TTC標準
Standard

JF-IR008.10 (E)

赤外線通信インタフェース
腕時計型端末用赤外線通信仕様

IrWW (Infrared Wrist Watch) Specifications

第 1.0 版

2001 年 11 月 27 日制定

社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

Telecommunication
Technology
Committee

全体構成

参考

第１部　一般アクセスプロファイル

１．序文
　１．１　IrWW のプロファイル構成
　１．２　用語定義
　１．３　シンボル定義
　１．４　参考文献
２．利用モデル
　２．１　時刻同期利用モデル
　　２．１．１　範囲
　　２．１．２　概要
　　２．１．３　相互接続性
　　２．１．４　有用性
　　２．１．５　設定
　　２．１．６　信頼性
　　２．１．７　付加情報
　２．２　対向送信型オブジェクトプッシュ
　　２．２．１　範囲
　　２．２．２　概要
　　２．２．３　相互接続性
　　２．２．４　有用性
　　２．２．５　設定
　　２．２．６　信頼性
　　２．２．７　付加情報
３．IrWW 一般アクセスプロファイル
　３．１　ユーザ要求
　　３．１．１　範囲
　　３．１．２　概要
　３．２　プロファイル概要
　　３．２．１　設定と役割
　　３．２．２　プロトコルスタック
　　３．２．３　適合性
　３．３　ユーザインタフェース
　　３．３．１　モード選択(サーバ)

３．３．２　機能選択(クライアント)
３．３．３　アプリケーションの使用法
３．４　アプリケーション層
３．４．１　概要
３．４．２　一般バイナリオブジェクト
３．４．３　bWATCH
３．４．３．１　プロパティ
３．４．３．２　プロパティ一覧
３．４．３．３　プロパティパラメーター一覧
３．４．３．４　プロパティ値一覧
３．４．３．５　プロパティ順序
３．４．３．６　プロパティ制限
３．４．３．７　Ultraプロトコル使用時のプロパティ順序
３．４．４　bWatch情報記録
３．４．４．１　プロパティ一覧
３．４．４．２　X-IRWWフィールド
３．４．４．３　プロパティ順序
３．５　アプリケーション手続き
３．５．１　単位オブジェクトプッシュ
３．５．１．１　Ultra IrWWの単位オブジェクトプッシュ
３．５．１．２　IrOBEXの単位オブジェクトプッシュ
３．５．１．３　IrOBEXの単位オブジェクトプル
３．５．２　一般オブジェクト交換
３．６　OBEX
３．６．１　OBEXの動作
３．６．２　OBEXヘッダ
３．６．３　OBEX接続の確立
３．６．４　オブジェクトの読み出し
３．６．５　オブジェクトの書き込み
３．６．６　OBEX接続の切断
３．７　Ultra IrWW
３．８　IrWW製品の為のプロトコル実装ガイドライン
３．８．１　接続型非接続型の連携動作
３．９　 Tiny TP
３．１０　IrLMP
３．１１　IAS

３．１１．１　LsapSel
３．１１．２　パラメーター一覧
３．１１．３　デバイス情報
３．１２　サービスヒントビット
３．１３　IrLAP(必要最低条件)
３．１４　物理層

第２部　時刻同期プロファイル

１．序文
１．１　IrWWのプロファイル構成
１．２　シンボル定義
１．３　参考文献
２．時刻同期利用モデル
２．１　範囲
２．２　概要
２．３　相互接続性
２．４　有用性
２．５　設定
２．６　信頼性
２．７　付加情報
３．時刻同期プロファイル
３．１　ユーザ要求
３．１．１　範囲
３．１．２　概要
３．２　プロファイル概要
３．２．１　設定と役割
３．２．２　プロトコルスタック
３．２．３　適合性
３．３　ユーザインタフェース
３．３．１　モード選択(サーバ)
３．３．２　機能選択(クライアント)
３．３．３　アプリケーションの使用法
３．４　アプリケーション層
３．４．１　機能概要 (時刻同期の基本構想)
３．４．２　コンテントタイプ
３．４．３　時刻同期オブジェクト

３．４．４　アラームオブジェクト
３．４．５　タイマーオブジェクト
３．４．６　ストップウォッチオブジェクト
３．５　　Ultra IrWW
３．６　　ウルトラプロトコル
３．７　　IrLMP
３．８　　IrLAP
３．９　　物理層
４．添付:既存 PC/PDA 向け時刻同期プロファイル
４．１　　ユーザ要求
４．１．１　範囲
４．１．２　概要
４．２　　プロファイル概要
４．２．１　設定と役割
４．２．２　プロトコルスタック
４．２．３　適合性
４．３　　ユーザインタフェース
４．３．１　モード選択(サーバ)
４．３．２　機能選択(クライアント)
４．３．３　アプリケーションの使用法
４．４　　アプリケーション層
４．４．１　機能概要
４．４．２　コンテントタイプ
４．４．２．１　時刻同期オブジェクト
４．４．２．２　アラームオブジェクト
４．４．２．３　アラーム情報記録オブジェクト
４．４．２．４　タイマーオブジェクト
４．４．２．５　タイマー情報記録オブジェクト
４．４．２．６　ストップウォッチオブジェクト
４．４．２．７　ストップウォッチ情報記録オブジェクト
４．４．３　　アプリケーション手続き
４．４．３．１　IrOBEX での単位オブジェクトプル
４．４．３．２　IrOBEX での単位オブジェクトプッシュ
４．４．３．３　IrOBEX でのリードオール/ライトオールオブジェクト
４．５　　　OBEX
４．５．２　　OBEX の動作

４．５．３　OBEX ヘッダ
４．５．４　OBEX 接続の確立
４．５．５　オブジェクトの読み出し
４．５．６　オブジェクトの書き込み
４．５．７　OBEX 接続の切断
４．６　Tiny TP
４．７　IrLMP
４．８　IAS
４．８．２　パラメーター一覧
４．８．３　デバイス情報
４．９　サービスヒントビット
４．１０　IrLAP
４．１１　物理層

第３部　簡易オブジェクト交換プロファイル

１．序文
１．１　IrWW のプロファイル構成
１．２　用語定義
１．３　シンボル定義
１．４　参考文献
２．対向送信型オブジェクトプッシュ
２．１　範囲
２．２　概要
２．３　相互接続性
２．４　有用性
２．５　設定
２．６　信頼性
２．７　付加情報
３．簡易オブジェクト交換プロファイル
３．１　ユーザ要求
３．１．１　範囲
３．１．２　概要
３．２　プロファイル概要
３．２．１　設定と役割
３．２．２　プロトコルスタック
３．２．３　適合性

３．３　ユーザインタフェース
３．３．１　モード選択(サーバ)
３．３．２　機能選択(クライアント)
３．３．３　アプリケーションの使用法
３．３．３．１　単位プッシュ
３．３．３．２　リードオール/ライトオール
３．４　アプリケーション層
３．４．１　機能概要
３．４．２　コンテントタイプ
３．４．２．１　単位プッシュオブジェクト
３．４．２．１．１　時刻記録オブジェクト
３．４．２．１．２　名刺オブジェクト
３．４．２．１．３　スケジュールオブジェクト
３．４．２．１．４　仕事オブジェクト
３．４．２．１．５　ノートオブジェクト
３．４．２．２　リードオール/ライトオールオブジェクト
３．４．２．２．１　装置情報
３．４．２．２．２　時刻記録オブジェクト
３．４．２．２．３　時刻記録の情報記録オブジェクト
３．４．２．２．４　所有者名刺オブジェクト
３．４．２．２．５　名刺オブジェクト
３．４．２．２．６　名刺情報記録オブジェクト
３．４．２．２．７　スケジュールオブジェクト
３．４．２．２．８　スケジュール情報記録オブジェクト
３．４．２．２．９　仕事オブジェクト
３．４．２．２．１０　仕事情報記録オブジェクト
３．４．２．２．１１　ノートオブジェクト
３．４．２．２．１２　ノート情報記録オブジェクト
３．４．３　アプリケーション手続き
３．４．３．１　単位プッシュオブジェクトのアプリケーション手続き
３．４．３．２　リードオールオブジェクトのアプリケーション手続き
３．５　OBEX
３．５．１　OBEXの動作
３．５．２　OBEXヘッダ
３．５．３　OBEX接続の確立
３．５．４　オブジェクトのプッシュ

３．５．５　オブジェクトの読み出し

３．５．６　オブジェクトの書き込み

３．５．７　OBEX接続の切断

３．６　　Tiny TP

３．７　　IrLMP

３．８　　IAS

３．９　　サービスヒントビット

３．１０　IrLAP

３．１１　物理層

４．付録

４．１　バイナリvCard特有なコード割当表

４．２　バイナリvCalendar特有なコード割当表

４．３　バイナリvNote特有なコード割当表

第４部　一般バイナリオブジェクト(ＧＢＯ)仕様

１．序文

１．１　範囲

１．２　IrWWプロファイルとの関係

１．３　シンボル定義

１．４　参考文献

１．５　コンテント書式

２．一般バイナリオブジェクトの構成

２．１　ＧＢＯの基本構成

２．１．１　一般バイナリオブジェクトの構文

２．２　ＧＢＯの基本要素

２．２．１　入れ子になったＧＢＯ構成の取扱い

２．２．２　プロパティオペレーションコード

２．２．３　プロパティパート、グループパート、グループコードの表現

２．２．４　プロパティパートの表現

２．２．５　プロパティパラメータパートの要素

２．２．６　共通プロパティパラメータコード

２．２．６．１　共通プロパティパラメータコードの詳細

２．２．６．２　プロパティパートと構文フローの詳細

２．２．６．３　値に関する基本コーディングルール

２．２．６．４　固定値の表現

２．２．６．５　バイトストリーム値の表現

２．２．６．６　数値表現
２．２．６．７　複数の値と相関関係の表現方法
２．３　 オブジェクト関係図式
２．４　 オブジェクトストリーム識別子（オブジェクトマジックコード）
２．５　 装置情報オブジェクト
３．IrWW オブジェクト（bWatch）
３．１　bWATCH 構成識別子
３．２　bWATCH オブジェクト識別子
３．３　共通 IrWW プロパティコード（OUPC）
３．３．１　IrWW 独特な時間に関する値表現
３．３．１．１　VTHM 値識別子 VC=0x81
３．３．１．２　VHTMS 値識別子 VC=0x82
３．３．１．３　VTHMSL 値識別子 VC=0x83
３．３．１．４　VTRHM 値識別子 VC=0x84
３．３．１．５　VTRMS 値識別子 VC=0x85
３．３．１．６　VTRL 値識別子 VC=0x86
３．３．１．７　VDYMD 値識別子 VC=0x88
３．３．１．８　VDYMDW 値識別子 VC=0x89
３．３．１．９　VTUTC 値識別子 VC=0x8c
３．３．１．１０　VDTL 値識別子 VC = 0x8D
３．３．１．１１　VDTU 値識別子 VC = 0x8E

第５部　Ultra IrWW プロトコル仕様

１．序文
１．１　範囲
１．２　IrWW プロファイルとの関係
１．３　シンボル定義
１．４　参考文献
２．Ultra IrWW の詳細
２．１　プロトコルスタック
２．１．１　IrLMP の要求条件
２．１．２　IrLAP の要求条件
２．１．３　物理層の要求条件
２．２　 Ultra IrWW のフレーム構造
２．２．１　フレーム書式
２．２．２　フレーム構成

２．２．３　IrWW ペイロードデータ構造
２．３　 通信手続き

第６部　時刻同期プロファイル相互接続性試験

１．序文
１．１　目的
１．２　範囲
１．３　参考文献
２．環境
２．１　ハードウェア
２．２　ソフトウェア
３．相互接続性試験の概要
３．１　この製品にはどんな試験が適応されるのか
３．１．１　クライアント/サーバ
３．１．２　Ultra IrWW 転送試験
３．１．３　オブジェクト
３．１．４　その他の試験
３．２　 一般的な試験方法
４．Ultra IrWW 転送試験
４．１　クライアント
４．２　サーバ
５．エラー応答試験
５．１　クライアント
５．２　サーバ
６．その他の試験
６．１　内容
６．１．１　クライアント
６．１．２　サーバ
７．試験結果記入用テンプレート

第７部　付記 既存 PC/PDA における時刻同期プロファイル相互接続性試験

１．序文
１．１　目的
１．２　範囲
１．３　用語定義
１．４　参考文献

２．環境
２．１　ハードウェア
２．２　ソフトウェア
３．相互接続性試験の概要
３．１　この製品にはどんな試験が適応されるのか
３．１．１　クライアント/サーバ
３．１．２　プロトコル試験
３．１．３　リードオール/ライトオール試験
３．１．４　単位プッシュ試験
３．１．５　単位プル試験
３．１．６　オブジェクト
３．１．７　その他の試験
３．２　一般的な試験方法
４．プロトコル試験
４．１　ディスカバリヒントビット
４．１．１　クライアント
４．１．２　サーバ
４．２　IAS エントリ
４．２．１　クライアント
４．２．２　サーバ
４．３　装置情報
４．３．１　クライアント
４．３．２　サーバ
４．４　情報記録
４．４．１　クライアント
４．４．２　サーバ
５．リードオール/ライトオール試験
５．１　クライアント
５．２　サーバ
６．単位プッシュ試験
６．１　クライアント
６．２　サーバ
７．単位プル試験
７．１　クライアント
７．２　サーバ
８．エラー応答試験

８．１　クライアント
８．２　サーバ
９．その他の試験
９．１　内容
９．１．１　クライアント
９．１．２　サーバ
１０．試験結果記入用テンプレート

第８部　簡易オブジェクト交換プロファイル相互接続性試験

１．序文
１．１　目的
１．２　範囲
１．３　この文書作成の背景と動機
１．４　参考文献
２．環境
２．１　ハードウェア
２．２　ソフトウェア
３．相互接続性試験の概要
３．１　この製品にはどんな試験が適応されるのか
３．１．１　クライアント/サーバ
３．１．２　プロトコル試験
３．１．３　リードオール/ライトオール試験
３．１．４　単位プッシュ試験
３．１．５　オブジェクト
３．１．６　その他の試験
３．２　一般的な試験方法
４．プロトコル試験
４．１　ディスカバリヒントビット
４．１．１　クライアント
４．１．２　サーバ
４．２　IASエントリ
４．２．１　クライアント
４．２．２　サーバ
４．３　装置情報
４．３．１　クライアント
４．３．２　サーバ

４．４　　情報記録
４．４．１　クライアント
４．４．２　サーバ
５．リードオール/ライトオール試験
５．１　クライアント
５．２　サーバ
６．単位プッシュ試験
６．１　クライアント
６．２　サーバ
７．エラー応答試験
７．１　クライアント
７．２　サーバ
８．その他の試験
８．１　内容
８．１．１　クライアント
８．１．２　サーバ
９．試験結果記入用テンプレート

＜付録＞

＜参考＞

１．英文記述の適用レベル

適用レベル：Ｅ３

本標準の本文、付属資料および付録の文章および図に英文記述を含んでいる。

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１月に採択された標準IrWW（Infrared Wrist Watches）Version1.0 に基づいて定めたものである。IrWWは以下の８つのドキュメントから構成される。

・IrWW (Infrared Wrist Watch) Generic Access Profile Version 1.0 (December 26, 1999)

・IrWW (Infrared Wrist Watch) Time Synchronization Profile Version 1.0 (December 10, 1999)

・IrWW (Infrared Wrist Watch) Tiny Object Exchange Profile Version 1.0 (December 26, 1999)

・IrWW (Infrared Wrist Watch) Generic Binary Object (GBO) Specification Version 1.0 (December 26, 1999)

・Ultra IrWW Time Synchronization and Time Data Exchange using Ultra Protocol Version 1.0 (December 10,1999)

・IrWW (Infrared Wrist Watch) Time Synchronization Profile Compliance Tests for Ultra IrWW Version 1.0 (December 10, 1999)

・IrWW (Infrared Wrist Watch) Annex: Time Synchronize Profile for legacy PC/PDA Compliance Tests Version 1.0 (December 21, 1999)

・IrWW (Infrared Wrist Watch) Tiny Object Exchange Profile Compliance Tests Version 1.0 (December 26, 1999)

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　ナショナルマター決定項目

なし

３．３　先行している項目

なし

３．４　追加した項目

なし

３．５　削除した項目

なし

３．６　国際勧告に対する修正内容

なし

３．７　その他

なし

４．参照した国際勧告との章立ての構成の相違

（１）国際勧告と本標準の部番号は次のとおり対応している

本標準	国際勧告	備考
第１部　一般アクセスプロファイル	IrWW (Infrared Wrist Watch) Generic Access Profile	
第２部　時刻同期プロファイル	IrWW (Infrared Wrist Watch) Time Synchronization Profile	
第３部　簡易オブジェクト交換プロファイル	IrWW (Infrared Wrist Watch) Tiny Object Exchange Profile	
第４部　一般バイナリオブジェクト(GBO)仕様	IrWW (Infrared Wrist Watch) Generic Binary Object (GBO) Specification	
第５部　Ultra IrWWプロトコル仕様	Ultra IrWW Time Synchronization and Time Data Exchange using Ultra Protocol	
第６部　時刻同期プロファイル相互接続性試験	IrWW (Infrared Wrist Watch) Time Synchronization Profile Compliance Tests for Ultra IrWW	
第７部　付記　既存 PC/PDAにおける時刻同期プロファイル相互接続性試験	IrWW (Infrared Wrist Watch) Annex: Time Synchronize Profile for legacy PC/PDA Compliance Tests	
第８部　簡易オブジェクト交換プロファイル相互接続性試験	IrWW (Infrared Wrist Watch) Tiny Object Exchange Profile Compliance Tests	

（２）国際勧告と本標準の図および表は次のとおり対応している。

第１部 一般アクセスプロファイル（IrWW Generic Access Profile）

国際勧告	本標準
Chapter3 Figure3-1	図 1-3-1／JF-IR008.10(E)
Chapter3 Figure3-16	図 1-3-2／JF-IR008.10(E)
Chapter3 Figure2	図 1-3-3／JF-IR008.10(E)
Chapter3 Table1	表 1-3-1／JF-IR008.10(E)
Chapter3 Table2	表 1-3-2／JF-IR008.10(E)
Chapter3 Table3	表 1-3-3／JF-IR008.10(E)
Chapter3 Table4	表 1-3-4／JF-IR008.10(E)
Chapter3 Table5	表 1-3-5／JF-IR008.10(E)
Chapter3 Table6	表 1-3-6／JF-IR008.10(E)
Chapter3 Table3-11	表 1-3-7／JF-IR008.10(E)
Chapter3 Table3-12	表 1-3-8／JF-IR008.10(E)
Chapter3 Table3-13	表 1-3-9／JF-IR008.10(E)
Chapter3 Table3-14	表 1-3-10／JF-IR008.10(E)
Chapter3 Table7-15	表 1-3-11／JF-IR008.10(E)
Chapter3 Table3-16	表 1-3-12／JF-IR008.10(E)
Chapter3 Table3-17	表 1-3-13／JF-IR008.10(E)
Chapter3 Table3-18	表 1-3-14／JF-IR008.10(E)
Chapter3 Table3-19	表 1-3-15／JF-IR008.10(E)
Chapter3 Table8	表 1-3-16／JF-IR008.10(E)

