

IrTran-P (Infrared Transfer Picture) Specification

IrDA Application for Consumer Digital Cameras

Version 1.0
October 1997



Contributing Companies:

NTT Corporation

CASIO COMPUTER CO., LTD.

SHARP Corporation

Sony Corporation

OKAYA SYSTEMWARE CO., LTD.

Authors:

Yuichiro "Tuck" Takagawa (NTT Corporation)

Ken-ichiro Shimokura (NTT Corporation)

Takashi Imaeda (NTT Corporation)

Tatsuo Arai (CASIO COMPUTER CO., LTD.)

Shuichiro Ono (SHARP Corporation)

Norihiro Kumagai (SHARP Corporation)

Takahiro Imai (SHARP Corporation)

Akihiro Kikuchi (Sony Corporation)

Jun Nagai (Sony Corporation)

Toshiharu Kobayashi (Sony Corporation)

Gontaro Kitazumi (OKAYA SYSTEMWARE CO., LTD.)

Document Status: Version 1.0

Version 1.0 was approved at the IrDA General Meeting October 1997

INFRARED DATA ASSOCIATION (IrDA) - NOTICE TO THE TRADE -

SUMMARY:

Following is the notice of conditions and understandings upon which this document is made available to members and non-members of the Infrared Data Association.

- Availability of Publications, Updates and Notices
- Full Copyright Claims Must be Honored
- Controlled Distribution Privileges for IrDA Members Only
- Trademarks of IrDA - Prohibitions and Authorized Use
- No Representation of Third Party Rights
- Limitation of Liability
- Disclaimer of Warranty
- Product Testing for IrDA Specification Conformance

IrDA PUBLICATIONS and UPDATES:

IrDA publications, including notifications, updates, and revisions, are accessed electronically by IrDA members in good standing during the course of each year as a benefit of annual IrDA membership. Electronic copies are available to the public on the IrDA web site located at irda.org. Requests for publications, membership applications or more information should be addressed to: Infrared Data Association, P.O. Box 3883, Walnut Creek, California, U.S.A. 94598; or e-mail address: info@irda.org; or by calling John LaRoche at (510) 943-6546 or faxing requests to (510) 934-5600.

COPYRIGHT:

1. Prohibitions: IrDA claims copyright in all IrDA publications. Any unauthorized reproduction, distribution, display or modification, in whole or in part, is strictly prohibited.
2. Authorized Use: Any authorized use of IrDA publications (in whole or in part) is under NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

TRADEMARKS:

1. Prohibitions: IrDA claims exclusive rights in its trade names, trademarks, service marks, collective membership marks and trademark marks (hereinafter collectively "trademarks"), including but not limited to the following trademarks: INFRARED DATA ASSOCIATION (wordmark alone and with IR logo), IrDA (acronym mark alone and with IR logo), IR logo, and MEMBER IrDA (wordmark alone and with IR logo). Any unauthorized use of IrDA trademarks is strictly prohibited.
2. Authorized Use: Any authorized use of a IrDA collective membership mark or trademark mark is by NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

NO REPRESENTATION of THIRD PARTY RIGHTS:

IrDA makes no representation or warranty whatsoever with regard to IrDA member or third party ownership, licensing or infringement/non-infringement of intellectual property rights. Each recipient of IrDA publications, whether or not an IrDA member, should seek the independent advice of legal counsel with regard to any possible violation of third party rights arising out of the use, attempted use, reproduction, distribution or public display of IrDA publications.

IrDA assumes no obligation or responsibility whatsoever to advise its members or non-members who receive or are about to receive IrDA publications of the chance of infringement or violation of any right of an IrDA member or third party arising out of the use, attempted use, reproduction, distribution or display of IrDA publications.

LIMITATION of LIABILITY:

BY ANY ACTUAL OR ATTEMPTED USE, REPRODUCTION, DISTRIBUTION OR PUBLIC DISPLAY OF ANY IrDA PUBLICATION, ANY PARTICIPANT IN SUCH REAL OR ATTEMPTED ACTS, WHETHER OR NOT A MEMBER OF IrDA, AGREES TO ASSUME ANY AND ALL RISK ASSOCIATED WITH SUCH ACTS, INCLUDING BUT NOT LIMITED TO LOST PROFITS, LOST SAVINGS, OR OTHER CONSEQUENTIAL, SPECIAL, INCIDENTAL OR PUNITIVE DAMAGES. IrDA SHALL HAVE NO LIABILITY WHATSOEVER FOR SUCH ACTS NOR FOR THE CONTENT, ACCURACY OR LEVEL OF ISSUE OF AN IrDA PUBLICATION.

DISCLAIMER of WARRANTY:

All IrDA publications are provided "AS IS" and without warranty of any kind. IrDA (and each of its members, wholly and collectively, hereinafter "IrDA") EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND WARRANTY OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.

IrDA DOES NOT WARRANT THAT ITS PUBLICATIONS WILL MEET YOUR REQUIREMENTS OR THAT ANY USE OF A PUBLICATION WILL BE UN-INTERRUPTED OR ERROR FREE, OR THAT DEFECTS WILL BE CORRECTED. FURTHERMORE, IrDA DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING USE OR THE RESULTS OR THE USE OF IrDA PUBLICATIONS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN PUBLICATION OR ADVICE OF A REPRESENTATIVE (OR MEMBER) OF IrDA SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY.

LIMITED MEDIA WARRANTY:

IrDA warrants ONLY the media upon which any publication is recorded to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of distribution as evidenced by the distribution records of IrDA. IrDA's entire liability and recipient's exclusive remedy will be replacement of the media not meeting this limited warranty and which is returned to IrDA. IrDA shall have no responsibility to replace media damaged by accident, abuse or misapplication. ANY IMPLIED WARRANTIES ON THE MEDIA, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF DELIVERY. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM PLACE TO PLACE.

COMPLIANCE and GENERAL:

Membership in IrDA or use of IrDA publications does NOT constitute IrDA compliance. It is the sole responsibility of each manufacturer, whether or not an IrDA member, to obtain product compliance in accordance with IrDA Specifications.

All rights, prohibitions of right, agreements and terms and conditions regarding use of IrDA publications and IrDA rules for compliance of products are governed by the laws and regulations of the United States. However, each manufacturer is solely responsible for compliance with the import/export laws of the countries in which they conduct business. The information contained in this document is provided as is and is subject to change without notice.

Table of Contents

1. Introduction	1
1.1. Foreword	1
1.2. Scope of IrTran-P Standard	1
1.3. SCEP and bFTP	1
1.4. Image Format UPF (Uni Picture Format)	2
1.5. Study of Approach of UPF.....	2
2. Usage Model and Operations of IrTran-P	4
2.1. Usage model 1 Simple model.....	5
2.2. Usage model 2 The case of sending the non-mandatory size	5
2.3. Usage Model 3 The case of sending the plural pictures	6
2.4. Closing Remark.....	7
3. Protocol (SCEP & bFTP)	8
3.1. Introduction.....	8
3.1.1. Overview	8
3.1.2. Terminology	8
3.1.3. Service Model.....	8
3.1.4. Bit and Byte Ordering	9
3.1.5. References	9
3.2. SCEP (Simple Command Execute Protocol)	11
3.2.1. Service Definition	11
3.2.2. SCEP Protocol Data Units	17
3.2.3. State Definition and Transitions.....	28
3.3. bFTP (binary File Transfer Protocol).....	32
3.3.1. Service Definition	32
3.3.2. bFTP Protocol Data Units.....	35
3.3.3. State definition and transitions.....	42
3.4. IrCOMM and IrLMP IAS Objects	44
3.4.1. Recommendation of IrCOMM Operation.....	44
4. Appendix Uni Picture Format -	45
4.1. Introduction.....	45
4.1.1. Scope and Format Abbreviations.....	45
4.1.2. Terminology	45
4.1.3. Bit and Byte Ordering	45
4.1.4. References	46
4.2. Specifications.....	46
4.2.1. Signal format	46
4.2.2. File Specifications.....	50
4.2.3. Application rule.....	64
4.3. Additions	66

1. Introduction

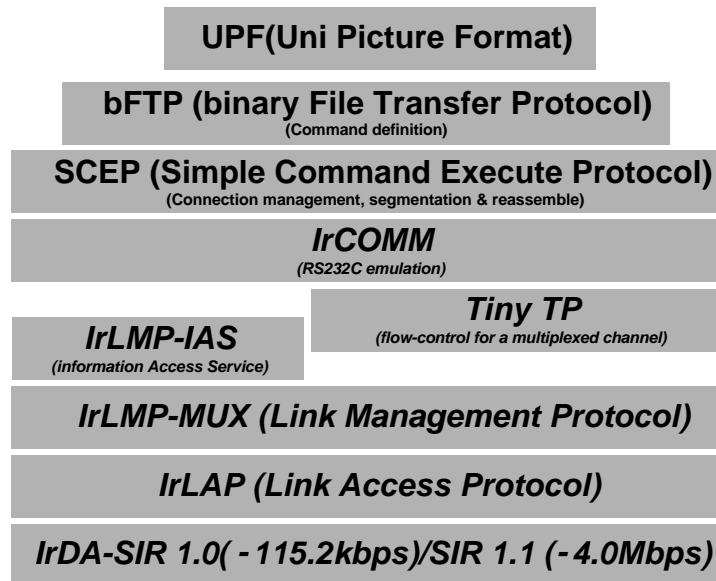
1.1. Foreword

IrTran-P(Infrared Transfer Picture) is an image communication scheme for a digital camera based on the Infrared Communication Standard specification created by IrDA. The IrTran-P specification is to be largely used together with the IrDA standard specifications.

1.2. Scope of IrTran-P Standard

IrTran-P is placed on the upper layer of IrSIR, IrLAP, IrLMP, TinyTP and IrCOMM which is already established as IrDA standard specifications. SCEP (Simple Command Execute Protocol) and a bFTP (Binary File Transfer Protocol) are necessary for exchanging an image between devices and mutually exchanging properties of the devices. An image format (file) called UPF (Uni Picture Format) is exchanged on such an entity(UPF is image format out of the category of IrDA, and will be treated as an appendix.). IrTran-P is a generic name given to all of these components.

This section is written to clarify and position the respective standards adopted by IrTran-P, in what context the standards are adopted, and to facilitate understanding of the reason for adoption as well. As for technical data on SCEP, bFTP and UPF, please refer to the sections individually written for SCEP, bFTP and UPF in the later part of this document.



1.3. SCEP and bFTP

SCEP establishes a session on IrCOMM and provides a transparent session which notifies an upper layer of a command. The procedure of SCEP is developed by a lower layer's making use of an advantage that an IrDA protocol is "error free", as a high speed session layer matching the IrDA protocol.

As is apparent from its name, bFTP provides a service for transferring a binary file. The bFTP assumes a virtual file system together with a communication protocol. The bFTP has an aspect that it can be easily implemented, because it assumes such a simple file system that will allow "a binary file to be stored with its name".

Moreover, bFTP is characterized by a query function which allows to query about functions and properties of a device and the image format available in the theme of this section, i.e., the image transfer. This query function simplifies the user interface of a digital camera, and allows the most suitable data of an image to be transferred between the digital cameras or printers faced with each other. In addition, this function makes it possible for the user to transfer, communicate or print suitable image data regardless of the difference in platform or model just by “selecting a photograph to be sent and pushing a transmission button”.

1.4. Image Format UPF (Uni Picture Format)

As mentioned earlier, UPF is the standard of an image format not included in the category of the IrDA standard. The IrDA standards are originally provided for defining and standardizing a protocol in connection with infrared communications. Accordingly, it is out of the scope to define the contents of an image transfer. However, in order to ensure mutual connectivity as an application of a digital camera, it is required to decide an image format so that image data sent via infrared communication is reliably reproduced. Therefore, in advocating IrTran-P as a standard to IrDA, the specific contents of an image format of IrTran-P are defined and described in an appendix.

UPF is an image file format based on the JPEG base line. JFIF, which is a JPEG file, makes an image of various color forms available and employs a high level of compression scheme. For this reason, JFIF may be regarded as the industry standard of an image file format today. Since JPEG is a format enabling a variety of color forms, a compromise is required to some extent in order to realize the standard at a low cost, such as adopting only a part of the format as the standard. In UPF, among the formats included in the base line of JPEG, the format reliably allowing the devices at least to display and mutually transfer an image is defined as an indispensable one, and others are regarded as an option. For more details, please refer to the sections of UPF in the later part of this document.

1.5. Study of Approach of UPF

As well known, it is characteristic of a digital camera that all the data accompanying a photograph taken by a digital camera, such as a photo-taking date/time and the orientation (direction) of an image and other additional data, cannot be covered by the data within the JPEG format. In view of such a background, UPF is designed so that data is separated and stored on its own header arranged in the file without changing the image data scheme of JPEG Base Line at all. In addition, the header has expandability and allows a vendor-unique function to be added thereto. This makes it possible to separate the data necessary for a digital video camera from the data necessary for display and expansion of an image, which is advantageous in that the existing JPEG techniques can be used as it is. In a compact device like a digital camera, when using existing hardware or software, e.g. in the case where an algorithm of JPEG compression/expansion or the like is performed by hardware or is fixedly used as firmware, it is undesirable to change JPEG itself.

As a further advanced step, UPF is designed so that additional data on an ambiguous point within the data of JPEG scheme is arranged in the header part. The additional data includes factors such as white level, black level and color-difference signal, necessary for reproducing an image with correct brightness and color.

Though the format of a digital camera is being examined by various organizations, a conclusive decision has not been made yet. In many cases, there is proposed an arrangement such as newly addition of a tag to JPEG or the like. However, it will take a long time to reach the conclusion satisfactory for all the companies concerned, which is not a timely manner in view of the movements of the market today. The approach of making the best use of existing standard, wherein the data necessary for a digital camera is separated and added so as to assure expandability, is more realistic than the approach of waiting for the standard to be decided at last.

Though UPF is defined as an appendix, it is indispensable for the IrTran-P standard to be able to support an image format of UPF scheme.

2. Usage Model and Operations of IrTran-P

This section describes how “operations” of a user are reflected onto “SCEP/bFTP”, or data transfer procedures of IrTran-P, when IrTran-P transfers picture data.

In IrTran-P, an operation which transfers picture data from a digital camera is started by a sender.

(1) Operation by User

A user operates a digital camera of the sender to cause the digital camera to be in a transmission state, with the use of “selection of a specific picture” and a “transmission button”.

It is supposed that the device of a receiver is always in a receiving state or caused to be in a picture data receiving state by a “reception button”.

(2) Establishment of Session by SCEP

The digital camera of the sender carries out a discovery procedure by IrDA protocols and performs a connection for physical to IrCOMM layers of IrDA protocols in accordance with IrDA protocols. When a transmission path of IrDA is established, SCEP makes a “session establishment request” from the sender toward a digital camera, printer or PC of the receiver. If the receiver is implemented with SCEP, it must make a response of either “session established” or “session establishment rejected”.

