiquitous ID Center, “ucode Resolution Protocol (Standard version),” 910-S221/UID-00008, 2006.
- [3] T-Engine Forum, Ubiquitous ID Center, “Ubiquitous Code: ucode,” 930-S101/UID-00010, 2006.
- [4] T-Engine Forum, Ubiquitous ID Center, “ucode Resolution Gateway,” 930-S204/UID-00007, 2006.
- [5] RFC3986, “Uniform Resource Identifier (URI) : Generic Syntax, ” 2005.
- [6] RFC952, “DOD Internet Host Table Specification, ” 1985.

Definition of Terms

- ucode
An identifier (ID) for the purpose of identifying "objects" and "locations" in the real world.
- Ubiquitous Communicator (UC)
A user terminal that communicates with ucode tags and that provides a function of information services based on ucodes.

1. Objective

1.1. Definition

The ucode resolution protocol (ucodeRP) is defined as a database search protocol for retrieving the storage location or access method of information associated with the identified 'objects' or 'locations' from the ucodes which identify 'objects' or 'locations' in the real world. This "information" includes IPv4 and IPv6 addresses, a URL, an e-mail address, a telephone number, and etc.

1.2. Objective

In the ubiquitous ID architecture, the ucode written in the ucode tag realizes the identification of 'objects' or 'locations.' The realization of the identification is not goal in itself. The large number of applications identify 'objects' or 'locations', and then they will go on to look for the information associated with the identified 'object' or 'location' (for example, the content such as 'how to use the object' or 'a map of the location'). As the ucode tag does not have a large storage capacity in general, contents are often stored outside of the tag, for example, in a server with a large storage capacity on the network. The ucode resolution server manages a mapping database to associate ucodes with storage locations of content called a ucode resolution database and provides a mechanism to resolve content locations by ucodes. The ucode resolution protocol (ucodeRP) is specified as an access protocol to the ucode resolution server.

The user can access the ucode resolution server using the ucodeRP, retrieve location information and an access method of the content stored outside the tag, and access the actual content based on the retrieved information.

2. ucode Resolution Server

2.1. ucode Resolution Server

The ucode resolution server takes an important role in the ubiquitous ID architecture, which manages association between each ucode and content location. Due to the ucode characteristic of identifying the “objects” or “locations”, the ucode resolution server needs to manage a unlimitedly large number of ucodes. The ucode resolution server consequently offers the mechanism to support the explosive increase of ucodes by multiple servers' distributed management. The ucode resolution servers have a multi-layered tree configuration, and upper servers are assumed to be operated by countries or the ISO, and lower servers by enterprises or individuals.

Note that association between the ucode and its content location is referred to as ucode resolution information in this document. The ucode resolution information is stored in a ucode resolution database.

2.2. ucode Resolution Database

2.2.1. Data Model

The ucode resolution database, which manages the ucode resolution information, consists of two elements: “data file” and its record, namely, “data entry.”

The ucode space to be each managed is allocated to the respective distributed ucode resolution servers. Each ucode resolution server has a “data file” which indicates a ucode space to be managed, and specific ucode resolution information is registered in a record called a “data entry,” in the data file (Figure 1). One or more data entries are registered in the data file, and the ucode resolution information of each data entry should be within the ucode space of the data file.

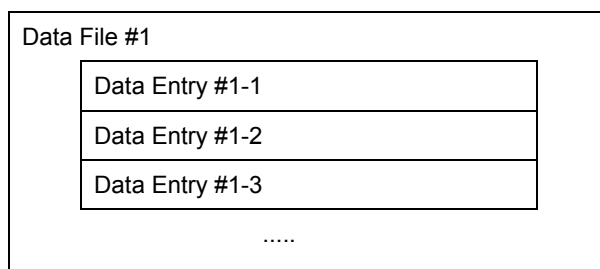


Figure 1: Relationship between Data File and Data Entry

In the ucode resolution database, the management of a large ucode space is decentralized with a hierarchical tree consisting of “data files” (Figure 2). For example, data files managing a larger ucode space such as a top level domain (TLD) are placed in the upper layer, and data files managing smaller ucode spaces such as a class code (CC), a second level domain (SLD), or identification code (IC) are placed in the lower layer. Each ucode resolution server can manage one or more data files.

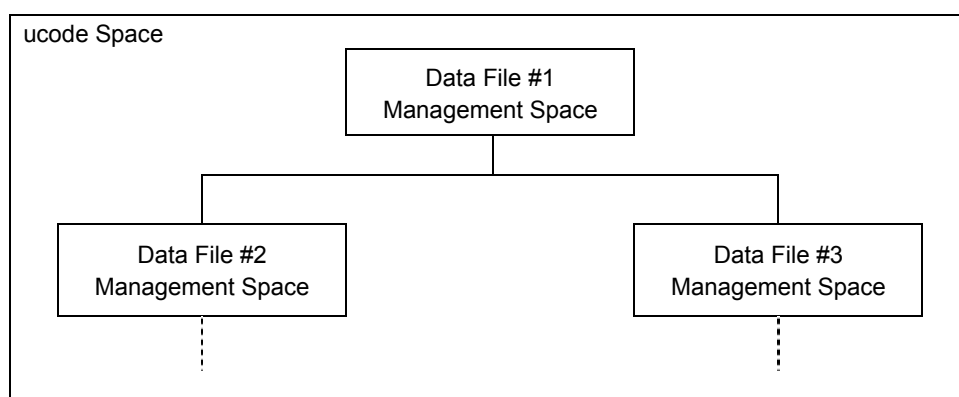


Figure 2: Layering of Data Files

Table 1 shows an example of data file which the ucode resolution server manages. In this example, the server manages two data files. In addition, Table 2 shows data entries registered within the data file shown in Table 1. In this example, three data entries are registered.