第２部 時刻同期プロファイル（IrWW Time Synchronization Profile）

国際勧告	本標準
Section 3.2 Figure1	図 2-3-1／JF-IR008.10(E)
Section 3.4 Figure1	図 2-3-2／JF-IR008.10(E)
Chapter 4 Figure2	図 2-4-1／JF-IR008.10(E)
Chapter 3 Table1	表 2-3-1／JF-IR008.10(E)
Chapter 3 Table2	表 2-3-2／JF-IR008.10(E)
Chapter 3 Table3	表 2-3-3／JF-IR008.10(E)
Chapter 3 Table4	表 2-3-4／JF-IR008.10(E)
Chapter 3 Table5	表 2-3-5／JF-IR008.10(E)
Chapter 3 Table6	表 2-3-6／JF-IR008.10(E)
Chapter 3 Table7	表 2-3-7／JF-IR008.10(E)
Chapter 3 Table8	表 2-3-8／JF-IR008.10(E)
Chapter 4 Table1	表 2-4-1／JF-IR008.10(E)
Chapter 4 Table2	表 2-4-2／JF-IR008.10(E)
Chapter 4 Table3	表 2-4-3／JF-IR008.10(E)
Chapter 4 Table4	表 2-4-4／JF-IR008.10(E)
Chapter 4 Table5	表 2-4-5／JF-IR008.10(E)
Chapter 4 Table6	表 2-4-6／JF-IR008.10(E)

Chapter 4 Table7	表 2-4-7／JF-IR008.10(E)
Chapter 4 Table8	表 2-4-8／JF-IR008.10(E)
Chapter 4 Table9	表 2-4-9／JF-IR008.10(E)
Chapter 4 Table10	表 2-4-10／JF-IR008.10(E)
Chapter 4 Table11	表 2-4-11／JF-IR008.10(E)
Chapter 4 Table12	表 2-4-12／JF-IR008.10(E)
Chapter 4 Table13	表 2-4-13／JF-IR008.10(E)
Chapter 4 Table14	表 2-4-14／JF-IR008.10(E)
Chapter 4 Table15	表 2-4-15／JF-IR008.10(E)
Chapter 4 Table16	表 2-4-16／JF-IR008.10(E)
Chapter 4 Table17	表 2-4-17／JF-IR008.10(E)
Chapter 4 Table18	表 2-4-18／JF-IR008.10(E)
Chapter 4 Table19	表 2-4-19／JF-IR008.10(E)
Chapter 4 Table20	表 2-4-20／JF-IR008.10(E)
Chapter 4 Table21	表 2-4-21／JF-IR008.10(E)

第３部 簡易オブジェクト交換プロファイル（IrWW Tiny Object Exchange Profile）

国際勧告	本標準
Chapter 3 Figure1	図 3-3-1／JF-IR008.10(E)
Chapter 3 Figure2	図 3-3-2／JF-IR008.10(E)
Chapter 3 Table1	表 3-3-1／JF-IR008.10(E)
Chapter 3 Table2	表 3-3-2／JF-IR008.10(E)
Chapter 3 Table3	表 3-3-3／JF-IR008.10(E)
Chapter 3 Table4	表 3-3-4／JF-IR008.10(E)
Chapter 3 Table5	表 3-3-5／JF-IR008.10(E)
Chapter 3 Table6	表 3-3-6／JF-IR008.10(E)
Chapter 3 Table7	表 3-3-7／JF-IR008.10(E)
Chapter 3 Table8	表 3-3-8／JF-IR008.10(E)
Chapter 3 Table9	表 3-3-9／JF-IR008.10(E)
Chapter 3 Table10	表 3-3-10／JF-IR008.10(E)
Chapter 3 Table11	表 3-3-11／JF-IR008.10(E)
Section 3.4 Table12	表 3-3-12／JF-IR008.10(E)
Chapter 3 Table13	表 3-3-13／JF-IR008.10(E)
Chapter 3 Table14	表 3-3-14／JF-IR008.10(E)
Chapter 3 Table15	表 3-3-15／JF-IR008.10(E)
Chapter 3 Table16	表 3-3-16／JF-IR008.10(E)
Chapter 3 Table17	表 3-3-17／JF-IR008.10(E)
Chapter 3 Table18	表 3-3-18／JF-IR008.10(E)
Chapter 3 Table19	表 3-3-19／JF-IR008.10(E)
Chapter 3 Table20	表 3-3-20／JF-IR008.10(E)
Chapter 3 Table21	表 3-3-21／JF-IR008.10(E)
Chapter 3 Table22	表 3-3-22／JF-IR008.10(E)
Chapter 3 Table23	表 3-3-23／JF-IR008.10(E)

Chapter 3 Table24	表 3-3-24／JF-IR008.10(E)
Chapter 3 Table25	表 3-3-25／JF-IR008.10(E)
Chapter 3 Table26	表 3-3-26／JF-IR008.10(E)
Section 3.5 Table12	表 3-3-27／JF-IR008.10(E)
Chapter 3 Table27	表 3-3-28／JF-IR008.10(E)

第４部 一般バイナリオブジェクト(GBO)仕様

（IrWW Generic Binary Object(GBO) Specification）

国際勧告	本標準
Chapter1 Figure1	図 4-1-1／JF-IR008.10(E)
Chapter2 Figure2-1	図 4-2-1／JF-IR008.10(E)
Chapter2 Figure2-2	図 4-2-2／JF-IR008.10(E)
Section2.1 Figure2-3	図 4-2-3／JF-IR008.10(E)
Section2.2 Figure2-3	図 4-2-4／JF-IR008.10(E)
Chapter2 Figure2-4	図 4-2-5／JF-IR008.10(E)
Chapter2 Figure2-5	図 4-2-6／JF-IR008.10(E)
Chapter2 Figure2-6	図 4-2-7／JF-IR008.10(E)
Chapter2 Figure3-7	図 4-2-8／JF-IR008.10(E)
Chapter2 Figure2-8	図 4-2-9／JF-IR008.10(E)
Chapter2 Figure2-9	図 4-2-10／JF-IR008.10(E)
Chapter2 Figure2-10	図 4-2-11／JF-IR008.10(E)
Chapter2 Figure2-11	図 4-2-12／JF-IR008.10(E)
Chapter2 Figure2-12	図 4-2-13／JF-IR008.10(E)
Chapter2 Figure2-13	図 4-2-14／JF-IR008.10(E)
Chapter2 Figure2-14	図 4-2-15／JF-IR008.10(E)
Chapter3 Figure3-4	図 4-3-1／JF-IR008.10(E)
Chapter2 Table2-1	表 4-2-1／JF-IR008.10(E)
Chapter2 Table2-2	表 4-2-2／JF-IR008.10(E)
Chapter2 Table2-3	表 4-2-3／JF-IR008.10(E)
Chapter2 Table2-4	表 4-2-4／JF-IR008.10(E)
Chapter2 Table2-5	表 4-2-5／JF-IR008.10(E)
Chapter2 Table2-6	表 4-2-6／JF-IR008.10(E)
Chapter2 Table2-7	表 4-2-7／JF-IR008.10(E)
Chapter2 Table2-8	表 4-2-8／JF-IR008.10(E)
Chapter2 Table2-9	表 4-2-9／JF-IR008.10(E)
Chapter2 Table2-10	表 4-2-10／JF-IR008.10(E)
Chapter2 Table2-11	表 4-2-11／JF-IR008.10(E)
Chapter2 Table2-12	表 4-2-12／JF-IR008.10(E)
Chapter2 Table2-13	表 4-2-13／JF-IR008.10(E)
Chapter2 Table2-14	表 4-2-14／JF-IR008.10(E)
Chapter3 Table3-1	表 4-3-1／JF-IR008.10(E)
Chapter3 Table3-2	表 4-3-2／JF-IR008.10(E)
Chapter3 Table3-13	表 4-3-3／JF-IR008.10(E)

第５部 Ultra IrWW プロトコル仕様
（Ultra IrWW Time Synchronization and Time Data Exchange using Ultra Protocol）

国際勧告	本標準
Chapter1 Figure1	図 5-1-1／JF-IR008.10(E)
Chapter2 Figure2	図 5-2-1／JF-IR008.10(E)
Chapter2 Figure3	図 5-2-2／JF-IR008.10(E)
Chapter2 Figure4	図 5-2-3／JF-IR008.10(E)
Chapter2 Figure5	図 5-2-4／JF-IR008.10(E)
Chapter2 Figure6	図 5-2-5／JF-IR008.10(E)
Chapter2 Table1	表 5-2-1／JF-IR008.10(E)
Chapter2 Table2	表 5-2-2／JF-IR008.10(E)

５．改版の履歴

版数	制定日	改版内容
第１版	２００１年１１月２７日	制定

６．工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、ＴＴＣホームページでご覧になれます。

７．その他

(1)参照勧告、標準等

IrDA 標準:

IrPHY (Serial Infrared Physical Layer Link Specification)

IrLAP (Serial Infrared Link Access Protocol)

IrLMP (Serial Infrared Link Management Protocol)

TinyTP (A Flow-Control Mechanism for use with IrLMP)

IrOBEX (IrDA Object Exchange Protocol)

IrMC (Infrared Mobile Communications)

Ultra (Guidelines for Ultra Protocols)

IAS (IrLMP Service Hint Bit Assignments and Known IAS Definitions)

その他標準:

vCard (The Electronic Business Card Exchange Format)

vCalendar (The Electronic Calendaring and Scheduling Exchange Format)

８．標準作成部門

第四部門委員会　第四専門委員会

第1部

一般アクセスプロファイル

Infrared Data Association

IrWW (Infrared Wrist Watch)
Generic Access Profile

Version 1.0

December 26, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Ken-ichiro Shimokura	k.shimokura@east.ntt.co.jp
Gontaro Kitazumi	kitazumi@osw.co.jp

Significant Contributors:

History:
Version 0.5: Infrared Data Association IrWW IrDA for Wrist Watches for Directional Vote
Version 0.8: IrWW Generic Access Profile for Draft Vote.
Version 1.0: IrWW Generic Access Profile for Final Vote.

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 Introduction 1

1.1 IrWW Profile Structure 1

1.2 Definitions and Acronyms 2

1.3 Symbols and Conventions 2

1.4 References 2

2 Usage Models 4

2.1 Time Synchronization Usage Model (New) 4

2.1.1 Scope 4
2.1.2 User Scenario 4
2.1.3 Interoperability 4
2.1.4 Usability 4
2.1.5 Configuration 4
2.1.6 Reliability 4
2.1.7 Additional Information 4

2.2 Point and Shoot Object Push 5

2.2.1 Scope 5
2.2.2 User Scenario 5
2.2.3 Interoperability 5
2.2.4 Usability 6
2.2.5 Configuration 6
2.2.6 Reliability 6
2.2.7 Additional Information 6

3 IrWW Generic Access Profile 7

3.1 User Requirements 7

3.1.1 Scope 7
3.1.2 User Scenarios 7

3.2 Profile Overview 8

3.2.1 Configuration and Roles 8
3.2.2 Protocol Stack 8
3.2.3 Conformance 9

3.3 User Interface Aspects 10

3.3.1 Mode Selection (Server) 10
3.3.2 Function Selection (Client) 10
3.3.3 Application Usage 10

3.4 Application Layer 11

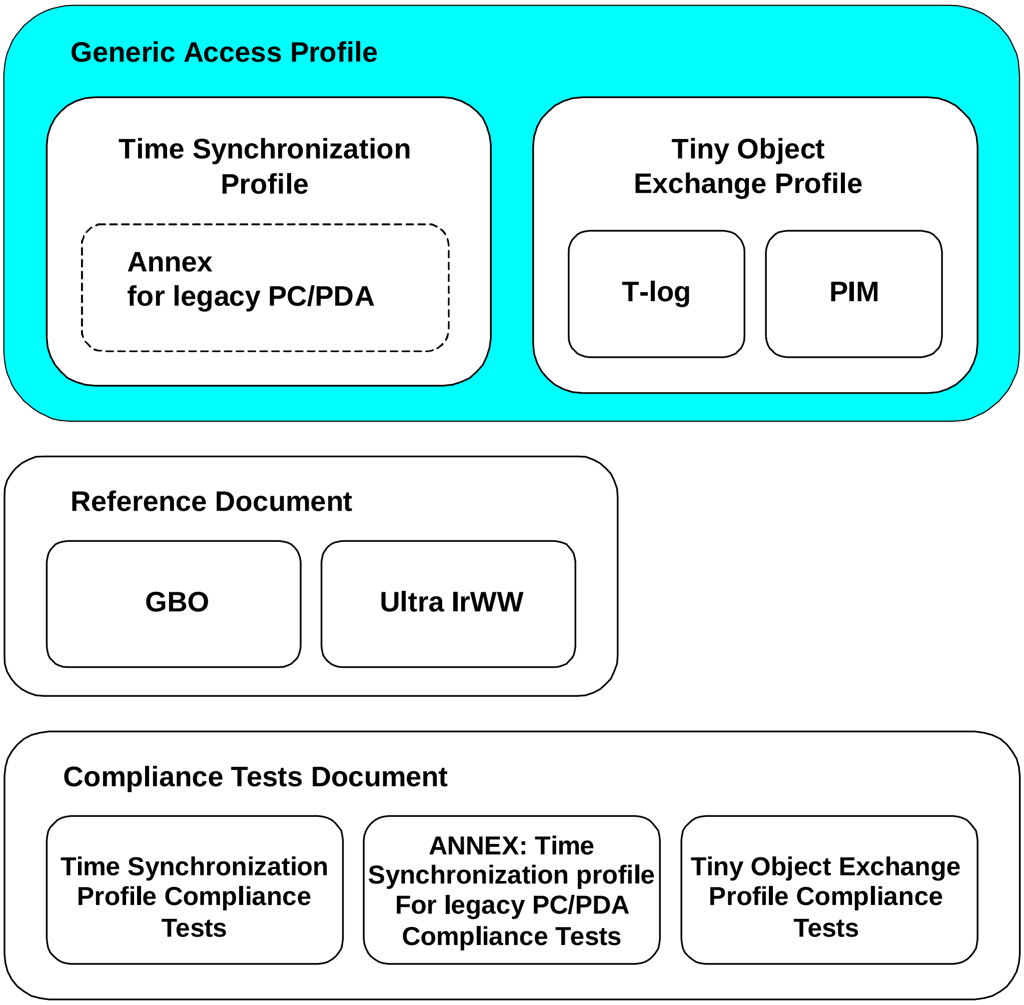
3.4.1 Overview 11
3.4.2 Generic Binary Object (GBO) 11
3.4.3 bWATCH 11
3.4.3.1 Property 11
3.4.3.2 Properties 12
3.4.3.3 Property Parameters 13
3.4.3.4 Property Values 14
3.4.3.5 Property Order 15
3.4.3.6 Property use restrictions 17

3.4.3.7 GBO property order for Ultra IrWW ... 18
3.4.4 bWatch Information log ... 19
3.4.4.1 Properties ... 19
3.4.4.2 X-IRWW-Fields ... 20
3.4.4.3 Property Order ... 20
3.5 Application Procedure ... 20
3.5.1 Single object push ... 20
3.5.1.1 Single Object Push with Ultra IrWW ... 20
3.5.1.2 Single Object Push with IrOBEX ... 21
3.5.1.3 Single Object Pull with IrOBEX ... 22
3.5.2 Generic Object Exchange ... 22
3.6 OBEX ... 23
3.6.1 OBEX Operations ... 23
3.6.2 OBEX Headers ... 24
3.6.3 Establishing an OBEX session ... 24
3.6.4 Reading Objects ... 26
3.6.5 Writing Objects ... 27
3.6.6 Disconnecting an OBEX session ... 27
3.7 Ultra IrWW ... 28
3.8 Protocol Implementation Guideline for IrWW Products ... 28
3.8.1 Combination of Connection and Connectionless ... 28
3.9 Tiny TP ... 29
3.10 IrLMP ... 29
3.11 IAS ... 29
3.11.1 LsapSel ... 29
3.11.2 Parameters ... 29
3.11.3 DeviceInfo ... 34
3.12 Service Hint Bit ... 35
3.13 IrLAP (minimum requirements) ... 35
3.14 Physical Layer ... 35

1 Introduction

IrWW (IrDA for Wristwatches) provides time-based data communication scheme for a wristwatch by using IrDA Communication Standards. IrWW profiles are to be largely used together with the IrDA standard specifications.

1.1 IrWW Profile Structure


Generic Access Profile
Time Synchronization Profile
Annex for legacy PC/PDA
Tiny Object Exchange Profile
T-log
PIM
Reference Document
GBO
Ultra IrWW
Compliance Tests Document
Time Synchronization Profile Compliance Tests
ANNEX: Time Synchronization profile For legacy PC/PDA Compliance Tests
Tiny Object Exchange Profile Compliance Tests

Relationship between profile, Usage model and objects

Profile Name	Time Synchronization Profile		
Usage Model	Time Synchronization Usage Model		
Protocol Stack	Ultra IrWW (Mandatory)		IrLAP+IrLMP+TinyTP+IAS+ OBEX (for Annex)
Support Object	Mandatory (M)	Option (O)	Option (O)
Time-Sync	GBO for Ultra IrWW		bWatch1.0 (for Annex)
Alarm		GBO for Ultra IrWW	bWatch1.0 (for Annex)
Timer		GBO for Ultra IrWW	bWatch1.0 (for Annex)
Stopwatch		GBO for Ultra IrWW	bWatch1.0 (for Annex)

Profile Name	Tiny Object Exchange Profile	
Usage Model	Point and Shoot Object Push Usage Model	
Protocol Stack	IrLAP+IrLMP+TinyTP+IAS+OBEX	
Support Object	Mandatory (M)	Option (O)
Time-Log		bWatch1.0
Business Card		bvCard2.1
Schedule		bvCalendar 1.0
To do		bvCalendar 1.0
Notes		bvNote1.1

1.2 Definitions and Acronyms

Inbox A generic Object Store on an IrWW Device that can hold objects of various formats, such as Business Cards, Schedules, To dos and Notes. It is typically used as a temporary holding area for objects received from other IrWW Devices. Often, the user can inspect the items in the Inbox, and file them away or delete them.

Alarm In a wristwatch, normally it means daily alarm function. Daily alarm notifies the same setting time everyday. One shot alarm notifies the setting time just once.

Timer Timer Function to count down to the presetting period of time.

Stopwatch Stopwatch Function to count up with a second resolution or under a second besides displaying current time.

Split Split Time Function to show elapsed time from the start point one by one while keeping count up.

