Internet Draft
draft-ietf-rescap-proto-format-01.txt
August 8, 2000
Expires in six months

Paul Hoffman
Internet Mail Consortium

# The rescap Request and Response Format

## Status of this memo



## Abstract

The rescap protocol is a general client-server resolution protocol that translates resource identifiers to a list of attributes. For instance, a rescap client can ask a rescap server for the attributes of a particular mail user. rescap is very light-weight and acts only as a resolution protocol, not a directory service. This document describes the format of rescap requests and responses.

## 1. Introduction

The rescap protocol is described in [RESCAP-MAIN]. This document specifies the format the request and response messages. It also defines the required rescap items that must be supported.

This document and others relating to the rescap protocol are being discussed in the recap WG of the IETF on the rescap@cs.utk.edu mailing list. To subscribe, send a message to rescap-request@cs.utk.edu. An archive of the mailing list is available at <ftp://cs.utk.edu/pub/rescap>.

### 1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [MUSTSHOULD].

Hexadecimal values are indicated as "xNN" or "xNNNN". For example, xA0B1 corresponds to the octet xA0 followed by the octet xB1.

## 2. Request and Response Format

Rescap requests and responses have a simple format. The format is optimized for simple processing in small clients. It is also optimized for size so that it is more likely that requests and responses can fit in single UDP datagrams.

The basic unit of a rescap request or response is the item. An item consists of exactly three parts:

- a two-octet tag
- a two-octet length
- content whose length and structure are defined by the tag

Each of the three parts of an are in network byte order, as is the entire item.

The tag values are given in this document and in other documents that profile rescap for various resource types. The length is the length of the content of the item, measured in octets. Some tags define fixed-length content, while other tags define variable-length content.

A program parsing a request or response can easily skip over items whose tags it doesn't recognize by reading the tag, reading the length, and skipping over the number of octets given in the length to get to the beginning of the next item.

There are two types of items: request-type items and response-type items. The type of the item is given in each item's specification.

### 2.1 Long content

The most significant bit of the two-octet length part of an item is a continuation marker. If the continuation marker is "1", the item has been split into two or more fragments. The item following an item whose continuation marker is "1" MUST have the same tag value as the preceding item. The first fragment that has a continuation marker of "0" is the last fragment in the item. When reconstructing a fragmented item, the content of all fragments are appended in the same sequence that they appeared in the data steam.

A consequence of this design is that any item whose length is greater than x7FFF MUST be split into at least two packets. Any item MAY be split into fragments. A process interpreting a stream of rescap items MUST be able to correctly handle long content. That is, if an item has a continuation marker of "1" in the length part, the process MUST read the next item, check that the tag matches that of the preceding item, and append the content of the second item to that of the first item, until the process comes to an item with the continuation marker set to "0". It is an error if the process comes to an item with a different tag value than the preceding item if that preceding item had a continuation marker of "1".

Design note: the method of using a two-octet length and checking for

the special value for the continuation marker was chosen instead of using a four-octet length in order to keep the size of the rescap response shorter, and therefore make it more likely that a response would fit in a single UDP datagram. The vast majority of items will probably have lengths less than x7FFF. The continuation marker allows a program to marshal fewer than x7FFF octets if it has memory constraints. The cost of having to check for the special case of a continuation marker seems to be worth the tradeoff of making every item two octets longer.

## 3. Basic requests and responses

The items in this section are included in a rescap client's request to a rescap server and in the server's response to the client. A rescap request is always a FullRequest item; that item contains other request-type items. A rescap response is always a FullResponse item; that item contains other response-type items.

### 3.1 Basic request types

Clients MUST be able to emit the FullRequest and BaseURI types, and SHOULD be able to emit ItemsToReturn. Clients MAY implement the AuthInTheClear, AuthIP, and PrivUseRequest types. Servers MUST be able to interpret the FullRequest, BaseURI, and ItemsToReturn types, although servers do not have to comply with the request in ItemsToReturn. For instance, a server may not want to return certain items due to lack of authorization or due to the response becoming too long.

#### 3.1.1 FullRequest

The FullRequest item (tag x0001) is used to encapsulate the other request-type items. A rescap client's request to the server MUST be a single FullRequest item. The structure of the item is a two-octet integer that specifies the number of items that follow that are part of the FullRequest. If the number of items that follows the FullRequest item is larger than that indicated in the FullRequest item, the server MUST ignore them.

The items that are encapsulated by the FullRequest item MUST be request-type items. A FullRequest item MUST NOT encapsulate another FullRequest item. Note that some items contain other items; the prohibition against response-type items is only for items directly encapsulated in the FullRequest item, not items that are encapsulated in lower-level items.

A FullRequest item MUST contain exactly one instance of a BaseURI item, and MAY contain zero or one instances of each of the other request-type items.