(3) Query Operation by bFTP (Query function)

When a session by SCEP is established, the digital camera of the sender issues a Query request in order to recognize picture processing functions of the receiver. The information mutually exchanged by the Query request includes the transmittable/receivable picture size, the picture compression format and the basic picture size of the device. Since this information is exchanged before transfer of picture data, the picture data can be transferred in “the most reasonable format” between devices of different platforms.

In IrTran-P, a “mandatory format” is defined among the picture data formats of both sides, whereby a picture can be reliably exchanged between device of different grades or manufactures.

Furthermore, it is possible to query about the power supply condition of the device, the receivable data capacity and the like. This makes it possible to deal with applications of a portable system.

(4) Transfer of Picture Data by bFTP

Transfer of picture data is started since the most appropriate picture format for both of the sender and the receiver is determined by Query.. SCEP performs the data transfer at a high transmission rate by making use of IrDA protocols. After the file transfer is completed, next picture data may be subsequently transmitted, or a session may be disconnected by SCEP. (Accordingly, even a simple model can transmit more than one pictures in succession.)

(5) Completion of Session by SCEP

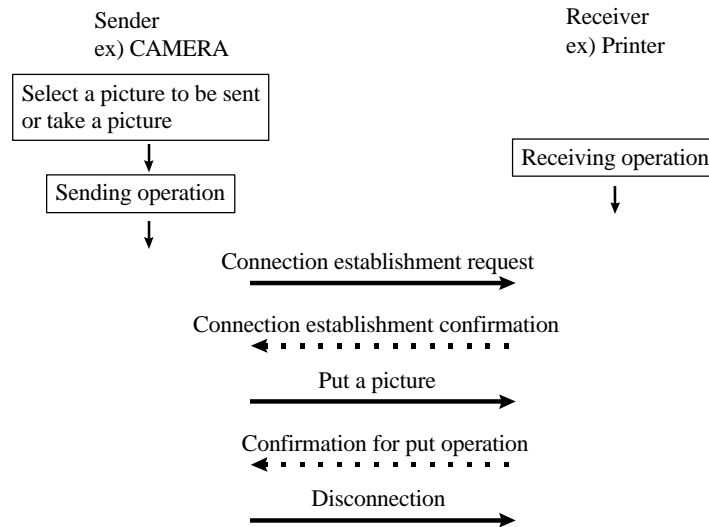
When the picture transfer has been completed, the digital camera of the sender disconnects a session by SCEP. Thereafter, a disconnection request is issued for IrCOMM and lower layers of IrDA protocols, and the picture transfer operation is completed.

Next, three exemplary simple operations will be specifically described using services of SCEP/bFTP.

2.1. Usage model 1 Simple model

Following chart is the simplest usage model. This model describes the case of picture sending machine transmits only 1 picture and doesn't inquire the receivable picture format, rest of the memory nor the reminding battery life.

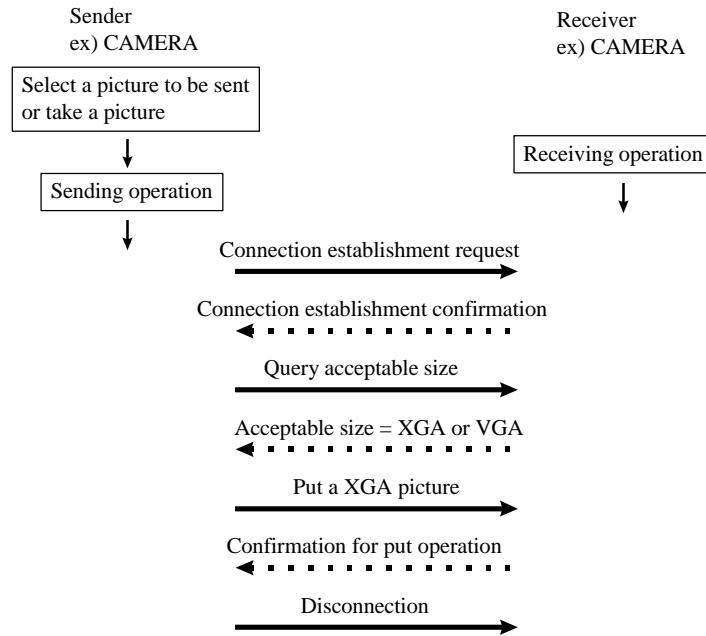
Sender is possible to send the mandatory picture format only.



2.2. Usage model 2 The case of sending the non-mandatory size

When the sender have some picture format possible to send and different from the mandatory format, it is possible to use the query service for obtaining the receivable picture format of the receiver and select the picture format for the actual picture send.

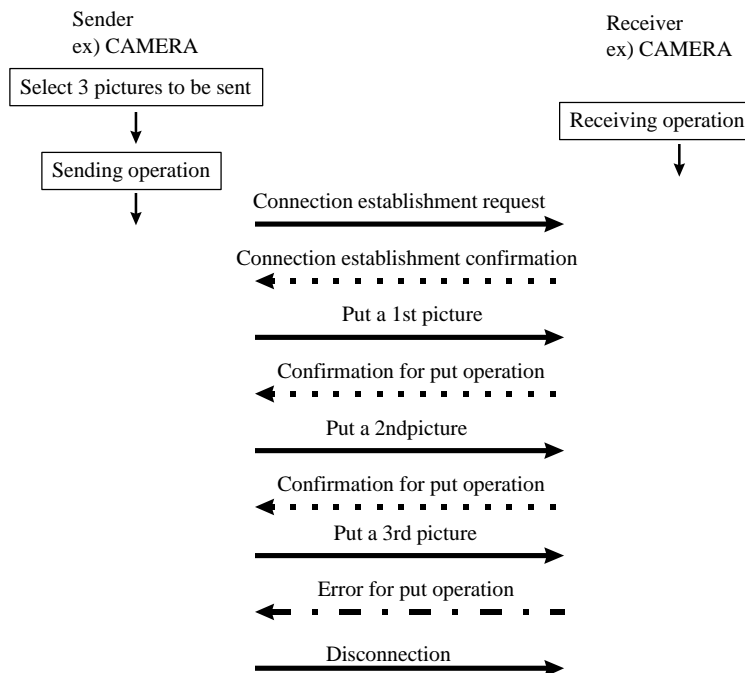
Following chart is the usage model when the XGA size picture was transferred. In this case, the receiver can receive not only the mandatory format picture but also the XGA size format picture. Sender can send not only the mandatory format picture but also the XGA size format picture. Sender recognizes that the receiver can receive the XGA size picture from the information in the query command response, and sends the XGA picture. Though it is left to the picture sending machine which picture format to be sent, in the most of the case, the highest priority of the receivable picture format should be chosen.



2.3. Usage Model 3 The case of sending the plural pictures

Put command can be sent repetitively for sending more than one pictures. If the picture receiving machine can't receive pictures over the certain number, then return the error code on the put response. The sender which has received an error response should terminate communication.

Following chart is the usage model when the plural pictures were transferred. In this case, the receiver can receive up to 2 pictures, but sender tried to send 3 pictures. Since receiver can't receive the 3rd picture, it skips to read the all of 3rd picture data and return the error response. The sender resigned to send 3rd picture and terminate communication.



2.4. Closing Remark

As apparent from above description, the picture data exchange by IrTran-P is though quite simple, yet powerful as well. Within the application range of a digital camera in the consumer market, this implementation is sufficiently effective by itself. As for bFTP, the definition is such that it can be expanded to support formats other than this simple one, and therefore will grow with functional development in the future.

3. Protocol (SCEP & bFTP)

IrTran-P is to place a SCEP and a bFTP which are necessary for exchanging an image between devices and mutually recognizing properties of the devices, as the upper layer of IrSIR, IrLAP, IrLMP and IrCOMM.

3.1. Introduction

SCEP offers a connection management and command management service on a reliable stream-type transport layer. bFTP is a definition of a protocol for providing file transfer service for SCEP.

3.1.1. Overview

The connection management service of SCEP provides a user with a function of invoking the PDU(Protocol Data Unit) size receivable at a time and an authentication function using a password encoded by the user's name and MD5(Message Digest 5). The command management service provides with a user functions of returning the result of command execution to the user, interrupting command execution, and segmenting or reassembling PDU so as to be receivable by the other side.

The file transfer service is performed by a file transmission function and a file server function. The file transmission function is the sub-set of the file server function. This document defines the functions for realizing PUT model. Put Command to transmit a page of file is solely defined as the file transmission function. Query Command to inquire about processing abilities of an application on the responding side is solely defined as the file server function.

3.1.2. Terminology

The following terms are used throughout this section.

Primary the entity that requests establishment of a SCEP connection.

Secondary the entity that responds to the request for establishment of a SCEP connection.

Requester the entity that transmits command request by using an established connection.

Responder the entity that receives command request by using an established connection.

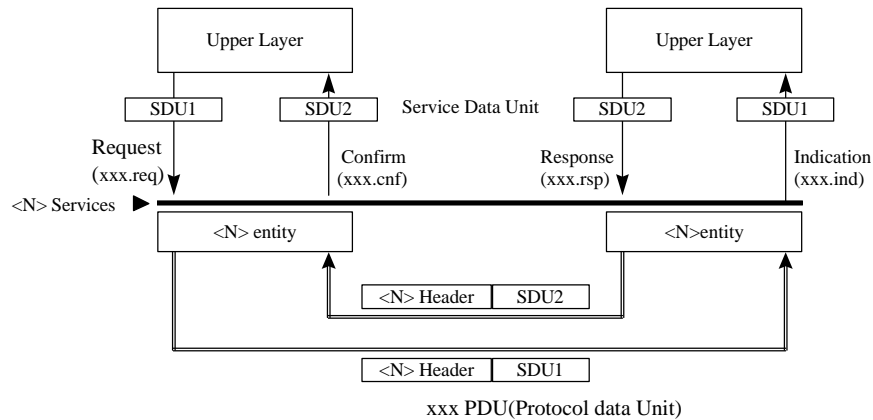
3.1.3. Service Model

SCEP and bFTP employs four generic types of service primitive:

1. Request: Passed from the Upper Layer to invoke a service.
2. Indication: Passed from <N> entity to the Upper Layer to indicate an event or to notify the Upper Layer of an <N> entity initiated action.
3. Response: Passed from the Upper Layer to acknowledge some procedure invoked by an indication primitive.
4. Confirm: Passed from <N> entity to the Upper Layer to convey the results of the previous service request.

<N> entity means SCEP or bFTP. <N> entity uses these primitives to communicate with the upper layer in order to manage the communications processes.

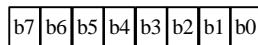
These primitives are shown graphically here.



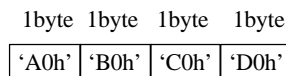
3.1.4. Bit and Byte Ordering

This section regards frames as collections of bytes (octets) with each byte being composed of 8 bits numbered 0-7. Bit 0 is always the least significant bit (LSB) and bit 7 is always the most significant bit (MSB). Bytes are represented throughout this section in the following forms:

- Diagrammatic - a byte is represented by a rectangle. In some cases bit fields have special meaning and are indicated for clarity. The most significant bit is the bit on the left and the least significant bit is the bit on the right. An example is given below.



- Hexadecimal - a byte is represented with two hex digits with the least significant nibble on the right, the most significant nibble on the left, and both digits suffixed by 'h'. An example is the value 5 which is written as 05h.
- Multiple bytes form - is represented as a rectangle with slots for each byte. The least significant byte is on the right and the most significant byte is on the left. The multiple bytes example shows a four bytes sequence of 'A0h', 'B0h', 'C0h', 'D0h' :



3.1.5. References

- [1] ISO7498, "Information processing systems - Open Systems Interconnection - Basic Reference Model", 1984
- [2] Infrared Data Association, 'IrCOMM': Serial and Parallel Port Emulation over IR (Wire Replacement)", October 1995

- [3] RFC1321, "The MD5 Message-Digest Algorithm", 1992
- [4] IEEE EUI-64, "Extended Unique Identifier, 64bits"
- [5] ISO/IEC646, "Information Technology - 7-bit coded character set for information interchange", 1991
- [6] ISO8859-1, "Information Processing - 8-bit Single-byte coded graphic character sets - Part 1: Latin alphabet No. 1", 1987
- [7] ISO/IEC 10918-1, "Information technology- Digital compression and coding of continuous-tone still images: Requirements and guidelines (JPEG)", 1994

3.2. SCEP (Simple Command Execute Protocol)

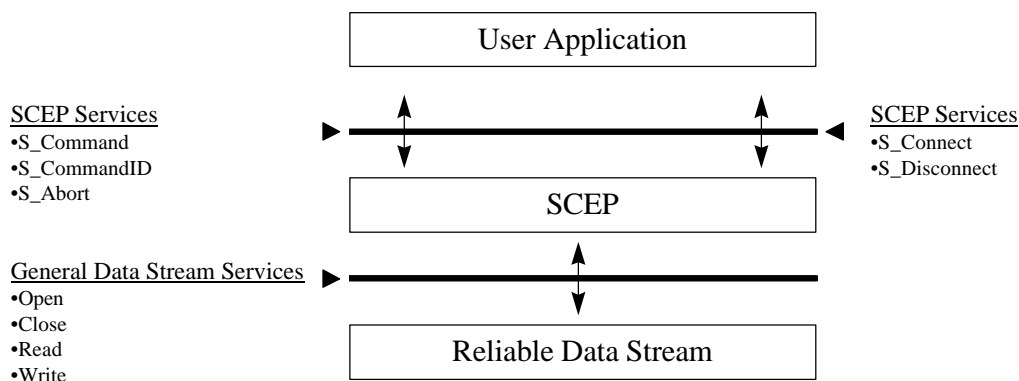
This section defines SCEP, which execute communication job (command) and reports back the results between applications.

3.2.1. Service Definition

SCEP is intended to define a protocol to manage a connection and commands.

3.2.1.1. Service Model

The figure below shows a model of how SCEP fits into a typical system. This figure represents the SCEP reference model.



The elements for the SCEP reference model are described below.

SCEP Services SCEP Service primitives which are provided to the SCEP user.

SCEP Provides a connection management, command management and segmentation & reassemble mechanism.

Command The element executed on server application.

CommandID Identifier to manage the executing command. The executing commands have different CommandID.

MachineID Identification number to tell one machine from another, and must be described in IEEE EUI64 format. In the case of a machine not requiring individual recognition, the machine does not need to have MachineID. As for the machine without MachineID, 00h is entered in the field of MachineID (eight octets).

PID Identifier to distinguish the server process which is one of the application.

General Transport Services Service which is provided by reliable data stream.

Reliable Transport Layer Provide a reliable data stream mechanism. An example is the IrCOMM defined by IrDA.

3.2.1.2. SCEP Service Primitives

The SCEP interface provides the following services.