Table 1: Example of Data File

	dbucode	dbmask	cascade_mode
(1)	0efffec000000000 000000000040000	fffffffffffffffff ffffffffffff0000	UIDC_NOCS
(2)	0efffec000000000 000000000050000	fffffffffffffffff ffffffffffff0000	UIDC_CS

*1 Each record indicates one data file.

Table 2: Example of Data Entry (registered within the data file shown in Table 1)

	ucode	ucodemask	attribute	ver	tll	datatype	data
(1)	0efffec000000000 000000000050100	fffffffffffffffff fffffffffffff00	UIDC_ATTR_SS	1	0	UIDC_DATA TYPE_URL	http://www.uidcenter.org/ getinfo.cgi
(2)	0efffec000000000 000000000050200	fffffffffffffffff fffffffffffff00	UIDC_ATTR_RS	1	3600	UIDC_DATA TYPE_IPV4	192.168.10.1
(3)	0efffec000000000 000000000050300	fffffffffffffffff fffffffffffff00	UIDC_ATTR_RS	2	0	UIDC_DATA TYPE_IPV4	192.168.10.2

*2 Each record indicates one data entry.

In the data file, the ucode space to be managed is specified by two values: a dbucode and dbmask. The value of dbucode & dbmask (& hereinafter means a bitwise logical AND) indicates the ucode space. For example, the data file shown in Table 1 indicates managing the ucode with the space of 0efffec00...0040000 - 0efffec00...004ffff.

In the data entry, similarly, the ucode space which the entry points is specified by two values of the ucode and ucodemask. The value of ucode & ucodemask indicates the ucode space. For example, the data entry shown in Table 2 indicates the ucode with the space of 0efffec00...0050200 - 0efffec00...00502ff. ucode resolution information for the ucode space specified by ucode & ucodemask is registered for each attribute from "attribute" to "data." In this manner, ucode resolution information is registered in the data entry not for each ucode but for the ucode space.

Table 3 and Table 4 show the definition of a data file attribute and attribute value. Table 5 and Table 6 show the definition of a data entry attribute and attribute value.

Table 3: List of Data File Attributes

Data File Attribute	Meaning
dbucode	ucode managed by data file
dbmask	Bitmask value indicating the effective bit of dbucode
cascade_mode	Supporting mode in cascade connection

Table 4: Definition of Data File Attribute Values

Data File Attribute	Attribute Value	Macro	Meaning
cascade_mode	0x01	UIDC_NOESC	Does not provide cascade search function
	0x02	UIDC_CSC	Provides cascade search function for the ucode resolution server

Table 5: List of Data Entry Attributes

Data Entry Attribute	Meaning
ucode	ucode indicated by data entry
ucodemask	Bitmask value indicating the effective bit of ucode
attribute	Type of the node which the data points
version	Data entry version
tll	Cache expiration date of data entry (seconds)
datatype	Data display style
data	Storage location or access method of information associated with the ucode

ucode resolution information

↑

↓

Table 6: Definition of Data Entry Attribute Values

Data Entry Attribute	Attribute Value	Macro	Meaning
attribute	0x0001	UIDC_ATTR_RS	ucode resolution server address
	0x0002	UIDC_ATTR_SS	Information server address
	0x0003	UIDC_ATTR_SIGS	Signature server address
	0x00ff	UIDC_ATTR_USER	User defined address
datatype	0x0001	UIDC_DATATYPE_UCODE_128	128bit type ucode
	0x0002	UIDC_DATATYPE_UCODE_256	256bit type ucode
	0x0003	UIDC_DATATYPE_UCODE_384	384bit type ucode
	0x0004	UIDC_DATATYPE_UCODE_512	512bit type ucode
	0x0011	UIDC_DATATYPE_IPV4	IPv4 address
	0x0012	UIDC_DATATYPE_IPV6	IPv6 address
	0x0013	UIDC_DATATYPE_URL	URL
	0x0014	UIDC_DATATYPE_HOST	Hostname
	0x0021	UIDC_DATATYPE_EMAIL	E-mail address
	0x0031	UIDC_DATATYPE_PHONE	Telephone number
	0x00fe	UIDC_DATATYPE_TXT	Text information
	0x00ff	UIDC_DATATYPE_USER	User-defined type

The ucode resolution information registered in the data entry is mapping information to associate with the ucode and its content location, typically holding the address of the server where content is stored. This server is called an information server in this document.

As the data file is managed in the multilevel hierarchical structure, the information server address is not necessarily registered directly in the data entry. The ucode resolution server address which manages the data file of a lower layer to be searched next may be registered in the data file of an upper layer.

