TTC標準
Standard

JF-IR009．10

赤外線通信インタフェース
ポイントアンドシュートプロファイル
およびテスト仕様

Point and Shoot Profile with Test Specification

第 1.0 版

2002 年 11 月 28 日制定

社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

TTC Telecommunication Technology Committee

目　　次

＜参考＞

１．英文記述の適用レベル

適用レベル：Ｅ１

２．国際勧告等との関連

本標準は、赤外線通信標準化団体 IrDA(Infrared Data Association)において２０００年３月に採択された標準 Point and Shoot Profile Version1.1 および Point and Shoot Application Profile Test SpecificationVersion1.0 に基づいて定めたものである。

以下の２つのドキュメントから構成される。

・ Point and Shoot Profile Version 1.1 (March 20,2000)

・Point and Shoot Application Profile Test SpecificationVersion1.0(March 30,2000)

３．上記国際勧告等に対する追加項目等

３．１ オプション選択項目

なし

３．２ ナショナルマター項目

なし

３．３ その他

（１）先行している項目

なし

（２）追加した項目

なし

（３）削除した項目

なし

３．４ 上記国際勧告等に対する変更事項

なし

３．５ 参照した国際勧告との章立て構成の相違

ＴＴＣ標準	ＩｒＤＡ標準	備考
第１部 ポイントアンドシュートプロファイル	Point and Shoot Profile	
第２部 ポイントアンドシュート応用プロファイルテスト仕様	Point and Shoot Application Profile Test Specification	

４．改版の履歴

版数	制定日	改版内容
第１版	２００２年１１月２８日	制定

５．その他

(1)参照勧告、標準等

ＴＴＣ標準：

IrPHY　JF-IR001.10(E) 赤外線通信インタフェース 物理層プロトコル
(Serial Infrared (SIR) Physical Layer Link Specification)

IrLAP　JF-IR002.20(E) 赤外線通信インタフェース データリンク層プロトコル
(Serial Infrared Link Access Protocol)

IrLMP　JF-IR003.10(E) 赤外線通信インタフェース リンク層管理プロトコル
(Serial Infrared Link Management Protocol)

TinyTP　JF-IR004.10(E) 赤外線通信インタフェース 簡易トランスポートプロトコル
('Tiny TP': A Flow-Control Mechanism for use with IrLMP)

IrOBEX　JF-IR007.10(E) 赤外線通信インタフェース オブジェクト交換プロトコル
(Serial Infrared (SIR) Object Exchange Protocol)

ＩｒＤＡ標準:

IrDA Lite（Minimal IrDA Protocol Implementation）

IrMC　(Infrared Mobile Communications)

その他標準:

vCard (The Electronic Business Card Exchange Format)

vCalendar (The Electronic Calendaring and Scheduling Exchange Format)

RFC822 (Standard for the Format of the Arpa Internet Text Messages)

RFC2045 (Multipurpose Internet Mail Extension(MIME) Part One : Format of Internet Message Bodies）

第1部

ポイントアンドシュートプロファイル

Infrared Data Association

Point and Shoot Profile

Version 1.1

March 20, 2000

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 (925) 943-6546 or faxing requests to (925) 943-5600.

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 feature 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 an IrDA collective membership mark or feature trademark 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 DOCUMENT STATUS

Point and Shoot Working Group Convenor:
Bryce Jeppsen, Hewlett-Packard bryce_jeppsen@am.exch.hp.com

Document Editor:
Charles Knutson, Oregon State University knutson@cs.orst.edu

Contributors:
Yoshinobu Akimoto, Open Interface
Pete Bramhall, Hewlett-Packard
Eric Edwards, Sony
Melinda Grant, Hewlett-Packard
Gontaro Kitazumi, Okaya Systemware
Yusuke Kushida, NTT DoCoMo
Rob Lockhart, Motorola
Tim Looney, Kodak
Lars Novak, Ericsson
Hiroshi Ono, NEC
Gary Parham, Hewlett-Packard
Gavin Peacock, 3Com
James Scales, Nokia
Mike Sloane, Hewlett-Packard
Richard Stow, Philips
David Suvak, Extended Systems

History:
Version 1.0: First release, January 12, 2000
Version 1.1: Extensive changes to section 4.1.4 (Device Capabilities), plus other editorial changes.

2 INTRODUCTION

This document presents the IrDA Point and Shoot Application Profile. Section 3 describes the Point and Shoot Usage Model upon which the Application Profile is based. Section 4 presents the Point and Shoot Application Profile.

2.1 References

[IrLAP] "Serial Infrared Link Access Protocol, IrLAP, Version 1.1," Infrared Data Association
http://www.irda.org/standards/specifications.asp

[IrLMP] "Link Management Protocol, IrLMP, Version 1.1," Infrared Data Association
http://www.irda.org/standards/specifications.asp

[IrPHY] "Serial Infrared Physical Layer Link Specification, IrPHY, Version 1.3," Infrared Data Association
http://www.irda.org/standards/specifications.asp

[TTP] "Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1," Infrared Data Association
http://www.irda.org/standards/pubs/Tinytp11.PDF

[LITE] "Minimal IrDA Protocol Implementation, IrDA Lite, Version 1.0," Infrared Data Association
http://www.irda.org/standards/pubs/litever10.pdf

[IrOBEX] "IrDA Object Exchange Protocol, IrOBEX, Version 1.2," Infrared Data Association
http://www.irda.org/standards/pubs/IrOBEX12.pdf

[IrMC] "IrMC (Ir Mobile Communications) Specification, Version 1.1," February 1999, Infrared Data Association
http://www.irda.org/standards/specifications.asp

[VCARD] "vCard – The Electronic Business Card Exchange Format, Version 2.1," September 1996, The Internet Mail Consortium
http://www.imc.org/pdi/vcard-21.doc

[VCAL] "vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0," September 1996, The Internet Mail Consortium
http://www.imc.org/pdi/vcal-1.0.doc

[VNOTE] "IrMC (Ir Mobile Communications) Specification, Version 1.1," February 1999, Infrared Data Association
http://www.irda.org/standards/specifications.asp

[VMSG] "IrMC (Ir Mobile Communications) Specification, Version 1.1," February 1999, Infrared Data Association
http://www.irda.org/standards/specifications.asp

[TEXT] RFC 822: "Standard for the Format of the Arpa Internet Text Messages"
RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies"

[PCL3] Available through Hewlett-Packard: http://www.hp.com/go/solutions

[PCL5] Available through Hewlett-Packard: http://www.hp.com/go/solutions

[PS] Postscript Language Reference, Adobe Systems, Inc., Addison-Wesley, 1999.

[ESCP] EPSON ESC/P Reference Manual ESC/P-80 December 1999
http://www.ercipd.com/isv2/escp80.htm

[EXIF] "Digital Still Camera Image File Format Standard (Exif), Version 2.1," Japan Electronic Industry Development Association
http://www.jeida.or.jp/document/standard/index-e.html

[FILE] This data type refers to generic file types that can be stored by a generic file system.