Servers conforming to this specification MUST check the tag of the first item in the request, and MUST reject a request that starts with an item whose tag is not x0001. The tag in the FullRequest item acts like a version number for requests. If this specification is updated in the future in an incompatible fashion, the future specification will

use a different tag for the first request item.

### 3.1.2 BaseURI

The BaseURI item (tag x0002) specifies the Internet resource for which the rescap client wants information. The structure of the item is a URI string as defined in [URI] for a single resource. Every FullRequest MUST include exactly one BaseURI item.

### 3.1.3 ItemsToReturn

The ItemsToReturn item (tag x0003) lists the response-type items that the client would like the server to return. It is a request, not a demand; the server is allowed to return any response-type item it chooses. The client can use this item to specify that it only is interested in a limited number of response-type items, and that the server should not waste its time or bandwidth returning any items not listed in the ItemsToReturn item. The structure of the item is a sequence of tag values, each of which is two octets long.

If the ItemsToReturn item is not included in a request, the server may return whatever information it pleases about the resource named in the BaseURI item. If the ItemsToReturn item is included and its length is zero, the client wants the server to return all the information it has about the resource named in the BaseURI item. If the ItemsToReturn item is included and its length is greater than zero, the client wants the server to return only response-type items of the type listed in the ItemsToReturn item.

Note that it is not an error for the client to list items in the ItemsToReturn item that cannot be returned to by the server.

### 3.1.4 AuthInTheClear

The AuthInTheClear item (tag x0005) holds a sequence of binary octets. The format of the sequence is determined by private agreement between the client and the server. This item MUST only be sent when using the secure form of rescap. It MUST NOT be used in the basic form of rescap.

### 3.1.5 AuthIP

The AuthIP item (tag x0008) is a zero-length item that indicates that the client requests that it gain some authorization simply based on the IP address of the client, which the server will determine by examining the IP address of the connection. This item MAY be used in either the basic form or the secure form of rescap.

### 3.1.6 PrivUseRequest00 through PrivUseRequest255

The PrivUseRequest00 through PrivUseRequest255 items (tags xFE00 through xFEFF) are reserved for private use as request-type items and will never be assigned by IANA. These tags may be used by protocol developers to test protocols that they are developing, such as during the process of preparing Internet Drafts that contain registration for future request-type items. The structure of the items is undefined.

### 3.2 Basic response types

Servers MUST implement the FullResponse and Status types, and SHOULD implement the Referral type. Servers SHOULD implement the TTLOfInfo, ExpirationOfInfo, and DateOfChange types. Servers MAY implement the PrivUseResponse types.

Clients MUST be able to interpret the FullResponse, Status, and Referral types; clients SHOULD be able to process Referrals by fetching the information from the referred-to host. A rescap client SHOULD NOT attempt to make any use of the user response-type items that it does not fully understand.

Note: clients MUST parse TTLOfInfo, ExpirationOfInfo, and DateOfChange items even if the clients do not know how to check the expiration or modification of information (such as a client that does not know when the information will be delivered to the user or a client that does not know the current time). These items encapsulate other response items that the client may want to deliver to the user regardless of the expiration status. Clients that cannot check the expiration or creation information MUST treat the enclosed response-type data as if it appeared outside of the time-specific types, and in the case of TTLOfInfo and ExpirationOfInfo, SHOULD indicate to the user that the server put an expiration on the information and that the expiration was not checked.

#### 3.2.1 FullResponse

The FullResponse item (tag x000C) is used to encapsulate the other response-type items. A rescap server's Response to a client MUST be a single FullResponse item. The structure of the item is a two-octet integer that specifies the number of items that follow that are part of the FullResponse item. If the number of items that follows the FullResponse item is larger than that indicated in the FullResponse item, the server MUST ignore them.

The items that are encapsulated by the FullResponse item MUST be response-type items. A FullResponse item MUST NOT encapsulate another FullResponse item. Note that some items contain other items; the prohibition against request-type items is only for items directly encapsulated in the FullResponse item, not items that are encapsulated in lower-level items.

A FullResponse item MUST contain one or more instance of a Status item, and MAY contain zero or more instances of each of other response-type items. Note that some response-type items further restrict the number of times those items can appear in a FullResponse item.

Clients conforming to this specification MUST check the tag of the first item in the response, and MUST reject a response that starts with an item whose tag is not x000C. The tag in the FullResponse item acts like a version number for responses. If this specification is updated in the future in an incompatible fashion, the future specification will use a different tag for the first response item.