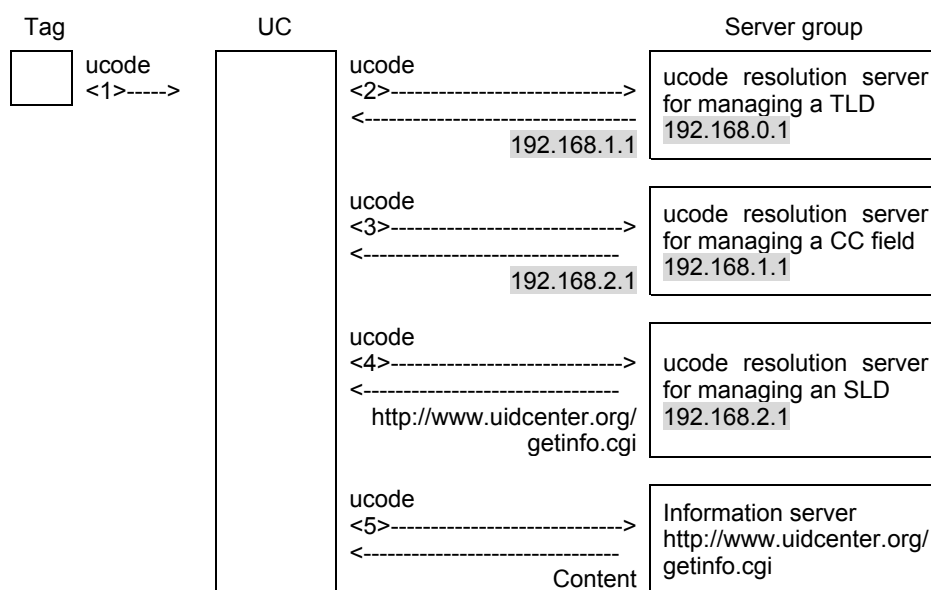
Thus, the ucode resolution database can be managed in the multilevel hierarchical structure and uses a dynamic structure which does not assign fixed information to a specific bit. In addition, the database is logically independent of the ucode allocation structure.

2.3. ucode Resolution

2.3.1. Basic Behavior

In the ubiquitous ID architecture, the ubiquitous communicator (UC) retrieves the ucode and then gets the information server address by searching the ucode resolution database, and accesses content. Therefore, the function to retrieve the ucode resolution database via network is required. This section describes a search mechanism of the ucode resolution database using the ucode as a search key.

Figure 3 shows the basic behavior of ucode resolution. If the ucode resolution server is in the multilevel hierarichical structure, UC recursively queries the ucode resolution server while the ucode resolution server address is returned. Figure 3 shows the behavior when the ucode space consists of three levels of a TLD, CC field, and SLD.



<1> UC reads a ucode from a tag.

<2> UC sends the ucode to the ucode resolution server for managing a TLD and gets the location information (192.168.1.1) of the ucode resolution server for managing a CC field.

<3> UC sends the ucode to the ucode resolution server for managing a CC field and gets the location information (192.168.2.1) of the ucode resolution server for managing an SLD.

<4> UC sends the ucode to the ucode resolution server for managing an SLD and gets the location information (http://www.uidcenter.org/getinfo.cgi) of the information server.

<5> UC sends the ucode to the information server and gets content.

Figure 3: Example of the Basic Behavior of ucode Resolution

The behavior of <2>-<4> is in accordance with the ucodeRP specification.

The ucode resolution server matches the received ucode with the holding

data file. If it is within the ucode space managed by the data file, then the server searches the data entry and returns the search result to UC. If the returned ucode resolution information is another ucode resolution server address, UC sends a ucode resolution request to the second ucode resolution server (<3>, <4>). If the returned ucode resolution information is an information server address, UC gets the content from the information server (<5>). Of these, the access protocol of <2>-<4> is specified as a ucodeRP.

2.3.2. Search Algorithm

This section describes a search algorithm for a data file and data entry when the ucode resolution server receives a ucode resolution request from UC.

UC sends three parameters shown in Table 7 as part of ucode resolution requests to the ucode resolution server. Similar to a data file and data entry, the ucode and maskbit should be specified for a search request.

Table 7: Parameters of a ucode Resolution Request

Parameter Name	Meaning
queryucode	Target ucode to be resolved
querymask	Bitmask which indicates the effective bit of a queryucode
queryattribute	ucode resolution information attribute to be acquired

When the ucode resolution server receives a ucode resolution request, first it retrieves the target data file and then retrieves the data entry in the retrieved data file. The matching criteria are as follows:

1. **Matching criteria of a data file**
 - A data file is chosen if (queryucode & querymask & dbmask) equals (dbucode & querymask & dbmask).
2. **Matching criteria of a data entry (all of the following conditions should be met.)**
 - A data entry is chosen for further matching if (queryucode & querymask & ucodemask) equals (ucode & querymask & ucodemask) and (querymask & ucodemask) equals ucodemask.
 - A data entry is chosen as a final result if the queryattribute is equal to the attribute. If there is no data entry whose attribute equals queryattribute, a data entry is chosen as a final result if the attribute equals to UIDC_ATTR_RS. However, if the queryattribute is UIDC_ATTR_ANONYMOUS, this data entry is unconditionally chosen.

The server returns a search failure if data file search fails. If data file search is successful, but data entry search produces no results in the data file, the server returns a search failure.

If data entry search is successful, the ucode resolution server returns ucode resolution information including the parameters shown in Table 8 to the client.