[JETSEND] JetSend Protocol on IrDA Application Note, Version 1.1, November1999, Infrared Data

Association
http://www.irda.org/standards/pubs/IrJetSendAppNoteV1.1.pdf

3 POINT AND SHOOT (OBJECT PUSH) USAGE MODEL

3.1 Introduction

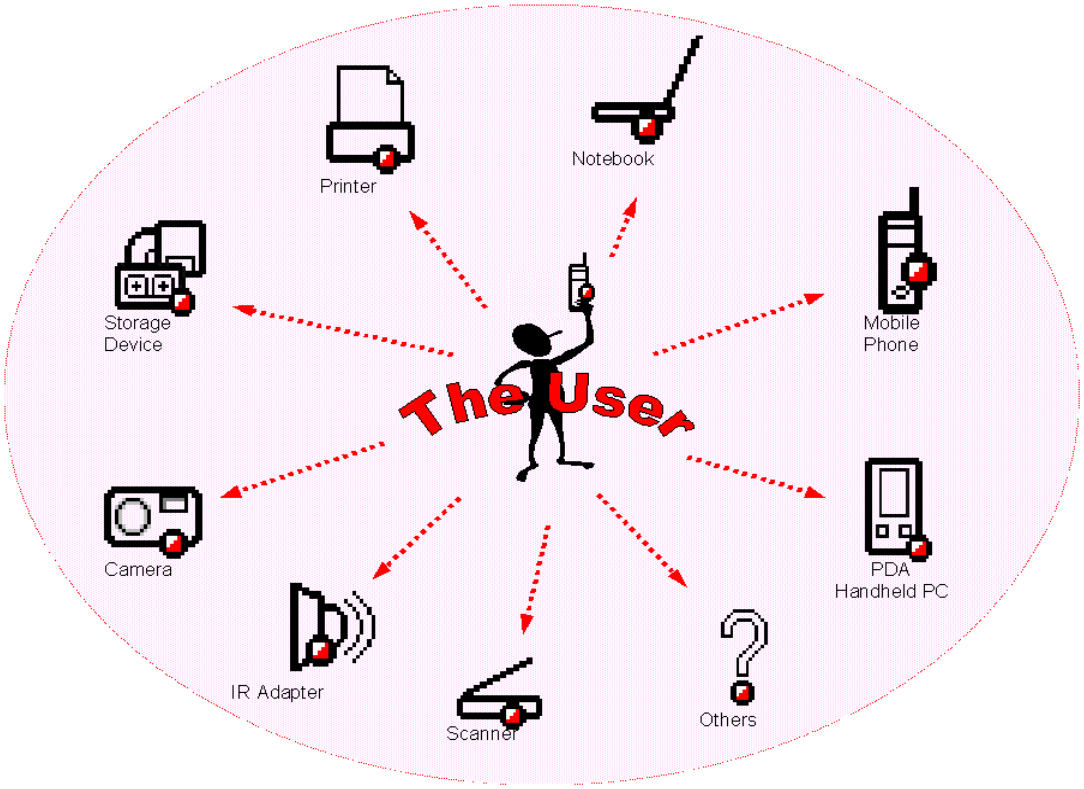
The IrDA Point and Shoot Usage Model is based on IrDA-Data that was initially defined and adopted in 1994. It is recommended for high speed, short range, line of sight, point-to-point wireless data transfer and is targeted at the IrDA 1.1 Fast Infrared Data (FIR) 4 Mbps components. The model also applies to IrDA 1.0 Serial Infrared Data (SIR), and will be the same model for the recently announced IrDA Very Fast Infrared (VFIR). This model is useful to over 50 million electronic devices including desktop, notebook and palm computers, printers, digital cameras, public phones/kiosks, cellular phones, pagers and other mobile devices.

3.2 Scope

The scope of the information presented in this section is based on the ability to exchange data between two IrDA-enabled devices. The focus of this Usage Model is on the user's experience. Many data exchange operations can be reduced to simple object push events, such as printing, faxing, business card exchange, image transfer, and file transfer. The Point and Shoot Usage Model is the universal way to move data objects between IrDA-enabled devices. The key to universal object exchange is support for standard object types (for example, vCard, JPEG (Exif), and text). Almost all IrDA devices will support this capability including PCs, printers, PDAs, cameras, phones, watches, pagers, storage devices, and kiosks.

3.3 User Scenario

Many user scenarios are covered by Point and Shoot object push. The picture below captures the power and simplicity of this Usage Model.



In the following scenarios, the user can:

- push a business card from his phone to another person's PDA or PC.
- pass the presentation stored on a PC to another PC.
- print his business card from his watch.
- fax a memo from his PDA.
- store images or sound files on a portable storage device.
- print or fax pictures directly from the storage device.
- dial a pay phone simply by pushing a business card from his PDA.
- send pictures over the Internet by pushing pictures from his camera to a kiosk or a cell phone.

The possibilities are endless and the user is control.

3.4 Interoperability

IrReady 2000 devices will have this Point and Shoot capability built-in. The use of standard object types will guarantee that objects are correctly understood on the other device. Devices will be able to alert the user when the other device will not understand an object being sent.

Below is a list of the different data types with examples of what the user may experience when pushing these objects from one device to another. This list is meant to be representative of the types of interoperable data exchange represented by the Point and Shoot Usage Model. This list is by no means exhaustive.

3.4.1 Types of Data Exchange:

- **Business Cards (vCard)**
  - Business Card Exchange
  - Phone List Exchange
  - Business Card Print

- **Appointments, To Do items, Alarms (vCalendar)**
  - Calendar item exchange
  - Calendar item print

- **Text Notes (vNote)**
  - Text item exchange
  - Text item print

- **Messages (vMessage)**
  - Email message exchange
  - Email message print

- **Digital Images**
  - Image exchange
  - Image print

- **Text Files**
  - Text file exchange
  - Text file print

- **Generic Files**
  - Exchange of files between file systems

3.5 Usability

Users will be able to transfer an object to another device by selecting the object and performing a simple operation (such as pressing a button). For example on a PC the user can send a file to another device by dragging the file and dropping it on an icon representing a remote device or an IR application. Another approach may be to select the object and perform a right mouse click operation that will bring up a menu. The user then selects the “send to IR” option and the object is sent. Sending your business card may be as simple as pushing a “send” button.

The short-range, narrow angle of IrDA-Data allows the user to aim, in a Point and Shoot style, at the intended recipient. Close proximity to the other device is natural in this type of data exchange situation, as is pointing one device at another. The limited range and angle of IrDA-Data allows others to simultaneously perform a similar activity nearby without interference. The short-range and narrow angle of IrDA-Data provides a simple form of security and a natural ease of use.

Other technologies with omni-directional capabilities are not as easy to use in this type of scenario. The user is not able to point at the intended recipient. Instead, the user must discover the other devices and

choose the appropriate recipient from a list. Close proximity to the intended recipient will usually not help, and choosing the proper device from a list may require special knowledge or additional information.

Point and shoot object exchange using IrDA-Data is the simplest way to transfer objects between two devices.

3.6 Configuration

By default, no configuration should be required for pushing or receiving objects. In some systems the user can select the location of the inbox and perhaps the behavior of prompts. But the device must be equipped to Point and Shoot “out of the box.” In situations where some configuration is required, it should require minimal effort by the user, and should quickly and easily render the device ready to perform an object push or receive where appropriate.

3.7 Reliability

Objects will be sent error free. Specific reliability standards are identified in the test specifications associated with the required enabling technology. Those details are addressed in the Point and Shoot Application Profile (Section 4).

3.8 Additional Information

As objects are received, they may be put into an appropriate data store on the device or delivered to an appropriate application. For example, on a PC received business cards could be placed directly into the user’s PIM. These features may require some configuration on the part of the user.

4 POINT AND SHOOT PROFILE

4.1 User Requirements

4.1.1 Scope

This Point and Shoot profile defines the minimum requirements for the protocols and procedures that shall be implemented in devices that support the Point and Shoot Usage Model. The most common devices implementing this Usage Model include PCs, notebooks, PDAs, mobile phones, printers and digital cameras.

4.1.2 User Scenarios

The basic scenario covered by this profile is an IrDA device pushing an object to another IrDA device (for example, a mobile phone pushing a business card to a PDA).

4.1.3 Data Object Types

It is necessary to define a set of standard object types for the Point and Shoot profile. The purpose of defining a standard set of data object types is to establish a baseline such that some level of interoperability can be reasonably achieved across a broad range of devices. This standard set of objects is NOT intended to span all of the possible data objects, nor is it intended to define a complete set of data objects to enable highly optimal application solutions. The intent is to define a minimal set of data objects which will establish a common denominator for specific classes of data types, so that interoperability between two devices will occur.

Having a standard set of data objects does not preclude a device manufacturer from supporting other data objects which provide improved value to the end customer, or provide a more optimal end solution. Additional data objects may be supported, but generic interoperability must still be guaranteed in these circumstances in order to comply with the Point and Shoot profile.

As an example, JPEG (Exif) is the default standardized data object for image data. This format is used to exchange image data on PCs, Digital Cameras, and other devices. Point and Shoot devices that need or require the support of sending or receiving image data should then support the processing of JPEG (Exif) files. It is entirely possible (and even probable) that future digital photography solutions will utilize new file formats for improved picture quality and/or efficiency. As these new file formats are developed, device manufacturers may include support of the new formats, but JPEG (Exif) support must still be the baseline requirement for interoperability.