### 3.2.2 Status

The Status item (tag x000D) gives status information to the client about its request. The structure of the status tag is a sequence of a one-octet main status code, a one-octet secondary status code, and an optional string that is a sequence of characters from the ISO/IEC 10646-1 character set encoded with the UTF-8 transformation format defined in [UTF8]. A FullResponse MUST have at least one Status item and MAY have more than one Status item. A rescap server SHOULD include only as many Status items as necessary for a client to process the response.

The optional string may be used to transmit status information, but it is optional. In fact, a rescap server that is trying to keep its responses as short as possible SHOULD NOT include a string at all.

The rescap client MAY choose to display the string to the client. However, because there is no way to know the languages understood by the user, the string may be of little or no use to them.

Note: If a client sent a request over UDP and receives a Status item of x0201, the client SHOULD sent the same request over TCP.

The values for the main status code (the first octet of the Status item) are based loosely on those in SMTP. They are:
x00 - positive completion
x01 - transient negative completion
x02 - permanent negative completion
x03 - informational

A rescap client MAY use just the main status code to decide how to display any results of the request to the user who made the request.

The complete list of status codes is:

x0000 The request was fully processable

x0100 Too busy; try again whenever you feel like it
x0101 Too busy; try again later than 10 seconds from now
x0102 Too busy; try again later than 60 seconds from now
x0103 Too busy; the Referral item in the response leads to another rescap server authoritative for this resource

x0200 The body of the request was longer than what was indicated in the length
x0201 The body of the request was shorter than what was indicated in the length
x0202 The request included items not of request-type
x0203 The request included more than one BaseURI item
x0204 This server is not authoritative for the resource named in the BaseURI item and no referral is available
x0205 This server is not authoritative for the resource named in the BaseURI item; the URI in the Referral item leads you to a server that this server believes is authoritative

x0300 Successful authorization through AuthInTheClear

x0301 Successful authorization through AuthIP
x0302 Some items requested in the ItemsToReturn were not returned due
to insufficient authentication or lack of authentication

### 3.2.3 Referral

The Referral item (tag x000E) tells the client that another rescap server has authoritative information for the requested resource. If some information may be available from several sources, an "authoritative" source is one whose response should be regarded as superseding all others in the event of any discrepancy. The structure of the Referral item is a URI string as defined in [URI] for a single resource. The scheme in the URI MUST be "rescap-basic" or "rescap-secure". If a FullResponse item contains a Referral item or a Referral item enclosed in a SignedResponse or an EncryptedResponse item, there MUST NOT be any other items in the FullResponse item other than Status items.

### 3.2.4 TTLOfInfo

The TTLOfInfo item (tag x0017) specifies how long the server assures that the information enclosed in the item will be valid. The TTLOfInfo item has the structure of a four-octet integer followed by a two-octet integer. The four-octet integer represents the number of seconds before the server no longer assures that the listed items are valid. Note: this value is approximate and probably will only be accurate within minutes or tens of seconds. The two-octet integer is the number of the following items that are covered by the TTL; these MUST be response-type items.

If a rescap client does not support expiration of items, it MUST treat the items in a TTLOfInfo item as if they appeared outside of a TTLOfInfo item. This allows a server to respond with TTLOfInfo without knowing whether or not the client handles the expiration time listed in the item.

### 3.2.5 ExpirationOfInfo

The ExpirationOfInfo item (tag x0018) specifies the final time in the future that the server assures that the information enclosed in the item will be valid. The ExpirationOfInfo item has the structure of a 14-octet string followed by a two-octet integer. The 14- octet ASCII string represents the date at Greenwich Mean Time in YYYYMMDDHHMMSS format after which the server no longer assures that the listed items are valid. The two-octet integer is the number of the following items that are covered by the expiration time; these MUST be response-type items.

If a rescap client does not support expiration of items, it MUST treat the items in a ExpirationOfInfo item as if they appeared outside of a ExpirationOfInfo item. This allows a server to respond with ExpirationOfInfo without knowing whether or not the client handles the expiration time listed in the item.

### 3.2.6 DateOfChange

The DateOfChange item (tag x001C) specifies the time that the server

last changed the value of the attribute (or, if it has never been changed, first created the value). The DateOfChange item has the structure of a 14-octet  string followed by a two-octet integer. The 14-octet ASCII string represents the date at Greenwich Mean Time as a string of ASCII characters in YYYYMMDDHHMMSS format. The two-octet integer is the number of the following items that are covered by the date of change; these MUST be response-type items.

If a rescap client does not support showing modification times of items, it MUST treat the items in a DateOfChange item as if they appeared outside of a DateOfChange item. This allows a server to respond with DateOfChange without knowing whether or not the client handles the display of modification time listed in the item.