Table 8: Parameters of ucode Resolution Information

Parameter Name	Meaning
dataattribute	Attribute of retrieved location information
dataversion	Version of ucode resolution information
ttl	Effective period of ucode resolution information (seconds)
datatype	Type and format of retrieved location information
data	Retrieved location information
returnmask	Bitmask value which indicates the bit used for resolution
resolvemode	Search mode of retrieved ucode resolution database

ucode resolution information involves the returnmask indicative of the bit used for the resolution and the resolvemode indicative of the search mode as well as ucode resolution information in the data entry.

Furthermore, if the search fails, the server may return the address of the server (higher-level server, etc.) with which another search attempt may succeed. This document does not specify the selection criteria when multiple data entries are retrieved.

2.3.3. Cascade Search

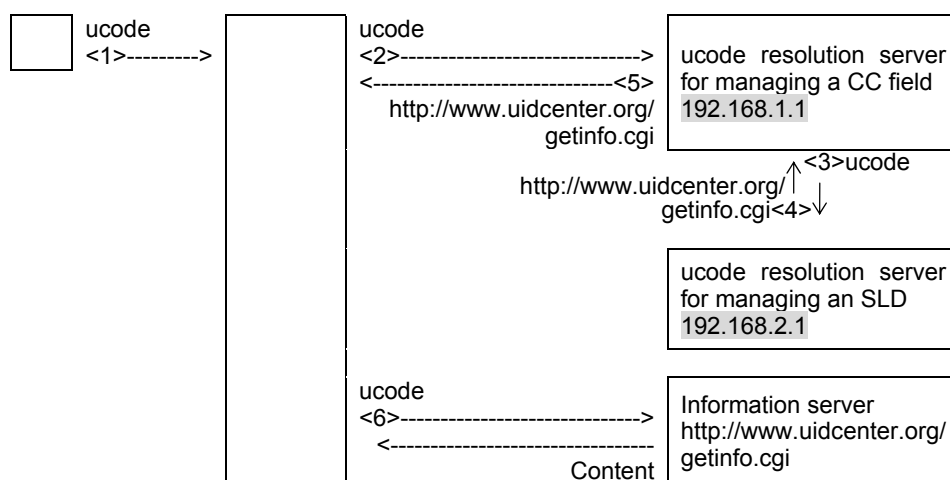
The ucode resolution server can support a cascade search function as well as basic behaviors. In the cascade search, the ucode resolution server retrieves the ucode resolution databases, allowing the client with a poor throughput speed or network connectivity to reduce the load. This section specifies the cascade search for the ucode resolution server.

Figure 4 shows an example of the behavior when the ucode resolution server for managing a CC field executes the cascade search. In the cascade search, UC can acquire an information server address with only one ucode resolution request.

Tag

UC

Server group



<1> UC reads a ucode from a tag.

<2> UC sends the ucode to the ucode resolution server for managing a CC field. When the ucode resolution server for managing a CC field performs ucode resolution by using the ucode, the server gets the address (192.168.2.1) of the ucode resolution server for managing an SLD.

<3> The ucode resolution server for managing a CC field sends the ucode to the ucode resolution server for managing an SLD. When the ucode resolution server for managing an SLD performs ucode resolution by using the ucode, the server gets the location information (<http://www.uidcenter.org/getinfo.cgi>) of the information server.

[4] The ucode resolution server for managing an SLD returns the search result (<http://www.uidcenter.org/getinfo.cgi>) to the ucode resolution server for managing a CC field.

[5] The ucode resolution server for managing a CC field returns the search result (<http://www.uidcenter.org/getinfo.cgi>) to UC.

[6] UC sends the ucode to the information server and gets content.

Figure 4: Example of Cascade Search

The process from <2> to <5> is in accordance with ucodeRP specification.

The ucode resolution server for managing a CC field directly sends a search request to the next ucode resolution server for managing an SLD to be queried (<3>). Then, the ucode resolution server for managing a CC field simply behaves as a resolution client against the ucode resolution server for managing an SLD. Note that Figure 4 shows an example that the ucode resolution server is cascaded in two-tier configuration, but in general, a n-tiered (n: positive integers) cascade is possible.

The ucode resolution server executes the cascade search if the following conditions (logical conjunction) are satisfied:

1. A cascade connection is demanded in the search request.
2. Target data file to be searched supports the cascade connection function. That is, the value of `cascade_mode` is `UIDC_CSC`.
3. The address retrieved as a search result is a ucode resolution server address.

If an error response is returned from the resolution server which performs the cascade search, the error response can be transferred to UC, or assuming the cascade search is not executed, a usual search result can be returned.

Note that this document does not specify a cascade connection to the information server.

2.3.4. Cache Search

For an efficient search, the ucode resolution server or UC can cache the ucode resolution information retrieved as a search result. Furthermore, cache-only nodes such as a cache server can be installed.

However, when returning a cached response, it should be clearly indicated that it is a cached response. In addition, the `tll` field of each data entry indicates a usable period of its cache, so the cache should not be utilized after the valid period expires. Note that the entry with `tll = 0` indicates a noncachable entry.

The client can specify whether or not returning a cached response is permitted. A ucode resolution server should not return a cached response against a search request that prohibits a cache response.

2.3.5. Proxy Server

In order to keep a UC network address, etc. confidential, you can access the ucode resolution server through a proxy server. This document does not specify a proxy server.