The standardized object types are shown in the table below. This table includes both required and optional object data types. Section 4.1.4 clarifies the types that are required for specific device capabilities.

Data Type	Format	Examples
Business Card	vCard [VCARD]	Business card exchange Phone list exchange Business card print
Calendar items, Appointments, To do items, Alarms	vCalendar [VCAL]	Exchange of calendar items Calendar print
Text Notes	vNote [VNOTE]	Exchange text notes Text note print
Messages and Emails	vMessage [VMSG]	Email exchange Email print
Text files	ASCII using CRLF [TEXT]	Exchange a text document Print a text document
Formatted Document Files*	PCL3 [PCL3] PCL5 [PCL5] Postscript [PS] ESC/P-80 [ESCP]	Print ready files Print driver output
Images	JPEG (Exif) [EXIF]	Image exchange Image print
Files	Any [FILE]	Generic file exchange

*Note: These "Formatted Document Files" are never required to be supported by any device in order to comply with the Point and Shoot profile. However, they are the most common file types that printers are designed to handle. We include them here as a reference for devices that want to extend their functionality by supporting these types over IR.

4.1.4 Device Capabilities

The following sections discuss the specific capabilities and requirements of the various device classes. It is expected that additional categories will be added to this section over time.

It is assumed that, unless explicitly mentioned in the sections that follow, IrReady Point and Shoot devices are capable of both sending and receiving objects. This means that both push clients and push servers are supported on these devices. It also means that the devices send and/or received objects in the ways described.

Many devices are multi-functional, meaning that they have capabilities that span device categories. For a device to be compliant with the Point and Shoot Profile, it must comply with the capabilities described in all the relevant tables in this section. As an example, a device may function as both a PDA and a cell phone. In this case the device would need to comply with the capability requirements of both the PDA and cell phone tables from the following sections.

The following terms are defined for these sections.

Required	*Mandatory support*. Devices of this class must support this data type in the way described in order to be compliant with the Point and Shoot profile.
Optional	*Optional support*. The table suggests the way in which this must be done if supported, but devices of this class are not required to support this data type.
Conditional	*Conditional support*. If the device has an application that supports the particular data type, and has the capability to send and receive, it must send and/or receive this data type in the manner prescribed.
Not Applicable	*No support under most conditions*. The particular data type makes little sense for the device class within the context of this profile. It should be noted that while support of the data type for this device class is not defined by this profile, it is not prohibited.

The following discussion is intended to help clarify the motivation and usage of these terms.

Requiring data types for a particular device class has the effect of prescribing a particular feature set for all devices of that class that wish to be compliant with the Point and Shoot profile. While some of these required data types are a natural outgrowth of the typical usage of these devices, others are not. For example, while it is natural for a digital camera to support the Image data type, it is not necessarily obvious (or inevitable) that a printer should support the Business Card data type. In either case, whenever a data type is required for a particular device class, it is done to enhance interoperability between devices in a natural and achievable way.

Optional data types may, over time become required as device capabilities grow, and as the usage of IrReady devices becomes more common. There are currently a number of data types that are not required for device classes despite the fact that the usage would be natural and valuable for such a device. In these situations, vendors may add considerable value by implementing support for these data types. Over time, many of these optional data types will become required for certain device classes.

Conditional support applies to device classes that support Point and Shoot, but whose specific complement of application support is not appropriately dictated by this profile. For example, cell phones have a common set of possible applications (phone list, calendar, etc.), and will beam these data objects according to the Point and Shoot profile. However, it is not appropriate to say that all cell phones must support a phone list (or a calendar) application to be compliant with this profile. However, it is essential that these devices behave consistently. In other words, if they have the capability to push objects, then they must do so for all the defined applications that they support. In addition, these devices must support at least one of the data types defined as Conditional in the capability table for that device.

4.1.4.1 LAPTOPS AND DESKTOPS

It should be noted that since Laptops and Desktops are required to support generic files, all of the other data types can conceivably be sent and received if treated as generic files. The purpose of the optional entries in this table is to outline appropriate behavior that brings a more meaningful experience to the user where possible. For example, if a desktop has an application (such as Outlook or Palm Desktop) that maintains a default phone list, support of the Business Card data type requires that the user be provided an interface from within the phone list application for sending and receiving Business Cards. It is not sufficient to import and export these data objects from the application and then send and receive them as generic files.

Data Type	Support	Expected Appropriate Usage
Business Card	Optional	***Push Server:*** Business Card is stored into the database of the default PIM. ***Push Client:*** User interface is provided to select and send Business Card from within the PIM application.
Calendar Item	Optional	***Push Server:*** Calendar Item is stored into the database of the default calendar or scheduling application. ***Push Client:*** User interface is provided to select and send Calendar Item from within the calendar or scheduling application.
Text Note	Optional	***Push Server:*** Text Note is stored into the database of the default PIM. ***Push Client:*** User interface is provided to select and send Text Note form within the PIM application.
Message	Optional	***Push Server:*** Message is stored as an email in the default email application. ***Push Client:*** User interface is provided to select and send Message from within the email application.
ASCII Text	Optional	***Push Server:*** Received object is stored as an ASCII Text file. ***Push Client:*** User interface is provided to select and send an ASCII Text file.
Formatted Document File	Optional	***Push Server:*** Received objects are stored as files. ***Push Client:*** User interface is provided to select and send formatted document files from the file system.
Image	Optional	***Push Server:*** Images are received by default image manipulation application. ***Push Client:*** User interface is provided to select and send Images from the default image manipulation application.
File	Required	***Push Server:*** Received object is stored as a file in the file system. ***Push Client:*** User interface is provided to select and send any file from the file system.

4.1.4.2 GENERAL FUNCTION PRINTERS

Since Printers are receive-only devices, they are only required to support Push Server.

Data Type	Support	Expected Appropriate Usage
Business Card	Required	***Push Server:*** Received Business Card is printed in a form that is reasonably understandable to the user.
Calendar Item	Required	***Push Server:*** Received Calendar Item is printed in a form that is reasonably understandable to the user.
Text Note	Optional	***Push Server:*** Received Text Note is printed in a form that is reasonably understandable to the user.
Message	Optional	***Push Server:*** Received Message is printed in a form that is reasonably understandable to the user.
ASCII Text	Required	***Push Server:*** Received text file is printed to the page as text.
Formatted Document File	Optional	***Push Server:*** Received document files are printed according the specifications for the particular document format.
Image	Required	***Push Server:*** Received Image object is printed to the page in a form that is reasonably understandable to the user.
File	Not Applicable	None

4.1.4.3 PHOTO PRINTERS

Since Printers are receive-only devices, they are only required to support Push Server.

Data Type	Support	Expected Appropriate Usage
Business Card	Optional	***Push Server:*** Received Business Card is printed in a form that is reasonably understandable to the user.
Calendar Item	Optional	***Push Server:*** Received Calendar Item is printed in a form that is reasonably understandable to the user.
Text Note	Optional	***Push Server:*** Received Text Note is printed in a form that is reasonably understandable to the user.
Message	Optional	***Push Server:*** Received Message is printed in a form that is reasonably understandable to the user.
ASCII Text	Optional	***Push Server:*** Received text file is printed to the page as text.
Formatted Document File	Optional	***Push Server:*** Received document files are printed according the specifications for the particular document format.
Image	Required	***Push Server:*** Received Image object is printed to the page in a form that is reasonably understandable to the user.
File	Not Applicable	None

4.1.4.4 LABEL OR ADDRESS PRINTERS

Since Printers are receive-only devices, they are only required to support Push Server.