- Connect
- Disconnect
- Command
- CommandID
- Abort

3.2.1.2.1. Connect services

S_Connect.req (Primary MachineID,
Secondary MachineID,
Primary CFLG,
Primary NegInf)

S_Connect.ind (Primary MachineID,
Secondary MachineID,
Primary CFLG,
Primary NegInf)

S_Connect.rsp (AckOrNackFlag,
Primary MachineID,
Secondary MachineID,
Secondary CFLG,
Secondary NegInf)

S_Connect.cnf (AckOrNackFlag,
Primary MachineID,
Secondary MachineID,
Secondary CFLG,
Secondary NegInf)

The Connect services are used to establish a communication path with a peer SCEP system. This is a confirmed service. Upon receipt of an S_Connect.ind primitive the Secondary must either accept or reject the incoming connection. Connections are accepted by an invocation of S_Connect.rsp (Cack) or are rejected by an invocation of S_Connect.rsp (Cnack) or S_Disconnect.req with a reason of 'User Disconnect'.

Parameter used in this definition are as follow.

PrimaryMachineID, SecondaryMachineID

MachineID of the side requesting Connection is referred to as Primary MachineID, and the MachineID of the side receiving Connection as Secondary MachineID. When Secondary MachineID is not specified, 00h is entered in the field of Secondary MachineID.

CFLG

A flag indicating whether or not being able to be Responder. If not being able to be Responder, neither Req PDU nor Rqs PDU is acceptable.

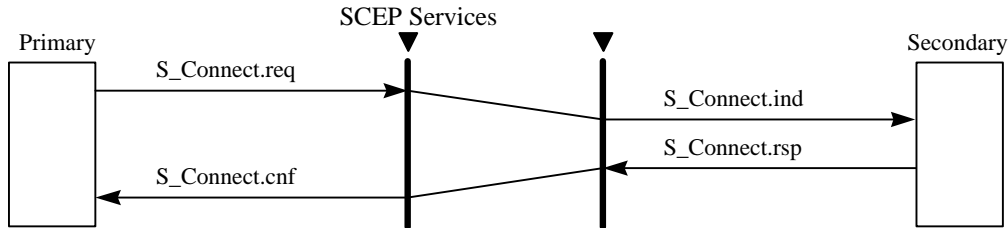
NegInf

Negotiation information of SCEP connection which is value of a receivable maximum PDU size and authentication data including the user name and password..

AckOrNackFlag A flag indicating a permission or rejection to the connection request

Ack Accepting a connection request

Nack Refusing a connection request



3.2.1.2.2. Disconnect services

S_Disconnect.req (ReasonCode)

S_Disconnect.ind (ReasonCode)

The Disconnect service is used to close the connection between SCEP entities. The Disconnect services are a non-confirmation type service. The user of SCEP is always permitted to use this service whenever it wishes to release the connection. The Disconnect service is used in these cases.

- If a SCEP user wishes to release or abort a SCEP connection with a peer SCEP entity, it will use this service.
- If the underlying communication path is disconnected, SCEP will notify the SCEP user via an S_Disconnect.ind.
- A SCEP user uses Disconnect service to refuse an incoming connection.
- An S_Disconnect.ind is issued if the underlying layer failed to establish a connection.

Parameter used in the Disconnect services are as follows.

ReasonCode

This parameter indicates the reason why a link is disconnected or why a connection is refused. 'ReasonCode' should be one of the following:

Unspecified Reason

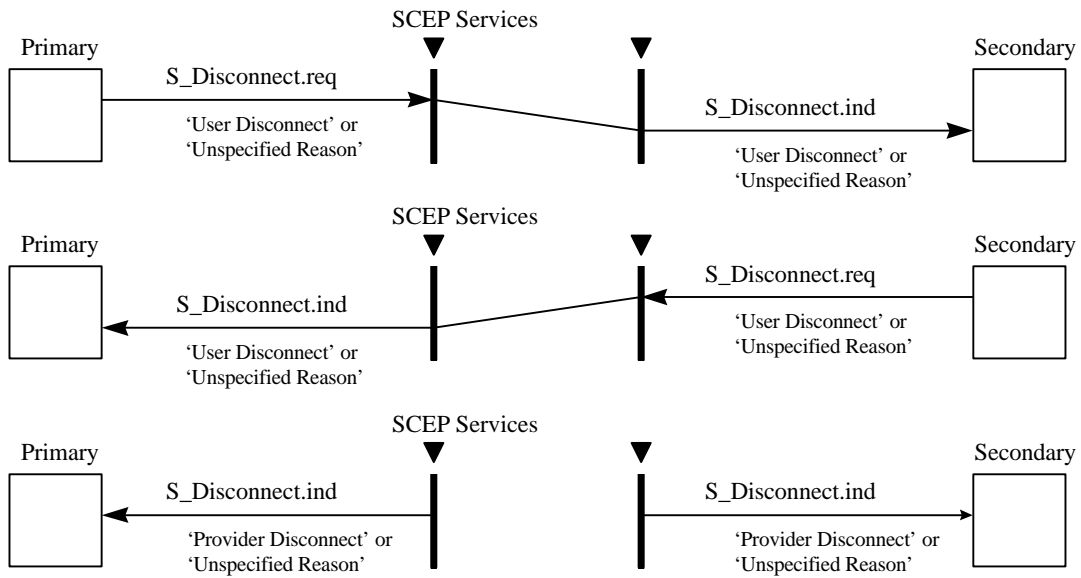
The reason is unspecified in this document.

User Disconnect

The Responder refuse to establish a SCEP connection, or a SCEP user wishes to disconnect the existing connection.

Provider Disconnect

The provider of SCEP connection (SCEP or an underlying protocol stack) causes a disconnection.



3.2.1.2.3. Command services

S_Command.req (Requester MachineID,
Responder MachineID,
Requester PID,
Responder PID,
UserData)

S_Command.ind (Requester MachineID,
Responder MachineID,
Requester PID,
Responder PID,
CmdID,
UserData)

S_Command.rsp (AckOrNackFlag,
Requester MachineID,
Responder MachineID,
Requester PID,
Responder PID,
CmdID,
UserData)

S_Command.cnf (AckOrNackFlag,
Requester MachineID,
Responder MachineID,
Requester PID,
Responder PID,
CmdID,
UserData)

The Command service is used to convey commands and results between SCEP users. The Command service is a confirmed service.

The parameter used in this service is as follows.

RequesterMachineID, ResponderMachineID

ID for identifying a machine. MachineID used in S_Connect service must be used. MachineID of the side requesting Command is referred to as Requester MachineID, and the MachineID of the side receiving Command as Responder MachineID.

RequesterPID, ResponderPID

ID for identifying a SCEP user. RequesterPID is ID by which a user of S_Command.req can be identified. S_Command.cnf is given to the user specified at RequesterPID. ResponderPID must be the ID which allows identification of a server (SCEP user) which can execute a command requested by Requester. bFTP server's PID = 8.

UserData

Data to be sent

AckOrNackFlag

indicates whether or not command execution is normally completed

Ack indicates that command execution is normally completed

Nack indicates that command execution is abnormally terminated

3.2.1.2.4. CommandID services

S_CommandID.ind (Requester PID,
Responder PID,
CmdID)

The protocol machine of the Requester which has received S_Command.req must generate and manage IDs identifying S_Command.req (i.e., CmdID). In addition, after generating an ID, this ID must be returned to a user of S_Command.req through S_CommandID.ind service. When receiving a command interrupt request and a command response, a specified CmdID must be deleted from a management table. When disconnecting the connection, every CmdID should be deleted from the management table. The algorithm realizing these depends on implementation.

The protocol machine of the Responder must preserve and manage the received CmdID. When receiving a command interrupt and a command response, a specified CmdID must be removed from a management table. When disconnecting the connection, every CmdID must be deleted from the management table. The algorithm realizing these depends on implementation.

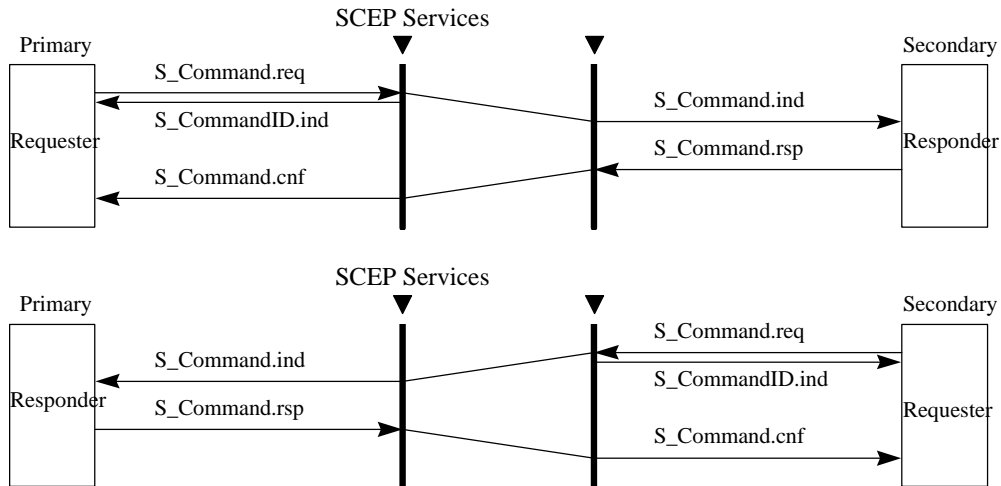
The parameter used in this service is as follows.

RequesterPID, ResponderPID

ID for identifying a SCEP user. RequesterPID is ID allowing identification of a user of S_Command.req. S_Command.ind is notified to a user specified at ResponderPID. ResponderPID must be ID allowing identification of a server (a SCEP user) which can execute a command requested by a Requester.

CmdID

ID for identifying a command. This is used when using S_Abort service.



3.2.1.2.5. Abort services

S_Abort.req (Requester MachineID,
 Responder MachineID,
 Requester PID,
 Responder PID,
 CmdID)

S_Abort.ind (Requester MachineID,
 Responder MachineID,
 Requester PID,
 Responder PID,
 CmdID)

It offers a function of interrupting command execution. The Abort services are a non-confirmation type service.

The parameter used in this service is as follows.

RequesterMachineID, ResponderMachineID

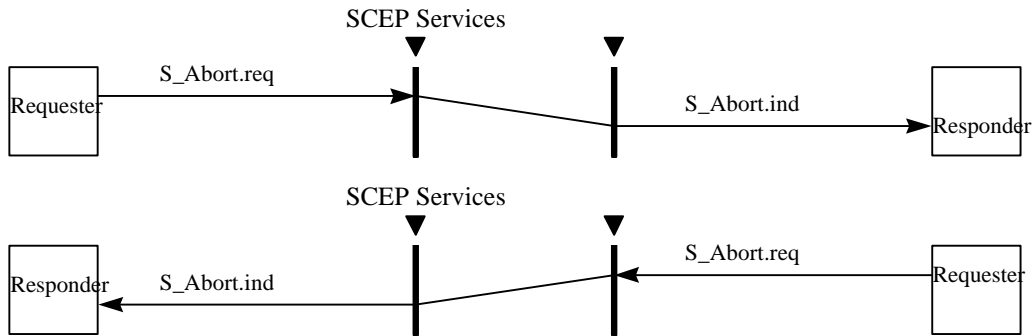
ID for identifying a machine. MachineID used in S_Connect service must be used.

RequesterPID, ResponderPID

ID for identifying a SCEP user. RequesterPID is the ID which allows identification of a user of S_Abort.req. ResponderPID must be ID which identifies a server (a SCEP user) which can execute a command requested by a user of S_Command service.

CmdID

ID for identifying the command of which execution must be interrupted



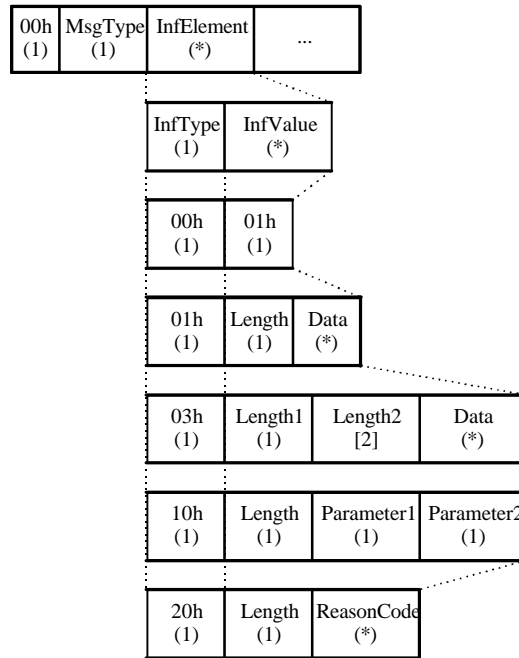
3.2.2. SCEP Protocol Data Units

3.2.2.1. Definitions

SCEP PDU is constructed by SCEP header, Command header and User data.

3.2.2.1.1. SCEP Header Structure

SCEP header is constructed by MsgType and InfElements. SCEP header structure is below:



MsgType should be one of the following:

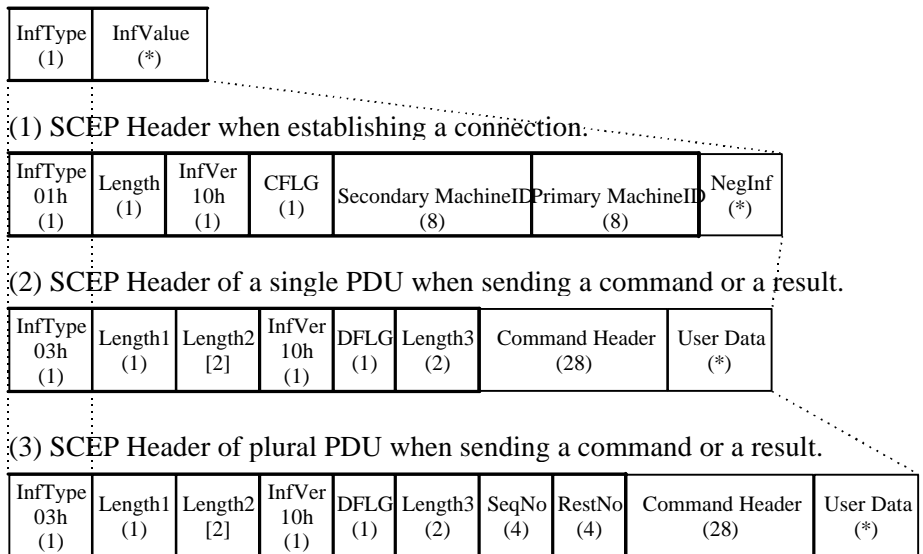
MsgType 10h: Connection establishment request
 11h: Connection establishment confirmation
 20h: Data (Command)
 30h: Disconnection
 Others: reserved

InfType should be one of the following:

InfType 00h: Version of MsgType
 Used only when MsgType is a connection establishment request
 01h: Negotiation Information
 Used only when MsgType is a connection establishment request or an acceptance of connection establishment request.
 03h: UserData
 Used only when a MsgType is data.
 Length2 can exist only when Length1 is FFh.
 10h: Extend in the future
 Used only when MsgType is connection establishment request
 20h: Reason
 Used only when MsgType is disconnection
 Others: reserved

(n): n bytes, *: variable length, []: optional

The fields included in this document are described in the network byte order (Big-endian).