2.3.6. ucode Resolution Gateway

The ucode resolution gateway can be installed to reduce the load on the client with a poor throughput speed or network. The ucode resolution gateway receives a resolution request from the client, and after a series of searches are executed on the ucode resolution server or information server, the last search result is returned to the client. Functions and access protocols for the ucode resolution gateway are defined on [4].

2.4. Retrieval Content

UC accesses the content, based on the information server address retrieved by the ucode resolution. UC can sometimes determine the access protocol from the ucode resolution information. For example, when the resolution information is a URL type, an access protocol can be determined by the scheme information such as "http://."

In many cases, it is expected that the content varies depending on the ucode, so the ucode can be passed as part of the query to the information server. The use of a CGI argument is one example of passing ucodes, etc. For example, UC can send the following HTTP request to the information server.

```
http://www.uidcenter.org/getinfo.cgi?ucode=0effec0000000000000000000050123
```


3. Packet Structure and Commands of ucode Resolution Protocol

3.1. Overview

ucodeRP is a protocol that provides access to the ucode resolution database. By using ucodeRP, UC can access the ucode resolution server and get the ucode resolution information. ucodeRP is a protocol at an application layer and is designed to be independent of the protocols at lower layers.

3.2. Command

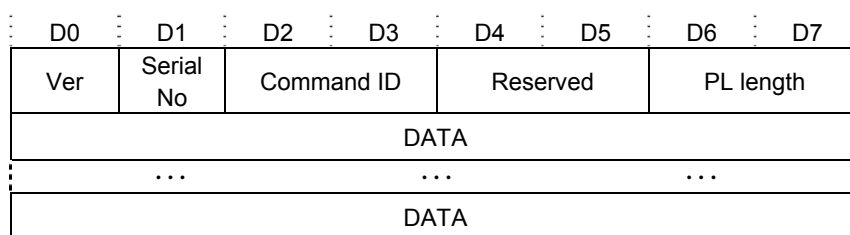
Table 9 shows the list of ucodeRP commands defined in this document.

Table 9: List of ucodeRP Command

Command name	Function
res_ucd	Retrieves a resolution result from the ucode resolution database

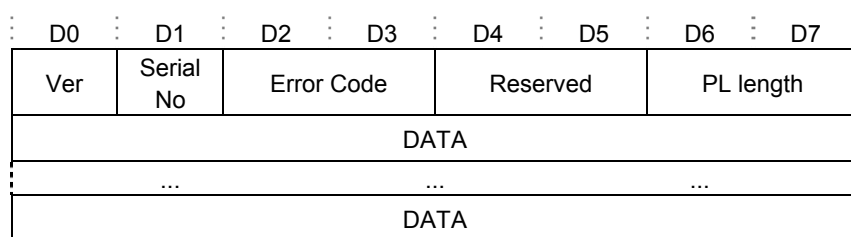
3.3. Packet Structure

The definition of the ucodeRP query packet structure is given in Figure 5, and a ucodeRP return packet structure in Figure 6. Parameters in all fields are stored in a network byte order.



Field	Meaning
Ver	Version of ucodeRP Fixed to 1
Serial No	Identification number of a ucodeRP request A sender side can set it freely.
Command ID	ucodeRP command code.
PL length	Total length of a ucodeRP packet (in 8 octet blocks) For example, if it is 48 octets, PL length is 0x06.
DATA	Command parameter specified for each command.
Reserved	Fixed to zero.

Figure 5: ucodeRP Query Packet



Field	Meaning
Ver	Version of ucodeRP Fixed to 1.
Serial No	Identification number of a ucodeRP response It should be the value of Serial No. + 1 of a corresponding ucodeRP request.
Error Code	Termination status of a ucodeRP command.
PL length	Total length of a ucodeRP packet (in 8 octet blocks) For example, if it is 48 octets, PL length is 0x0006.
DATA	Response parameter specified for each command.
Reserved	Fixed to zero.

Figure 6: ucodeRP Recieve Packet

3.4. Code Definition

(1) Command Code

The list of ucodeRP command codes is given in Table 10.

Table 10: List of a ucodeRP Command Code

Command code	Command name
0x0001	res_ucd

(2) Error code

The list of ucodeRP error codes is given Table 11.

Table 11: List of ucodeRP Error Codes

State	Value	Macro	Meaning
Normal termination	0x0000	E_UIDC_OK	Resolved successfully
Abort	0xffef	E_UIDC_NOSPT	Unsupported function
	0xffdf	E_UIDC_PAR	Parameter error
	0xffcc	E_UIDC_NOEXS	Entry nonexistent

* If resolvemode == UIDC_RSMODE_CASCADE, error code is returned from the resolution server which performs the cascade search.

3.5. Command Specification

3.5.1. res_ucd

(1) Function Overview

This function queries the ucode resolution database using by the specified ucode as a key and returns the ucode resolution information.

(2) Query Parameters

Figure 7 shows the definition of the query packet structure of a res_ucd command; Table 12 the list of query parameters; and Table 13 the parameter value definition.