Data Type	Support	Expected Appropriate Usage
Business Card	Required	***Push Server:*** Received Business Card is printed in a form that is reasonably understandable to the user.
Calendar Item	Optional	***Push Server:*** Received Calendar Item is printed in a form that is reasonably understandable to the user.
Text Note	Optional	***Push Server:*** Received Text Note is printed in a form that is reasonably understandable to the user.
Message	Optional	***Push Server:*** Received Message is printed in a form that is reasonably understandable to the user.
ASCII Text	Optional	***Push Server:*** Received text file is printed to the page as text.
Formatted Document File	Optional	***Push Server:*** Received document files are printed according the specifications for the particular document format.
Image	Optional	***Push Server:*** Received Image object is printed to the page in a form that is reasonable understandable to the user.
File	Not Applicable	None

4.1.4.5 DIGITAL CAMERAS

Data Type	Support	Expected Appropriate Usage
Business Card	Optional	***Push Server:*** Digital camera receives and stores Business Card. ***Push Client:*** User interface is provided to select and send Business Card.
Calendar Item	Optional	***Push Server:*** Digital camera receives and stores Calendar Item. ***Push Client:*** User interface is provided to select and send Calendar Item.
Text Note	Optional	***Push Server:*** Digital camera receives and stores Text Note. ***Push Client:*** User interface is provided to select and send Text Note.
Message	Optional	***Push Server:*** Digital camera receives and stores Message. ***Push Client:*** User interface is provided to select and send Message.
ASCII Text	Optional	***Push Server:*** Digital camera receives and stores Text file. ***Push Client:*** User interface is provided to select and send Text file.
Formatted Document File	Optional	***Push Server:*** Digital camera receives and stores Formatted Document File. ***Push Client:*** User interface is provided to select and send Formatted Document File.
Image	Required	***Push Server:*** Digital camera receives and stores Image. ***Push Client:*** User interface is provided to select and send Image.
File	Not Applicable	None

4.1.4.6 PDAs

Data Type	Support	Expected Appropriate Usage
Business Card	Required	***Push Server:*** Business Card is stored into the phone list. ***Push Client:*** User interface is provided to select and send Business Card from the phone list.
Calendar Item	Required	***Push Server:*** Calendar Item is stored into the default calendar application. ***Push Client:*** User interface is provided to select and send Calendar Item from the calendar application.
Text Note	Optional	***Push Server:*** Text Note is stored into the note list. ***Push Client:*** User interface is provided to select and send Text Note from the note list.
Message	Optional	***Push Server:*** Message is stored into the email application. ***Push Client:*** User interface is provided to select and send Message from the email application.
ASCII Text	Required	***Push Server:*** Received object is stored in the collection of text files. ***Push Client:*** User interface is provided to select and send text files.
Formatted Document File	Optional	***Push Server:*** Formatted Document File is stored. ***Push Client:*** User interface is provided to select and send Formatted Document File.
Image	Optional	***Push Server:*** Received object is stored in collection of Images. ***Push Client:*** User interface is provided to select and send Images.
File	Optional	***Push Server:*** Received object is stored in file system. ***Push Client:*** User interface is provided to select and send files from the file system.

4.1.4.7 CELL PHONES

Data Type	Support	Expected Appropriate Usage
Business Card	Conditional	***Push Server:*** Received object is stored in the phone list. ***Push Client:*** User interface is provided to select and send Business Card from phone list.
Calendar Item	Conditional	***Push Server:*** Received object is stored in the calendar. ***Push Client:*** User interface is provided to select and send calendar items.
Text Note	Optional	***Push Server:*** Received object is stored in collection of Text Notes. ***Push Client:*** User interface is provided to select and send Text Notes.
Message	Conditional	***Push Server:*** Received object is stored in the collection of email messages. ***Push Client:*** User interface is provided to select and send email messages.
ASCII Text	Optional	***Push Server:*** Received object is stored as a text file. ***Push Client:*** User interface is provided to select and send text file.
Formatted Document File	Optional	***Push Server:*** Formatted Document File is stored. ***Push Client:*** User interface is provided to select and send Formatted Document File.
Image	Optional	***Push Server:*** Received object is stored in collection of images. ***Push Client:*** User interface is provided to select and send Images.
File	Not Applicable	None

4.1.4.8 PAGERS

Data Type	Support	Expected Appropriate Usage
Business Card	Conditional	***Push Server:*** Received object is stored in the phone list. ***Push Client:*** User interface is provided to select and send Business Card from phone list.
Calendar Item	Conditional	***Push Server:*** Received object is stored in the calendar. ***Push Client:*** User interface is provided to select and send calendar items.
Text Note	Conditional	***Push Server:*** Received object is stored in collection of Text Notes. ***Push Client:*** User interface is provided to select and send Text Notes.
Message	Conditional	***Push Server:*** Received object is stored in the collection of email messages. ***Push Client:*** User interface is provided to select and send email messages.
ASCII Text	Conditional	***Push Server:*** Received object is stored as a text file. ***Push Client:*** User interface is provided to select and send text file.
Formatted Document File	Optional	***Push Server:*** Formatted Document File is stored. ***Push Client:*** User interface is provided to select and send Formatted Document File.
Image	Optional	***Push Server:*** Received object is stored in collection of images. ***Push Client:*** User interface is provided to select and send Images.
File	Not Applicable	None

4.1.4.9 HANDHELD SCANNERS

Data Type	Support	Expected Appropriate Usage
Business Card	Optional	***Push Server:*** Business Card is stored. ***Push Client:*** User interface is provided to select and send Business Card.
Calendar Item	Optional	***Push Server:*** Calendar Item is stored. ***Push Client:*** User interface is provided to select and send Calendar Item.
Text Note	Optional	***Push Server:*** Text Note is stored. ***Push Client:*** User interface is provided to select and send Text Note.
Message	Optional	***Push Server:*** Message is stored. ***Push Client:*** User interface is provided to select and send Message.
ASCII Text	Optional	***Push Server:*** Text file is stored. ***Push Client:*** User interface is provided to select and send text file.
Formatted Document File	Optional	***Push Server:*** Formatted Document File is stored. ***Push Client:*** User interface is provided to select and send Formatted Document File.
Image	Required	***Push Server:*** Scanner receives and stores Image. ***Push Client:*** User interface is provided to select and send Image.
File	Not Applicable	None

4.2 Profile Overview

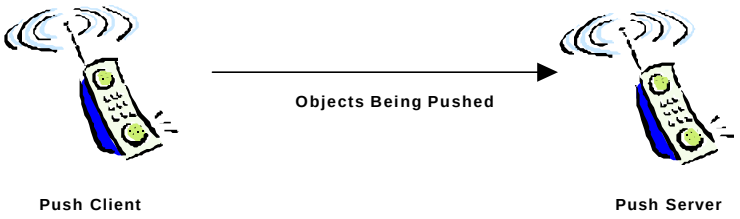
4.2.1 Configuration and Roles

The following roles are defined for this profile.

Push Server – The device that provides an object exchange server. The Push Server waits passively for the client to initiate the operation.

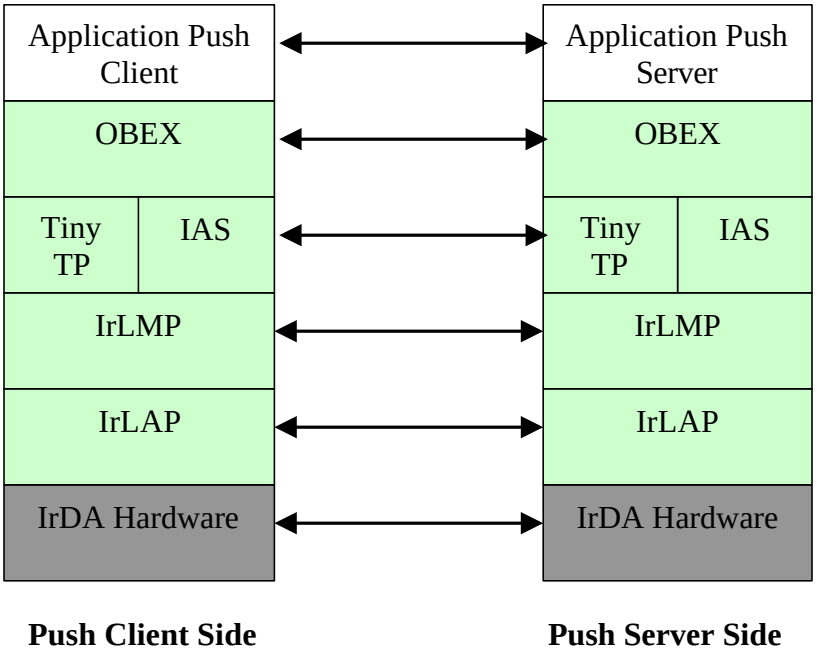
Push Client – The device that pushes the object to the Push Server. The Push Client initiates the operation.