1.3 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.
C	Conditional support. Refers to capabilities that shall be used in when certain other capabilities are also used.
X	Excluded. Refers to capabilities that may be supported by the device but shall never be used in this profile.
N/A	Not applicable in the given context. It is impossible to use this capability.

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.

1.4 References

[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

[TINYTP] Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, 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.

[ULTRA IRWW] Ultra IrWW Time Synchronization and Time Data Exchange using Ultra Protocol

[IrWW2] IrWW Time Synchronization Profile, Version 1.0, Infrared Data Association.

[IrWW3] IrWW Tiny Object Exchange Profile, Version 1.0, Infrared Data Association.

2 Usage Models

2.1 Time Synchronization Usage Model (New)

2.1.1 Scope

The scope of the information presented here is based on the ability to synchronize time with many IrDA devices.

2.1.2 User Scenario

A user can point device such as wristwatch to other devices and shoot time object. Receiver devices can synchronize time by using the received time object.

2.1.3 Interoperability

Devices will have this capability built-in. The use of specific 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 type of object data that the user may push from one device to another.

Time Synchronization Data:
Date and time that are considered to be correct and can be used as reference.
These data should be described with appropriate binary format.

2.1.4 Usability

Users will be able to synchronize time between IrDA equipped devices with simple operation (such as pressing a button). The short-range and narrow angle of IrDA-Data provides a simple form of security and a natural ease of use. Time synchronization using UI frame of IrDA-Data is the simplest way to synchronize time between many devices.

2.1.5 Configuration

No configuration is the default for time synchronization.

2.1.6 Reliability

Time object will be sent error free. Specific reliability standards will be identified in the test specifications associated with the required enabling technology.

2.1.7 Additional Information

Whether or not the device that received time object synchronizes time, it depends on application of the device.

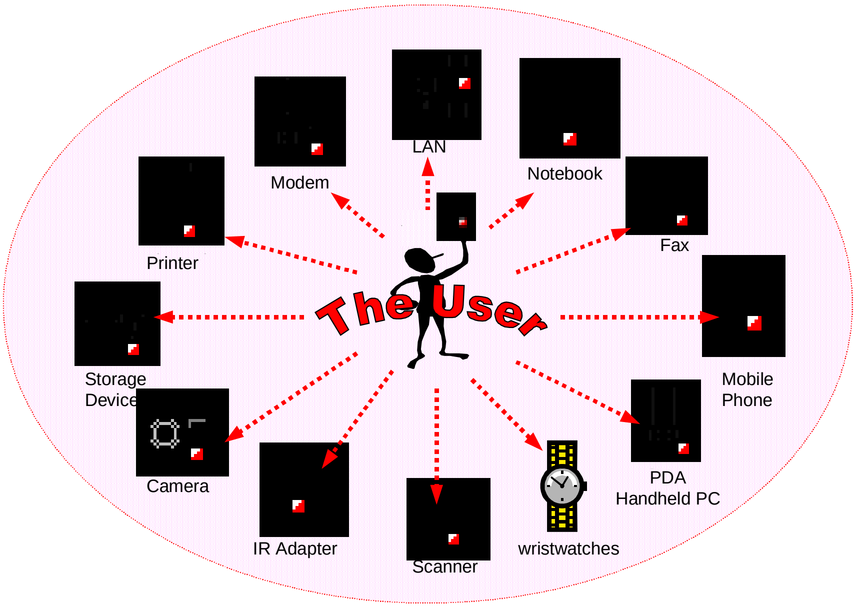
2.2 Point and Shoot Object Push

2.2.1 Scope

The scope of the information presented here is based on the ability to exchange data between two IrDA enabled devices. Our focus is on what the user will 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 model is the universal way to move data between IrDA enabled devices. The key to universal object exchange is support for standard object types such as vCard, JPEG, and text. Almost all IrDA devices will support this capability including PCs, printers, PDAs, cameras, phones, watches, pagers, storage devices, and kiosks.

2.2.2 User Scenario

Many user scenarios are covered by point and shoot object push. The picture below captures the power and simplicity of Point and Shoot Object Push.



The following scenarios are possible:

- The user can push his business card from his watch to another persons watch.
- The user can store time-log information in a multi-functional watch and transfer it to PC, PDA.

2.2.3 Interoperability

IrReady 2000 devices will have this 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.

Types of Data Exchange:

- **Generic Files**
  - Exchange of files in which Time-log informations are stored.
  - Exchange of files in which GBO based PIM objects (business card, schedule, to do list and note) are stored.

2.2.4 Usability

Users will be able to transfer an object to another device by simply 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 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.

2.2.5 Configuration

No configuration is the default for pushing objects. In some systems the user can select the location of the inbox and possibly the behavior of prompts.

2.2.6 Reliability

Objects will be sent error free. Specific reliability standards will be identified in the test specifications associated with the required enabling technology.

2.2.7 Additional Information

Objects received may be put into the appropriate data store on the device or sent to the appropriate application. For example, on a PC, received business cards could be placed directly into the user's PIM.

3 IrWW Generic Access Profile

3.1 User Requirements

3.1.1 Scope

Time Synchronization profile defines the requirements for the protocols and procedures that shall be used by applications implementing Time Synchronization usage model. The most common devices implementing this usage model include wristwatches.
Tiny Object Exchange profile defines the requirements for the protocols and procedures that shall be used by applications implementing Point and Shoot Object Push usage model. The most common devices implementing this usage model include wristwatches, PCs, and PDAs.

3.1.2 User Scenarios

The scenarios covered by this profile are:

Usage of the IrDA device to read IrWW objects from the IrDA device.
Usage of the IrDA device to write IrWW objects to the IrDA device.

IrWW objects are covered as shown in the table below

Object Type		Format	Examples
Time-Sync Object	Time-Sync Object	GBO for Ultra IrWW bWatch (for ANNEX)	Time-Adjust Object
	Time-related Object	GBO for Ultra IrWW bWatch (for ANNEX)	Alarm, Stopwatch, Timer
Tiny Objects	Time-log Object	bWatch	T-Log (Diving data, Trekking data …)
	Tiny PIM Object	bWatch	Business Card, Schedule, To Do, Note

Table 1 Time-related Objects

3.2 Profile Overview

3.2.1 Configuration and Roles

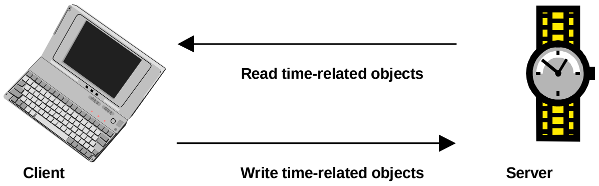


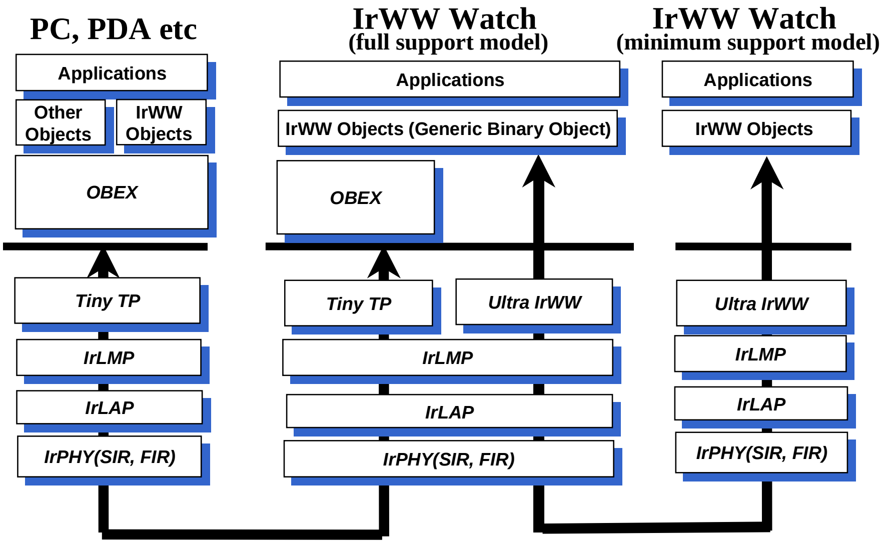
Figure 3-1 Time-related Object Transfer Example between wristwatches

The following roles are defined for this profile.

Server – The device provides an object exchange server to and from which data objects can be pushed and pulled. The Server waits passively for the client to initiate the operation.

Client – The device can push or/and pull data objects to and from the server.

3.2.2 Protocol Stack



IrDA Hardware is governed by the [IrPHY]

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

IrLMP is a multiplexing layer specified in [IrLMP]

Tiny TP provide flow control and is specified in [TINYTP]

IAS is the Information Access Service specified in [IrLMP]

Ultra and Ultra-IrWW is the Information Access Service specified in [Ultra]

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

Application Client and **Application Server** are the application entities, which provide the user interface and perform the operation of the Time-related Object Transfer profile.

3.2.3 Conformance

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner. This also applies for all optional and conditional capabilities for which support is indicated.

3.3 User Interface Aspects

3.3.1 Mode Selection (Server)

Server Mode is the state in which a server is ready to receive objects from a client or to send objects to a client.

It is mandatory that a server be in this mode whenever the physical IR port is enabled (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 must correspond to entering server mode.

3.3.2 Function Selection (Client)

The **Pull Function** initiates the reading of time-related objects from a server.
The **Push Function** initiates the writing of time-related objects to a server.

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.

3.3.3 Application Usage

When the user wants to read time-related objects from a server to a client or write time-related objects from a client to a server, the following scenario can be followed.

Client	**Server**
	The user sets the device into **Server Mode** if it is not already.
The user of the Client selects a kind of time-related object to pull or push.	
The user points the IR port of the Client device at the IR port of the Server device.	
The user selects the **Push Function or Pull Function** to read or write the selected object.	
	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.
It is recommended that the user be notified of the result of the operation.	

3.4 Application Layer

3.4.1 Overview

To achieve application level interoperability, content format is defined for time-related object transfer. Since there are limitations of CPU performance and memory space, objects should be compact and easy to handle. To cope with this requirement, a format for reducing object data size without losing any important information are introduced. In the following, basic scheme of format (Generic Binary Object : GBO) and application to time-related objects (bWATCH) are described.

3.4.2 Generic Binary Object (GBO)

Since Generic Binary Object (GBO) is based on format of the vCard, GBO can be applicable to other communication environment such as IrMC. Detail structure as well as coding mechanism and syntax of GBO can be seen in [GBO].

3.4.3 bWATCH

For describing the time-related objects such as Time, Alarm, Timer, Stopwatch or Time Log(T-LOG), the bWatch is used.
The bWatch is binary data which is described by GBO. See [GBO] for GBO and the bWatch’s binary code.
Here, the bWatch is explained using character based code for convenience' sake.
And see Profile [IrWW2] and [IrWW3] for the examples of the bWatch.

3.4.3.1 Property

The bWatch is the thing whereby a property which is signified in syntax such as the following was arranged.

The bWatch is the thing that properties are arranged. The property is the definition of an individual attribute describing bWatch object. The property is signified in syntax such as the following.

<Property Name> { **;** <Parameter> } **:** <Value>

3.4.3.2 Properties

The properties which are used in bWatch are following.

Property Name	Name	Purpose
BEGIN	Begin	This property specifies beginning of component.
END	End	This property specifies end of component.
VERSION	Version	This property specifies the identifier corresponding to the highest version number of the bWatch Specification supported by the implementation that created the bWatch object.
NOTE	Note	This property specifies non-processing information intended to provide a note the user.
X-	Non-standard Properties	This property provides a framework for defining non-standard properties.
TIME	Time	This property specifies a local time of day.
DATE	Date	This property specifies a local calendar date.
DATETIME	Date and Time	This property specifies a calendar date and time of day.
UTCOFFSET	UTC Offset	This property specifies an offset from UTC to local time.
PTIME	Period of Time	This property specifies a period of time of day.
PDATE	Period of Date	This property specifies a period of calendar date.
REFERENCE	Reference	This property specifies an information about source of transmitted time of T-Adjust object.
CATEGORY	Category	This property defines the category for an alarm object.
ACTIVE	Active	This property defines whether this function is activated in the receiver device or not.
ACTION	Action	This property defines the action to be invoked when an alarm is triggered.
STATE	State	This property defines the state to be invoked when an alarm is triggered.
DESCRIPTION	Description	This property defines a description for object.
REPEAT	Repeat	This property defines whether the timer is auto repeat timer or not.
TITLE	Title	This property specifies a title of T-log object.
SUBTITLE	Subtitle	This property specifies a subtitle of T-log object.
DATA	Data	This property specifies the definition of values in RECORD property.
RECORD	Record	This property specifies several measurement data of T-log object.

Table 2 The properties which are used in bWatch

3.4.3.3 Property Parameters

The property parameters which are used in bWatch are following.

Property	Parameter	Meaning
BEGIN	-	-
END	-	-
VERSION	-	-
NOTE	**ENCODING=XXX**	This parameter specifies an encoding. See [GBO] for XXX.
	CHARSET=XXX	This parameter specifies a character set. See [GBO] for XXX.
	LANGUAGE=XXX	This parameter specifies a language. See [GBO] for XXX.
X-	User defined.	-
TIME	**TYPE=VTHM**	This parameter specifies that the value type is hour/minute.
	TYPE=VTHMS	This parameter specifies that the value type is hour /minute/second.
	TYPE=VTHMSL	This parameter specifies that the value type is hour/minute/second/under-second.
	TYPE=VTRHM	This parameter specifies that the value type is hour/minute.
	TYPE=VTRMS	This parameter specifies that the value type is minute/second.
	TYPE=VTRL	This parameter specifies that the value type is under-second.
DATE	**TYPE=VDYMD**	This parameter specifies that the value type is year/month/day.
	TYPE=VDYMDW	This parameter specifies that the value type is year/month/day/week/daylight saving time.
DATETIME	**TYPE=VTDL**	This parameter specifies that the value type is local year/month/day/hour/minute/second.
	TYPE=VTDU	This parameter specifies that the value type is UTC year/month/day/hour/minute/second.
UTCOFFSET	-	-
PTIME	Same as TIME property.	Same as TIME property.
PDATE	Same as DATE property.	Same as DATE property.
REFERENCE	-	-
CATEGORY	-	-
ACTIVE	-	-
ACTION	**TYPE=SOUND**	This parameter specifies an alarm that causes a sound to be played to alert the user.
	TYPE=VIBRATION	This parameter specifies an alarm that causes a vibration to be played to alert the user.
	TYPE=DISPLAY	This parameter specifies an alarm that causes a message to be displayed to the user.
	Same as NOTE property.	Same as NOTE property.
STATE	-	-
DESCRIPTION	Same as NOTE property.	Same as NOTE property.
REPEAT	-	-
TITLE	Same as NOTE property.	Same as NOTE property.

SUBTITLE	Same as NOTE property.	Same as NOTE property.
DATA	**TYPE=NUMERIC**	This parameter specifies an item of record is a numeric data.
	TYPE=STRING	This parameter specifies an item of record is a string data.
	ITEMNAME=XXX	This parameter specifies an item name of record is XXX. XXX is string.
	UNIT= XXX	This parameter specifies an item's unit of record is XXX. XXX is string.
	Same as NOTE property.	Same as NOTE property.
RECORD	-	-

Note

1. The VALUE= of GBO common parameter can be used at all parameter field.

Table 3 The property parameters which are used in bWatch

3.4.3.4 Property Values

The property values which are used in bWatch are following.

Property Name	Value	Meaning
BEGIN	**BWATCH**	This value is bwatch component.
	T-SYNC	This value is t-sync component.
	ALARM	This value is alarm component.
	TIMER	This value is timer component.
	STOPWATCH	This value is stopwatch component.
	T-LOG	This value is t-log component.
END	Same as BEGIN property.	Same as BEGIN property.
VERSION	Numeric	This value is version number. This version is 1.0.
NOTE	String	This value is a note.
X-	User defined.	-
TIME	HH;MM	When the parameter is TYPE=VTHM. This value is hour/minute (from 00;00 to 99;59).
	HH;MM;SS	When the parameter is TYPE=VTHMS. This value is hour /minute/second (from 00;00;00 to 99;59;59).
	HH;MM;SS;LLLLLL	When the parameter is TYPE=VTHMSL. This value is hour/minute/second/under-second (from 00;00;00;000000 to 99;59;59;999999).
	HHHH;MM	When the parameter is TYPE=VTRHM. This value is hour/minute (from 0000;00 to 9999;59).
	MMMM;SS	When the parameter is TYPE=VTRMS. This value is minute/second (from 0000;00 to 9999;59).
	LLLLLL	When the parameter is TYPE=VTRL. This value is under-second (from 000000 to 999999).
DATE	YYYY;MM;DD	When the parameter is TYPE=VDYMD. This value is year/month/day. (from 0000;01;01 to 9999;12;31)
	YYYY;MM;DD;WW;DST	When the parameter is TYPE=VDYMDW. This value is year/month/day/week/daylight saving time.

DATETIME	YYYY;MM;DD;HH;MM;SS	When the parameter is TYPE=VTDL. This value is local year/month/day/hour/minute/second. (from 0000;01;01;00;00;00 to 9999;12;31;23;59;59)
	YYYY;MM;DD;HH;MM;SS	When the parameter is TYPE=VTDU. This value is UTC year/month/day/hour/minute/second. (from 0000;01;01;00;00;00 to 9999;12;31;23;59;59)
UTCOFFSET	(S)HH;MM	This value is sign hour/minute local time offset from UTC.
PTIME	Same as TIME property.	Same as TIME property.
PDATE	Same as DATE property.	Same as DATE property.
REFERENCE	RTIME (Reference Time)	
	YES	The time of T-sync is a time that made reference.
	NO	The time of T-sync is not a time that made reference.
	RTYPE (Reference Type)	
	ATOMIC	The time of T-sync is a time that made reference to an atomic clock.
	GPS	The time of T-sync is a time that made reference to a Global Positioning System.
	RADIO	The time of T-sync is a time that made reference to a radio controlled clock.
	TCXO	The time of T-sync is generated by a TCXO.
	QUARTZ	The time of T-sync is generated by a quartz crystal oscillator.
	UNDEFINED	The source of time of T-sync is undefined.
CATEGORY	**WAKE**	The time is time when you wake up.
	CALL	The time is time when you call.
	MEETING	The time is time when you hold a meeting.
	APPOINT	The time is time when you made an appointment.
	OTHER	The time is time when you do other matter.
ACTIVE	**ON**	The alarm function is activated.
	OFF	The alarm function is not activated.
ACTION	String	User defined String data.
STATE	**ON**	The switch is turned on when an alarm is triggered.
	OFF	The switch is turned off when an alarm is triggered.
DESCRIPTION	String	This value is a description for object.
REPEAT	**ON**	The timer automatically restarts the countdown from countdown time.
	OFF	The timer stops the countdown when zero is reached.
TITLE	String	This value is a title of T-log object.
SUBTITLE	String	This value is a subtitle of T-log object.
DATA	String	This value is same as PICTURE format of COBOL language.
RECORD	Time-related value, Numeric, String	This value form is value1; value2; value3 … Each value obeys the properties of field block.