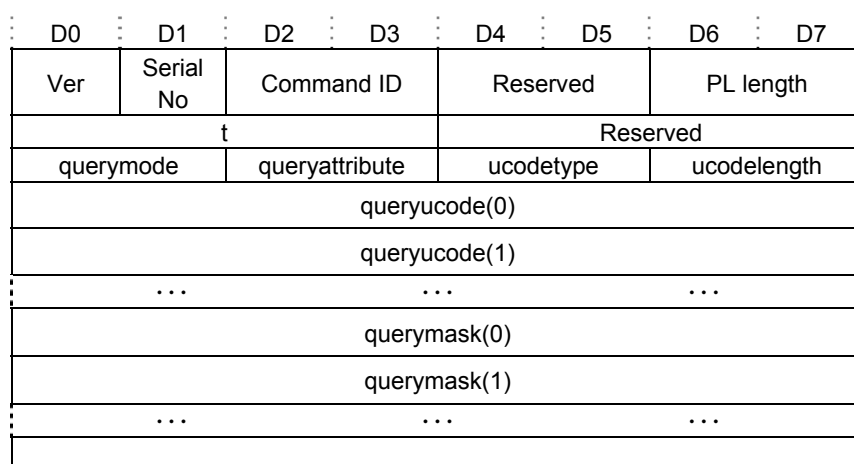


Figure 7: Query Packet of a res_ucd Command

Table 12: List of Query Parameters

Field	Meaning
t	Command send time (cumulative seconds since 0:00 AM, Jan. 1, 2000) (Greenwich Mean Time)
querymode	Search mode of the ucode resolution database
queryattribute	Data attribute to be retrieved
ucodetype	ucode type to be retrieved
ucodelength	Total length of a queryucode/querymask (byte)
queryucode	ucode to be retrieved
querymask	Bitmask value which indicates the effective bit of the ucode
Reserved	fixed to zero

Table 13: Definition of Query Parameter Values

Parameter	Value	Macro	Description
querymode (specified by logical disjunction)	0x0000	UIDC_RSMODE_RESOLUTION	Instructs ucode resolution
	0x0001	UIDC_RSMODE_CACHE	Permits a cache search
	0x0002	UIDC_RSMODE_CASCADE	Permits a cascade search
queryattribute	0x0000	UIDC_ATTR_ANONYMOUS	Attribute not specified
	0x0001	UIDC_ATTR_RS	ucode resolution server address
	0x0002	UIDC_ATTR_SS	Information server address
	0x0003	UIDC_ATTR_SIGS	Signature server address
	0x00ff	UID_USER	User defined address
ucodetype	0x0001	UID_128	128bit type ucode
	0x0002	UID_256	256bit type ucode
	0x0003	UID_384	384bit type ucode
	0x0004	UID_512	512bit type ucode
queryucode	<ucode binary>	-	ucode binary value
querymask	<ucode binary>	-	bitmask binary value

<ucode binary> ::= 16(%x00-ff)^{*1 *2}

*1 : n<element> represents that a same element is repeated a specific decimal number of times

*2 : %x represents the hexadecimal value.

(3) Response Parameter

Figure 10 shows the definition of the response packet structure of a res_ucd command; Table 14 the list of response parameters; and Table 15 the parameter value definition.

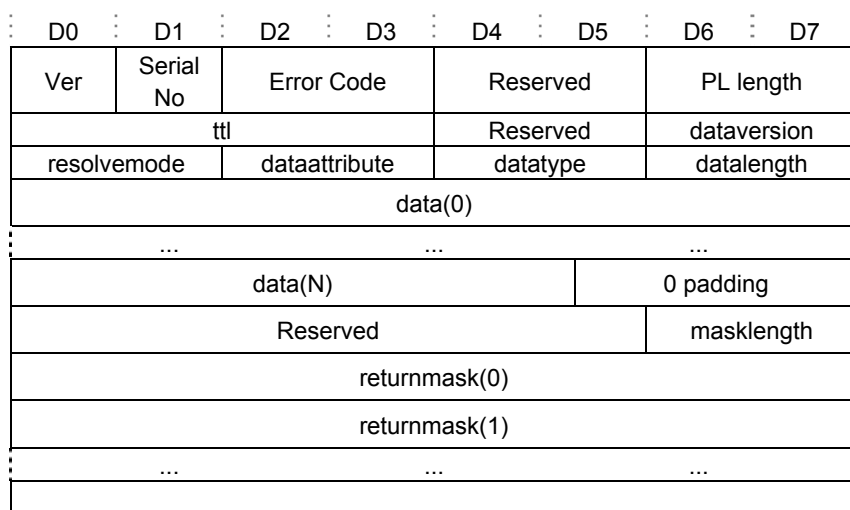


Figure 8: Recieve Packet of a res_ucd Command

Table 14: List of Recieve Parameters

Field	Meaning
ttl	Expiration date of retrieved data (seconds)
dataversion	Search data version
resolvemode	Search mode of the retrieved database
dataattribute	Retrieved data attribute
datatype	Retrieved data type and format
datalength	Retrieved data length
data	Retrieved data
masklength	Number of bytes of a returnmask
returnmask	Bitmask value which indicates the bit used for resolution
Reserved	Fixed to zero