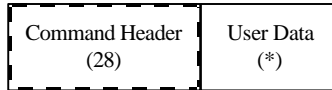
- The details of Data included in InfValue, when InfType is Negotiation Information and UserData.
- InfVer: Version of InfType
- Length2 must exist only when Length1=FFh. If Length1 has a value other than FFh, the Length2 field must not exist.
- When InfType is UserData, PDU exceeding the maximum receivable size requested at the time of connection establishment must be segmented so as to be accommodated within the size, and

SCEP Header shown with (3) must be used. In the case of single PDU, SCEP Header shown with (2) must be used.

- When Length3=00h, the Command Header field and the succeeding UserData field are not present.

3.2.2.1.2. Command Structure

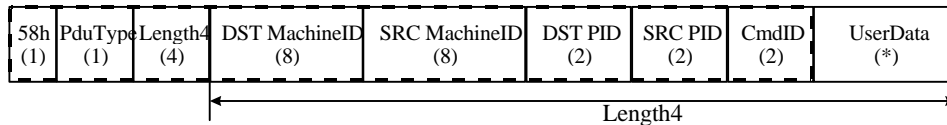
Command is constructed Command structure is below:



- The data structure to request for command execution and the result of command execution will be described.
- This structure is composed of Command Header and User Data.
- User Data is given through S_Command service.
- Information necessary for the command execution is stored on User Data.

3.2.2.1.3. Command Header Structure

Command header structure is below:



- PduType**
- b7b6 00 : *Req PDU*
 - b7b6 01 : *Rpl PDU(ACK)*
When command execution is normally completed,
Command Header + User Data stores results of the command execution.
 - b7b6 10 : *Rpl PDU(Nack)*
When command execution is failed,
Command Header + User Data stores causes of failure.
 - b7b6 11: *Abt PDU*
Command execution is interrupted.
 - b0 - b5: reserved

- DST MachineID: MachineID of the side receiving PDU including Command Header
- SRC MachineID: MachineID of the side sending PDU including Command Header
- DST PID: Program ID of the side receiving User Data succeeding Command Header
- SRC PID: Program ID of the side sending User Data succeeding Command Header

- **CmdID:** Command ID. ID for identifying a command, needed to be generated and managed by a SCEP protocol machine. A user uses this ID to identify a command when using an S_Abort service.

3.2.2.1.4. Parameters

CFLG The below combinations are available.

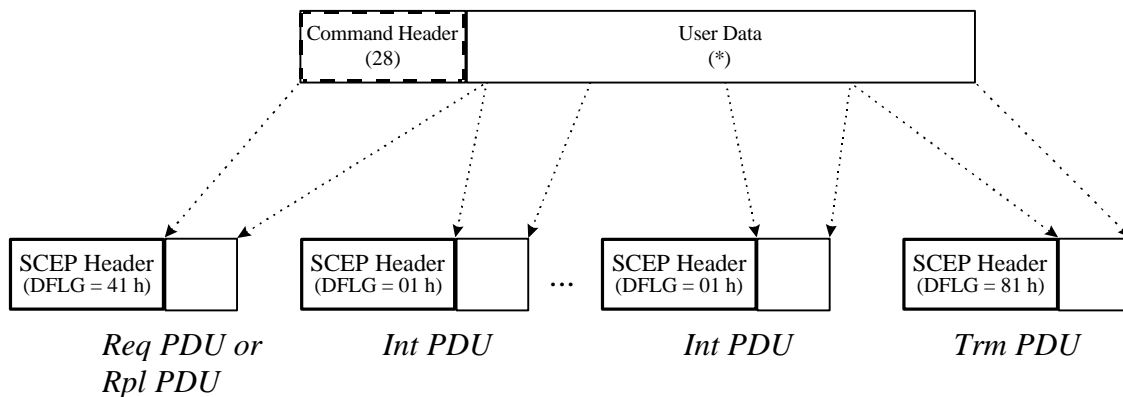
- CFLG = 00h: the machine can issue a command but cannot execute a command.
- CFLG = 04h: the machine can issue and execute a command.
- Others: reserved.

DFLG Permitted Combinations and their meanings

- DFLG = C1h: When PDU is not segmented (Single PDU)
- DFLG = 41h: The first segmented PDU
- DFLG = 01h: Intermediate segmented PDUs
- DFLG = 81h: The last segmented PDU
- DFLG = C2h: Communication Interruption
- DFLG = C3h: Reject to request connection
- Others: reserved.

3.2.2.1.5. Segmentation and Reassembling

If the sending PDU size is greater than the receivable maximum PDU size which is negotiated at connection establishment, the sending PDU must be segmented not greater than the receivable maximum PDU size. Con PDU, Cnack PDU, Cnack PUD, Dis PDU, Abt PDU, Stp PDU must not be segmented.



- Command Header + User Data is segmented into plural *PDU* s to be sent out. DFLG is used for identifying the first, intermediate and last PDU.
- Only when MsgType included in SCEP Header is 20h, it can be segmented. Otherwise, it must not be segmented. (Only the length equal to or less than the maximum receivable size is permitted.)

Attribute is omitted, 512bytes becomes the maximum receivable PDU size. The maximum acceptable PDU size of sender may be different from that of the receiver. The following sizes are defined.

n = `1': 512bytes (default), `2': 1024bytes,
`3': 2048bytes, `4': 4096bytes

id: (Products Identification Character String)<CR><LF>

Products Identification Character String: The character string specified by the products vendor. It is recommended that the type of machine (model number, type code e.g.) is suffixed to the character string of the company name.

nm: (User Name)<CR><LF>

User Name: It is possible to specify any bytes string except <CR><LF>. 32 bytes at maximum.

pw: (Password)<CR><LF>

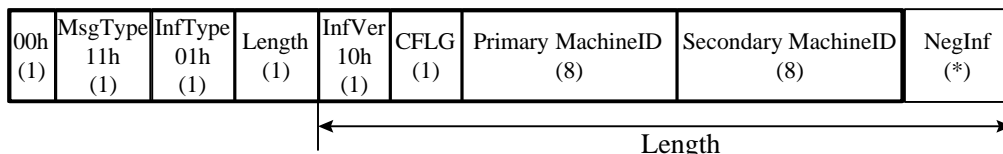
Password: The hexadecimal expression of a fingerprint (16bytes) fetched by encoding the password character string (ASCII code) entered by a user by MD5. In the hexadecimal expression, a space must not be inserted between characters. (For example, the form of `FE 80 FE 80' is not permitted. It should be the form of `FE80FE80')

If the same Attribute name appears at plural times, the Attribute value appearing later becomes effective.

The NegInf are limited to a total encoded size of 228 bytes.

3.2.2.3. Cack PDU

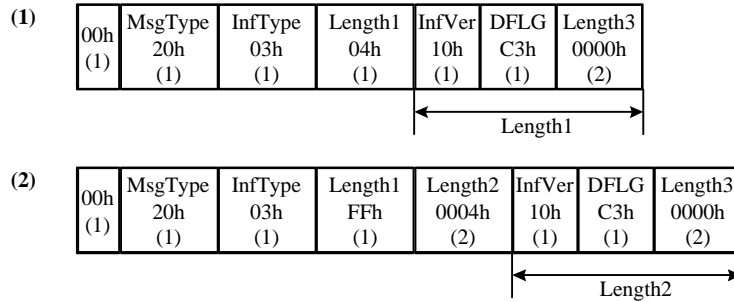
SCEP protocol machine writes Cack PDU packet to the data stream by an invocation of S_Connect.rsp which connection is accepted at the Secondary. Con PDU packet structure is below:



- Connection Establishment Certification PDU
- NegInf are similar to Con PDU.

3.2.2.4. Cnack PDU

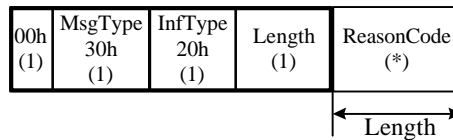
SCEP protocol machine writes Cack PDU packet to the data stream by an invocation of S_Connect.rsp which connection is rejected at the Secondary. Cnack PDU packet structure is below:



- SCEP Connection refusal PDU
Format is (1) or (2).

3.2.2.5. Dis PDU

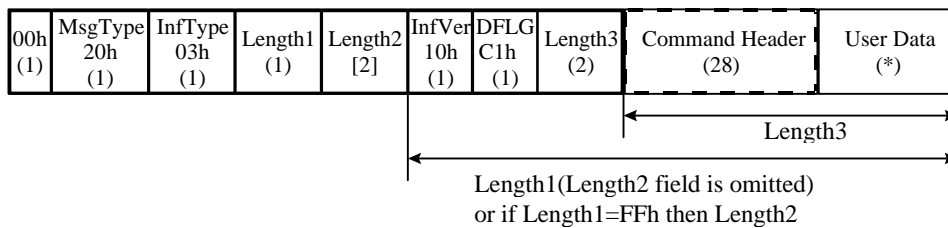
SCEP protocol machine writes Dis PDU packet to the data stream by an invocation of S_Disconnect.req. Dis PDU packet structure is below:



- SCEP Connection Disconnect PDU
- ReasonCode
 - 0000h: Unspecified Reason
 - 0001h: User Disconnect
 - 0002h: Provider Disconnect
 - Others: Reserved

3.2.2.6. Rqs PDU

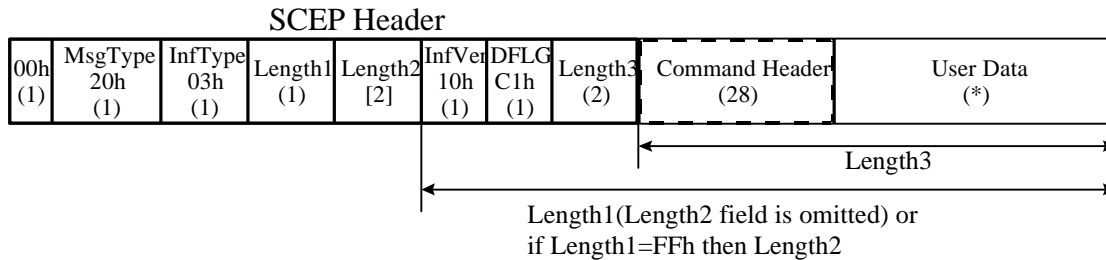
SCEP protocol machine writes Rqs PDU packet to the data stream by an invocation of S_Command.req which the PDU size is not greater than the receivable maximum PDU size of the Responder. Rqs PDU packet structure is below:



- Command Request PDU (Single PDU)
 - DFLG = C1h
 - PduType in Command Header = 00h
- Length1, Length2
 - If Length1 has a value FFh, next two bytes (Length2) indicate the length.
- Length3
 - To indicate the length of Command Header + User Data
- If the whole length exceeds the maximum receivable PDU size, it must be segmented into plural parts to be *Req PDU*, *Int PDU* and *Trm PDU*, respectively.

3.2.2.7. Rps PDU

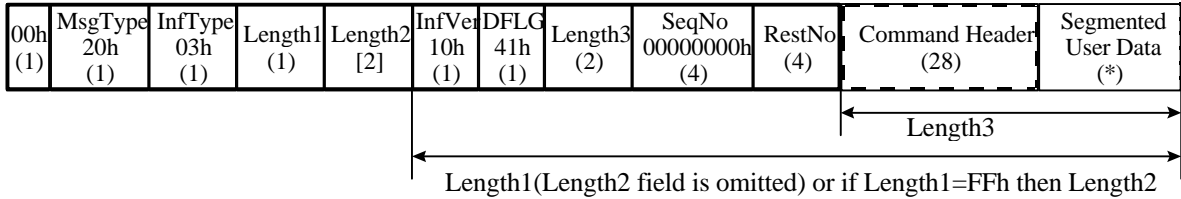
SCEP protocol machine writes Rps PDU packet to the data stream by an invocation of S_Command.rsp which the PDU size is not greater than the receivable maximum PDU size of the Requester. Rqs PDU packet structure is below:



- Result of the Command Execution PDU (Single PDU)
 - DFLG = C1h
 - PduType = 40h(Ack) or 80h(Nack) in Command Header
- Length1, Length2
 - If Length1 has a value FFh, next two bytes (Length2) indicate the length.
- Length3
 - To indicate the length of Command Header + User Data
- If the whole length exceeds the maximum receivable PDU size, it must be segmented into plural parts to be *Rpl PDU*, *Int PDU* and *Trm PDU*, respectively.

3.2.2.8. Req PDU

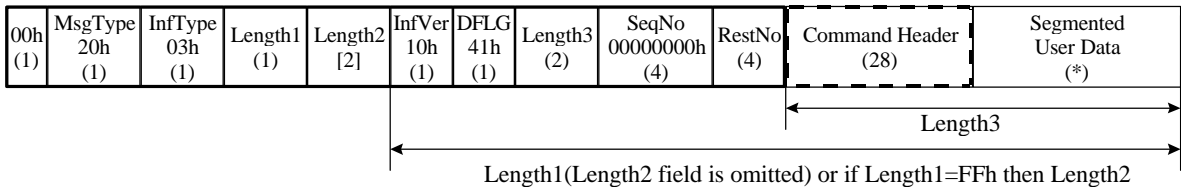
SCEP protocol machine writes Req PDU packet to the data stream by an invocation of S_Command.req which the PDU size is greater than the receivable maximum PDU size of the Responder. When the sending PDU is greater than the receivable maximum PDU size of the Responder, the sending PDU is segmented to Req PDU, Int PDUs and Trm PDU. Req PDU packet structure is below:



- Command Request PDU (The first PDU when User Data is segmented into plural PDUs)
DFLG = 41h
PduType = 00h in Command Header
- Length1, Length2
If Length1 has a value FFh, next two bytes (Length2) indicate the length.
- Length3
To indicate the length of Command Header + Segmented User Data.
- SeqNo: The sequence number of this PDU. 0 is specified at the first PDU.
- RestNo: The remaining number of segmented PDUs. 1 is specified at the last PDU.

3.2.2.9. Rpl PDU

SCEP protocol machine writes Rpl PDU packet to the data stream by an invocation of S_Command.rsp which the PDU size is greater than the receivable maximum PDU size of the Requester. When the sending PDU is greater than the receivable maximum PDU size of the Requester, the sending PDU is segmented to Rpl PDU, Int PDUs and Trm PDU. Rpl PDU packet structure is below:



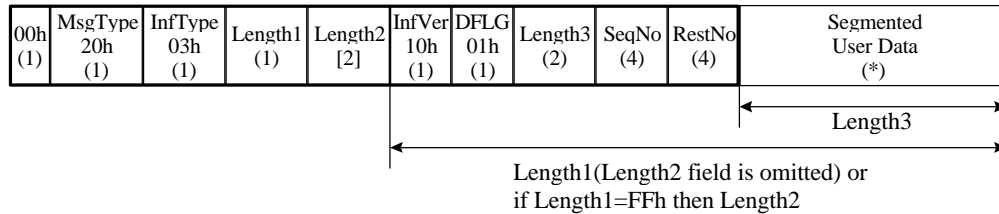
- Command Execution Result Return PDU (The first PDU when User Data is segmented into plural PDUs)
DFLG = 41h
PduType = 40h(Ack) or 80h(Nack) in Command Header
- Length1, Length2
If Length1 has a value FFh, next two bytes indicate the length (Length2).
- Length3
To indicate the length of Command Header + Segmented User Data.
- SeqNo: The sequence number of the PDU. 0 is specified at the first PDU.
- RestNo: The remaining number of segmented PDUs. 1 is specified at the last PDU.