Table 4 The property values which are used in bWatch

3.4.3.5 Property Order

The bWatch's properties are arranged in order of the following.

T-Sync / Alarm / Timer / Stopwatch Object is the following.

```
BEGIN: BWATCH
   VERSION:1.0
   { <NOTE property> | <X- property> }
   { <T-Sync Component> | <Alarm Component> | <Timer Component> | <Steopwatch
   Component> | <T-log Component> }
END: BWATCH
```

<T-Sync / Alarm / Timer / Stopwatch Component> is the following.

```
BEGIN: T-SYNC (, ALARM, TIMER or STOPWATCH)
   <Properties>
END: T-SYNC (, ALARM, TIMER or STOPWATCH)
```

<T-log Component> is the following.

```
BEGIN: T-LOG
   <Properties>
   BEGIN: FIELD
      <Properties>
   END: FIELD
   BEGIN: MATRIX
      <Properties>
   END: MATRIX
END: T-LOG
```

3.4.3.6 Property use restrictions

The bWatch's Property use restrictions are following.

Property	bWatch component			
		T-Sync component	Alarm component	Timer component
VERSION	M (1)	-	-	-
NOTE	O (n)	O (n)	O (n)	O (n)
X-	O (n)	O (n)	O (n)	O (n)
TIME	-	M (1) Only "TYPE=VTHMSL" **The current time.**	M (1) Only "TYPE=VTHM" **The alarm time.**	-
DATE	-	M (1) Only "TYPE=VDYMDW" **The current time.**	M (1) Only "TYPE=VDYMDW" **The alarm time.**	-
DATETIME	-	-	-	-
UTCOFFSET	-	M (1) **The current time.**	O (1) **The alarm time.**	-
PTIME	-	M (1) Only "TYPE=VTRL" **The delay time.**	-	M (1) Only "TYPE=VTHMS" **The repeat time.**
PDATE	-	-	-	O (1) **The repeat time.**
REFERENCE	-	M (1)	-	-
CATEGORY	-	-	O (1) Default "OTHER".	-
ACTIVE	-	-	O (1) Default "ON".	-
ACTION	-	-	O (n)	O (n)
STATE	-	-	O (1)	O (1)
DESCRIPTION	-	-	O (1)	O (1)
REPEAT	-	-	-	M (1)
TITLE	-	-	-	-
SUBTITLE	-	-	-	-
DATA	-	-	-	-
RECORD	-	-	-	-

Property	bWatch component			
	Stopwatch component	T-log component		
			Field	Matrix
VERSION	-	-	-	-
NOTE	O (n)	O (n)	O (n)	O (n)
X-	O (n)	O (n)	O (n)	O (n)
TIME	O (1) **The time when the split time was recorded.**	O (1) **The time when the log was recorded.**	O (n)	-
DATE	O (1) **The time when the split time was recorded.**	O (1) **The time when the log was recorded.**	O (n)	-
DATETIME	O (1) **The time when the split time was recorded.**	O (1) **The time when the log was recorded.**	O (n)	-
UTCOFFSET	O (1) **The time when the split time was recorded.**	O (1) **The time when the log was recorded.**	O (n)	-
PTIME	M (1) Only "TYPE=VTHMSL" **The split time.**	-	O (n)	-
PDATE	O (1) **The split time.**	-	O (n)	-
REFERENCE	-	-	-	-
CATEGORY	-	-	-	-
ACTIVE	-	-	-	-
ACTION	-	-	-	-
STATE	-	-	-	-
DESCRIPTION	O (1)	O (1)	-	-
REPEAT	-	-	-	-
TITLE	-	O (1)	-	-
SUBTITLE	-	O (1)	-	-
DATA	-	-	O (n)	-
RECORD	-	-	-	O (n)

Note

(1) means to be able to use only one property.

(n) means to be able to use property many times.

Table 5 The bWatch's Property use restrictions

3.4.3.7 GBO property order for Ultra IrWW

If the time-related objects are sent using Ultra protocol, GBO properties for Ultra IrWW are used. GBO properties for Ultra IrWW use the bWatch's properties and are arranged in order of the following. Mandatory property order is fixed.

T-Sync Object

OID for T-Adjust
DATE; TYPE=VDYMDW: Value
TIME; TYPE=VTHMSL: Value

UTCOFFSET: Value
PTIME; TYPE=VTRL: Value
REFERENCE: Value
< Option properties>

Alarm Object

OID for Alarm
DATE; TYPE=VDYMDW: Value
TIME; TYPE=VTHM: Value
< Option properties>

Timer Object

OID for Timer
PTIME; TYPE=VTHMS: Value
REPEAT: Value
< Option properties>

Stopwatch Object

OID for Stopwatch
PTIME; TYPE=VTHMSL: Value
< Option properties>

3.4.4 bWatch Information log

The Information Log is object that contain general information about specific Object Store. The Information Log must be supported by all devices that support read-all/write-all. This specification is same as IrMC's Information Log. See [IrMC].

The bWatch's Information Log is binary data which is described by GBO. See [GBO] for GBO and the bWatch's Information Log binary code. Here, the bWatch's Information Log is explained using character based code for convenience' sake.
And see Profile [IrWW2] and [IrWW3] for the examples of the information log.

3.4.4.1 Properties

The properties which are used in bWatch's Information Log are following.

Property	Parameter	Value	Purpose	State
Total-Records	-	Numeric	The total number of Objects (or records) within the Object Store.	O
Maximum-Records	**-**	Numeric	The maximum number of objects (or records) that can be stored in the Object Store. If there is no particular limit, then an asterisk "*" will be used in place of a number.	M
IEL **(Information Exchange Level)**	**-**	Numeric	The level of Information Exchange supported by the Object Store. The value is the following. Single Push Support: 1 Read-all/Write-all Support: 2	O
X-IRWW-Fields	**-**	**-**	This property provides an overview of all the fields supported. See next Section.	M

Table 6 The properties which are used in bWatch's Information Log

3.4.4.2 X-IRWW-Fields

The X-IRWW-FIELDS extension property will represent all the supported field properties, property parameters and types utilized within the IrWW Object Store.

This specification is same as IrMC's Information Log. See [IrMC].

3.4.4.3 Property Order

The Information Log Object is the following.

{
<**Total Records** property> |
<**Maximum Records** property> |
<**IEL** property> |
<X-irww fields object> |
}

<X-irww fields object> is the following.

X-IRWW-Fields:
<Begin>
{ <x- irww property> }
<End>

3.5 Application Procedure

3.5.1 Single object push

In this case, a client can send a single Object to a Server. When received, the Server stores the object in a generic inbox with the original name of the object. In this application, flow of the object is uni-directional. Therefore, PUSH operation is generally carried out. Either connectionless or connection-oriented data transfer, such as Ultra IrWW or IrOBEX is used.

In this case, a client can send a single Object to a Server. When received, the Server stores the object in a generic inbox with the original name of the object. In this application, flow of the object is uni-directional. Therefore, PUSH operation is generally carried out. Either connectionless or connection-oriented data transfer, such as Ultra IrWW or IrOBEX is used.

3.5.1.1 Single Object Push with Ultra IrWW

Figure 3-16 shows an example of time sequence of Ultra IrWW communication. The advantage of the Ultra IrWW is that an IrWW object generated in the DEVICE1 UI level can be sent to the DEVICE 2 UI level almost directly. This characteristic is effective for transmitting time-based information with little time-delay.

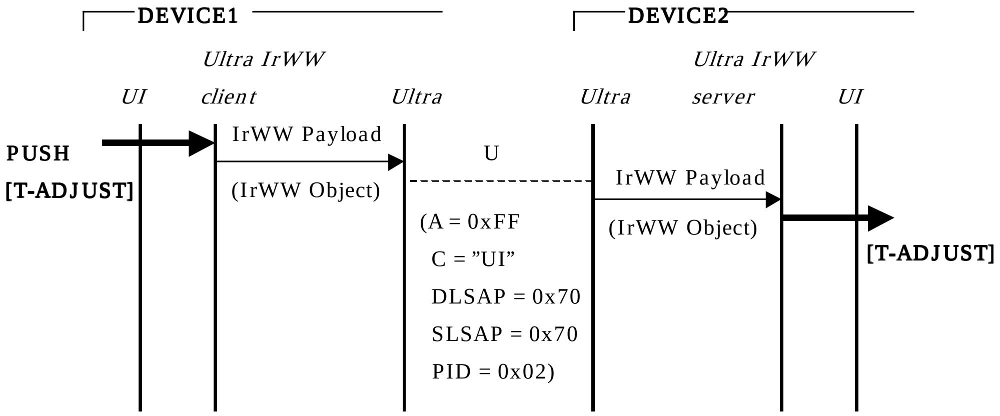


Figure 3-16 Example of Ultra IrWW time sequence

3.5.1.2 Single Object Push with IrOBEX

Both Push Client and Push Server are built on top of the OBEX application framework. A Push Client uses OBEX_PUT to push single object to the inbox of a Push Server. The Push Client connects to the Push Server's inbox and pushes single object to it. The Push Client only knows that the object are successfully received. It does not know the layout or construction of the Push Server's inbox.

A Push Server's inbox must holds objects. But, it is possible to automatically dispatch objects from inbox to store. For example, if single Alarm is received it can be dispatched to the Alarm.

Table 3-11 shows an example of the application procedure required by the Push Client to push object to a Push Server.

Table 3-11 Application layer procedure for Push object

Push Client	Details
OBEX CONNECT.	
Push the object using OBEX PUT.	
OBEX DISCONNECT.	

In Time-Related object push , following file names which distinguish from Time-Related Object transfer are used.

Table 3-12 Object name assignment rule of Time-Related Object Push

Object	**File name**
Alarm	Xxx.wal
Timer	Xxx.wtm
Stopwatch	Xxx.wsw

3.5.1.3 Single Object Pull with IrOBEX

Both Pull Client and Pull Server are built on top of the OBEX application framework. A client uses OBEX_GET to pull single object with the IrWW specified object name from a server. The client connects to the server and pulls single object from it.

Table 3-13 shows an example of the application procedure required by the client to pull object from a server.

Table 3-13 Application layer procedure for Pull object

Push Client	**Details**
OBEX CONNECT.	
Push the object using OBEX GET.	
OBEX DISCONNECT.	

Table 3-14 Object name assignment rule of Time-Related Object Push

Object	**File name**
Time-Sync	Xxx.wsy

As shown in the table3-14, only time-adjust object can be pulled from the server.

3.5.2 Generic Object Exchange

This application provides a read-all/write-all functionality. Servers with this model support create or act upon an object stream containing all the objects in the devices. Note that inbox objects are not taken into account when building this stream of objects. This model mainly applies communications between watches and PC or PDAs. Objects sent to the PC or the PDA can be edited and then sent back to the watch and updated. Thus, there appears bi-directional data flow between the client and the server devices. Both a Client and Server are built on top of the OBEX application framework. A client uses OBEX PUT to write-all objects to the store of a server. And the client uses OBEX GET to read-all objects from the store of the server. Transferring files requires a single PUT or GET operation per file. The client connects to the Server's storage and transfers all objects. Note that inbox isn’t used in these objects.

Information Log contains general information about specific Object Stores. Before the client reads or writes generic objects from or to the server, It becomes help of processing of information by reading Information Log.

Table 3-15 shows an example of the application procedure required by the client to read all objects from the Server.

Table 7-15 Application procedure for Read-all objects

Client	**Details**
OBEX CONNECT.	
Read the Information Log using OBEX GET.	In accordance with read Information Log, memory is prepared in which the data are stored.
Read all objects using OBEX GET.	

OBEX DISCONNECT.	

It is not necessary for a client to always use Information Log.

Table 3-16 shows an example of the application procedure required by the client to write all objects to the server.

Table 3-16 Application procedure for push objects

Client	**Details**
OBEX CONNECT.	
Read the Information Log using OBEX GET.	
Write all objects using OBEX PUT.	In accordance with read Information Log, only necessary data are sent.
OBEX DISCONNECT.	

It is not necessary for a client to always use Information Log.

In Generic object transfer, file names shown in table 3-17 which distinguish from single object push are used.

Table 3-17 File name assignment rule of Generic Object Transfer

Object	**Object name**	**Contents**
Time-Related Objects	/watch/alarm.wal	Alarm Data
	/watch/timer.wtm	Timer Data
	/watch/stop.wsw	Stop Watch Data
Time-Log Object	/watch/t-log.wlg	
PIM Objects	/watch/pim/ob.bcf /watch/pim/bc.bcf /watch/pim/sd.bet /watch/pim/td.btd /watch/pim/nt.bnt	Owner Business Card Bussiness Card Schedule ToDo Note , Memorundom

3.6 OBEX

3.6.1 OBEX Operations

Table 3-18 shows the OBEX operations, which are used in the Single Object Push or Generic Object Transfer profile.

Table 3-18 OBEX Operations

Operation no.	OBEX Operation	Client	Server
1	Connect	M	M
2	Disconnect	O	M
3	Put	M	M
4	Get	M	M
5	Abort	M	M
6	Set Path*	N/A	N/A

* IrWW Object Name must be used absolute object path name.

3.6.2 OBEX Headers

Table 3-19 shows the OBEX headers used in the Generic Object Transfer profile.

Table 3-19 OBEX Headers

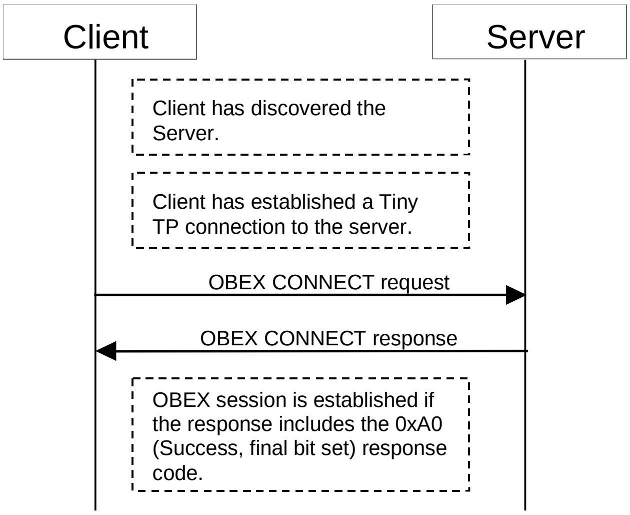
Header no.	OBEX Headers	Read-all/Write-all Client	Read-all/Write-all Server
1	Count	O	O
2	Name	M	M
3	Type	O	O
4	Length	O	O
5	Time	O	O
6	Description	O	O
7	Target	X	X
8	HTTP	N/A	N/A
9	Body	M	M
10	End of Body	M	M
11	Who	X	X
12	Connection ID	X	X
13	Authenticate Challenge	X	X
14	Authenticate Response	X	X
15	Application Parameters	X	X
16	Object Class	X	X

3.6.3 Establishing an OBEX session

Setting up an OBEX session for Tiny TP Generic Object Transfer involves three steps.

1. Client discovers the server device (see section 3.6.1).
2. Client establishes a Tiny TP connection to the server device (see section 3.6.2).
3. Client performs an OBEX connect operation to the server.

The figure below shows how the OBEX session is established.


Client
Server
Client has discovered the Server.
Client has established a Tiny TP connection to the server.
OBEX CONNECT request
OBEX CONNECT response
OBEX session is established if the response includes the 0xA0 (Success, final bit set) response code.

The OBEX connect request must contain the following fields.

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	0x07	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**	**Status**	**Explanation**
Field	Response code for CONNECT request	0x0A	M	0xA0 for success
Field	Connect Response Packet Length	0x07	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-

Connection procedure can be neglected. But for IrWW communication, it is necessary to negotiate PDU size. Therefore, connection procedure must be carried out before put requests.

3.6.4 Reading Objects

All Objects are pulled from the server using the OBEX GET operation. Pulling object can take one or more OBEX packets. The initial GET packet must include the following fields and headers.

Field/ Header	**Name**	**Value**	**Status**	**Explanation**
Field	Opcode for GET	0x03 or 0x83	M	-
Field	Packet Length	Varies	M	-
Header	Name	Varies	M	The header value is the name of all Objects

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

Field/ Header	**Name**	**Value**	**Status**	**Explanation**
Field	Response code for GET	0x90 or 0xAO	M	0x90 for continue, 0xA0 for success.
Field	Packet Length	Varies	M	-
Header	Length	Varies	O	Length of the object.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

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

3.6.5 Writing Objects

All Objects are pushed to the server using the OBEX PUT operation. Pushing a object can take one or more OBEX packets. The initial PUT packet must include the following fields and headers.

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for PUT	0x02 or 0x82	M	-
Field	Packet Length	Varies	M	-
Header	Name	Varies	M	The header value is the name of all objects.
Header	Length	Varies	O	Length of the object.
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	Status	Explanation
Field	Response code for PUT	0x90, 0xAO or 0xCD	M	0x90 for continue, 0xA0 for success, 0xCD if the object is too large.
Field	Packet Length	Varies	M	-

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

Writing Empty Object
If the object consists of only GBO Object Stream Identifier (OBJECT MAGIC CODE), the write-all action is regarded as to make Object Store empty. Any previous information is cleared.

3.6.6 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. Read-all/Write-all Server’s must be able to handle both methods of disconnect.

The OBEX DISCONNECT request must contain the following fields.

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for DISCONNECT	0x81	M	-
Field	Connect Packet Length	Varies	M	-

Other headers such as **Description**, which can be optionally used, are found in [OBEX].

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

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

3.7 Ultra IrWW

Ultra IrWW is mandatory protocol for Time Synchronization profile. Details of the Ultra IrWW are shown in the [ULTRA IRWW].

3.8 Protocol Implementation Guideline for IrWW Products

3.8.1 Combination of Connection and Connectionless

If a client device supports both connections-less and connection-oriented profiles for same application such as time-related object push, the device must adhere to the following guidelines.

1. Discovery-1: If a correct discovery response is not received from the server after running the discovery process a third time, the client shall assume that there is no connection oriented server present, wait at least 500 ms and send the object connection less (Ultra). According to the media access rules there shall be delay before the sending of the ultra frame, this will mean that the total delay from the last XID frame to the first Ultra frame will be about 1 sec.

2. Discovery-2: If the server returns a proper discovery response, the client will try to connect to the OBEX server. If no OBEX server can be found, the object is not sent using connection less either. This is because the client assumes that server does not support OBEX connection less if it also does not supported connection oriented. A server implementing OBEX and supporting connection oriented services is required to support OBEX connection-oriented.

3. Connection-1: If a connection is established to the server, the IAS is then queried for the OBEX server, if an OBEX server is not present, the client disconnects from the server. If an OBEX server is present a connection is established and the object is transferred.