Table 15: Definition of Recieve Parameter Values

Parameter	Value	Macro	Description
resolvemode (specified by logical disjunction)	0x0000	UIDC_RSMODE_RESOLUTION	Usual ucode resolution response
	0x0001	UIDC_RSMODE_CACHE	Response by a cache search
	0x0002	UIDC_RSMODE_CASCADE	Response by a cascade search
dataattribute	0x0001	UIDC_ATTR_RS	ucode resolution server address
	0x0002	UIDC_ATTR_SS	Information server address
	0x0003	UIDC_ATTR_SIGS	Signature server address
	0x00ff	UID_USER	User defined address
datatype	0x0001	UIDC_DATATYPE_UCODE_128	128bit type ucode
	0x0002	UIDC_DATATYPE_UCODE_256	256bit type ucode
	0x0003	UIDC_DATATYPE_UCODE_384	384bit type ucode
	0x0004	UIDC_DATATYPE_UCODE_512	512bit type ucode
	0x0011	UIDC_DATATYPE_IPV4	IPv4 address
	0x0012	UIDC_DATATYPE_IPV6	IPv6 address
	0x0013	UIDC_DATATYPE_URL	URL
	0x0014	UIDC_DATATYPE_HOST	Hostname
	0x0021	UIDC_DATATYPE_EMAIL	E-mail address
	0x0031	UIDC_DATATYPE_PHONE	Telephone number
	0x00fe	UIDC_DATATYPE_TXT	Text information
0x00ff	UIDC_DATATYPE_USER	User defined type	
data	<data value>	-	Retrieved data itself

```

<data value> ::= <ucode binary> | ; for UIDC_DATATYPE_UCODE_*
                <IPv4 address binary> | ; for UIDC_DATATYPE_IPV4
                <IPv6 address binary> | ; for UIDC_DATATYPE_IPV6
                <URI notation ASCII *1> | ; for UIDC_DATATYPE_URL
                <Host notation ASCII *2> | ; for UIDC_DATATYPE_HOST
                *CHAR ; for other types

<IPv4 address binary> ::= 4(%x00-ff)
<IPv6 address binary> ::= 16(%x00-ff)
CHAR = <US-ASCII character>

*1 : Refer to [5]
*2 : Refer to [6]

```

Appendix A. Construction/Behavior Example of ucode Resolution Server

A.1. Construction Example of ucode Resolution Server

Here is an example of configuration when a ucode space is managed in the three layers of a TLD (upper 20bits), CC field (Class Code: upper 24bits), and SLD (upper 112bits), which respective data files are managed in different ucode resolution servers. The hierarchical structure of the server group is given in Figure 9.

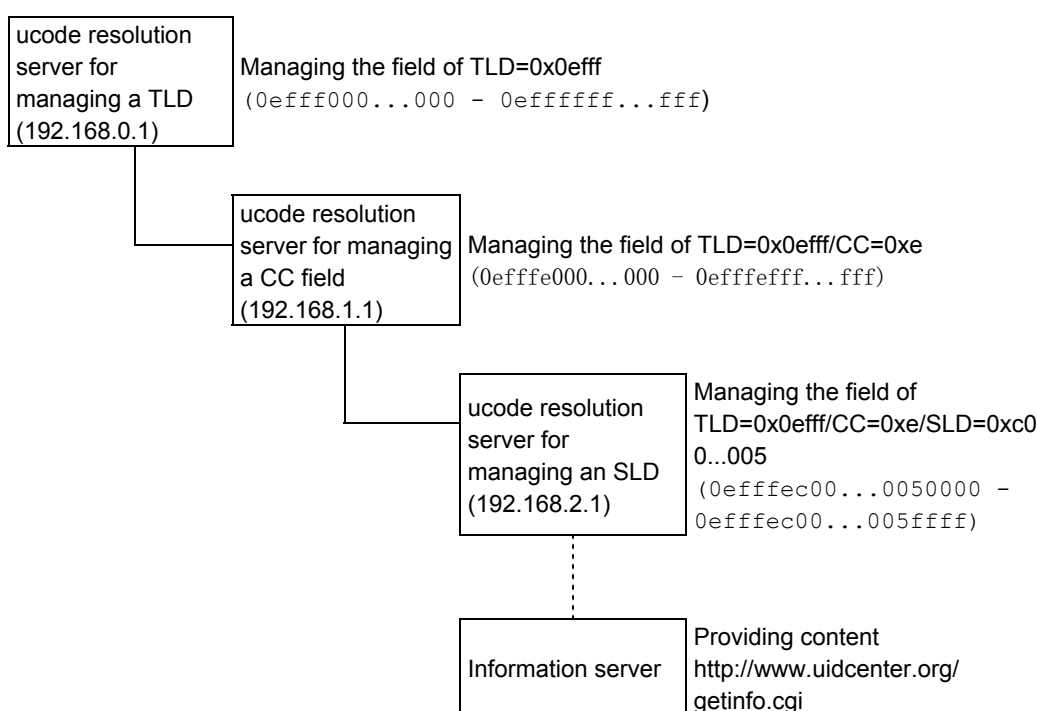


Figure 9: Hierarchical Structure of the Server Group

/

(1) ucode Resolution Server for Managing a TLD

Shown is an example of construction of a data file managed by the ucode resolution server for managing a TLD. This data file manages the ucode space of 0efff000...000 - 0effffff...fff. In the data entry, the ucode space of 0efffe000...000 - 0efffefff...fff is associated with the IPv4 address "192.168.1.1" which points at an IP address of the ucode resolution server at a lower layer.