3.2.2.10. Int PDU

SCEP protocol machine writes Int PDU packet to the data stream by an invocation of PDUConf which is internal event of the segmentation mechanism. When the sending PDU is greater than the receivable maximum PDU size of the Responder, the sending PDU is segmented to Req PDU, Int PDUs and Trm PDU by an invocation of S_Command.req. When the sending PDU is greater than the receivable maximum PDU

size of the Requester, the sending PDU is segmented to Rpl PDU, Int PDUs and Trm PDU by an invocation of S_Command.rsp.

The segmentation mechanism invokes PDUConf event after sending out Req PDU or Int PDU until sending out the last PDU which is Trm PDU by an invocation of S_Command.req. The segmentation mechanism invokes PDUConf event after sending out Rsp PDU or Int PDU until sending out the last PDU which is Trm PDU by an invocation of S_Command.rsp. Int PDU packet structure is below:

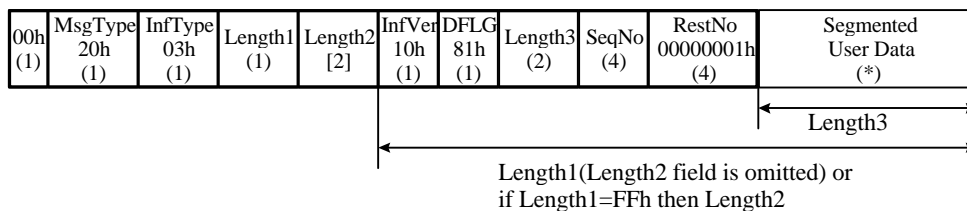


- Intermediate PDUs of command request or result of the command execution
(Intermediate PDUs when User Data is segmented into plural PDUs)
DFLG = 01h
- Length1, Length2
If Length1 has a value FFh, next two bytes (Length2) indicate the length.
- Length3
To indicate the length of User Data
- SeqNo: The sequence number of the PDU. 0 is specified at the first PDU.
- RestNo: The remaining number of segmented PDUs. 1 is specified at the last PDU.

3.2.2.11. Trm PDU

SCEP protocol machine writes Trm PDU packet to the data stream by an invocation of PDUConf which is internal event of the segmentation mechanism. When the sending PDU is greater than the receivable maximum PDU size of the Responder, the sending PDU is segmented to Req PDU, Int PDUs and Trm PDU by an invocation of S_Command.req. When the sending PDU is greater than the receivable maximum PDU size of the Requester, the sending PDU is segmented to Rpl PDU, Int PDUs and Trm PDU by an invocation of S_Command.rsp.

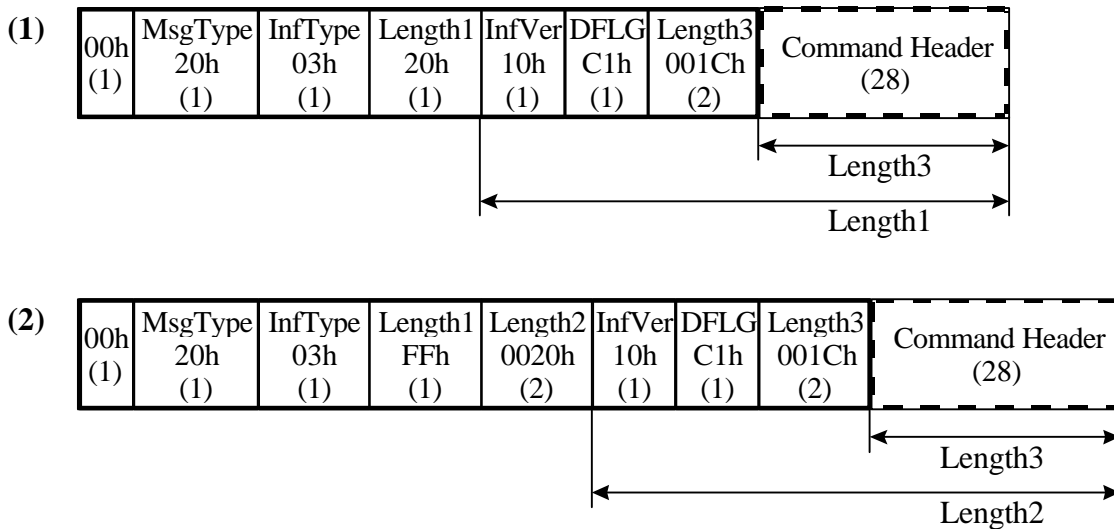
The segmentation mechanism invokes PDUConf event after sending out Req PDU or Int PDU until sending out the last PDU which is Trm PDU by an invocation of S_Command.req. The segmentation mechanism invokes PDUConf event after sending out Rsp PDU or Int PDU until sending out the last PDU which is Trm PDU by an invocation of S_Command.rsp. Trm PDU packet structure is below:



- The last PDU of Command Issue or Command Execution Result (The last PDU when User Data is segmented into plural PDUs)
DFLG = 81h
- Length1, Length2
When Length1 has a value FFh, next two bytes (Length2) indicate the length.
- Length3
To indicate the length of Segmented User Data
- SeqNo: The sequence number of the PDU. 0 is specified at the first PDU.
- RestNo: The remaining number of segmented PDUs. 1 is specified at the last PDU.

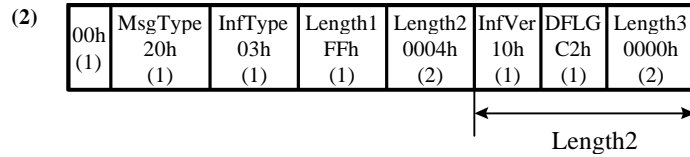
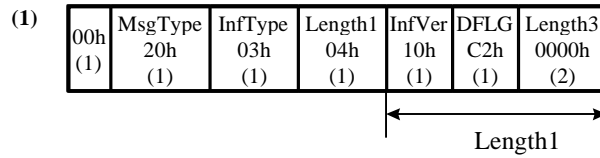
3.2.2.12. Abt PDU

SCEP protocol machine writes Abt PDU packet to the data stream by an invocation of S_Abort.req. Abt PDU packet structure is below:



- Command execution abort PDU
The format is (1) or (2).
DFLG = C1h
PduType = C0h
- To interrupt the execution of SCEP command specified at DST PID and CmdID in Command Header.
PDU when Abort.req is used after all the PDUs concerned with command request have been sent out.

3.2.2.13. Stp PDU



- Transmission interrupt PDU of command execution or result of the command execution.
Format is (1) or (2).
DFLG = C2h
- During transmission of PDU concerned with SCEP command execution, it is sent out to inform the transmission interruption.
- When Abt PDU is sent out and the Responder tries to interrupt the SCEP command execution specified at DST PID and CmdID, if a part of PDU concerned with the execution result has been already sent out, it is sent out to interrupt this result and to inform the side started receiving the result that the receiving be interrupted.

3.2.3. State Definition and Transitions

This section contains a state transition table based on the SCEP service primitives described above. Descriptions of the states, events and actions are included.

3.2.3.1. State Transition Table

The state transition table of SCEP is given below. Initial state is CLOSED. When the action is not described, the input event is ignored and the state dose not transit.

State Transition Table of Connection Management

	<i>CLOSED</i>	<i>WFCC</i>	<i>WFCR</i>	<i>OPEN</i>
SCONreq	CON <i>WFCC</i>			
SCONrsp			p1:CAACK <i>OPEN</i> ; p2:CNACK <i>CLOSE</i> ;	
SDISreq		DIS <i>CLOSED</i>	DIS <i>CLOSED</i>	DIS <i>CLOSED</i>
CAACK		p4:SDISind DIS <i>CLOSED</i> ; not p4:SCONcnf(ACK) <i>OPEN</i> ;		
CNACK		SCONcnf(NACK) <i>CLOSED</i>		
CON	p3:CNACK <i>CLOSED</i> ; not p3:SCONind <i>WFCR</i> ;	SDISind <i>CLOSED</i>		
DIS	<i>CLOSED</i>	SDISind <i>CLOSED</i>	SDISind <i>CLOSED</i>	SDISind <i>CLOSED</i>

State Transition Table of Command Execution

	<i>OPEN</i>	<i>SREQ</i>	<i>WRSP</i>	<i>RRSP</i>	<i>IRSP</i>	<i>RREQ</i>	<i>EXEC</i>	<i>SRSP</i>	<i>IREQ</i>
SCOMreq	P1:REQ SIDind PDUcnf <i>SREQ</i> ; notP1:RQS SIDind <i>WRSP</i> ;								
SCOMrsp						P6:RPS(Nack) <i>IREQ</i> ;	P3:RPL PDUcnf <i>SRSP</i> ; notP3:RPS <i>OPEN</i> ;		
SABTreq		STP <i>OPEN</i>	ABT <i>OPEN</i>	ABT <i>IRSP</i>	<i>IRSP</i>				
REQ	<i>RREQ</i>	STP <i>OPEN</i>	STP <i>OPEN</i>	SDISind DIS <i>CLOSED</i>		SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>
RPL	<i>IRSP</i>	SDISind DIS <i>CLOSED</i>	<i>RRSP</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>
INT	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	<i>RRSP</i>	<i>IRSP</i>	<i>RREQ</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	<i>IREQ</i>
TRM	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SCOMcnf(Ack) <i>OPEN</i>	<i>OPEN</i>	SCOMind <i>EXEC</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	<i>OPEN</i>
RQS	SCOMind <i>EXEC</i>	STP <i>OPEN</i>	STP <i>OPEN</i>	SDISind DIS <i>CLOSED</i>		SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>
RPS		P5:SDISind DIS <i>CLOSED</i> ; P6:SCOMcnf(Nack) STP <i>OPEN</i> ;	SCOMcnf <i>OPEN</i>	SDISind DIS <i>CLOSED</i>	<i>OPEN</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>
ABT						SDISind DIS <i>CLOSED</i>	SABTind <i>OPEN</i>	STP <i>OPEN</i>	
STP	SDISind DIS <i>CLOSED</i>		SDISind DIS <i>CLOSED</i>	SCOMcnf(Nack) <i>OPEN</i>	<i>OPEN</i>	<i>OPEN</i>	SDISind DIS <i>CLOSED</i>	SDISind DIS <i>CLOSED</i>	<i>OPEN</i>
PDUcnf		P2:INT PDUcnf <i>SREQ</i> ; notP2:TRM <i>WRSP</i> ;						P4:INT PDUcnf <i>SRSP</i> ; notP4:TRM <i>OPEN</i> ;	

3.2.3.2. State Definitions

The state definition for SCEP is given below.

States of Connection Management

<i>CLOSED</i>	Disconnection
<i>WFCC</i>	Wait for CackPDU Reception
<i>WFCR</i>	Wait for S_Connect.rsp
<i>OPEN</i>	SCEP Connection Already Set

The State of Command Execution

<i>SREQ</i>	Sending Request
<i>WRSP</i>	Waiting for Reply
<i>RRSP</i>	Reply being Received
<i>IRSP</i>	Ignoring Reply: Command Execution Interrupted by S_Abort.req Reply being Received, but Reply being Received is Discarded
<i>CLOSED</i>	Disconnection
<i>RREQ</i>	Request being Received
<i>EXEC</i>	Command being Executed
<i>SRPS</i>	Reply being Transmitted
<i>IREQ</i>	Ignoring Request

3.2.3.3. Event Descriptions

The input and output event for SCEP are given below.

Input Event of Connection Management

SCONreq	S_Connect.req
SCONrsp	S_Connect.rsp
SDISreq	S_Disconnect.req
CACK	Cack PDU
CNACK	Cnack PDU
CON	Con PDU
DIS	Dis PDU

Input Event

REQ	Req PDU
RPL	Rpl PDU
INT	Int PDU
TRM	Trm PDU
RQS	Rqs PUD
RPS	Rps PDU
ABT	Abt PDU
STP	Stp PDU
PDUcnf	Internal Event Occurring When transmission of Req, Rsp, Int or Trm PDU is completed
SCOMreq	S_Command.req
SCOMrsp	S_Command.rsp
SABTreq	S_Abort.req

3.2.3.4. Action Descriptions

The action description for SCEP is given below.

Predicates of Connection Management

p1	Connection Establishment Response
p2	Connection Establishment Rejection
p3	Not acceptable CON PDU
p4	Not Acceptable CACK PDU

Output Event of Connection Management

SCONind	S_Connect.ind
SCONcnf	S_Connect.cnf
SDISind	S_Disconnect.ind
SCOMind	S_Command.ind
CON	Con PDU
CACK	Cack PDU
CNACK	Cnack PDU
DIS	Dis PDU

Predicate of Command Execution

P1	Sum of SCEP Header, Command Header and SDU Exceeds Maximum Receivable PDU Size of Receiving Side
P2	Length of Remaining PDU Exceeds Maximum Receivable Size of Receiving Side
P3	Sum of SCEP Header, Command Header and SDU Exceeds Maximum Receivable PDU Size of Transmission Side
P4	Length of Remaining PDU Exceeds Maximum Receivable Size of Transmission Side
P5	Ack
P6	Nack

Output Event

REQ	Req PDU
INT	Int PDU
TRM	Trm PDU
RQS	Rqs PDU
ABT	Abt PDU
STP	Stp PDU
DIS	Dis PDU
PDUcnf	Internal Event Occurring When transmission of Req, Rsp, Int or Trm PDU is completed
SCOMcnf	S_Command.cnf
SCOMind	S_Command.ind
SDISind	S_Disconnect.ind
SIDind	S_CommandID.ind
SABTind	S_Abort.ind

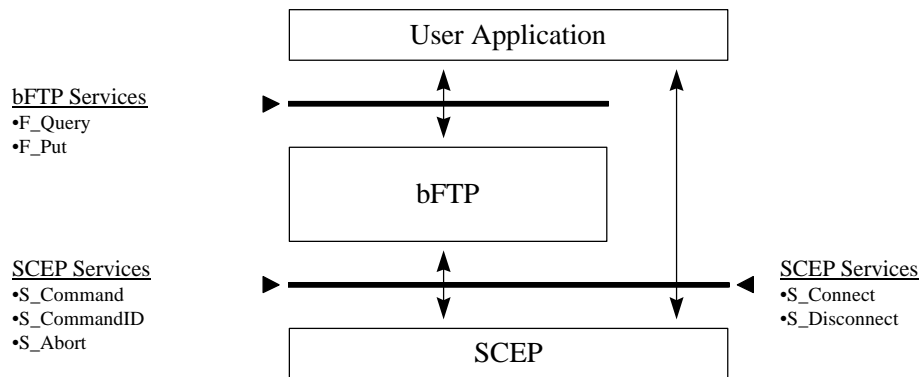
3.3. bFTP (binary File Transfer Protocol)

3.3.1. Service Definition

bFTP is intended to define a protocol that can be used to transfer files from peer to peer. This document describes the Query service and Put service.