The figure below shows an example of a Push between two mobile phones. In this example the phone on the left is a Push Client that pushes the object, and the phone on the right is a Push Server that receives the object.



4.2.2 Protocol Stack

The diagram below defines the minimum required protocol stack that shall be implemented by a device that conforms to this profile. Alternative stacks or components (such as JetSend [JETSEND]) may be implemented by a device in addition to this minimum.



IrDA Hardware is governed by the physical specification in [IrPHY].

IrLAP is the link level protocol specified in [IrLAP].

IrLMP is a multiplexing layer specified in [IrLMP].

Tiny TP provides flow control and is specified in [TTP].

IAS is the Information Access Service specified in [IrLMP].

OBEX includes both a session level protocol and an application framework. Both are specified in [IrOBEX].

Application Push Client and **Application Push Server** are the application entities, which provide the user interface and perform the operation of the Point and Shoot profile. They are discussed later in this document.

Ultra is not required for the Point and Shoot Profile. However, while not required, support is strongly encouraged in order to extend the interoperable reach of Point and Shoot devices [ULTRA].

4.2.3 Conformance

For a device to conform to this profile, all capabilities indicated as mandatory shall be supported in the specified manner. This also applies to all optional and conditional capabilities for which support is indicated. All supported capabilities are subject to verification as part of the IrReady 2000 certification program.

4.3 User Interface Aspects

4.3.1 Mode Selection

Push Server Mode is the state in which a Push Server is ready to receive an object from a Push Client. When entering this mode the Push Server must register the OBEX IAS entry and set the OBEX hint bit. It must be in a state where it is ready to respond to incoming Discovery frames and accept an incoming OBEX connection.

It is ideal that a Push Server be in this mode whenever the physical IR port is enabled (in other words, when the IR port is able to receive signals). In some devices the IR port is enabled whenever the device is turned on. For other devices the user must explicitly turn on the IR port. Turning on the IR port will ideally correspond to entering Push Server Mode.

However, it should be noted that for some devices it is either impractical or impossible to require that enabling the IR port will necessarily place the device into Push Server Mode. Where that ideal is not achievable, the user should be able to place the device into Push Server Mode as easily as possible, which is defined here as requiring only a single additional action on the part of the user.

To summarize, there are two scenarios in which a device will enter Push Server Mode:

Scenario One: The device automatically enters Push Server Mode when turned on. This is the recommended method.

Scenario Two: The device enters Push Server Mode when the user specifically enables it. This method may be required for devices with security or power usage constraints. The user should only be required to perform a single action via the user interface to cause the device to enter Push Server Mode.

4.3.2 Function Selection

The **Object Push Function** initiates the sending of an object to a Push Server. The user initiates this function. Typically, the function is selected for a specific object or group of objects. For example, in the Windows environment the user selects a file in the Explorer window, clicks the right mouse button and selects “IR recipient” from the “Send to” menu. When the selection is made the object is sent.

In most cases only one device will be available. The Point and Shoot Usage Model works best when the user selects the desired device by pointing at it. If multiple devices are in the IR space then the user must select from a list or be told to position the device so only one device is in range. The device may also use the hint bits of the discovered devices to identify those that support IrOBEX (since all Point and Shoot devices would support this hint bit). In this way a device could either present a more effective list to the user, or intelligently determine which of the devices is the most appropriate one to connect to.

4.3.3 Application Usage

When the user wants to push an object from a Push Client to a Push Server, the following is a typical scenario.

Push Client	**Push Server**
	The user sets the device into Push Server Mode if it is not already.
The user of the Push Client selects the object or objects to send.	
The user points the IR port of the Push Client device at the IR port of the Push Server device.	
The user selects the Object Push Function to send the selected object(s). (Objects being pushed should be of a type appropriate for the Push Server.) It is recommended that a status indicator show the progress of the operation.	
	It is recommended that user intervention be kept to a minimum on the Server device. It is possible that the user may be asked to accept or reject the object. Also if an object with the same name already exists the user may be asked if the existing object should be overwritten.
It is recommended that the user be notified of the result of the operation.	It is recommended that, where appropriate, the Push Server device notify its user of the result of the operation.

4.4 Application Layer

4.4.1 Feature Overview

A device following the Point and Shoot profile must be a Push Client, a Push Server or both and must support appropriate content types from those listed in the next section.

4.4.2 Content Types

To achieve application level interoperability, content formats are defined for Object Push. Sections 4.1.3 and 4.1.4 identify required and optional data types for Point and Shoot devices. If a device supports additional content formats for a given application, it must still support the required ones listed in these sections.

4.4.3 Generic File Push

Generic File transfer applications can send and receive files in any format. It is assumed that in this case both devices contain applications that understand the file format or that the file is simply stored on the device. It is also possible to send directory hierarchies containing Generic Files. Push Servers are not required to support Generic Files and if they do support them they are not required to support folder/directory hierarchies. The Push Client must verify that the Push Server supports folder/directory hierarchies by attempting to set the current folder of the receiving device to the root folder (see Section 4.5.5 for more details).

4.4.4 Application Architecture

The Push Client and Push Server are both built on top of the OBEX application framework. A Push Client uses OBEX to push objects to the inbox of a Push Server. The Push Client only knows that the objects are successfully received. It does not know the layout or construction of the Push Server's inbox.

A Push Server's inbox must take one of the following forms:

General Storage Location	*Holds objects of any type*. An example is a directory in a file system. It is possible to automatically dispatch objects from a general storage location to a database. For example, if a vCard is received it can be dispatched to the address book. The exception is when pushing a folder hierarchy. For example, vCards that are inside a folder being pushed are not automatically dispatched to the address book.
Database	*A data store or application that contains objects of a specific type*. An example is an address book in a mobile phone, which holds phone book items (vCards).
Process	*An application or program that processes the object as data*. An example is a printer, which will print the object. Processes are allowed to process the object as it is received. This means that the object does not have to be completely received before processing can begin. Since OBEX uses Tiny TP flow control the Push Server can properly pace the Push Client.

The table below shows the application procedure required by the Push Client for pushing one or more objects to a push server.

Push Client	**Details**
OBEX CONNECT	Target Header must not be used.
One or more OBEX PUTs for sending one or more objects.	
OBEX DISCONNECT	

The following table shows the application procedure required by the Push Client for pushing a folder hierarchy to a Push Server. This procedure is optional.

Push Client	Details
Set the current folder to the root using the SETPATH command. If this operation fails, then the Push Server does not support folders.	Name header is empty.
Create a new folder (if it does not already exist) in the Push Server's current folder using SETPATH. The current folder is changed to this new folder.	Name header is set to the name of the new folder.
Push all files to the new folder using a PUT command for each file.	The Name header is set to the name of the file.
Folders are created using SETPATH.	Name header is set to folder name. This application procedure is applied recursively to each folder until the folder hierarchy is sent. Setting the current folder to the root is only performed once at the beginning.
Set the current folder back to the parent folder using SETPATH.	The Backup flag is set and no Name header is sent.

4.5 OBEX Requirements

This section details the use of IrOBEX in the Point and Shoot Profile.

4.5.1 Symbols and Conventions

The Application Profile must use the following scheme to define the support for individual features. The following symbols are used:

M	Mandatory support. Refers to capabilities that shall be used in the profile.
O	Optional support. Refers to capabilities that can be used in the profile, but are not required.
X	Excluded. Refers to capabilities that may be supported by the device but shall not be used in this profile.

Some excluded capabilities are capabilities that, according to the relevant IrDA specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a device is operating as a device within this profile.

4.5.2 OBEX Operations

The table below shows the OBEX operations that are used in the Point and Shoot profile.

OBEX Operation	**Push Client**	**Push Server**
Connect	M	M
Disconnect	O	M
Put	M	M
Get	X	X
Set Path	O	O
Abort	M	M
Reserved	X	X
User Definable	X	X

4.5.3 OBEX Headers

The table below shows the OBEX headers used in the Point and Shoot profile.