Data File

dbucode	= 0efff0000000000000000000000000000
dbmask	= fffff00000000000000000000000000000
cascade_mode	= UIDC_NOCS

Data Entry

ucode	= 0efffe0000000000000000000000000000
ucodemask	= fffff00000000000000000000000000000
attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80101 // Binary notation of "192.168.1.1"

(2) ucode Resolution Server for Managing a CC field

Next is an example of construction of a data file managed by the ucode resolution server for managing a CC field. This data file manages the ucode space of 0efffe000...0000 - 0efffefff...fff. In the data entry, the ucode space of 0efffec00...0050000 - 0efffec00...005ffff is associated with the IPv4 address "192.168.2.1" which points at the ucode resolution server of at a lower layer.

Data File

dbucode	= 0efffe0000000000000000000000000000
dbmask	= fffff00000000000000000000000000000
cascade_mode	= UIDC_NOCS

Data Entry

ucode	= 0efffec0000000000000000000000050000
ucodemask	= ffffffffffffffffffffffffffffff0000
attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80201 // Binary notation of "192.168.2.1"

(3) ucode Resolution Server for Managing an SLD

Here is an example of construction of a data file managed by the ucode resolution server for managing an SLD. This data file manages the ucode space of 0efffec00... 0050000 - 0efffec00...005ffff. In the data entry, the ucode space of 0efffec00...0050100 - 0efffec00... 00501ff is associated with the URL "http://www.uidcenter.org/getinfo.cgi" which is a CGI script of the information server.

Data File

dbucode	= 0efffec000000000000000000050000
dbmask	= ffffffffffffffffffffffffffffffffff0000
cascade	= UIDC_NOCS

Data Entry

ucode	= 0efffec000000000000000000050100
ucodemask	= ffffffffffffffffffffffffffffffffff00
attribute	= 0x0002 (UIDC_ATTR_SS)
version	= 0x0001
tll	= 0x00000000
datatype	= 0x0013 (UIDC_DATATYPE_URL)
data	= "http://www.uidcenter.org/getinfo.cgi"

A.2. Example of Behaviors of ucode Resolution

Here is an example of behaviors of ucode resolution of the configuration in Figure 9. The ucode to be passed to the resolution server is 0efffec000000000000000000050123.

(1) ucode Resolution Server for Managing a TLD

Here is an example of sending a ucode resolution request to the ucode resolution server for managing a TLD and getting the address of the ucode resolution server for managing a CC field.

Data Entry

ucode	= 0efffe000000000000000000000000
ucodemask	= fffffff00000000000000000000000
attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80101 // Binary notation of "192.168.1.1"

Search Request

queryucode	= 0efffec000000000000000000050123
querymask	= ffffffffffffffffffffffffffffffffff

Search Result

attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80101
returnmask	= fffffff00000000000000000000000
resolvemode	= 0x0000 (UIDC_RSMODE_RESOLUTION)

The ucode resolution is performed using the upper 24bit, and the IPv4 address "192.168.1.1" of the ucode resolution server for managing a CC field is returned as a search result.

(2) ucode Resolution Server for Managing a CC Field

Here is an example of sending a ucode resolution request to the ucode resolution server for managing a CC, and getting the address of the ucode resolution server for managing an SLD.

Data Entry

ucode	= 0efffec000000000000000000050000
ucodemask	= ffffffffffffffffffffffffffffff0000
attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80201 // Binary notation of "192.168.2.1"

Search Request

queryucode	= 0efffec000000000000000000050123
querymask	= ffffffffffffffffffffffffffffff

Search Result

attribute	= 0x0001 (UIDC_ATTR_RS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0011 (UIDC_DATATYPE_IPV4)
data	= 0xC0A80201
returnmask	= ffffffffffffffffffffffffffffff0000
resolvemode	= 0x0000 (UIDC_RSMODE_RESOLUTION)

The ucode resolution is performed using the upper 112bit, and the IPv4 address "192.168.2.1" of the ucode resolution server for managing an SLD is returned as a search result.

(3) ucode Resolution Server for Managing an SLD

Here is an example of sending a ucode resolution request to the ucode resolution server for managing an SLD and getting the address of the information server.

Data Entry

ucode	= 0efffec0000000000000000000000000050100
ucodemask	= ffffffffffffffffffffffffffffffffffffff00
attribute	= 0x0002 (UIDC_ATTR_SS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0013 (UIDC_DATATYPE_URL)
data	= "http://www.uidcenter.org/getinfo.cgi"

Search Request

queryucode	= 0efffec0000000000000000000000000050123
querymask	= ffffffffffffffffffffffffffffffffffffff

Search Result

attribute	= 0x0002 (UIDC_ATTR_SS)
version	= 0x0001
tll	= 0x00015180
datatype	= 0x0013 (UIDC_DATATYPE_URL)
data	= "http://www.uidcenter.org/getinfo.cgi"
returnmask	= ffffffffffffffffffffffffffffffffffffff00
resolvemode	= 0x0000 (UIDC_RSMODE_RESOLUTION)

The ucode resolution is performed using the upper 120bit, and the URL, "http://www.uidcenter.org/getinfo.cgi" of the information server is returned as a search result.

Index

C

Cascade Search13

D

data entry 7

data file..... 7

P

Proxy Server.....15

U

unicode Resolution Database 7

unicode Resolution Gateway.....15

unicode Resolution Protocol..... 6

unicode Resolution Server 7