4. Connection-2: It is assumed with an error when trying to PUT or GET an IrWW object but not found.

Example: The client receives a proper discovery response to its discovery command from the server side. It then sends a SNRM command frame to the server. If the server decides to accept, it sends an UA frame as response to the SNRM command. The client will then connect to the IAS LSAP of the server, searching for a suiting IAS entry. If such an entry is not found it means that the server does not support this service. The client should then disconnect from the IAS LSAP, followed by disconnect from the physical link.

3.9 Tiny TP

Support flow control and byte data stream mode only. OBEX does not use connection and disconnection payloads.

3.10 IrLMP

Number of LSAPs：

Connection type 3 (IAS Server，IAS Client，Application)

Connection less type 1 (LSAP 0x70 is used by Ultra)

IAS Server：Support GetValueByClass function

IAS Client：Support GetValueByClass function

3.11 IAS

The information needed for accessing the IrDA IrWW applications is included in the IAS class WATCH. The following table defines the attributes associated with this class. The information about IAS entries and access is described in [IrLMP].

Note that IAS should be the primary means of identifying the services supported by a device. There is no need for OBEX's "who" or "target" headers when dealing with these objects.

Class OBEX

Attributes		
	IrDA:TinyTP:LsapSel	Integer (0x01)

Class IrDA:WATCH

Attributes		
	Parameters	Octet Sequence (0x02)
	DeviceInfo	Octet Sequence (0x02)

3.11.1LsapSel

A link service access point selector is the address of an application. In the IrLMP LM-MUX, all IrWW services are accessed through the OBEX LSAP selector whose value is defined in the IAS class OBEX. The information about this IAS class is described in [OBEX].

3.11.2Parameters

The Parameters attribute uniquely identifies the IrWW services provided by a device. All IrWW service information is packed into this one attribute to allow a single IAS GetValueByClass query.

Parameters attribute is an octet sequence, which consists of one or more 3-tuples with the following format:

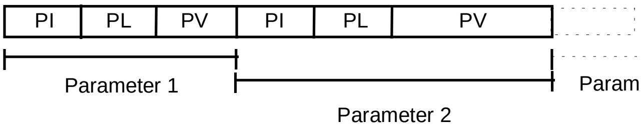


Figure 2 The structure of the parameters attribute

The fields in the 3-tuples are

Field	**Value Type**	**Description**
PI – Parameter Identifier	UINT8	A unique parameter name. If bit 7 is set, the parameter is regarded as critical. Critical parameters are used to identify special services that only work properly with peers.
PL – Parameter Length	UINT8	The length of the PV field in bytes.
PV – Parameter Value	UINT8 sequence	Value, whose meaning depends on the PI.

In Table 8, the Parameters of IrWW Object are listed.

PI	PI name	PL	PV data type	PV Description	Default Value	Status
0x00	T-Adjust Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Single Pull Support		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x01	Alarm Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Single Pull Support		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x02	Timer Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		

			bit 1	Single Pull Support		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x03	Stopwatch Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Single Pull Support		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x04	T-log Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Single Pull Support		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		

				Minor version number		
				0x00 is undefined version		
0x05	Business Card Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Undefined		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x06	Schedule Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Undefined		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x07	To Do Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Undefined		

			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		
0x08	Notes Support	2		The first octet =	0	Necessary if this mode is supported
				Support Level		
				(bit mask)		
			bit 0	Single Push Support		
			bit 1	Undefined		
			bit 2	Write-all Support		
			bit 3	Read-all Support		
				The second octet :	(1,0)	
				Upper side nibble(4bit)		
				Major version number		
				Lower side nibble(4bit)		
				Minor version number		
				0x00 is undefined version		

Table 8 The Parameters of IrWW Object

If some objects are not supported in some devices, corresponding parameters can be omitted.

3.11.3DeviceInfo

PI	PI name	PL	PV data type	PV Description	Default Value	Status
0x00	Manufacturer	n	Strings	Manufacturer Name	None	M
0x01	Model	n	Strings	Product Model Name	None	M
0x02	OEM	n	Strings	OEM Name	None	O
0x03	Firmware Version	n	Strings		None	O
0x04	Firmware	n	Strings		None	O

	Date					
0x05	Software Version	n	Strings		None	O
0x06	Software Date	n	Strings		None	O
0x07	Hardware Version	n	Strings		None	O
0x08	Hardware Date	n	Strings		None	O
0x09	Serial Number	n	Strings		None	O

Total packet length of the DeviceInfo must be limited within 40 bytes.

3.12 Service Hint Bit

The Watch service hint bit (bit XX) of the IrLMP service hints is used in the device discovery to inform about the IrWW application capabilities of the device. The information about service hint bit is described in [IrLMP].

It should be noted that the Watch bit does not indicate the type of the device in question. It only points out that the device supports some of the IrWW services, for example T-Sync or Alarm. Consequently, all PCs , PDAs, and other non-watch devices are also expected to indicate their IrWW capability with the Watch bit. IrWW device as which connection-oriented data transfer is used, should set the OBEX IrLMP service hint bit (value of 0x20 in the second hint byte.).

3.13 IrLAP (minimum requirements)

Discovery
Number of slot：1
Negotiation parameters
Band Rate：9600bps
Data size：64byte
Window size：1
Mode: Primary and Secondary should be implemented
Flow control: executed in the TinyTP
Disconnect warning: necessary (3sec，Disconnection: 5sec)
Data Transmission timing：same as IrDA DATA

3.14 Physical Layer

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

第2部

時刻同期プロファイル

Infrared Data Association

IrWW (Infrared Wrist Watch) Time Synchronization Profile

Version 1.0

December 10, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Tomozumi Saruwatari saruwatari@sii.co.jp
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)

Significant Contributors:

History:
Version 0.5: Infrared Data Association IrWW IrDA for Wrist Watches for Directional Vote
Version 0.8: IrWW Time-Adjust Object Push Profile for Draft Vote.
Version 0.8: IrWW Time-Adjust Object Pull Profile for Draft Vote.
Version 0.8: Connectionless-based profile: IrWW Time-related Object Push Profile for Draft Vote.
Version 0.8: Connection-based profile: IrWW Time-related Object Push Profile for Draft Vote.
Version 0.8: IrWW Time-related Object Exchange Profile for Draft Vote.
Version 1.0: IrWW Time Synchronization Profile for Final Vote.

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 INTRODUCTION......1

1.1 IrWW Profile Structure......1
1.2 Symbols and Conventions......1
1.3 References......2

2 TIME SYNCHRONIZATION USAGE MODEL......3

2.1 Scope......3
2.2 User Scenario......3
2.3 Interoperability......3
2.4 Usability......3
2.5 Configuration......3
2.6 Reliability......3
2.7 Additional Information......3

3 TIME SYNCHRONIZATION PROFILE......4

3.1 User Requirements......4
3.1.1 Scope......4
3.1.2 User Scenarios......4
3.2 Profile Overview......4
3.2.1 Configuration and Roles......4
3.2.2 Protocol Stack......5
3.2.3 Conformance......5
3.3 User Interface Aspects......6
3.3.1 Mode Selection(Server)......6
3.3.2 Function Selection(Client)......6
3.3.3 Application Usage......6
3.4 Application Layer......7
3.4.1 Feature Overview - Basic Concept of Time Synchronization -......7
3.4.2 Content Types......8
3.4.3 Time-Sync Object......8
3.4.4 Alarm Object......8
3.4.5 Timer Object......9
3.4.6 Stopwatch Object......10
3.5 Ultra IrWW......11
3.6 Ultra Protocol......11
3.7 IrLMP......11
3.8 IrLAP......12
3.9 Physical Layer......12

4 ANNEX: TIME SYNCHRONIZATION PROFILE FOR LEGACY PC/PDA......13

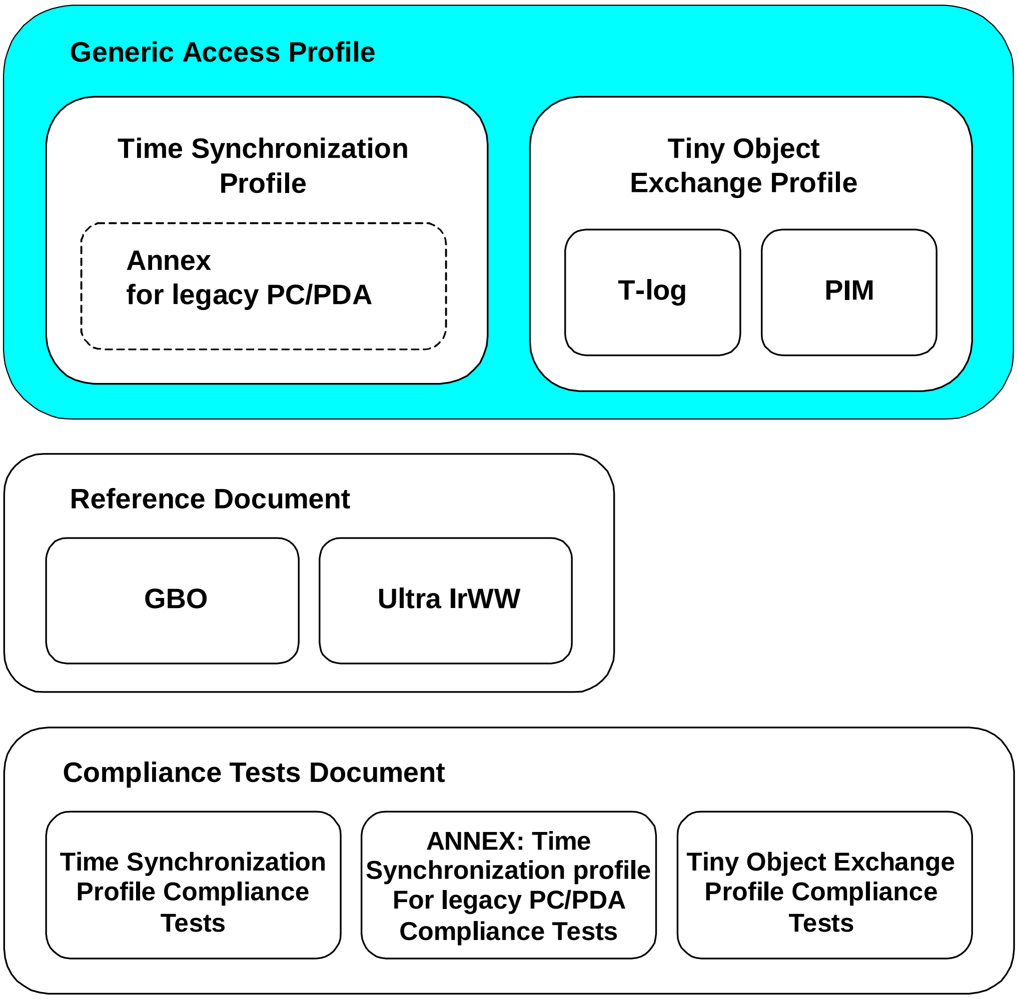
4.1 User Requirements......13
4.1.1 Scope......13
4.1.2 User Scenarios......13
4.2 Profile Overview......13
4.2.1 Configuration and Roles......13
4.2.2 Protocol Stack......13
4.2.3 Conformance......14
4.3 User Interface Aspects......15
4.3.1 Mode Selection (Server)......15
4.3.2 Function Selection (Client)......15
4.3.3 Application Usage......15
4.4 Application Layer......16

4.4.1 Feature Overview 16
4.4.2 Content Types 16
4.4.2.1 Time-Sync Object 16
4.4.2.2 Alarm Object 17
4.4.2.3 Alarm Information Log Object 18
4.4.2.4 Timer Object 19
4.4.2.5 Timer Informaiton Log Object 20
4.4.2.6 Stopwatch object 21
4.4.2.7 Stopwatch Information Log Object 22
4.4.3 Application Procedure 23
4.4.3.1 Single Object Pull with IrOBEX 23
4.4.3.2 Single Object Push with IrOBEX 23
4.4.3.3 Read-all/Write-all Objects with IrOBEX 24
4.5 OBEX 26
4.5.1 OBEX Operations 26
4.5.2 OBEX Headers 27
4.5.3 Establishing an OBEX session 27
4.5.4 Reading Objects 27
4.5.5 Writing Objects 27
4.5.6 Disconnecting an OBEX session 27
4.6 Tiny TP 27
4.7 IrLMP 27
4.8 IAS 27
4.8.1 Parameters 27
4.8.2 DeviceInfo 31
4.9 Service Hint Bit 31
4.10 IrLAP 31
4.11 Physical Layer 31

1 Introduction

IrWW (IrDA for Wristwatches) provides Time-based data communication scheme for a wristwatch by using IrDA Communication Standards. This profile is to be largely used together with the IrDA standard specifications.

1.1 IrWW Profile Structure


Generic Access Profile
Time Synchronization Profile
Annex for legacy PC/PDA
Tiny Object Exchange Profile
T-log
PIM
Reference Document
GBO
Ultra IrWW
Compliance Tests Document
Time Synchronization Profile Compliance Tests
ANNEX: Time Synchronization profile For legacy PC/PDA Compliance Tests
Tiny Object Exchange Profile Compliance Tests

1.2 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.
C	Conditional support. Refers to capabilities that shall be used in when certain other capabilities are also used.

X	Excluded. Refers to capabilities that may be supported by the device but shall never be used in this profile.
N/A	Not applicable in the given context. It is impossible to use this capability.

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.

1.3 References

[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

[Ultra] Guidelines for Ultra Protocols, Version 1.0, Infrared Data Association.

[IrWW] IrWW Generic Access Profile, Version 0.8, Infrared Data Association.

[GBO] Generic Binary Object, Version 0.8, Infrared Data Association

[Ultra IrWW] Ultra Protocols for IrWW, Version 0.8, Infrared Data Association

2 Time Synchronization Usage Model

2.1 Scope

The scope of the information presented here is based on the ability to synchronize time with many IrDA devices.

2.2 User Scenario

A user can point device such as wristwatch to other devices and shoot time object. Receiver devices can synchronize time by using the received time object.

2.3 Interoperability

Devices will have this capability built-in. The use of specific 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 type of object data that the user may push from one device to another.

Time Synchronization Data:
Date and time that are considered to be correct and can be used as reference.
These data should be described with appropriate binary format.

2.4 Usability

Users will be able to synchronize time between IrDA equipped devices with simple operation (such as pressing a button). The short-range and narrow angle of IrDA-Data provides a simple form of security and a natural ease of use. Time synchronization using UI frame of IrDA-Data is the simplest way to synchronize time between many devices.

2.5 Configuration

No configuration is the default for time synchronization.

2.6 Reliability

Time object will be sent error free. Specific reliability standards will be identified in the test specifications associated with the required enabling technology.

2.7 Additional Information

Whether or not the device that received time object synchronizes time, it depends on application of the device.

3 Time Synchronization Profile

3.1 User Requirements

3.1.1 Scope

The Time Synchronization profile defines the requirements for the protocols and procedures that shall be used by applications implementing the Time Synchronization usage model. The most common devices implementing this usage model include wristwatches, clocks, PCs and PDAs.

3.1.2 User Scenarios

The scenarios covered by this profile are:

- Usage of an IrDA device to synchronize the time with another IrDA device by using Time-Sync object.
- Usage of an IrDA device to synchronize the time-related data with another IrDA device by using Time-Related object.

Object Type	Format	Examples of Synchronization
Time-Sync Object (CL)	GBO for Ultra IrWW format	Time
Time-Related Object (CL)	GBO for Ultra IrWW format	Alarm, Timer, Stopwatch

Note: (CL) is Connection-Less Protocol

Table 1 Objects handled in Time Synchronization Profile

3.2 Profile Overview

3.2.1 Configuration and Roles

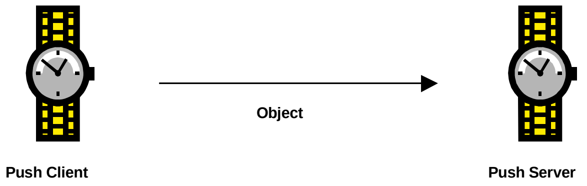


Figure 1 Synchronization Example between two wristwatches

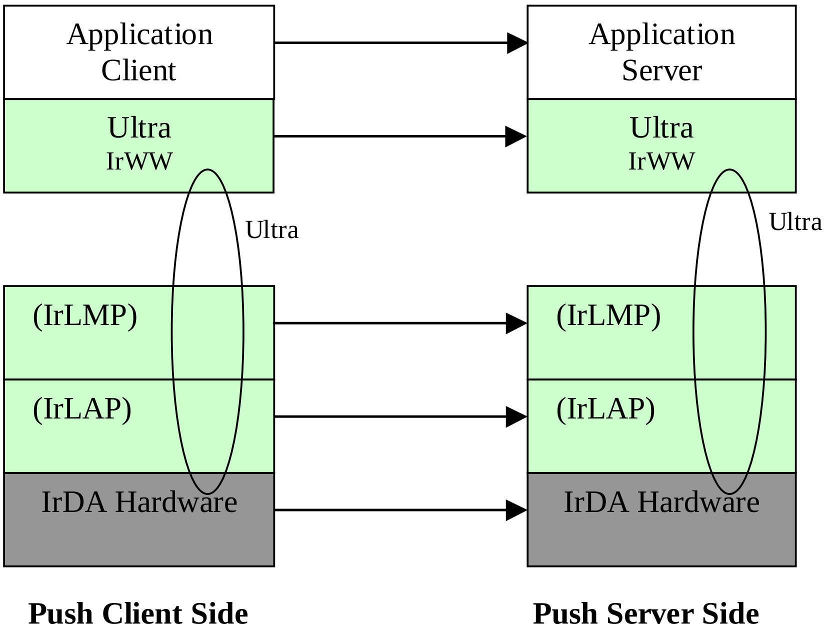
The following roles are defined for this profile.

Push Server – The device that provides an object push server. The Push Server receives Time-Sync object, which is used to synchronize the time with the client.

Push Client – The device that pushes Time-Sync object to the Push Server.

Time-Related objects are used to synchronize the time-related data, such as alarm, timer and stopwatch, between the Push Server and the Push Client as well as Time-Sync object.

3.2.2 Protocol Stack



IrDA Hardware is governed by the [IrPHY]

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

IrLMP is a multiplexing layer specified in [IrLMP]

Ultra is connectionless protocol specified in [Ultra]

Ultra IrWW is placed on the upper layer Ultra protocol. Specified in [Ultra]

Application Client and **Application Server** are the application entities, which provide the user interface and perform the operation of the Time Synchronization profile.

3.2.3 Conformance

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner. This also applies for all optional and conditional capabilities for which support is indicated.

3.3 User Interface Aspects

3.3.1 Mode Selection(Server)

Push Server Mode is the state in which a Server is ready to receive an object from a Push Client.

It is mandatory that a Push Server be in this mode whenever the physical IR port is enabled (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 must correspond to entering Server Mode.

3.3.2 Function Selection(Client)

The **Push Function** initiates the sending of an object to a Push Server.