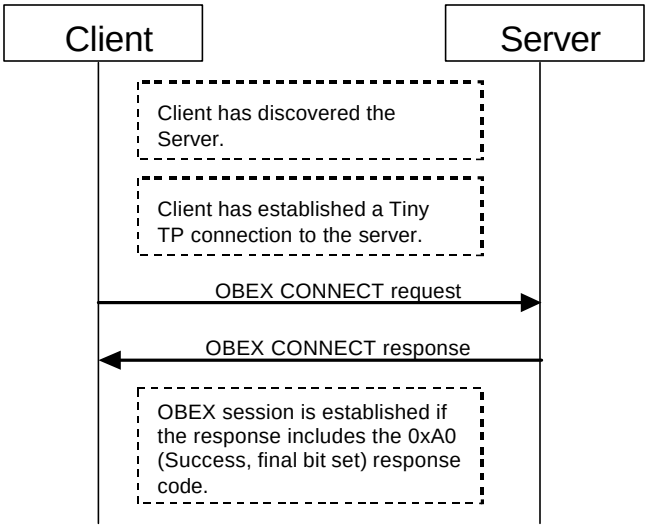
OBEX Headers	Push Client	Push Server
Count	O	O
Name	M	M
Type	O	O
Length	O	O
Time	O	O
Description	O	O
Target	X	X
HTTP	O	O
Body	M	M
End of Body	M	M
Who	X	X
Connection ID	X	X
Application Parameters	X	X
Authenticate Challenge	X	X
Authenticate Response	X	X
Object Class	X	X
Reserved	X	X
User Definable	X	X

4.5.4 Establishing an OBEX session

Setting up an OBEX session for Point and Shoot involves three steps:

1. Push Client discovers the Push Server device.
2. Push Client establishes a Tiny TP connection to the Push Server device. (See Section 4.6, Tiny TP/IrLMP Operations.)
3. Push Client performs an OBEX connect operation to the Push Server.

The figure below shows how the OBEX session is established.



The OBEX connection is established to the inbox service so no targeting information is used. The OBEX connect request must contain the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for CONNECT	0x80	M	
Field	Packet Length	Varies	M	
Field	OBEX Version Number	Varies	M	
Field	Flags	Varies	M	
Field	Max OBEX Packet Length	Varies	M	

The OBEX connect response must contain the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for CONNECT request	0x0A	M	0xA0 for success
Field	Packet Length	Varies	M	
Field	OBEX Version Number	Varies	M	
Field	Flags	Varies	M	
Field	Max OBEX Packet Length	Varies	M	

4.5.5 Pushing Objects

Objects are pushed to the Push Server using the OBEX PUT operation. Pushing an object can take one or more OBEX packets. Compliant OBEX Push Servers should be able to receive multiple sequential PUT requests.

The PUT packet must include the following fields and headers.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for PUT	0x02 or 0x82	M	0x02 is used for packets previous to the last put packet. 0x82 (which is 0x02 with the high bit set) is used for the last put packet.
Field	Packet Length	Varies	M	
Header	Name	Varies	M	The header value is the name of a single object, object store, or log information.
Header	Type	Varies	O	The MIME type of the object. This header is optional but highly recommended.
Header	Length	Varies	O	Length of the object. This header is optional but highly recommended.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

The response to the PUT request has the following fields and headers.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for PUT	0x90, 0xAO, 0xCD or 0xCF	M	0x90 for continue 0xA0 for success 0xCD if the object is too large 0xCF if the object type is not supported
Field	Packet Length	Varies	M	

Other headers, which can be optionally used, are found in [IrOBEX].

The type of an object is distinguished in two ways, first by using an extension in the name and second by sending a **Type** header. Sending a **Type** header is optional but highly recommended. Sending a **Name** header is mandatory. The table below shows the MIME types and name extensions required for each of the content types.

Object	MIME encoding (Type)	Name extension
vCard 2.1	text/x-vcard	.vcf
vCalendar 1.0	text/x-vcalendar	.vcs
vNote 1.1	text/x-vnote	.vnt
vMessage 1.1	text/x-vmessage	.vmg
Plain ASCII text	text/plain	.txt
PCL3 and PCL5	application/vnd.hp-pcl	.pcl
Postscript	application/postscript	.ps
ESC/P-80	application/octet-stream	
JPEG (Exif) Image	Image/jpeg	.jpg

4.5.6 Folder Operations

4.5.6.1 Setting the Current Folder to the Root

Setting the current folder to the root requires the SETPATH operation. The SETPATH request must include the following fields and headers.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for SETPATH	0x82	M	
Field	Packet Length	Varies	M	
Field	Flags	0x02	M	"Backup level" flag is set to 0 and "Don't Create" flag is set to 1
Field	Constants	0x00	M	Constants are not used and must be set to 0
Header	Name	Empty	M	Name header is empty

The response to the SETPATH request for setting the current folder to the root has the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for SETPATH	0xA0, 0xC3 or 0xD1	M	0xA0 for success 0xC3 if SETPATH is not supported 0xD1 if folders in the inbox are not supported

Field	Packet Length	Varies	M	

4.5.6.2 CREATING A FOLDER

Creating a new folder requires the SETPATH operation. The SETPATH request must include the following fields and headers.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for SETPATH	0x82	M	
Field	Packet Length	Varies	M	
Field	Flags	0x00	M	"Backup level" flag is set to 0 and "Don't Create" flag is set to 0
Field	Constants	0x00	M	Constants are not used and must be set to 0
Header	Name	Varies	M	Name of the folder

The response to the SETPATH request for creating a new folder has the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for SETPATH	0xA0 or 0xCD	M	0xA0 for success 0xCD if there is not enough room for a new folder
Field	Packet Length	Varies	M	

4.5.6.3 SETTING THE CURRENT FOLDER BACK TO THE PARENT

Setting the current folder back to the parent folder requires the SETPATH operation. The SETPATH request must include the following fields and headers.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for SETPATH	0x82	M	
Field	Packet Length	Varies	M	
Field	Flags	0x03	M	"Backup level" flag is set to 1 and "Don't Create" flag is set to 1
Field	Constants	0x00	M	Constants are not used and must be set to 0

Response to the SETPATH request for setting the current folder back to the parent has the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for SETPATH	0xA0 or 0xC4	M	0xA0 for success or 0xC4 if the current folder is the root.
Field	Packet Length	Varies	M	

4.5.7 Disconnecting an OBEX session

An OBEX session can be disconnected in two ways. First, the OBEX connection can be disconnected using the DISCONNECT procedure. Second, the underlying Tiny TP connection can be disconnected. Normally after all objects have been pushed, the Tiny TP connection to the OBEX server is disconnected so there is really no need to perform an OBEX DISCONNECT procedure. If the Tiny TP connection is used for other purposes as well (such as synchronization or file transfer) then the Tiny TP connection must be left up and an OBEX DISCONNECT should be issued. Push Servers must be able to handle both methods of disconnect.

The OBEX DISCONNECT request must contain the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Opcode for DISCONNECT	0x81	M	
Field	Packet Length	Varies	M	

Other headers (such as **Description)** which can be optionally used are found in [IrOBEX].

The response to an OBEX DISCONNECT request must contain the following fields.

Field/ Header	Name	Value	M/O	Explanation
Field	Response code for DISCONNECT	0xAO	M	0xA0 for success
Field	Packet Length	Varies	M	

4.6 Tiny TP/IrLMP Requirements

This section details the use of Tiny TP and IrLMP in the Point and Shoot Profile.

4.6.1 Tiny TP/IrLMP Operations

Tiny TP and IrLMP combine to form the IrDA transport layer. The steps involved in setting up a Tiny TP connection to a Push Server are as follows:

1. Push Client discovers the Push Server device and establishes an IrLAP connection.
2. Push Client queries the IAS of the Push Server for the LSAP-SEL entry of the OBEX IAS entry (see Section 4.7 IAS for more details).
3. Push Client performs a Tiny TP connect request to the LSAP-SEL retrieved in step 2.

If the Push Client already has an IrLAP connection to the Push Server then step 1 can be skipped and the Push Client should start at step 2.