3.3.1.1. Service Model

The figure below shows a model of how bFTP fits into a typical system, i.e., the bFTP reference model.



The elements for the bFTP reference model are described below.

bFTP Services bFTP Service primitives which are provided by bFTP.

bFTP protocol providing a file transfer and virtual file server mechanism.

SCEP Services SCEP Service primitives which are provided by SCEP.

SCEP protocol providing a connection management, command management and segmentation & reassemble mechanism.

3.3.1.2. bFTP Service Primitives

3.3.1.2.1. Query Service

F_Query.req (Responder MachineID,
Requester MachineID,
Requester PID,
What)

```

F_Query.ind ( Responder MachineID,
              Requester MachineID,
              Requester PID,
              What )
F_Query.rsp ( AckOrNackFlag,
              Responder MachineID,
              Requester MachineID,
              Requester PID ,
              Result )
F_Query.cnf ( AckOrNackFlag,
              Responder MachineID,
              Requester MachineID,
              Requester PID ,
              Result )

```

Query service is used to obtain the processing ability of the application on the responder. This is confirmed service. F_Query service is provided by using S_Command services of SCEP. Responder MachineID, Requester MachineID, Requester PID and AckOrNackFlag respectively correspond to the parameters of S_Command services. Responder PID which is one of the parameter of S_Command services should be 8.

Requester PID

identifies the bFTP-user that has issued F_Query.req.

What

indicates what category of processing abilities of the Responder. The value of 'What' should be one of the following:

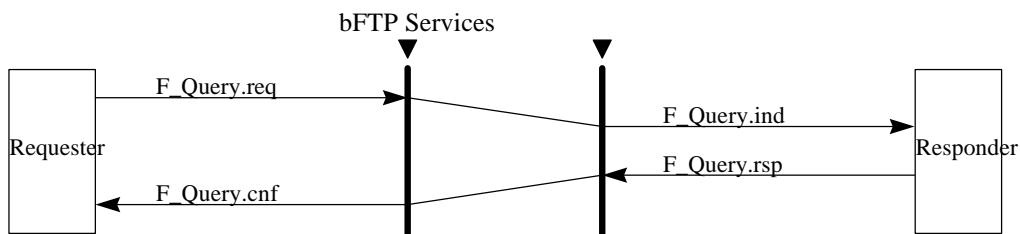
- RIMG** to inquire information of a still-image which can be processed by the Responder.
- RINF** to inquire a status of the Responder.
- RCMD** to inquire commands which can be executed by the Responder.

AckOrNackFlag

indicates the status whether the command execution is success or failure. The value of the flag is Ack or Nack respectively.

Result

indicates the results of the command execution. If AckOrNackFlag = Ack, it indicates the processing ability of the responder which is specified at 'What'. If AckOrNackFlag = Nack, it indicates an error code.



3.3.1.2.2. Put services

F_Put.req (Responder MachineID,
Requester MachineID,
Requester PID,
FileName,
UserFileName,
Time, FileHeader,
Thumbnail,
File)

F_Put.ind (Responder MachineID,
Requester MachineID,
Requester PID,
FileName,
UserFileName,
Time,
FileHeader,
Thumbnail,
File)

F_Put.rsp (AckOrNackFlag,
Responder MachineID,
Requester MachineID,
Requester PID ,
Result)

F_Put.cnf (AckOrNackFlag,
Responder MachineID,
Requester MachineID,
Requester PID ,
Result)

Put service is used for sending a named file to the Responder. The Put services are a confirmed-service. F_Put service is provided by using the S_Command services of SCEP. Responder MachineID, Requester MachineID, Requester PID and AckOrNackFlag respectively correspond to the parameters of S_Command services. Responder PID which is one of the parameter of S_Command services should be 8.

Requester PID

indicates ID number to identify the bFTP-user which has issued F_Query.req.

FileName

indicates the name of the file. The file name must be a character string of ASCII 8.3 format. The maximum length is 31 bytes.

UserFileName

indicates the long file name of the file. The file name must be a character string in the format of SJIS(Shifted-JIS Code), ASCII or ISO8859-1.

Time

indicates the time at which the file is created or modified. This must be a character string expressed in the 'YYYYMMDDHHMMSS' (year, month, day, hour, minute, second) format.

FileHeader

indicates the information of File or Thumbnail. This is not used in this document.

Thumbnail

indicates the scale-down image when the contents of the file is a still-image. This is not used in this document.

File

indicates a file itself to be transferred.

AckOrNackFlag

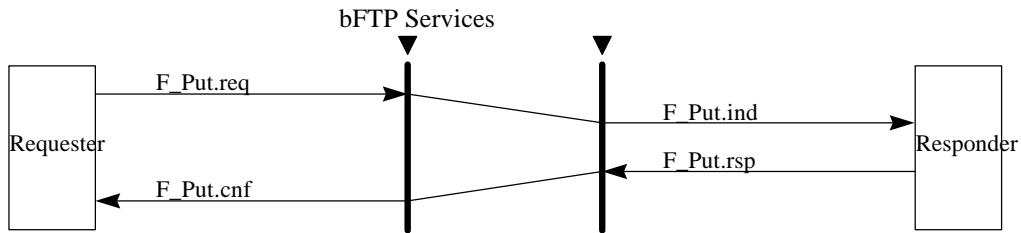
indicates in what state the command execution is finished.

Ack indicates that the command execution is successfully completed.

Nack indicates that the command execution is abnormally terminated.

Result

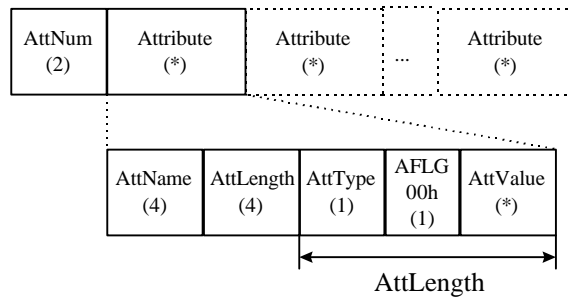
indicates the results of the command execution. If AckOrNackFlag = Ack, it indicates the name of File given when File is stored by the Responder. The file name must be a character string of ASCII 8.3 format. The maximum length is 31 bytes. If AckOrNackFlag = Nack, it indicates an error code.



3.3.2. bFTP Protocol Data Units

3.3.2.1. Attribute Structure

bFTP protocol data units are carried as the user data of SCEP PDU. The bFTP protocol data unit has a structure as shown below.



(n): n bytes, *: variable length, []: optional

The bFTP protocol data unit is composed of AttNum field and some Attribute fields. AttNum specifies the number of Attribute fields which are included the PDU. An Attribute field includes AttName field, AttLength field, AttType field, AFLG field and AttValue field.

- AttNum** The number of Attribute fields.
- AttName** Attribute field Name
- AttLength** The Length of Attribute

AttType	Attribute Type 00h Binary Type 01h Character Type, SJIS, ISO8859-1 or ASCII 06h Time Type, a character string of `YYYYMMDDHHMMSS` format Others reserved
AFLG	Supplementary Data 00h No Supplementary Data Others reserved
AttValue	Attribute Real Data

3.3.2.2. Attributes

AttName field represents characteristics of each Attribute field. AttName should be one of the following:

- File Name

AttName	“FIL0”
AttType	Character Type
AttValue	File name. It must be written in a character string of ASCII 8.3 format.

- User File Name

AttName	“LFL0”
AttType	Character Type
AttValue	The long file name of the file. It must be written in a character string of SJIS or ISO8859-1.

- Time

AttName	“TIM0”
AttType	Time Type
AttValue	The time at which the file is created or modified. It must be written in a character string of `YYYYMMDDHHMMSS` format.

- File Header

AttName	“TYP0”
AttType	Binary Type
AttValue	The information of the File or Thumbnail.

- Thumbnail

AttName	“TMB0”
AttType	Binary Type
AttValue	The scale-down image.

- Body

AttName	“BDY0”
AttType	Binary Type or Character Type
AttValue	

- Command
 - AttName “CMD0”
 - AttType Binary Type
 - AttValue Command Name

- Category to be Queried
 - AttName “WHT0”
 - AttType Character Type
 - AttValue Category Name

- Error
 - AttName “ERR0”
 - AttType Binary Type
 - AttValue ERRCODE

- Result
 - AttName “RPL0”
 - AttType Character Type
 - AttValue Stored File Name

3.3.2.3. File Information

The file information is expressed as the list of Attributes where attribute data and contents of File are joined together. The Attributes must appear in the order as shown in the below table, but Attributes with '(optional)' can be omitted.

AttName	Attlength	AttType	AFLG	AttValue	
“LFL0” (4)	(4)	01h (1)	00h (1)	(*)	•UserFileName (optional)
“TIM0” (4)	(4)	06h (1)	00h (1)	(*)	•Time (optional)
“TYP0” (4)	(4)	00h (1)	00h (1)	(*)	•FileHeader (optional)
“TMB0” (4)	(4)	00h (1)	00h (1)	(*)	•Thumbnail (optional)
“BDY0” (4)	(4)	00h (1)	00h (1)	(*)	•Body (mandatory)

3.3.2.4. Query_Req PDU

Query_Req PDU is configured by setting the parameter What of F_Query.req primitive to AttValue of AttName=WHT0”. Query_Req PDU is UserData of S_Command.req and S_Command.ind.

AttNum 0002h (2)					
AttName "CMD0" (4)	Attlength 00000006h (4)	AttType 00h (1)	AFLG 00h (1)	AttValue 00010040h (4)	
AttName "WHT0" (4)	Attlength 00000006h (4)	AttType 01h (1)	AFLG 00h (1)	AttValue (4)	

•The number of Attributes (mandatory)

•Command (mandatory)

•Category to be Queried (mandatory)

AttValue of AttName="WHT0"

indicates what category of the processing ability of the responder is desired to query about. 'Attribute' should be one of the following.

"RIMG"

is used when it is desired to know the data on a still image which can be processed by the responder.

"RINF"

is used when it is desired to know the data on the responder system which is related to the responder.

"RCMD"

is used when it is desired to know the data on the commands which can be processed by the responder.

3.3.2.5. Query_Rpl PDU

When the command execution is successfully completed, i.e., at the time of the parameter AckOrNackFlag of F_Query.rps primitive = Ack, Query_Rpl PDU is configured by setting the parameter Result of F_Query.rsp primitive to AttValue of AttName="BDY0". Query_Rpl PDU is UserData of S_Command.rsp and S_Command.cnf.

AttNum 0001h (2)					
AttName "BDY0" (4)	Attlength (4)	AttType 00h (1)	AFLG 00h (1)	AttValue (*)	

•Query Result Data (mandatory)

AttValue

indicates the processing ability of the responder which is specified at Query_Req. This should follow the below-listed BNF representation.

Representation of Symbols

- <> variable
- [] option
- * repeatable zero or more times
- + repeatable one or more times
- := definition

(...) explanation

At the time of WHT0 = "RIMG"

AttValue = <rimg>

<rimg> := [[<tag-org-size><original-size>]
[<tag-acc-size><num-acc-size>{<accept-size>}+]
[<tag-org-samp><original-sampling>]
[<tag-acc-samp><num-acc-samp>{<accept-sampling>}+]
[<tag-acc-filesize><accept-filesize>]{<tag-option><option>}*

<tag-pix-aspect> := 00h

<pix-aspect> := units in width (1byte) x units in height (1byte)

indicates the aspect ratio of pixels. The default value of 0101h stands for the pixels at the ratio of 1 : 1. With FFFFh, the aspect ratio is ignored.

<tag-org-size> := 01h

<original-size> := <QVGA> | <VGA> | <SVGA> | <XGA> | <SXGA> | <FREE>

indicates the original lattice size of the Responder.

If the Requester can transmit an image suitable for this field, the image can be transmitted without conversion.

If <accept-size> field is present, there is no default. If not, the default is <VGA>.

<tag-acc-size> := 02h

<num-acc-size> := (the number of <accept-size>, hexadecimal, 1byte)

<accept-size> := <QVGA> | <VGA> | <SVGA> | <XGA> | <SXGA> | <FREE>

The transmission side has to send an image of the size of the lattice included in this field. The default is <VGA>.

<tag-org-samp> := 03h

<original-sampling> := <compressed-420>

indicates the sampling method of the Responder. If the Responder can transmit an image suitable for this field, the image can be transmitted without conversion.

If <accept-sampling> field is present, there is no default.

If not, the default is <compressed-420>.

<tag-acc-samp> := 04h

<num-acc-samp> := (the number of <accept-sampling>, hexadecimal, 1byte)

<accept-sampling> := <compressed-420>

The transmission side has to send the image of sampling included in the <accept-sampling> field. The default is <compressed-420>.

<tag-acc-filesize> := 05h

<accept-filesize> := (the maximum receivable size, the bytes number divided by 256 is entered, 4bytes)

The transmission side must send the image of a size equal to or smaller than the size included in the <accept-filesize> field.

The receiving size has to assure receiving of images of this size.

The default is 00000200h(128Kbytes). FFFFFFFFh indicates that any size of image can be received.

<tag-option> := FEh | FFh

It is possible to include vendor-unique information.

Definition of Lattice Size

<QVGA> := (320 x 240)
 <VGA> := (640 x 480)
 <SVGA> := (800 x 600)
 <XGA> := (1024 x 768)
 <SXGA> := (1280 x 960)
 <FREE> := (m x n)

Size is expressed in hexagonal notation by allocating 2 bytes to each of Width and Height.

Width (2bytes)	Height (2bytes)
----------------	-----------------

(FFFFh,FFFFh) denotes an arbitrary size, which is used for the appliance that does not need to restrict the size, such as printers or PCs.

Definition of Sampling

<compressed-420> := C420h

At the time of WHT0 = "RINF"

AttValue = <rinf>

<rinf> := [<tag-mem><memory>][<tag-batt><battery>]

<tag-mem> := 10h

<memory> := (the amount of memory available for receiving. FFFFh indicates that the remaining memory is large enough. The values other than FFFFh serve as just a rough intimation. Unit is Kbytes. 2bytes)

At default, it is assumed that enough memory is available.

<tag-batt> := 11h

<battery> := (the remaining amount of battery. FFFFh indicates that the remaining battery is high enough. 0000h indicates low battery. Otherwise, it serves as just a rough indication. Unit is minute. 2bytes)

At default, it is assumed, but not assured, that the remaining amount is enough.

At the time of WHT0 = "RCMD"

AttValue = <rcmd>

<rcmd> := [<tag-opt-func><num-opt-func>{<opt-func>}+]

<tag-opt-func> := 20h

<num-opt-func> := (the number of <opt-func>, hexadecimal, 2byte)

<opt-func> := <func-multi-command>

<func-multi-command> := 0001h

The responder can execute a PUT command more than two times while the connection of the SCEP layer is established.

If this parameter is not present in the <opt-func> field, only one command can be executed.