3.3.3 Application Usage

When the user wants to send Time-Sync object or Time-Related object from a Push Client to a Push Server for synchronization, the following scenario can be followed.

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 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 **Push Function** to send the selected object.	
	When received, the Push Server immediately synchronizes its own time, alarm, timer or stopwatch using the received object. It is recommended that the user be notified of the success of the operation.

3.4 Application Layer

3.4.1 Feature Overview - Basic Concept of Time Synchronization -

Time synchronization is regarded as the most important application. But highly accurate time synchronization result is required. In general, it is necessary to synchronize watches to accuracy of 10 ms or less. In order to guarantee highly accurate time synchronization, it is necessary to minimize transmission delay. If following sequence is carried out, delay time can be minimized.
1) When periodic interrupt signal, that is generated every one second, is received, timer data is captured and set to current time buffer.
2) An object frame is generated in which the current time data is involved.
3) The object frame is transmitted to a receiver right after process 2) is over.
The period between the beginning of process 1) and the end of process 3) correspond to delay time for time synchronization. Figure 1 depicts schematic of the process.
If the delay time is estimated, the sender can transmit estimated delay time as well as current time. And the receiver can compensate the current time with the delay time.
To precisely estimate delay time, it is necessary to eliminate unpredictable time loss that happens during above process. Time loss often occur in the transition from process 2) to process 3), if each process is implemented as independent task.
Taking into account the condition above, a protocol in which transmitting process is carried out immediately after framing process is suitable for time synchronization. Ultra protocol satisfies above requirement and accuracy of 104 microseconds is expected.

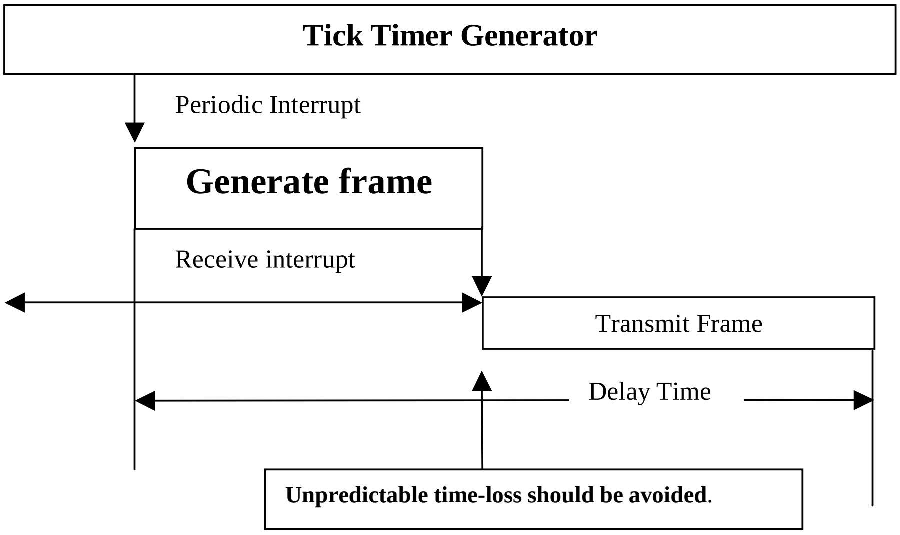


Figure 1 Time Synchronization Concept

A device following this profile must be a Push Client, Push Server or both. Table 2 below shows objects in this profile. Time-Sync object must be supported.

Object	Push Client	Push Server
Time-Sync	M	M
Alarm	O	O
Timer	O	O
Stopwatch	O	O

Table 2 Objects

3.4.2 Content Types

To achieve application level interoperability, content formats are defined for synchronization. The content formats are dependent on key application classes.

3.4.3 Time-Sync Object

Content format of the Time-Sync object is specified in [GBO].

Following is an example of the Time-Sync object written down with character strings.

```
(1) BEGIN: bWATCH
(2)    VERSION:1.0
(3)    BEGIN:T-Sync
(4)       DATE;TYPE=VDYMDW:value                      ⋯Mandatory
(5)       TIME;TYPE=VTHMSL:value                      ⋯Mandatory
(6)       UTCOFFSET:value(VTUTC)                      ⋯Mandatory
(7)       PTIME;TYPE=VTRL:value                       ⋯Mandatory
(8)       REFERENCE:RTIME;RTYPE                       ⋯Mandatory
(9)    END:T-Sync
(10) END:bWATCH
```

In the particular watches which have world time ability, UTCOFFSET property is used to indicate the time zone explicitly. In this case, local time data is transmitted in DATE and TIME properties with UTCOFFSET value. The watches not having world time ability transmit time data with 0xFF value in UTCOFSET property. In line (7), delay time is defined. And REFERENCE property is added to give receiver information about source of transmitted time. All properties are defined mandatory.

And the object translated into GBO format for Ultra IrWW Protocol. Table 3 shows Time-Sync object and corresponding GBO Time-Sync object for Ultra IrWW. OID for Time-Sync is 0x00 and mandatory properties are set in order. This example occupies 55 of 60 Bytes in the Ultra IrWW payload capability (refer to [Ultra IrWW]).

OID	**0x00**								
DATE:TYPE=VDYMDW:value	**0x28**	**0x10**	**0x30**	**0x09**	**0x01**	**0x89**	**0x05**	**99999**	**0x00**
TIME;TYPE=VTHMSL:value	**0x20**	**0x10**	**0x30**	**0x03**	**0x01**	**0x83**	**0x06**	**999999**	**0x00**
UTCOFFSET:value	**0x24**	**0x01**	**0x8c**	**0x02**	**99**	**0x00**			
PTIME;TYPE=VTRL:value	**0x21**	**0x10**	**0x30**	**0x06**	**0x01**	**0x86**	**0x03**	**999**	**0x00**
REFERENCE:RTIME;RTYPE	**0x40**	**0x01**	**0x20**	**0x01**	**0x01**	**0x20**	**0x01**	**0x02**	**0x00**

Table 3 Time-Sync Object example for Ultra IrWW

3.4.4 Alarm Object

Content format of Alarm Object is specified in [GBO].

Following is an example of the Alarm Object written down with character strings.

```
1)  BEGIN : BWATCH
2)     VERSION:1.0
3)     BEGIN : ALARM
4)        DATE;TYPE= VDYMDW:value          …Mandatory
5)        TIME;TYPE=VTHM:value             …Mandatory
6)        UTCOFFSET:value
7)        CATEGORY:OTHER
8)        ACTIVE:ON
9)        ACTION;TYPE=SOUND:(value)
10)       NOTE:value
11)    END ALARM
12) END BWATCH
```

DATE (year/month/day/week type) and TIME (hour/minute type) are minimum set of property for alarm set. So, these two are defined mandatory.
UTCOFFSET: This property is used to indicate the time zone explicitly. If this property is ignored, Date and time is regarded as local time.
CATEGORY: This property indicates for what purpose the alarm is used. If this property is ignored, default CATEGORY; OTHER is set.
ACTIVE: This property define whether this function is activated in the receiver device or not. The purpose of this function is to let the owner of the receiver know the specified time. So default value of this property should be ON. Also it is recommended that when sending this object, ACTIVE parameter should be set to ON or ACTIVE parameter should not be described.
ACTION: This property specifies function to be carried out at the target time. There are three property parameters; DISPLAY, SOUND and VIBRATION that are basic functions for watch. More than one function can be listed. If this property is ignored, default function of the receiver watch is activated.
NOTE: By using this property, short message can be inserted. By setting this property, Alarm object can be utilized as simple schedule or to do event.

And the object translated into GBO format for Ultra IrWW Protocol. Table 4 shows Alarm Object and the corresponding GBO Alarm Object for Ultra IrWW. OID for Alarm is 0x10 and mandatory properties are set in order. Any other optional properties can be put at random following the mandatory properties until the object size will be 60Bytes. The object size is 24 Bytes in this example, 36 Bytes are available for optional properties.

OID	0x10								
DATE:TYPE=VDYMDW:value	0x28	0x10	0x30	0x09	0x01	0x89	0x05	99999	0x00
TIME;TYPE=VTHM:value	0x20	0x10	0x30	0x01	0x01	0x81	0x02	99	0x00

Table 4 Alarm Object example for Ultra IrWW

3.4.5 Timer Object

Content format of Timer Object is specified in [GBO].

Following is an example of the Timer Object written down with character strings.

```
1)    BEGIN : BWATCH
2)      VERSION:1.0
3)      BEGIN : TIMER
4)        PTIME;TYPE=VTHMS:value           …Mandatory
```

```
       REPEAT:ON                  …Mandatory
       NOTE:value
     END : TIMER
   END : BWATCH
```

Type of the PTIME property in this case is hour/minute/second.
REPEAT: The device can repeat time count continuously when the status of this property is on. REPEAT is generic function of the timer.

And the object translated into GBO format for Ultra IrWW Protocol. Table 5 shows Timer Object and the corresponding GBO Timer Object for Ultra IrWW. OID for Timer is 0x20 and mandatory properties are set in order. Any other optional properties can be put at random following the mandatory properties until the object size will be 60Bytes. The object size is 19 Bytes in this example, 41 Bytes are available for optional properties.

OID	0x20								
PTIME;TYPE=VTHMS:value	0x21	0x10	0x30	0x02	0x01	0x82	0x03	999	0x00
REPEAT:ON	0x52	0x10	0x20	0x01	0x01	0x00	0x00		

Table 5 Timer Object example for Ultra IrWW

3.4.6 Stopwatch Object

Content format of Stopwatch Object is specified in [GBO] .

Following is an example of the Stopwatch Object written down with character strings.

```
BEGIN : BWATCH
  VERSION:1.0
  BEGIN : STOPWATCH
     DATE;TYPE=VDYMDW:value
     PTIME;TYPE=VTHMSL:value         …Mandatory
     NOTE:value
  END : STOPWATCH
END : BWATCH
```

DATE property has year/month/day/week type value.
Type of the PTIME property in this case is hour/minute/second/under-second and only PTIME property is defined mandatory.

And the object translated into GBO format for Ultra IrWW Protocol. Table 6 shows Stopwatch Object and the corresponding GBO Stopwatch Object for Ultra IrWW. OID for Stopwatch is 0x30 and mandatory properties are set in order. Any other optional properties can be put at random following the mandatory properties until the object size will be 60Bytes. The object size is 15 Bytes in this example, 45 Bytes are available for optional properties.

OID	0x30								
PTIME;TYPE=VTHMSL:value	0x21	0x10	0x30	0x03	0x01	0x83	0x06	999999	0x00

Table 6 Stopwatch Object example for Ultra IrWW

3.5 Ultra IrWW

Ultra IrWW is based on Ultra Protocol specified in [Ultra].
IrDA IrWW Devices must follow the recommendation in the Ultra specification that limits the maximum service data field size to 62 octets in length. Connectionless service is thus limited to use frames with maximum payload size of 60 octets.

OID specifies IrWW objects that are transferred with Ultra protocol. Unique number is given to each object.

Object	OID
Time-Adjust	0x00
Alarm	0x10
Timer	0x20
Stopwatch	0x30

Table 7 IrWW Object ID

The details of Ultra IrWW are described in [Ultra IrWW].

3.6 Ultra Protocol

PID is protocol ID of the high rank layer using ultra protocol. 0x01 is reserved for Ultra-OBEX currently. And 0x02 is reserved for Ultra-IrWW.

PID Octet	
PID No	**Function**
0x00	Reserved
0x01	For Ultra OBEX
0x02	**For Ultra IrWW**
0x03 – 0x7f	TBD
MSB(bit 7)	For extention

Note: All PID values are reserved and assigned by IrDA.

Table 8 The PID Octet Encoding

In the Ultra IrWW, object data size is less than 60bytes so that communication procedure should be completed with one UI frame. Therefore, SAR is always set to 00h, and Ultra IrWW Frame consists of only one frame.
The details of Ultra Protocol are described in [Ultra].

3.7 IrLMP

A device must support connectionless LSAP(both DLSAP and SLSAP are 0x70), and the data packet must be delivered to the upper layer transparently.

3.8 IrLAP

The data packet must be delivered to IrLMP or upper layer.

3.9 Physical Layer

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

4 ANNEX: Time Synchronization Profile for Legacy PC/PDA

4.1 User Requirements

4.1.1 Scope

The Time Synchronization profile (Connection-based) defines the requirements for the protocols and procedures that shall be used by applications implementing the Time Synchronization usage model. The most common devices implementing this usage model include PCs and PDAs.

4.1.2 User Scenarios

The scenarios covered by this profile are:

Usage of an IrDA device to synchronize the time with another IrDA device by using Time-Sync object.
Usage of an IrDA device to synchronize the time-related data with another IrDA device by using Time-Related object.

Object Type	Format	Examples
Time-Sync Object (CO)	BWATCH 1.0	Time
Time-Related Object (CO)	BWATCH 1.0	Alarm, Timer, Stopwatch

Note: (CO) is Connection-Oriented Protocol.

Table 1 Objects for Time Synchronization Profile

4.2 Profile Overview

4.2.1 Configuration and Roles

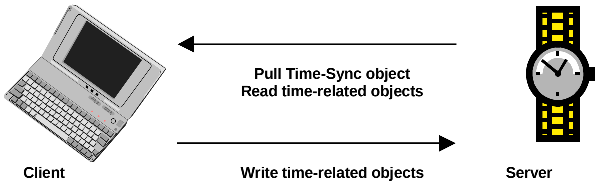


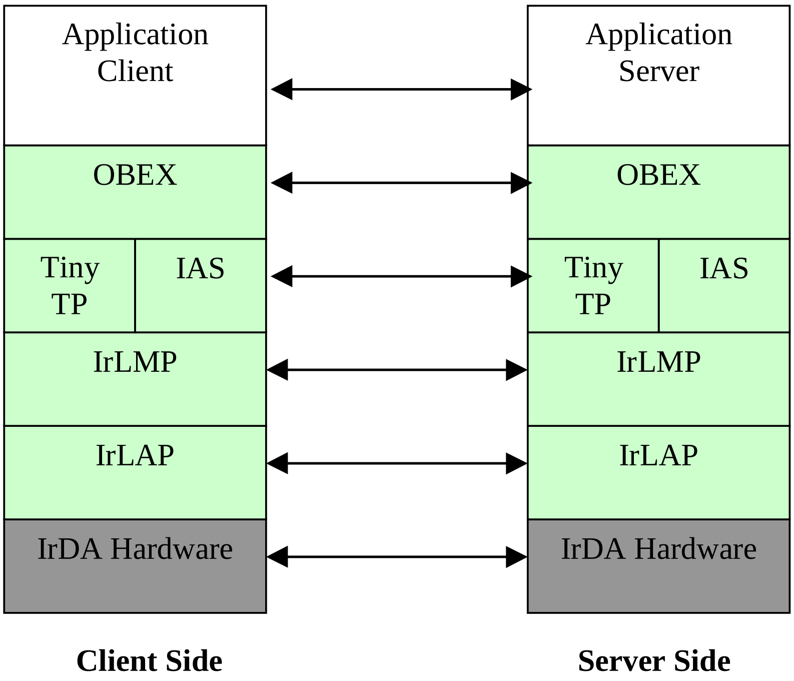
Figure 2 Time Synchronization Example between wristwatch and PC

The following roles are defined for this profile.

Server – The device provides an object exchange server to and from which data objects can be pushed and pulled. The Server waits passively for the client to initiate the operation.

Client – The device can push or/and pull data objects to and from the server.

4.2.2 Protocol Stack



IrDA Hardware is governed by the [IrPHY]

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

IrLMP is a multiplexing layer specified in [IrLMP]

Tiny TP provide flow control and is specified in [TINYTP]

IAS is the Information Access Service specified in [IrLMP]

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

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

4.2.3 Conformance

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner. This also applies for all optional and conditional capabilities for which support is indicated.

4.3 User Interface Aspects

4.3.1 Mode Selection (Server)

Server Mode is the state in which a server is ready to receive objects from a client or to send objects to a client.

It is mandatory that a server be in this mode whenever the physical IR port is enabled (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 must correspond to entering server mode.

4.3.2 Function Selection (Client)

The **Pull Function** initiates the reading of Time-Sync object or time-related objects from a server.
The **Push Function** initiates the writing of time-related objects to a server.

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.

4.3.3 Application Usage

When the user wants to read Time-Sync object or time-related objects from a server to a client or write time-related objects from a client to a server, the following scenario can be followed.

Client	**Server**
	The user sets the device into **Server Mode** if it is not already.
The user of the Client selects a kind of time-related object to pull or push.	
The user points the IR port of the Client device at the IR port of the Server device.	
The user selects the **Push Function or Pull Function** to read or write the selected object.	
	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.
It is recommended that the user be notified of the result of the operation.	

4.4 Application Layer

4.4.1 Feature Overview

A device following this profile must be a Client, Server or both and must support one of the content types listed in the next section.

4.4.2 Content Types

To achieve application level interoperability, content formats are defined for Time Synchronization. The content formats are dependent on key application classes.

4.4.2.1 Time-Sync Object

Content format of Time-Sync object is specified in [IrWW].

Following is an example of the Time-Sync object written down with character strings.

```
1) BEGIN: BWATCH
2)    VERSION:1.0
3)    BEGIN:T-SYNC
4)       DATE;TYPE=VDYMDW:value              …Mandatory
5)       TIME;TYPE=VTHMSL:value              …Mandatory
6)       UTCOFFSET:value(VTUTC)              …Mandatory
7)       PTIME;TYPE=VTRL:value               …Mandatory
8)       REFERENCE:RTIME;RTYPE               …Mandatory
9)    END:T-SYNC
10) END:BWATCH
```

To correct time zone automatically, UTCOFFSET property is involved. In line (6), delay time is defined. And REFERENCE property is added to give receiver information about source of transmitted time. All properties are defined mandatory.

And the object translated into GBO format. Table 2 shows Time-Sync Object and the corresponding GBO Time-Sync Object for this profile.

Time-Sync Object	bWATCH Time-Sync Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:BWATCH	0x08		0x01	0x20 0x01 0x80	0x00
VERSION:1.0	0x0a		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:T-SYNC	0x08		0x01	0x20 0x03 0xff 0x80 0x01	0x00
DATE;TYPE=VDYMDW:value	0x28	0x10 0x30 0x09	0x01	0x89 0x05 99999	0x00
TIME;TYPE=VTHMSL:value	0x20	0x10 0x30 0x03	0x01	0x83 0x06 999999	0x00
UTCOFFSET:value	0x24		0x01	0x8c 0x02 99	0x00
PTIME;TYPE=VTDL:value	0x21	0x10 0x30 0x06	0x01	0x86 0x03 999	0x00
REFERENCE:RTIME;RTYPE	0x40	0x01 0x20 0x01	0x01	0x20 0x01 0x02	0x00
END:T-SYNC	0x09		0x01	0x20 0x03 0xff 0x80 0x01	0x00
END:BWATCH	0x09		0x01	0x20 0x01 0x80	0x00

Table 2 Time-Sync object example

4.4.2.2 Alarm Object

Content format of Alarm Object is specified in [IrWW].