For the purposes of this Profile, a Point and Shoot device is not expected to be capable of sending and receiving simultaneously. However, a Point and Shoot device is expected to handle such a request from another device without dropping the connection.

Note: If the Push Client has Push Server capability and there is already a Tiny TP connection to the Push Server of the Push Client device, then there may be problems if the Push Client attempts to establish a Tiny TP connection to the Push Server device. The reason is because the Push Server device may be built using IrDA Lite [LITE]. IrDA Lite does not require support of Tiny TP/IrLMP disconnect and some IrDA Lite devices may only allow one Tiny TP connection to OBEX at a time. This means that these devices cannot support two Tiny TP connections to OBEX at once (one from its Client to the Server of another device and one from the Client of another device to its server). If one of these devices receives an incoming Tiny TP connection when it already has an outgoing Tiny TP connection then it may perform an IrLAP disconnect which will disconnect all Tiny TP/IrLMP connections.

4.6.2 Discovering the Push Server

The Push Client device must discover the Push Server device using the IrLMP discovery service described in [IrLMP]. If the Push Client device already has an IrLAP connection to a another device, then it is assumed that this other device is the Push Server.

4.6.3 Establishing a Tiny TP Connection

The Push Client must establish a Tiny TP connection to the Push Server using the Connect request procedure described in [TTP].

4.7 IAS Requirements

The Push Server must have an IAS entry for a default OBEX server as described in [IrOBEX].

4.8 IrLAP Requirements

There are no special issues concerning IrLAP.

4.9 Physical Layer Requirements

Devices are allowed to support the short-range option as described in [IrPHY].

第2部

ポイントアンドシュート応用プロファイルテスト仕様

Point and Shoot Application Profile Test Specification

Version 1.0

March 30, 2000

Table of Contents

1 DOCUMENT STATUS

Point and Shoot Working Group Convenor:
Bryce Jeppsen, Hewlett-Packard bryce_jeppsen@am.exch.hp.com

Document Editor:
Glade Diviney, Extended Systems gladed@extendsys.com

Contributors:
Keming Yeh, ACTiSYS

History:
Version 0.1: Initial draft.

2 INTRODUCTION

This document describes the tests that an approved IrDA Test Lab must perform on a device to ensure compliance with the Point and Shoot Profile described in [PNS]. For a device to become a candidate for IrReady Point and Shoot Profile certification, it must pass all of the tests as specified by this document.

2.1 Acronyms and Abbreviations

{PDU = Protocol Data Unit}
{DUT = Device Under Test}
{bps = Bits Per Second}

2.2 References

[Revise as necessary]

[IrLAP]	Serial Infrared Link Access Protocol, IrLAP, Version 1.1, Infrared Data Association
[IrLMP]	Link Management Protocol, IrLMP, Version 1.1, Infrared Data Association
[IrPHY]	Serial Infrared Physical Layer Link Specification, IrPHY, Version 1.3, Infrared Data Association
[PHYTEST]	IrReady 2000 and IrDA Reference Device Physical Test Guideline, Version 0.2, Infrared Data Association
[TINYTP]	Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, Infrared Data Association
[LITE]	Minimal IrDA Protocol Implementation, IrDA Lite, Version 1.0, Infrared Data Association
[OBEX]	IrDA Object Exchange Protocol, IrOBEX, Version 1.2, Infrared Data Association
[IrMC]	IrMC (Ir Mobile Communications) Specification, Version 1.1, February 1999, Infrared Data Association
[VCARD]	VCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996, The Internet Mail Consortium
[VCAL]	VCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996, The Internet Mail Consortium
[JETSEND]	JetSend Protocol on IrDA Application Note, , Version 1.1, November1999, Infrared Data Association
[PNS]	IrDA Point and Shoot Profile, Version 1.0, January 2000, Infrared Data Association
[INTEROP]	Interoperability Test Plan and Process, Version 1.2, September 1998, Infrared Data Association

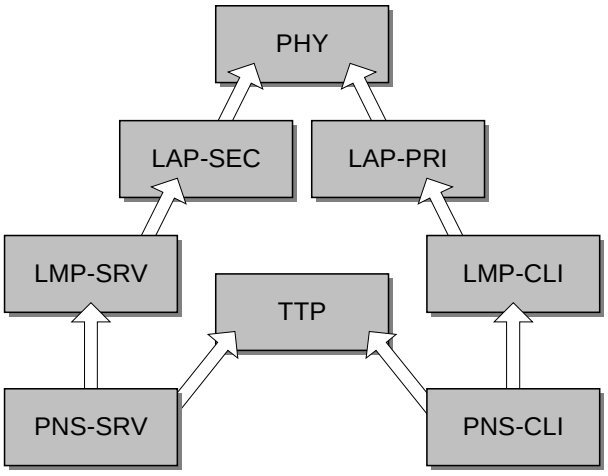
3 TEST PROFILES

[PNS] defines the Point and Shoot "usage model" and the Point and Shoot "application profile". This test specification defines a number of "test profiles". For a candidate device to be awarded IrReady 2000 Point and Shoot compliance, an IrReady Test Lab must demonstrate that the candidate device complies with all applicable test profiles.

For example, [PNS] defines an application profile with two independent modes: Push Client and Push Server. They are independent because a compliant device may be a Push Client, a Push Server, or both. These two modes correspond to separate Test profiles (PNS-SRV and PNS-CLI).

Furthermore, Push Client and Push Server behavior depend on typical infrared implementation layers such as IrLMP, IrLAP, and physical. For organizational purposes, this document defines a hierarchy of Test Profiles for these layers. It may be convenient for other profiles' test specifications to refer to the test profiles defined here.

The following diagram illustrates the dependencies of test profiles. For example, The PNS-CLI (Point and Shoot Push Client) profile depends on the TTP and LMP-CLI test profiles, which in turn depend on LAP-PRI and PHY test profiles. These test profiles are defined in more detail later in this chapter.



It is important to note that a demonstration of test profile compliance need not be exhaustive: tests will cover only a subset of the possible behaviors indicated by the protocol specifications (such as [IrLAP], [IrLMP], and the like). However, results from these tests are representative enough for an IrDA-approved test lab to determine whether the device is IrReady 2000 compliant.

The test profiles required by the Point and Shoot application profile are defined in the sections below.

3.1 Physical Layer Test Profile (PHY)

A device that can demonstrate a minimal level of compliance with [IrPHY] complies with this test profile. It is understood that not *all* physical characteristics need be tested, but test results should provide a representative sample to demonstrate interoperability. The tests and report formats described in the [PHYTEST] document are sufficient to demonstrate compliance with this PHY test profile.

3.2 IrLAP Secondary Test Profile (LAP-SEC)

A device that can demonstrate a minimal level of interoperability with IrLAP primary devices complies with this LAP-SEC test profile. Compliance with LAP-PRI also implies compliance with the PHY test profile.

Candidates must be able to demonstrate the following procedures:

1) Respond to discoveries from a primary device. The candidate's responses must include appropriate hint bits and a nickname. Responses must be formatted correctly and comply with all limitations and requirements as specified in [IrLAP].

2) Respond to IrLAP connection requests from a primary device. The candidate must respond with properly formatted connection parameters and honor the primary's requested connection settings.

3) Exchange data reliably during a connection. The candidate must discard improperly formatted data. Nr and Ns counts must be handled as specified by [IrLAP].

LAP-SEC test reports must include data captured from the session that shows timing, packet framing (XBOF, FCS, etc) and packet payload data from all devices involved in the demonstration. Any publicly available IrDA device or test tool may be used as the secondary device, as long as all of the procedures above are demonstrated.

3.3 IrLAP Primary Test Profile (LAP-PRI)

A device that can demonstrate a minimal level of interoperability with IrLAP secondary devices complies with this LAP-PRI test profile. Compliance with LAP-PRI also implies compliance with the PHY test profile.

Candidates must demonstrate the following procedures:

1) Initiate discoveries periodically or when a connection is requested by the user. The candidate's discovery frames must be formatted correctly and comply with all limitations and requirements as specified in [IrLAP].

2) Initiate IrLAP connections to discovered secondary devices. The candidate must transmit a properly formatted connection request, and honor the connection settings requested by the secondary.