Default Rule

1. If the reply of Query command is abnormal, (including the case where the Responder has not dealt with Query command), it is assumed that the Requester can send a picture with the aspect ratio = 1 : 1, <VGA> and <compressed-420>.
2. If the reply of Query command is normal, the transmission side searches the optimum transmittable form by sequentially reading tags. If there is no transmittable form, it is assumed that the transmission side can send <VGA> and <compressed-420>.

The default rule of each field is noted in the description of each field.

When the command execution is abnormally terminated, that is, at the time of the parameter AckOrNackFlag of F_Query.rps primitive = Nack, Query_Rpl PDU is configured by setting the parameter Result of F_Query.rsp primitive to Attvalue of AttName="ERR0".

AttNum 0001h (2)						•The number of Attributes (mandatory)
AttName "ERR0" (4)	Attlength 00000004h (4)	AttType 00h (1)	AFLG 00h (1)	AttValue ERRCODE (2)		•Result (mandatory) •AttValue is ERRCODE

3.3.2.6. Put_Req PDU

Put_Req PDU is configured by setting each parameter of F_Put.req primitive to AttValue :

FileName : "FIL0"
 Time : "TIM0"
 FileHeader : "TYP0"
 Thumbnail : "TMB0"
 File : "BDY0".

Put_Req PDU is UserData of S_Command.req and S_Command.ind.

AttNum (2)						•The Number of Attributes
AttName "CMD0" (4)	Attlength 00000006h (4)	AttType 00h (1)	AFLG 00h (1)	AttValue 00000000h (4)		•Command (mandatory)
AttName "FIL0 " (4)	Attlength (4)	AttType 01h (1)	AFLG 00h (1)	AttValue (4)		•File Name (mandatory)
File Information						•File Information (mandatory)

3.3.2.7. Put_Rpl PDU

When the command execution is normally completed, if the parameter AckOrNackFlag is `Ack`, Put_Rpl PDU is configured by setting the parameter `Result` of F_Put.rsp to AttValue of "BDY0". Put_Rpl PDU is UserData of S_Command.rsp and S_Command.cnf.

AttNum (2)						•The number of Attributes (mandatory)
AttName "RPL0" (4)	Attlength (4)	AttType 01h (1)	AFLG 00h (1)	AttValue (*)		•Result (mandatory) AttValue is a stored File Name

When the command execution is abnormally terminated, if the parameter AckOrNackFlag of F_Put.rps primitive is `Nack`, Put_Rpl PDU is configured by setting the parameter `Result` of F_Put.rsp to AttValue of "ERR0".

The scheme of PDU is similar to that of Query_Rpl PDU.

3.3.2.8. ERRORCODE

ERRCODE should employ some of the following.

- 0001h: Illegal data received
- 0002h: Unsupported PID received
- 0010h: Illegal attribute received
- 0011h: Unsupported command received
- 0020h: File system is full
- 0021h: No corresponding file or directory
- 0030h: Low Battery error
- 0031h: Abort execution of a command
- 0000h: Undefined error
- Others: reserved

3.3.3. State definition and transitions

This section contains a state transition table based on the bFTP service primitives described above. Descriptions of the states, events and actions are included.

3.3.3.1. State Transition Table

The state transition table for bFTP is given below. The first state is *NOEXIST*. When the action is not described, the input event is ignored and the state dose not transit.

State Transition Table of bFTP

	<i>NOEXIST</i>	<i>WRSP</i>	<i>EXEC</i>
F_Query.req	Query_Req <i>WRSP</i>		
F_Query.rsp			Query_Rpl <i>NOEXIST</i>
Query_Req	F_Query.ind <i>EXEC</i>		
Query_Rpl		F_Query.cnf <i>NOEXIST</i>	
F_Put.req	Put_Req <i>WRSP</i>		
F_Put.rsp			Put_Rpl <i>NOEXIST</i>
Put_Req	F_Put.ind <i>EXEC</i>		
Put_Rpl		F_Put.cnf <i>NOEXIST</i>	

3.3.3.2. State Definitions

The state definition for bFTP is given below.

States of bFTP

<i>NOEXIST</i>	Not Exist: there is no commad
<i>WRSP</i>	Waiting for Reply
<i>EXEC</i>	Under Command Execution

3.3.3.3. Event Descriptions

The event description for bFTP is given below.

Input Events

F_Query.req	Request from the upper layer to send Query command
F_Query.rsp	Response from the upper layer to send the result of Query command execution
Query_Req	Request from peer to send the Query command
Query_Rpl	Response from peer to send the result of Query command execution
F_Put.req	Request from the upper layer to send the Put command
F_Put.rsp	Response from the upper layer to send the result of Put command execution
Put_Req	Request from peer to send the Put command
Put_Rpl	Response from peer to send the result of Put command execution

3.3.3.4. Action Description

The action description for bFTP is given below.

Output Events

F_Query.ind	Query indication to upper layer
F_Query.cnf	Confirmation of query command execution from peer
Query_Req	Send the Query command to peer
Query_Rpl	Query_Rpl PDU Send the result of command execution to peer
F_Put.ind	Put indicate to upper layer
F_Put.cnf	Confirmation of Put command execution from peer
Put_Req	Send the Put command to peer
Put_Rpl	Put_Rpl PDU Send the result of command execution to peer

3.4. IrCOMM and IrLMP IAS Objects

This section describes the recommendation to use IrCOMM which is one of the reliable data stream.

The IAS is a database of infrared services, a sort of yellow pages listing what a device can provide. An IAS Object consists of a classname and one or more attributes that serve to advertise a service or group of related services on a device.

LsapSel (Link Service Access Point Selector) is the unique “address” or id of their service within the context of one device, and is needed to connect to that service.

LsapSel attribute of IrDA:IrCOMM IAS entry should be IrDA:TinyTP:LsapSel for the cooed service types (3-Wire or 9-Wire).

InstanceName is used to help distinguish among otherwise idntical IAS objects. Use of this attribute is recommended at this document which is IrTran-P. This document recommends to set “IrTran-P” at IrDA:IrLMP:InstanceName.

3.4.1. Recommendation of IrCOMM Operation

- 1) For category of IrCOMM connection, 9W or 3W of IrCOMM is available. 3W-RAW of IrCOMM is unavailable for connections because it does not use TinyTP.
- 2) For handling of control signal packets in 9W and 3W, control packets for baud rate setting, signal line control and the like are exchanged in addition to data. However, SCEP ignores these control data.
- 3) Flow control should be performed by using the credit of TinyTP.
- 4) When a packet of SCEP is larger than a packet of IrLAP, segmentation and reassembling of a packet is performed between SCEP and IrCOMM.

4. Appendix Uni Picture Format -

Notice

1. The specifications of this format are subject to change.
2. Version number

The number of the version of this format is indicated in the following form.

Version A. BC

- A : Number will increase each time the specification is updated.
BC : Numbers will increase by one each time a difficulty is cleared up or an application rule is updated.

4.1. Introduction

4.1.1. Scope and Format Abbreviations

This format is applied to still image data in “IrTran-P”.
The format name is “Uni Picture Format”, and the abbreviation will be “UPF”.

4.1.2. Terminology

The following terms are used throughout this section

4:2:0 Image component factor and Pixel sampling (see section 4.2.1.1.4)

APEX Data recording unit for camera setting information (see Appendix A)

4.1.3. Bit and Byte Ordering

This section regards frames as collections of bytes (octets) with each byte being composed of 8 bits numbered 0-7. Bit 0 is always the least significant bit (LSB) and bit 7 is always the most significant bit (MSB). Bytes are represented throughout this section in the following forms.

Diagrammatic - a byte is represented by a bit number. In some cases bit fields have special meaning and are indicated for clarity. The most significant bit is the bit on the left and the least significant bit is the bit on the right. An example is given below.

[MSB] bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 [LSB]

Hexadecimal - a byte is represented with two hex digits with the least significant nibble on the right, the most significant nibble on the left, and both digits suffixed by ‘h’. An example is the value 5 which is written as 05h.

Two bytes form - two bytes represented with four hex digits with the least significant nibble on the right, the most significant nibble on the left, and every digits suffixed by 'h'. An example is the value $256 \times 3 = 768$ which is written as 0300h.

4.1.4. References

- [1] ISO/IEC 10918-1 : “Digital Compression and Coding of Continuous-Tone Still Images”, Part 1 : Requirements and Guidelines, 1993
- [2] IEEE EUI-64,”<http://standards.ieee.org/db/oui/tutorials/EUI64.html>

4.2. Specifications

The method of data representation and file structure is defined in this section. Whether the specification is mandatory, optional or recommended is also specified.

4.2.1. Signal format

The signal format is specified as follows.

4.2.1.1. Video signal format

The video signal format and compression method is specified as follows.

4.2.1.1.1. Pixel aspect

The pixel aspect ratio of an image is 1: 1.
The aspect ratio is the ratio of width to height in an image.

Mandatory

4.2.1.1.2. Size of index image

The size of index image in this format is specified as below.

Horizontal x vertical	Name	Aspect ratio
80 x 60	INDEX	4 : 3

4.2.1.1.3. Size of image in Query

The size of image in Query is specified as below.

Horizontal x vertical	Name	Aspect ratio
320 x 240	QVGA	4 : 3
640 x 480	VGA	4 : 3
800 x 600	SVGA	4 : 3
1024 x 768	XGA	4 : 3
1280 x 960	SXGA	4 : 3
FREE	FREE	FREE

4.2.1.1.4. Image component factor and Pixel sampling

Image components are Y, Cb, Cr of one luminance and two color-difference signals.

Mandatory
Mandatory

Monochrome image is included in the above.
(See Section 4. 2. 1. 1. 8. 1.)

The pixel sampling ratio is 4 : 2 : 0 .

Mandatory
Mandatory

The sampling points of pixel is shown below.

The line is scanned from left to right and from top to bottom.
4 : 2 : 0 sampling points

Mandatory

Y	Y	Y	Y	Y	Y	
C		C		C		
Y	Y	Y	Y	Y	Y	
C		C		C		
Y	Y	Y	Y	Y	Y	
C		C		C		
Y	Y	Y	Y	Y	Y	

C : Cb / Cr

Fig. 4.2.1.1.4

4.2.1.1.5. Gamma and color management

Gamma and colors are managed to make color representation possible on the below supposed monitor.

Mandatory

Characteristics of the supposed monitor

- 1) Gamma is defined using the following reverse characteristics :

$$\begin{aligned} V &= 1.099L^{0.45} - 0.099 & 1 \geq L \geq 0.018 \\ V &= 4.500L & 0.018 > L \geq 0 \end{aligned}$$

L : input V : output of gamma compensation

- 2) Primary chromaticities

$$\begin{array}{lll} \text{red} & x = 0.640 & y = 0.330 \\ \text{green} & x = 0.300 & y = 0.600 \\ \text{blue} & x = 0.150 & y = 0.060 \end{array}$$

x and y are the CIE chromaticity coordinates.

- 3) Chromaticities of reference white

$$\text{D65} \quad x=0.3127 \quad y=0.3290$$

x and y are the CIE chromaticity coordinates.

- 4) Coefficients of color conversion

$$\begin{pmatrix} E'_R \\ E'_G \\ E'_B \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.701 & -0.587 & -0.114 \\ -0.299 & -0.587 & 0.886 \end{pmatrix}^{-1} \begin{pmatrix} E'_Y \\ E'_R - E'_Y \\ E'_B - E'_Y \end{pmatrix}$$

E'_R , E'_G and E'_B are gamma-compensated signals of R,G,B.

Regarding E'_Y , $E'_R - E'_Y$ and $E'_B - E'_Y$ (see Section 4.2.1.1.7.)

4.2.1.1.6. Number of bits of the image data

Y, Cb and Cr of the image data are 8 bits.

Mandatory

4.2.1.1.7. Image level

- 1) Y signal

$$Y = 219 (E'_Y) + 0$$

Mandatory

2) Color difference signal

$$Cr = 224 \{0.713 (E'_R - E'_Y)\} + 128$$
$$Cb = 224 \{0.564 (E'_B - E'_Y)\} + 128$$

Mandatory

Mandatory

that is

$$Cr = 160(E'_R - E'_Y) + 128$$
$$Cb = 126(E'_B - E'_Y) + 128$$

E'_Y, E'_R, E'_B are gamma-compensated signals of Y, R, B.

4.2.1.1.8. Image coding method

Image compression is subject to the JPEG baseline.
(ISO/IEC 10918-1)

Mandatory

The index image is subject to the same method.

Mandatory

4.2.1.1.8.1. Restriction factor of JPEG

- 1) Block-interleave only
- 2) With a monochrome image, Cb and Cr are compressed as 128.
- 3) Huffman table is fixed to JPEG recommended table.

Mandatory

Mandatory

Mandatory

4.2.1.1.8.2. Definition of MCU

The block of MCU (Minimum Coded Unit) is defined as below.

Mandatory

4 : 2 : 0

MCU0 : Y₀₀, Y₀₁, Y₁₀, Y₁₁. Cb₀₀. Cr₀₀
 MCU1 : Y₀₂, Y₀₃, Y₁₂, Y₁₃. Cb₀₁. Cr₀₁

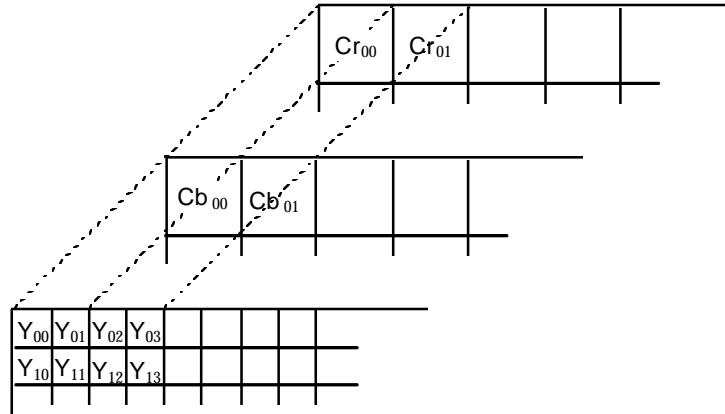


Fig. 4. 2. 1. 1. 8. 2

4.2.2. File Specifications

A file is specified as below.

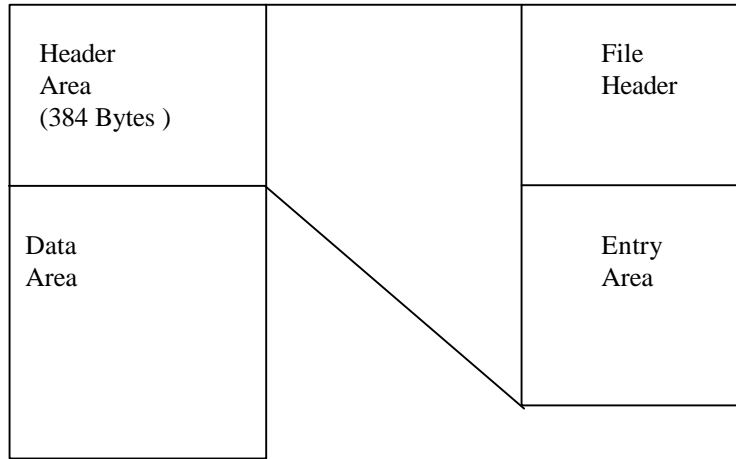
A file name extension must be "UPF".