### 3.2.7 PrivUseResponse00 through PrivUseResponse255

The PrivUseResponse00 through PrivUseResponse255 items (tags xFF00 through xFFFF) are reserved for private use as response-type items and will never be assigned by IANA. These tags may be used by protocol developers to test protocols that they are developing, such as during the process of preparing Internet Drafts that contain registration for future response-type items. The structure of the items is undefined.

## 4. Security Considerations

This document inherits all of the security considerations of [RESCAP-MAIN].

A server may choose to allow authorization through two mechanisms. AuthInTheClear can only be used in secure form rescap, is simple client identification; AuthIP, can be used in either basic form or secure form. Both provide simple authentication based on a single identifier for the client.

## 5. References

[MUSTSHOULD] "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119.

[RESCAP-MAIN] "The rescap Resolution Protocol", draft-ietf-rescap-proto-main.

[RESCAP-REQUIRE] "ResCap Requirements", draft-ietf-rescap-req.

[URI] "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396.

[UTF8] "UTF-8, a transformation format of ISO 10646", RFC 2279.

## A. IANA Considerations

### A.1 Item registry

IANA will maintain a registry of all rescap items. The registry will be

populated by the items in this specification and any other RFC. The RFCs must give sufficient detail so that interoperability between independent implementations is possible. The registry will contain the name of the item, the tag value, the type, and the RFC which defines the item. Each name in the registry must be unique, and each tag value must also be unique. The types in the registry are "request-type" and "response-type". IANA should also list the RFC in which the items are defined so that protocol designers know where to find the document defining the items.

Each RFC that defines rescap items must include an application of the form:

To: iana@iana.org
From: <author's address>
Subject: Registration of rescap items

The following are the rescap items defined in this document.

Item Type Tag

### A.2 Status value registry

IANA will maintain a registry of all rescap status values. The registry will be populated by the status values in this specification and any other RFC. The RFCs must give sufficient detail so that interoperability between independent implementations is possible. The registry will contain the value of the status value, a brief description of the meaning of the value, and the RFC which defines the value. Each value in the registry must be unique. IANA should also list the RFC in which the status values are defined so that protocol designers know where to find the document defining the status values.

Each RFC that defines rescap status values must include an application of the form:

To: iana@iana.org
From: <author's address>
Subject: Registration of rescap status values

The following are the rescap status values defined in this document.

Value Meaning

## B. Registration of rescap Types and Status Values

### B.1 Registration of rescap Types

To: iana@iana.org
From: Paul Hoffman <phoffman@imc.org>
Subject: Registration of rescap items
The following are the rescap items defined in this document.

Item Type Tag
FullRequest request x0001

```
BaseURI                 request   x0002
ItemsToReturn                   request   x0003
AuthInTheClear                  request   x0005
AuthIP                                  request   x0008
FullResponse                    response  x000C
Status                                  response  x000D
Referral                                response  x000E
TTLOfInfo                               response  x0017
ExpirationOfInfo        response  x0018
DateOfChange            response  x001C
PrivUseRequest00 through
    PrivUseRequest255    request   xFE00-xFEFF
PrivUseResponse00 through
    PrivUseResponse255   response  xFF00-xFFFF
```

### B.2 Registration of rescap Status Values

```
To: iana@iana.org
From: Paul Hoffman <phoffman@imc.org>
Subject: Registration of rescap status values

The following are the rescap status values defined in this document.

Value Meaning
x0000 The request was fully processable
x0001 Successful authorization through AuthInTheClear
x0002 Successful authorization through AuthDigest

x0100 Too busy; try again whenever you feel like it
x0101 Too busy; try again later than 10 seconds from now
x0102 Too busy; try again later than 60 seconds from now
x0103 Too busy; the Referral item in the response leads to another
       rescap server authoritative for this resource

x0200 The body of the request was longer than what was indicated in the
       length
x0201 The body of the request was shorter than what was indicated in
          the length
x0202 The request included items not of request-type
x0203 The request included more than one BaseURI item
x0204 This server is not authoritative for the resource named in the
       BaseURI item and no referral is available
x0205 This server is not authoritative for the resource named in the
       BaseURI item; the URI in the Referral item leads you to a server
       that this server believes is authoritative

x0300 Successful authorization through AuthInTheClear
x0301 Successful authorization through AuthIP
x0302 Some items requested in the ItemsToReturn were not returned due
       to insufficient authentication or lack of authentication
```

## C. Acknowledgments

Graham Klyne provided suggestions for early versions of the first draft.

Chris Newman devised the method of encapsulation that works with the method of fragmentation.

## D. Changes Between -00 and -01 Versions of This Document

Added notes in 3.1.1 and 3.2.1 about the tag value acting like a version number.

Changed the URL schemes in 3.2.3.

## E. Author Contact Information

Paul Hoffman
Internet Mail Consortium
127 Segre Place
Santa Cruz, CA  95060 USA
phoffman@imc.org