Following is an example of the Alarm object written down with character strings.

```
1)  BEGIN : BWATCH
2)     VERSION:1.0
3)     BEGIN : ALARM
4)        DATE;TYPE= VDYMDW:value          …Mandatory
5)        TIME;TYPE=VTHM:value             …Mandatory
6)        UTCOFFSET:value
7)        CATEGORY:OTHER
8)        ACTIVE:ON
9)        ACTION;TYPE=SOUND:(value)
10)       DESCRIPTION:value
11)    END : ALARM
12) END : BWATCH
```

DATE (year/month/day/week type) and TIME (hour/minute type) are minimum set of property for alarm set. So, these two are defined mandatory.
UTCOFFSET: This property is used for time zone correction. If this property is ignored, Date and time is regarded as local time.
CATEGORY: This property indicates for what purpose the alarm is used. If this property is ignored, default CATEGORY; OTHER is set.
ACTIVE: This property define whether this function is activated in the receiver device or not. The purpose of this function is to let the owner of the receiver know the specified time. So default value of this property should be ON. Also it is recommended that when sending this object, ACTIVE parameter should be set to ON or ACTIVE parameter should not be described.
ACTION: This property specifies function to be carried out at the target time. There are three property parameters;　DISPLAY, SOUND and VIBRATION that are basic functions for watch. More than one function can be listed. If this property is ignored, default function of the receiver watch is activated.
DESCRIPTION: By using this property, short message can be inserted. By setting this property, Alarm object can be utilized as simple schedule or to do event.

And the object is translated into GBO format. In Table 3, Alarm Object and the corresponding GBO Alarm Object are listed.

Alarm Object	bWATCH Alarm Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:BWATCH	0x08		0x01	0x20 0x01 0x80	0x00
VERSION:1.0	0x0a		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:ALARM	0x08		0x01	0x20 0x03 0xff 0x80 0x02	0x00
DATE;TYPE=VDYMDW:value	0x28	0x10 0x30 0x09	0x01	0x89 0x05 99999	0x00
TIME;TYPE=VTHM:value	0x20	0x10 0x30 0x01	0x01	0x81 0x02 99	0x00
UTCOFFSET:value	0x24		0x01	0x8c 0x02 99	0x00
CATEGORY:OTHER	0x48		0x01	0x20 0x01 0xff	0x00
ACTIVE:ON	0x50		0x01	0x20 0x01 0x01	0x00
ACTION:TYPE=SOUND:(value)	0x58	0x10 0x30 0x01	0x01	(VC VL VD)	0x00
DESCRIPTION:value	0x31		0x01	0x30 0x05 ‘abcde’	0x00
END:ALARM	0x09		0x01	0x20 0x03 0xff 0x80 0x02	0x00
END:BWATCH	0x09		0x01	0x10 0x01 0x80	0x00

Table 3 Alarm Object example

Following is an example of Alarm objects.

```
BEGIN:BWATCH
VERSION:1.0
BEGIN:ALARM
  DATE;TYPE=VDYMDW:value
  TIME;TYPE=VTHM:value
  ACTIVE:ON
  …
END:ALARM
BEGIN:ALARM
  DATE;TYPE=VDYMDW:value
  TIME;TYPE=VTHM:value
  ACTIVE:ON
  …
END:ALARM
                  ⋮
BEGIN:ALARM
  DATE;TYPE=VDYMDW:value
  TIME;TYPE=VTHM:value
  ACTIVE:ON
  …
END:ALARM
END:BWATCH
```

4.4.2.3 Alarm Information Log Object

Alarm Information Log object content format specified in [IrWW].

The Read-all/Write-all Client can send a request to get Information Log Object that contains information of properties existing in the Alarm Object. When the client device request GET /watch/alarm/info.log file, the server should respond to the request and send Information Object to the client. An example of Alarm Information Log Object is as follows;

```
(IEL:2)
X-IRWW-FIELDS:
<Begin>
Version:
DATE;TYPE=VDYMDW:
TIME;TYPE=VTHM:
UTCOFFSET:
CATEGORY;VALUE=WAKE;CALL;OTHER:
ACTION;TYPE=SOUND:
DESCRIPTION:=10
<End>
Total-Records:2
Maximum-Records:3
```

This object contains following information.
(The device supports Usage Model 2.)
The device supports DATE type VDYMDW (YYYY.MM.DD.WW).
The device supports TIME type VTHM (HH:MM).
The device supports UTCOFFSET.
The device supports WAKE, CALL and OTHER for CATEGORY value.
The device supports SOUND for ACTION.
The device supports up to 10 bytes comment (DESCRIPTION).
The Total Number of alarm Records is 2.
The Maximum Number of alarm Records that can be stored is 3.

And the object is translated into GBO format. In Table 4, Alarm Information Log Object and the corresponding GBO Alarm Information Log Object are listed.

Alarm Information Log Object	bWATCH Alarm Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x30 0x01 ‘2’	0x00
X-IRWW-FIELDS:	0x80				0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
DATE;TYPE=VDYMDW:	0x2a	0x10 0x30 0x09	0x01		0x00
TIME;TYPE=VTHM:	0x20	0x10 0x30 0x01	0x01		0x00
UTCOFFSET:	0x24		0x01		0x00
CATEGORY:WAKE	0x48		0x01	0x20 0x01 0x01	0x00
CATEGORY:CALL	0x48		0x01	0x20 0x01 0x02	0x00
CATEGORY:OTHER	0x48		0x01	0x20 0x01 0x05	0x00
ACTION;TYPE=SOUND	0x58	0x10 0x30 0x01	0x01		0x00
DESCRIPTION:=10	0x31		0x01	0x30 0x03 ‘=10’	
<END>	0x89				0x00
Total-Records:2	0xf4		0x01	0x30 0x01 ‘2’	0x00
Maximum-Records:3	0xf6		0x01	0x80 0x01 ‘3’	0x00

Table 4 Alarm Information Log object example

4.4.2.4 Timer Object

Content format of Timer Object is specified in [IrWW].

Following is an example of the Timer Object written down with character strings.

```
1)    BEGIN : BWATCH
2)      VERSION:1.0
3)      BEGIN : TIMER
4)        PTIME;TYPE=VTHMS:value          …Mandatory
5)        REPEAT:ON                       …Mandatory
6)        DESCRIPTION:value
7)      END : TIMER
8)    END : BWATCH
```

Type of the PTIME property in this case is hour/minute/second.
REPEAT: The device can repeat time count continuously when the status of this property is on. REPEAT is generic function of the timer.

And the object is translated into GBO format. In Table 5, Timer Object and the corresponding GBO Timer Object are listed.

Timer Object	bWATCH Timer Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:BWATCH	0x08		0x01	0x20 0x01 0x80	0x00
VERSION:1.0	0x0a		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:TIMER	0x08		0x01	0x20 0x03 0xff 0x80 0x03	0x00
PTIME;TYPE=VTHMS:value	0x21	0x10 0x30 0x02	0x01	0x82 0x03 999	0x00
REPEAT:ON	0x52	0x10 0x20 0x01	0x01		0x00
DESCRIPTION:value	0x31		0x01	0x30 0x05 ‘abcde’	0x00
END:TIMER	0x09		0x01	0x20 0x03 0xff 0x80 0x03	0x00
END:BWATCH	0x09		0x01	0x10 0x01 0x80	0x00

Table 5 Timer Object example

Following is an example of Timer objects.

```
BEGIN:BWATCH
VERSION:1.0
BEGIN:TIMER
   PTIME;TYPE=VTHMS:value
   REPEAT:ON
   …
END:TIMER
BEGIN:TIMER
   PTIME;TYPE=VTHMS:value
   REPEAT:ON
   …
END:TIMER
                         ⋮
BEGIN: TIMER
   PTIME;TYPE=VTHMS:value
   REPEAT:ON
   …
END:TIMER
END:BWATCH
```

4.4.2.5 Timer Informaiton Log Object

Timer Information Log object content format specified in [IrWW].

The Read-all/Write-all Client can send a request to get Information Log Object that contains information of properties existing in the Timer Object. When the client device request GET /watch/timer/info.log file, the server should respond to the request and send Information Object to the client. An example of Timer Information Log Object is as follows;

```
(IEL:2)
X-IRWW-FIELDS:
<Begin>
Version:
DATE;TYPE=VDYMDW:
PTIME;TYPE=VTHMS:
REPEAT:
DESCRIPTION:=10
<End>
Total-Records:1
Maximum-Records:2
```

This Object contains following information.
(The device supports Usage Model 2.)
The device supports DATE type VDYMDW (YYYY:MM:DD:WW).
The device supports PTIME type VTHMS (HH:MM:SS).
The device supports REPEAT timer.
The device supports up to 10 bytes comment (DESCRIPTION).
The Total Number of timer Records is 1.
The Maximum Number of timer Records that can be stored is 2.

And the object is translated into GBO format. In Table 6, Timer Information Log Object and the corresponding GBO Timer Information Log Object are listed.

Timer Information Log Object	bWATCH Timer Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xF0		0x01	0x30 0x01 ‘2’	0x00
X-IRWW-FIELDS:	0x80				0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
DATE;TYPE=VDYMDW:	0x28	0x10 0x30 0x09	0x01		0x00
PTIME;TYPE=VTHMS:	0x21	0x10 0x30 0x02	0x01		0x00
REPERT:	0x52		0x01		0x00
DESCRIPTION:=10	0x31		0x01	0x30 0x03 ‘=10’	0x00
<END>	0x89				0x00
Total-Records:1	0xF4		0x01	0x30 0x01 ‘1’	0x00
Maximum-Records:2	0xF6		0x01	0x30 0x01 ‘2’	0x00

Table 6 Timer Information log Object

4.4.2.6 Stopwatch object

Content format of Stopwatch Object is specified in [IrWW].

Following is an example of the Stopwatch Object written down with character strings.

```
1) BEGIN : BWATCH
2)   VERSION:1.0
3)   BEGIN : STOPWATCH
4)      DATE;TYPE=VDYMDW:value
5)      PTIME;TYPE=VTHMSL:value          …Mandatory
6)      DESCRIPTION:value
7)   END : STOPWATCH
8) END : BWATCH
```

DATE property has year/month/day/week type value.
Type of the PTIME property in this case is hour/minute/second/under-second and only PTIME property is defined mandatory.

And the object is translated into GBO format. In Table 7, Stopwatch Object and the corresponding GBO Stopwatch Object are listed.

Stopwatch Object	bWACTH Stopwatch Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:BWATCH	0x08		0x01	0x10 0x01 0x80	0x00
VERSION:1.0	0x0a		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:STOPWATCH	0x08		0x01	0x20 0x03 0xff 0x80 0x04	0x00
DATE;TYPE=VDYMDW:value	0x28	0x10 0x30 0x09	0x01	0x89 0x05 99999	0x00
PTIME;TYPE=VTHMSL:value	0x21	0x10 0x30 0x03	0x01	0x83 0x06 999999	0x00
DESCRIPTION:value	0x31		0x01	0x30 0x05 ‘abcde’	0x00
END:STOPWATCH	0x09		0x01	0x20 0x03 0xff 0x80 0x04	0x00
END:BWATCH	0x09		0x01	0x10 0x01 0x80	0x00

Table 7 Stopwatch Object example

```
BEGIN:BWATCH
VERSION:1.0

BEGIN:STOPWATCH
  DATE;TYPE=VDYMDW:value
  PTIME;TYPE=VTHMSL:value
  …
END:STOPWATCH

BEGIN: STOPWATCH
  DATE;TYPE=VDYMDW:value
  PTIME;TYPE=VTHMSL:value
  …
END:STOPWATCH

          ⋮

BEGIN: STOPWATCH
  DATE;TYPE=VDYMDW:value
  PTIME;TYPE=VTHMSL:value
  …
END:STOPWATCH

END:BWATCH
```

Following is an example of Stopwatch objects.

4.4.2.7 Stopwatch Information Log Object

Stopwatch Information Log object content format specified in [IrWW].

The Read-all/Write-all Client can send a request to get Information Log Object that contains information of properties existing in the Stopwatch Object. When the client device request GET /watch/stop/info.log file, the server should respond to the request and send Information Object to the client. An example of Stopwatch Information Log Object is as follows;

```
(IEL:2)
X-IRWW-FIELDS:
<Begin>
Version:
DATE;TYPE=VDYMDW:
PTIME;TYPE=VTHMSL:
DESCRIPTION:=10
<End>
Total-Records:1
Maximum-Records:2
```

This Object contains following information.
(The device supports Usage Model 2.)
The device supports DATE type VDYMDW (YYYY.MM.DD.WW).
The device supports PTIME type VTHMSL (HH:MM:SS:LLLLLL).
The device supports up to 10 bytes comment (DESCRIPTION).
The Total Number of timer Records is 1.
The Maximum Number of timer Records that can be stored is 2.

And the object is translated into GBO format. In Table 8, Stopwatch Information Log Object and the corresponding GBO Stopwatch Information Log Object are listed.

Stopwatch Information Log Object	bWATCH Stopwatch Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xF0		0x01	0x30 0x01 '1'	0x00
X-IRWW-FIELDS:	0x80				0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A				0x00
DATE;TYPE=VDYMDW	0x2A	0x10 0x30 0x09	0x01		0x00
PTIME;TYPE=VTHMSL	0x21	0x10 0x30 0x03	0x01		0x00
DESCRIPTION:=10	0x31		0x01	0x30 0x03 '=10'	0x00
<END>	0x89				0x00
Total-Records:1	0xF4		0x01	0x30 0x01 '1'	0x00
Maximum-Records:2	0xF6		0x01	0x30 0x01 '2'	0x00

Table 8 Stopwatch Information Log Object

4.4.3 Application Procedure

4.4.3.1 Single Object Pull with IrOBEX

Both Pull Client and Pull Server are built on top of the OBEX application framework. A client uses OBEX_GET to pull single object with the IrWW specified object name from a server. The client connects to the server and pulls single object from it.

Table 9 shows an example of the application procedure required by the client to pull object from a server.

Push Client	Details
OBEX CONNECT.	
Pull the object using OBEX GET.	
OBEX DISCONNECT.	

Table 9 Application layer procedure for Pull object

In single object pull, following file names are used.

Object	File name
Time-Sync	Xxx.wsy

Table 10 Object name assignment rule of Time-Sync Object Pull

As shown in the table10, only time-Sync object can be pulled from the server.

4.4.3.2 Single Object Push with IrOBEX

Both Push Client and Push Server are built on top of the OBEX application framework. A Push Client uses OBEX_PUT to push single object to the inbox of a Push Server. The Push Client connects to the Push Server's inbox and pushes single object to it. The Push Client only knows that the object are successfully received. It does not know the layout or construction of the Push Server's inbox.

A Push Server's inbox must hold objects. But, it is possible to automatically dispatch objects from inbox to store. For example, if single Alarm is received it can be dispatched to the Alarm.

Table 11 shows an example of the application procedure required by the Push Client to push object to a Push Server.

Push Client	Details
OBEX CONNECT.	
Push the object using OBEX PUT.	
OBEX DISCONNECT.	

Table 11 Application layer procedure for Push object

In single object push , following file names are used.

Object	File name
Alarm	Xxx.wal
Timer	Xxx.wtm
Stopwatch	Xxx.wsw

Table 12 Object name assignment rule of Time-Related Object Push

4.4.3.3 Read-all/Write-all Objects with IrOBEX

Both a Read-all/Write-all Client and Read-all/Write-all Server are built on top of the OBEX application framework. A Read-all/Write-all Client uses OBEX PUT to write-all objects to the store of a Read-all/Write-all Server. And a Read-all/Write-all Client uses OBEX GET to read-all objects from the store of a Read-all/Write-all Server. Transferring files requires a single PUT or GET operation per file. The Read-all/Write-all Client connects to the Read-all/Write-all Server's storage and transfers all objects. Note that inbox isn't used in Read-all/Write-all objects.

Information Log contains general information about specific Object Stores. Before Read-all/Write-all Client read or write Time-Related objects from or to Read-all/Write-all Server, It becomes help of processing of information by reading Information Log.

Table 13 shows an example of the application procedure required by the Read-all/Write-all Client to read all objects from a Read-all/Write-all Server.

Read-all/Write-all Client	Details
OBEX CONNECT.	
Read the Information Log using OBEX GET.	In accordance with read Information Log, memory is prepared in which the data are stored.
Read all objects using OBEX GET.	
OBEX DISCONNECT.	

Table 13 Application procedure for Read-all objects

It is not necessary for a client to always use Information Log.

Table 14 shows an example of the application procedure required by the Read-all/Write-all Client to write all objects to a Read-all/Write-all Server.

Read-all/Write-all Client	Details
OBEX CONNECT.	
Read the Information Log using OBEX GET.	
Write all objects using OBEX PUT.	In accordance with read Information Log, only necessary data are sent.
OBEX DISCONNECT.	

Table 14 Application procedure for Write-all objects

It is not necessary for a client to always use Information Log.

In Read-all/Write-all objecs, following file names are used.

Object	File name
Alarm Object	/watch/alarm.wal
Alarm Information Log Object	/watch/alarm/info.log
Timer Object	/watch/timer.wtm
Timer Information Log Object	/watch/timer/info.log
Stopwatch Object	/watch/stop.wsw
Stopwatch Information Log Object	/watch/stop/info.log

Table 15 Object name assignment rule of Read-all/Write-all Time-Related Objects

4.5 OBEX

4.5.1 OBEX Operations

The table below shows the OBEX operations, which are used in single object pull in Time Synchronization profile.

Operation no.	OBEX Operation	Pull Client	Pull Server
1	Connect	M	M
2	Disconnect	O	M
3	Put	N/A	N/A
4	Get	M	M
5	Abort	M	M
6	Set Path	N/A	N/A

Table 16 OBEX Operations used in single object pull

The table below shows the OBEX operations, which are used in single object push in Time Synchronization profile.

Operation no.	OBEX Operation	Push Client	Push Server
1	Connect	M	M
2	Disconnect	O	M
3	Put	M	M
4	Get	N/A	N/A
5	Abort	M	M
6	Set Path	N/A	N/A

Table 17 OBEX Operations used in single object push

The table below shows the OBEX operations, which are used in Read-all/Write-all in Time Synchronization profile.

Operation no.	OBEX Operation	Read-all/Write-all Client	Read-all/Write-all Server
1	Connect	M	M
2	Disconnect	O	M
3	Put	M	M
4	Get	M	M
5	Abort	M	M
6	Set Path	N/A	N/A

Table 18 OBEX Operations used in Read-all/Write-all objects

4.5.2 OBEX Headers

OBEX Headers follows the information described in Section 2.5.2 of [IrWW].

4.5.3 Establishing an OBEX session

Establishing an OBEX session follows the procedure described in Section 2.5.3 of [IrWW].

4.5.4 Reading Objects

Reading Object follows the procedure described in Section 2.5.4 of [IrWW].