3) Exchange data reliably during a connection. The candidate must discard improperly formatted data. Nr and Ns counts must be handled as specified by [IrLAP].

4) Close the IrLAP connection properly using DISC frames when a session is complete.

LAP-PRI test reports must include data captured from the session that shows timing, packet framing (XBOF, FCS, etc) and packet payload data from all devices involved in the demonstration. Any publicly available IrDA device or test tool may be used as the secondary device, as long as all of the procedures above are demonstrated.

3.4 IrLMP Server Test Profile (LMP-SRV)

To comply with the LMP-SRV test profile, a device must demonstrate a minimum level of interoperability with remote IrLMP and IAS client implementations. Compliance with LMP-SRV also implies compliance with the LAP-SEC test profile. (Compliance with the LAP-PRI test profile is not required for LMP-SRV test profile compliance but may be required by other higher-level test profiles.)

Candidates must demonstrate the following procedures:

1) Respond successfully to a request for an IAS connection according to the formats and procedures in [IrLMP].

2) Respond to requests for IAS attributes using the GetValueByClass query as specified in [IrLMP]. Specifically, the “Device” class, “DeviceName” attribute must be present. At least one other service must also be present, with an attribute that indicates its LSAP.

3) Respond successfully to a connection request for any advertised service.

4) Allow higher-layer data traffic to flow in both directions between the IrLMP client and the IrLMP server.

LMP-SRV test reports must include data captured from the session that shows data transmissions from both the client and server devices. Any publicly available IrDA device or test tool may be used as the IrLMP client, as long as all of the procedures above are demonstrated.

3.5 IrLMP Client Test Profile (LMP-CLI)

To comply with the LMP-CLI test profile, a device must demonstrate a minimum level of interoperability with remote IrLMP and IAS server implementations. Compliance with LMP-CLI also implies compliance with the LAP-PRI test profile, because Point and Shoot clients are generally responsible for opening the IrLAP connection.

Candidates must demonstrate the following procedures:

1) Request an IAS connection according to the formats and procedures in [IrLMP].

2) Generate at least one IAS GetValueByClass query to the IAS server. A request for the appropriate LSAP of a remote service must be made.

3) Initiate an IrLMP connection request using the LSAP provided by the server.

4) Allow higher-layer data traffic to flow in both directions between the IrLMP client and the IrLMP server.

LMP-CLI test reports must include data captured from the session that shows data transmissions from both the client and server devices. Any publicly available IrDA device or test tool may be used as the IrLMP server, as long as all of the procedures above are demonstrated.

3.6 Tiny TP Test Profile (TTP)

To comply with the TTP test profile, a device must demonstrate a minimum level of interoperability with remote Tiny TP implementations. Compliance with either LMP-SRV or LMP-CLI is required for compliance with the TTP test profile.

Candidates must demonstrate the following procedures:

1) Provide credits to a remote Tiny TP entity as defined by [TTP]. Credits must be advanced in the connection packet or shortly thereafter.

2) Prevent data from being transmitted when credits have been consumed.

3) Allow data to be transmitted when credits have been granted.

4) Grant an appropriate number of credits to the remote device to allow it to transmit data. Credits must not be advanced beyond the Tiny TP limitation of 127 credits, nor should credits be withheld for noticeably long periods of time.

TTP test reports must include data captured from the session that shows data transmissions from both the client and server devices. Any publicly available IrDA device or test tool may be used as the remote Tiny TP entity, as long as all of the procedures above are demonstrated.

3.7 Point and Shoot Push Server Test Profile (PNS-SRV)

To comply with the Point and Shoot Push Server test profile, a device must demonstrate a minimum level of interoperability with OBEX client entities. Compliance with TTP and LMP-SRV test profiles is required for compliance with the PNS-SRV test profile.

Candidates must demonstrate the following procedures:

1) Respond to an “OBEX” class, “IrDA:TinyTP:LsapSel” attribute IAS query with an LSAP that identifies the OBEX service.

2) Respond to “Connect”, “Put”, and “Disconnect” operations correctly as specified by [OBEX].

3) During a “Put”, accept data packets of any size up to the maximum specified in the Push Server’s “Connect” packet.

4) Accept “Abort” or link loss at any point during a “Put” to terminate a “Put”.

5) Accept multiple “Put” operations during the same OBEX connection.

6) Accept any of the following headers during a Put operation: Count, Name, Type, Length, Time, Description, HTTP, Body, and End of Body. Receipt of any of these headers should not cause a Put to fail unless they specify an object that the Push Server may legitimately reject. Note that this demonstration need not show that all received header is used, except for Body and End of Body headers.

7) Appropriately use objects received during a Put operation as determined by the Device Class of the candidate device. The appropriate use of received objects is detailed in section 4.1.4 of [PNS].

PNS-SRV test reports must include data captured from the session that shows data transmissions from both the client and server devices. Any publicly available IrDA device or test tool may be used as the Push Client, as long as all of the procedures above are demonstrated.

3.8 Point and Shoot: Push Client Test Profile (PNS-CLI)

To comply with the Point and Shoot Push Client test profile, a device must demonstrate a minimum level of interoperability with OBEX server entities. Compliance with TTP and LMP-CLI test profiles is required for compliance with the PNS-CLI test profile.

Candidates must demonstrate the following procedures:

1) Upon the user’s request to push an object or objects, establish an IrLAP/IrLMP/Tiny TP/OBEX transport connection.

2) Send the “Connect” operation and one or more “Put” operations using the formats and procedures specified by [OBEX].

3) Accept link loss at any point before, during, or after a “Put” operation.

4) Generate each type of data required for the Push Client’s device class, as specified in section 4.1.4 of [PNS].

PNS-SRV test reports must include data captured from the session that shows data transmissions from both the client and server devices. Any publicly available IrDA device or test tool may be used as the Push Client, as long as all of the procedures above are demonstrated.

＜付録＞

IrDA ポイントアンドシュートプロファイルおよびテスト仕様の概要紹介

本標準では、先に制定している IrOBEX（IrDA 赤外線オブジェクト交換プロトコル）に基づく端末アプリケーションの促進を図るため、IrDA 規格の赤外線データ通信における高速性、一定範囲に閉じる通信範囲、指向性、そして一対一の通信である事を最大限に生かす「ポイントアンドシュート」と呼ぶ操作手順モデルを提示し、その端末アプリケーションを実現するプロファイル、および当該プロファイルを実装する端末同士の相互接続性を確保するためのテスト仕様を規定している。

「ポイントアンドシュート」(Point and Shoot)とは、IrOBEX によるデータオブジェクトの送信(オブジェクトプッシュ)の際の典型的な操作手順を表している。例えば、自分の携帯電話上の名刺データを他の人の PDA(携帯情報端末)やパソコンに送信する場合、まず自分の携帯電話の赤外線ポートを相手に向け、名刺送信のボタンを押す、という簡単な操作で比較的少量のデータを瞬時に送信する手順を提供する。「ポイントアンドシュート」は、PDA や携帯電話等のモバイル端末における名刺交換やプリントアウト、fax 送信、ディジタル画像送信等、様々な端末アプリケーションに対応している。

ポイントアンドシュートプロファイルでは、「ポイントアンドシュート」を実現し、相互接続性を確保するために最低限必要なプロトコルおよび手順についての実装規約を規定している。

まず、対応する端末アプリケーションに対応するデータタイプとして、名刺(vCard)、スケジュール(vCalendar)、ノート(vNote)、メッセージ(vMessage)、テキストファイル、ドキュメントファイル、画像イメージ、一般ファイルを定義し、さらにモバイル端末および通信相手となる機器(パソコンやプリンタ等)の実装におけるデータタイプの実装必須またはオプションの規定、および動作条件を既定している。次に、プッシュサーバ(Push Server)およびプッシュクライアント(Push Client)の操作・動作モデルを示すとともに、端末アプリケーションにおける IrOBEX 実装の要求条件として、OBEX のオペレーション、OBEX ヘッダ、OBEX セッションの確立、オブジェクト送信等について、実装必須またはオプションの規定、および動作条件を規定している。

テスト仕様においては、相互接続性テストの仕様として、各プロトコルスタック毎のテスト手順と、プッシュサーバ、プッシュクライアントのテスト手順を規定している。