Mandatory

4.2.2.1. File structure

A file consists of Header Area and Data Area.
 The Data Area consists of single or plural data items.
 The start address of data is defined in Header.
 Data area has to start from an even-number address divisible by 4.
 Data more than 2 bytes is located in most significant byte first.
 The character string is terminated by null (00h).
 The data used in the Reserved area is 00h when Byte and 0 when Bit.

The basic structure of a file is shown as below.
 A file has Header Area and Data Area.
 The Header Area size is fixed to 384 Bytes.



4.2.2.2. Header organization

Header is composed of File Header and Entry Area.

	Size
File Header	240 Bytes
Entry Area	144 Bytes

4.2.2.2.1. File Header definition

Field Name	Size
General declaration	8 Bytes
File declaration	8
File ID	2
File Version	2
Making date	8
Editing date	8
Maker code, Model code	4
Edit Maker code, Model code	4
0-Reset reserve	16
The number of Data entries	1
Total number of tables	1
Reserve1	1
Character set code	1
Title	128
Reserve2	48

Field definitions

General declaration	“SSS V100” in ASCII (Between SSS and V100 is one Space code) (see section 4. 2. 3. 2. 1)
File declaration	“UPF V100” in ASCII (Between UPF and V100 is one Space code) (see section 4. 2. 3. 2. 1)
File ID	ID of UPF File : 0x0100
File Version	File Version : 0x0100 (see section 4. 2. 3. 2. 2)
Making date	Date of making this file (see section 4. 2. 2. 2. 1. 1)
Editing date	Date of editing this file (see section 4. 2. 2. 2. 1. 1)
Maker code, Model code	Code of maker who record this file Fill with FFFFh : Not defined (Reserved for Maker code, Model code)
Edit Maker code, Model code	Code of maker who edit this file Fill with FFFFh : Not defined (Reserved for Maker code, Model code)
0-Reset reserve	All bytes must be revised to 00h in each modifying In initial all bytes are 00h
Numbers of Data entry	Total numbers of entries Must be 2,3, or 4
Total number of tables	Total numbers of tables
Reserve1	Reserved : 00h

Character set code	Character set code of Title 00h : ASCII 01h : ISO-8859-1 02h : Shifted JIS FFh : No existence of Title string Other : Reserved
Title	String of Title Rest parts are 00h (NULL) String must be terminated by 00h(NULL)
Reserve2	Reserved All bytes are 00h

4.2.2.2.1.1. Date definition

A date is defined as below.

Field Name	Size	Definition
Difference in time	1 Byte	Difference in GMT time is expressed by complement Unit is 15 minute, from -12H to 12H Not defined : 80h
year	2	Christian Era. By binary Not defined : FFFFh
month	1	month by binary Not defined : FFh
day	1	day by binary Not defined : FFh
hour	1	hour by binary Not defined : FFh
minute	1	minute by binary Not defined : FFh
second	1	second by binary Not defined : FFh

for example + 9h,1997year,6month,26day,20hour,20minute,30second
: 24h 07CDh 06h 1Ah 14h 14h 1Eh (This example for Tokyo area)
for example (difference in time part) - 15minute : FFh - 1hour : FCh

4.2.2.2.2. Entry Area structure

Entry Area has 4 entries.

	Size
Entry 1	36 Bytes
Entry 2	36
Entry 3	36
Entry 4	36

Entry order must be same as data (existing in Data Area) order.

Each entry has 5 fields.

Field Name	Size
Start address	4 Bytes
Data size	4
Data type ID	1
Reserve	1
Information data	26

Field general definition

Start address	Start address of data Address is started from next address of Header end (address 0 means 384 Bytes from file top) No existence of data : FFFF FFFFh
Data size	Size of data 0000 0000h : no existence of data
Data type ID	ID for data type 00h : no existence of data 10h : image 11h : index image (thumb nail image) others : reserved
Reserve Information data	00h Information of data defined in each data type

4.2.2.2.3. Entry Area definition

Field Name	Size
Index image data start address	4 Bytes
Index image data size	4
Data type ID (index)	1
Reserve1	1
Index image information data	26
Image data Start address	4
Image data size	4
Data type ID (image)	1
Reserve2	1
Image Information data	26
Sub data 1 start address	4
Sub data 1 data size	4
Data type ID (Sub data 1)	1
Reserve3	1
Sub data 1 information data	26
Sub data 2 start address	4
Sub data 2 data size	4
Data type ID (Sub data 2)	1
Reserve4	1
Sub data 2 information data	26

Field definition

Index image data start address	Start address of index image FFFF FFFFh : no existence of index image
Index image data size	Size of index image data 0000 0000h : no existence of index image data
Data type ID (index)	11h (fixed) 00h : no existence of index image
Reserve1	00h
Index image information data	(See section 4.2.2.2.4.1)
Image data start address	Start address of image FFFF FFFFh : no existence of image
Image data size	Size of image data 0000 0000h : no existence of image data

4.2.2.2.4.2. Image information data

Field Name	Size
Image size (horizontal)	2 Bytes
Image size (vertical)	2
Image pixel configuration	1
Image rotation set ID	1
Wide ID	1
Compression ratio	1
White level information	1
Type of input device	1
Reserved1	3
Existence of dummy data	1
X-BEGIN in real data	2
Y-BEGIN in real data	2
X-SIZE in real data	2
Y-SIZE in real data	2
Non compression ID	1
Reserve2	3

Field definition

Image size	Size of image corresponds to number of pixels
Image pixel configuration	00h 4 : 2 : 0
	01h 4 : 2 : 0 (orthogonal*)

*" orthogonal " is when the head of Y and C correspond to the following.

Y / C	Y	Y / C
Y	Y	Y
Y / C	Y	Y / C
Y	Y	Y

Horizontal / vertical set of image b1, b0 : Information to rotate image counter-clockwise

0 0	: 0 degree
0 1	: 90 degree
1 0	: 180 degree
1 1	: 270 degree

b2 : Information to obtain mirror image (Right and left)

0	: None
1	: Reverse

Order of rotation and reversal are rotation as first, reversal as next.

Wide ID	00h : normal 01h : Cut off top and bottom of picture which corresponds to wide mode 10h : Wide indication of 16 : 9 in 4 : 3
Compression ratio	Compression ratio is expressed by number of bits in each pixel of picture. High position 4 bit : integer part Low position 4 bit : decimal part Not defined : FFh
White level information	219 or FFh : not defined
Type of input device	FFh Not defined First 4 bits 1h Television-related equipment Next 4 bits 0h : NTSC 1h : PAL 2h : SECAM 3h : HDTV First 4 bits 2h Camera Next 4 bits 0h : Original color filter 1h : Complementary color filter First 4 bits 3h Scanner Next 4 bits 0h : Print 1h : Negative film 2h : Positive film
Existence of dummy data	Existence / non-existence of dummy data 00h : non-existence 01h : existence
Position in real data	Position of real data is expressed by rectangle. X-BEGIN, Y-BEGIN Start position of horizontal, vertical real data (in pixel units) X-SIZE, Y-SIZE Size of real data Dummy data Content is not defined if dummy data non-existent.
Non compression ID	00h : JPEG others : reserved
Reserve2	00h

4.2.2.3. Data Area organization

A Data Area has plural data items.

Table data
Index image data
Image data
Sub data 1
Sub Data 2

4.2.2.3.1. Table area structure

The Table area is composed of several tables.
 The Table area is optional.
 The Table area must begin from Header end with no blank.
 Each table starts from an even-number address divisible by 4.
 The optional blank space between tables are allowed.
 The field data distribution in tables is shown in next section.
 The order of tables is free.
 Every Table has individual ID.

Table 1
Table 2
.....
Table N

4.2.2.3.2. Table structure

The table has 3 fields shown as below.

Field Name	Size
Table ID	1 Byte
Next table pointer	1
Table data	free (max.254)

Field definition

Table ID Type of Table

Next table pointer The Next table pointer is table size minus 2.
(In this case, table size includes the following blank area.)
The Next table pointer in the last table is also table size
minus 2.

The basic addressing of tables is shown as below.

Start address	Table data name	data
1	Table ID	
1 + 1	Next table pointer	m + n
1 + 2	Table data (m Bytes)	
1 + m + 2	Blank (n Bytes)	
1 + m + n + 2	Next Table ID	

4.2.2.3.3. Types of table

Table type	ID	See Section number
Comment table	12h	4. 2. 2. 3. 4. 1
Author information table	13h	4. 2. 2. 3. 4. 2
Camera information table	24h	4. 2. 2. 3. 4. 3
Transfer URL information table	80h	4. 2. 2. 3. 4. 4
Transfer TEL information table	81h	4. 2. 2. 3. 4. 5
Optional table	90h	4. 2. 2. 3. 4. 6

All IDs except the above are reserved.

4.2.2.3.4. Table definition

Tables are defined as follows.

4.2.2.3.4.1. Comment table

Field Name	Size
Table ID (12h)	1 Byte
Next table pointer	1
Character set code	1
Reserved	1
Comment	Free

Field definitions

Character set code 00h : ASCII
 01h : ISO-8859-1
 02h : Shifted JIS
 other : Reserved

Comment Comment is recorded.
 Maximum 252 bytes including last code of 00h

4.2.2.3.4.2. Author information table

Field Name	Size
Table ID (13h)	1 Byte
Next table pointer	1
Reserved	1
Character set code	1
Author information	32
Editor information	32
Reserved	4

Field definitions

Character set code 00h : ASCII
 01h : ISO-8859-1
 02h : Shifted JIS
 Other : Reserved

Author and editor information are optional. Last code is 00h.

4.2.2.3.4.3. Camera information table

Field Name	Size
Table ID (24h)	1 Byte
Next table pointer	1
Shutter speed	2
Aperture	2
Brightness	2
Exposure Bias	2
Max Aperture Ratio	2
Focal Length	2
Subject Distance	2
Metering Mode	1
Light Source	1
Flash	1
Reserved	1
Interval information	2
Reserved	2

Field definitions

Shutter speed APEX unit 1/100 unit 2's complement
 Aperture APEX unit 1/100 unit 2's complement
 Brightness APEX unit 1/100 unit 2's complement

Exposure Bias	APEX unit	1/100 unit	2's complement
Max. Aperture Ratio	APEX unit	1/100 unit	2's complement
	8000h = Unidentified in the above		
	APEX unit (see Appendix A)		
Focal Length	1/10 mm unit	FFFFh : Unidentified	
Subject Distance	1/10m unit	FFFEh : Infinite	
		FFFFh : Unidentified	
Metering Mode	00h : Average	01h : Center Weighted Average	
	02h : Spot	03h : MultiSpot	
	FFh : Unidentified		
Light Source	00h : Daylight	01h : Fluorescent light	
	02h : Tungsten Lamp		
	10h : Standard light source A		
	11h : Standard light source B		
	12h : Standard light source C		
	20h : D55	21h : D65	
	22h : D75		
	FFh : Not defined		
Flash	00h : No flash		
	01h : Flash		
	FFh : Not defined		
Interval information	Time of interval when continuous recording or recording at interval		
	bit 15 ~ bit 14	reserved	
	bit 13 ~ bit 12	0	0
		0	1
		1	0
		1	1
	bit 11 ~ bit 0	Data of interval	
	FFFFh : Not defined		

4.2.2.3.4.4. Transfer URL information table

Field Name	Size
Table ID (80h)	1 Byte
Next table pointer	1
Character set code	1
Reserved	1
Transfer URL information	Free

Field definitions

Character set code	00h : ASCII
	01h : ISO-8859-1
	02h : Shifted JIS
	Other : Reserved

URL Address for transfer is recorded as follows.

URL address for transfer <URL>url address information

Field Name	Size
Table ID (90h)	1 Byte
Next table pointer	1
Maker code	2
Model code	2
Maker code 2	3
Reserve	1
Optional data	max.246

Field definitions

Maker code	Fill with FFFFh : Not defined (Reserved for Maker code)
Model code	Fill with FFFFh : Not defined. (Reserved for Model code)
Maker code 2	Fill with EUI-64 company_id code
Reserve	00h
Optional data	Optional data less than 246 bytes

4.2.2.3.5. Main data structure

Plural data in Data Area are shown as below.

Tables
Blank
Index image data
Blank
Image data
Blank
Sub data 1
Blank
Sub data 2
Blank

Index image data and Image data are compressed by JPEG baseline.

Mandatory

4.2.3. Application rule

In order to ensure the system's interchangeability, the following rules are established.

4.2.3.1. Signal format matter

4.2.3.1.1. Image compression method

In Uni Picture Format , image data is compressed by JPEG baseline.

4.2.3.2. File management matter

4.2.3.2.1. Management of declaration

The declaration of file (see 4. 2. 2. 2. 1) is available for rejecting another file.

General declaration : “SSS“
File declaration : “UPF”

In each declaration, first 3 Bytes (see ahead) are available for this use.
Rest each 5 Bytes must be ignored.

4.2.3.2.2. Management of File version

The File version (see 4. 2. 2. 2. 1) is used for version up.

upper byte : shows integer part , if this byte is changed there is no interchangeability
from old version.

lower byte : shows decimal part

First 4 bits : inclement when feature is updated

Next 4 bits : inclement when minor change is done with keeping
interchangeability

for example Ver 1.12 : 01h 12h Ver 1.26 : 01h 26h

4.3. Additions

Appendix A : Data recording unit for camera setting information (APEX)

Uni Picture format uses a unit for camera setting information called “APEX” (Additive System of Photographic Exposure). APEX is a convenient unit for calculating the exposure value : Ev. The relationship between the conventional unit and the APEX unit is as follows.

$$\begin{aligned} \text{ApertureValue}(Av) &= 2\log_2(\text{FNumber}) \\ \text{ShutterSpeedValue}(Tv) &= -\log_2(\text{ExposureTime}) \\ \text{BrightnessValue}(Bv) &= \log_2(B/NK) \\ &\quad B:\text{cd/m}^2, N,K:\text{constant} \end{aligned}$$

The speed Value is as follows.

$$\text{SpeedValue}(Sv) = \log_2(\text{ASA}/3.125)$$

The exposure value is calculated as follows.

$$Ev = Av + Tv = Bv + Sv$$

Aperture Value (APEX)	F-Number
0	1
1	1.4
2	2
3	2.8
4	4
5	5.6
6	8
7	11
8	16
9	22
10	32

Brightness Value (APEX)	Foot lambert
-2	1/4
-1	1/2
0	1
1	2
2	4
3	8
4	15
5	30

Shutter Speed Value (APEX)	Exposure Time (second)
-5	30
-4	15
-3	8
-2	4
-1	2
0	1
1	1/2
2	1/4
3	1/8
4	1/15
5	1/30
6	1/60
7	1/125
8	1/250
9	1/500
10	1/1000
11	1/2000