4.5.5 Writing Objects

Writing Object follows the procedure described in Section 2.5.5 of [IrWW].

4.5.6 Disconnecting an OBEX session

Disconnecting an OBEX session follows the procedure described in Section 2.5.6 of [IrWW].

4.6 Tiny TP

Tiny TP follows the information described in Section 2.8 of [IrWW].

4.7 IrLMP

IrLMP follows the information described in Section 2.9 of [IrWW].

4.8 IAS

IAS follows the information described in Section 2.10 of [IrWW].

4.8.1 Parameters

In Table 19, the Parameters in single object pull are listed.

PI	PI name	PL	PV data type	PV Description	Status
0x00	Time-Sync Support	2		The first octet = Support Level (bit mask)	
			bit 0	Single Push Support	0
			bit 1	Single Pull Support	1
			bit 2	Write-all Support	0
			bit 3	Read-all Support	0
				The second octet :	
				Upper side nibble(4bit) Major version number	0001
				Lower side nibble(4bit) Minor version number 0x00 is undefined version	0000

Table 19 The Parameters of single Time-Sync Object pull

In Table 20, the Parameters in single object push are listed.

PI	PI name	PL	PV data type	PV Description	Status
0x01	Alarm Support	2		The first octet = Support Level (bit mask)	
			bit 0	Single Push Support	1
			bit 1	Undefined	0
			bit 2	Undefined	0 or 1
			bit 3	Undefined	0 or 1
				The second octet :	
				Upper side nibble(4bit) Major version number	0001
				Lower side nibble(4bit) Minor version number 0x00 is undefined version	0000
0x02	Timer Support	2		The first octet =	

				Support Level	
				(bit mask)	
			bit 0	Single Push Support	1
			bit 1	Undefined	0
			bit 2	Undefined	0 or 1
			bit 3	Undefined	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x03	Stopwatch Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	1
			bit 1	Undefined	0
			bit 2	Undefined	0 or 1
			bit 3	Undefined	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	

Table 20 Parameters of single Time-related Object push

In Table 21, the Parameters in Read-all/Write-all objects are listed.

PI	PI name	PL	PV data type	PV Description	Status
0x01	Alarm Support	2		The first octet =	
				Support Level	
				(bit mask)	

			bit 0	Single Push Support	0 or 1
			bit 1	Single Pull Support	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x02	Timer Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Single Pull Support	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x03	Stopwatch Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Single Pull Support	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	

				Lower side nibble(4bit) Minor version number 0x00 is undefined version	0000

Table 21 The Parameters of Read-all/Write-all Time-related Objects

4.8.2 DeviceInfo

DeviceInfo follows the information described in Section 2.10.3 of [IrWW].

4.9 Service Hint Bit

Service Hint Bit follows the information described in Section 2.11 of [IrWW].

4.10 IrLAP

IrLAP follows the information described in Section 2.12 of [IrWW].

4.11 Physical Layer

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

第3部

簡易オブジェクト交換プロファイル

Infrared Data Association

IrWW (Infrared Wrist Watch)
Tiny Object Exchange profile

Version 1.0

December 26, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Hiroshi Morohoshi h-moroho@tmgw.rd.casio.co.jp

Significant Contributors:

History:
Version 0.5: Infrared Data Association IrWW IrDA for Wrist Watches for Directional Vote
Version 0.8: IrWW Time-LOG Object Push Profile for Draft Vote.
Version 0.8: IrWW Time-LOG Objects Exchange Profile for Draft Vote.
Version 0.8: IrWW Tiny-PIM Object Push Profile for Draft Vote.
Version 0.8: IrWW Tiny-PIM Objects Exchange Profile for Draft Vote.
Version 1.0: IrWW Tiny Object Exchange Profile for Final Vote.

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 INTRODUCTION 1

1.1 IrWW Profile Structure 1
1.2 Definitions and Acronyms 1
1.3 Symbols and Conventions 2
1.4 References 2

2 POINT AND SHOOT OBJECT PUSH **3**

2.1 Scope 3
2.2 User Scenario 3
2.3 Interoperability 3
2.4 Usability 4
2.5 Configuration 4
2.6 Reliability 4
2.7 Additional Information 4

3 TINY OBJECT EXCHANGE PROFILE 5

3.1 User Requirements 5
3.1.1 Scope 5
3.1.2 User Scenarios 5
3.2 Profile Overview 5
3.2.1 Configuration and Roles 5
3.2.2 Protocol Stack 6
3.2.3 Conformance 6
3.3 User Interface Aspects 7
3.3.1 Mode Selection (Server) 7
3.3.2 Function Selection (Client) 7
3.3.3 Application Usage 7
3.3.3.1 Single Push 7
3.3.3.2 Read-all/Write-all 8
3.4 Application Layer 8
3.4.1 Feature Overview 8
3.4.2 Content Types 8
3.4.2.1 Single Push Object 8
3.4.2.1.1 Time Log Object 8
3.4.2.1.2 Business Card Object 13
3.4.2.1.3 Schedule Object 14
3.4.2.1.4 To do Object 14
3.4.2.1.5 Notes Object 15
3.4.2.2 Read-all/Write-all Object 15
3.4.2.2.1 Device Information 15
3.4.2.2.2 Time Log Object 16
3.4.2.2.3 Time Log's Information Log Object 16
3.4.2.2.4 Owner's Business Card Object 17
3.4.2.2.5 Business Card Object 18
3.4.2.2.6 Business Card’s Information Log Object 18
3.4.2.2.7 Schedule Object 19
3.4.2.2.8 Schedule’s Information Log Object 19
3.4.2.2.9 To do Object 20
3.4.2.2.10 To do’s Information Log Object 21
3.4.2.2.11 Notes Object 21
3.4.2.2.12 Notes Information Log Object 22
3.4.3 Application Procedure 22

3.4.3.1 Application procedure for Single Push object ... 22
3.4.3.2 Application procedure for Read-all objects ... 23
3.5 OBEX ... 24
3.5.1 OBEX Operations ... 24
3.5.2 OBEX Headers ... 25
3.5.3 Establishing an OBEX session ... 25
3.5.4 Pushing Object ... 25
3.5.5 Reading Objects ... 25
3.5.6 Writing Objects ... 25
3.5.7 Disconnecting an OBEX session ... 25
3.6 Tiny TP ... 25
3.7 IrLMP ... 25
3.8 IAS ... 25
3.9 Service Hint Bit ... 28
3.10 IrLAP ... 28
3.11 Physical Layer ... 28

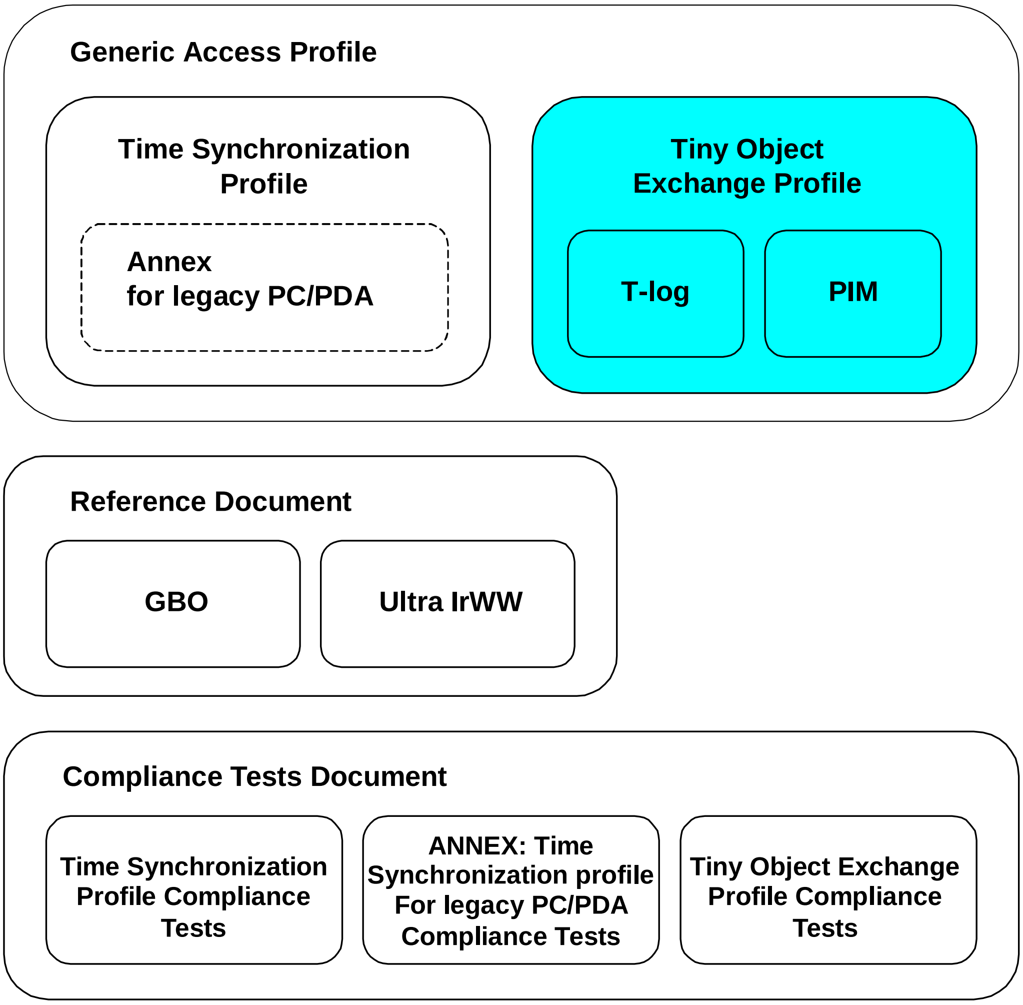
4 APPENDIX ... 29

4.1 Unique code assignment tables of the Binary vCard ... 29
4.2 Unique code assignment tables of the Binary vCalendar ... 30
Unique code assignment tables of the Binary vNote ... 31

1 Introduction

IrWW (IrDA for Wristwatches) provides Time-based data communication scheme for a wristwatch by using IrDA Communication Standards. This profile is to be largely used together with the IrDA standard specifications.

1.1 IrWW Profile Structure


Generic Access Profile
Time Synchronization Profile
Annex for legacy PC/PDA
Tiny Object Exchange Profile
T-log
PIM
Reference Document
GBO
Ultra IrWW
Compliance Tests Document
Time Synchronization Profile Compliance Tests
ANNEX: Time Synchronization profile For legacy PC/PDA Compliance Tests
Tiny Object Exchange Profile Compliance Tests

1.2 Definitions and Acronyms

Inbox A generic Object Store on an IrWW Device that can hold objects of various formats, such as Business Cards, Schedules, To dos and Notes. It is typically used as a temporary holding area for objects received from other IrWW Devices. Often, the user can inspect the items in the Inbox, and file them away or delete them.

Split Split Time Function to show elapsed time from the start point one by one while keeping count up.

1.3 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.
C	Conditional support. Refers to capabilities that shall be used in when certain other capabilities are also used.
X	Excluded. Refers to capabilities that may be supported by the device but shall never be used in this profile.
N/A	Not applicable in the given context. It is impossible to use this capability.

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.

1.4 References

[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
[TINYTP]	Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, 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.
[IrWW]	IrWW Generic Access Profile, Version 1.0, Infrared Data Association.

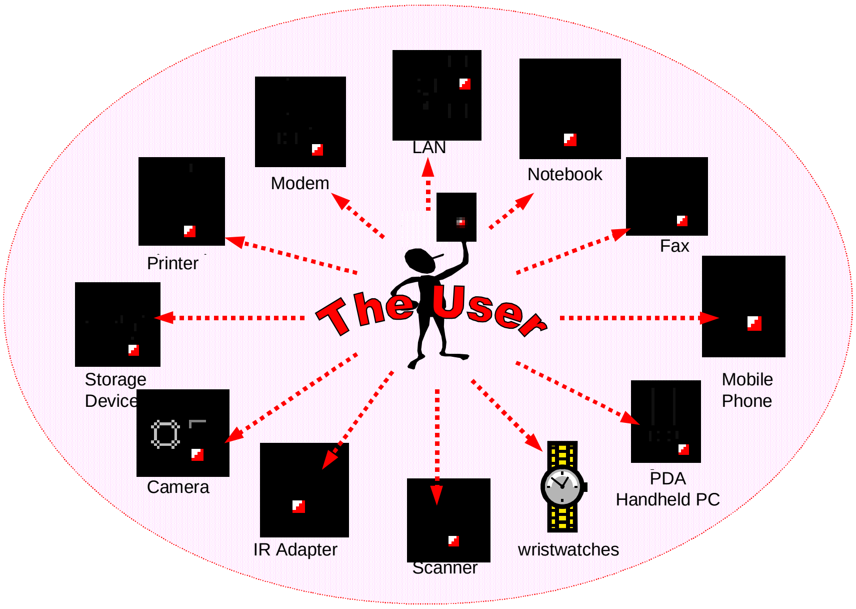
2 Point and Shoot Object Push

2.1 Scope

The scope of the information presented here is based on the ability to exchange data between two IrDA enabled devices. Our focus is on what the user will 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 model is the universal way to move data between IrDA enabled devices. The key to universal object exchange is support for standard object types such as vCard, JPEG, and text. Almost all IrDA devices will support this capability including PCs, printers, PDAs, cameras, phones, watches, pagers, storage devices, and kiosks.

2.2 User Scenario

Many user scenarios are covered by point and shoot object push. The picture below captures the power and simplicity of Point and Shoot Object Push.



The following scenarios are possible:

- The user can push his business card from his watch to another persons watch.
- The user can store time-log information in a multi-functional watch and transfer it to PC, PDA.

2.3 Interoperability

IrReady 2000 devices will have this 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.

Types of Data Exchange:

- **Generic Files**
    - Exchange of files in which Time-log informations are stored.
    - Exchange of files in which GBO based PIM objects (business card, schedule, to do list and note) are stored.

2.4 Usability

Users will be able to transfer an object to another device by simply 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 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.

2.5 Configuration

No configuration is the default for pushing objects. In some systems the user can select the location of the inbox and possibly the behavior of prompts.

2.6 Reliability

Objects will be sent error free. Specific reliability standards will be identified in the test specifications associated with the required enabling technology.

2.7 Additional Information

Objects received may be put into the appropriate data store on the device or sent to the appropriate application. For example, on a PC, received business cards could be placed directly into the user’s PIM.

3 Tiny Object Exchange Profile

3.1 User Requirements

3.1.1 Scope

Tiny Object Exchange profile defines the requirements for the protocols and procedures that shall be used by applications implementing the Point and Shoot usage model. The most common devices implementing this usage model include wristwatches, PCs and PDAs.

3.1.2 User Scenarios

The scenarios covered by this profile are:
Usage of a Client to push an object to a Server.
Usage of a Client to exchange objects with a Server.

Tiny Objects are covered as shown in the table below

Data Type	Format	Examples
Time log	T-log of bWatch 1.0	Time log exchange
Business Card	Binary vCard 2.1	Business card exchange
Schedule	EVENT of Binary vCalendar 1.0	Exchange of schedule items
To do	TODO of Binary vCalendar 1.0	Exchange of to do items,
Notes	Binary vNote (IrMC 1.1)	Memo exchange

Table 1 Tiny Objects

3.2 Profile Overview

3.2.1 Configuration and Roles

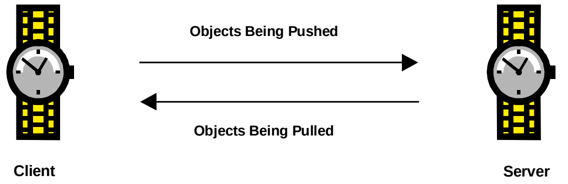


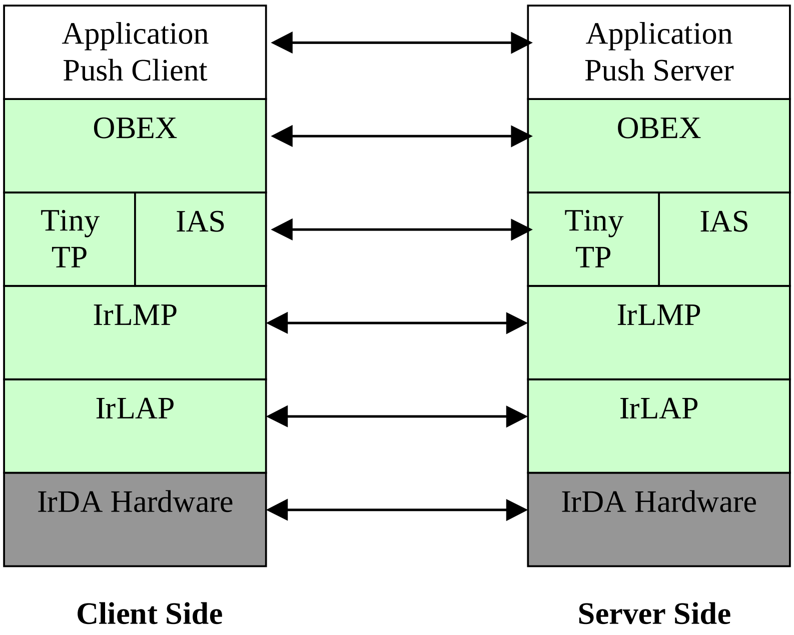
Figure 1 Tiny Object Exchange Example between two wristwatches

The following roles are defined for this profile.

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

Client – This is the client device, which pushes and pulls objects to and from the Server. The Client initiates the operation.

3.2.2 Protocol Stack



IrDA Hardware is governed by the [IrPHY]

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

IrLMP is a multiplexing layer specified in [IrLMP]

Tiny TP provide flow control and is specified in [TINYTP]

IAS is the Information Access Service specified in [IrLMP]

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

Application Client and **Application Server** are the application entities, which provide the user interface and perform the operation of the Tiny Object Exchange profile.

3.2.3 Conformance

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner. This also applies for all optional and conditional capabilities for which support is indicated.

3.3 User Interface Aspects

3.3.1 Mode Selection (Server)

Server Mode is the state in which a Server is ready to receive an object from a Client. When entering this mode the Server must register the IAS entry and set the hint bit (See Section 3.8). It must be in a state where it is ready to respond to incoming Discovery frames and accept an incoming OBEX connection.

It is mandatory that a Server be in this mode whenever the physical IR port is enabled (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 must correspond to entering Server Mode.

3.3.2 Function Selection (Client)

There are three different **functions** associated with Tiny Object Exchange Profile.
- Single Push function
- Read-all Function
- Write-all Function

The **Single Push Function** initiates the pushing of a single Tiny Object to a Server.
The **Read-all Function** initiates the reading of all Tiny Objects from a Server.
The **Write-all Function** initiates the writing of all Tiny Objects to a Server.

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.

3.3.3 Application Usage

In the following sections, the presented scenarios work as examples and variations in the actual implementations are possible and allowed.

3.3.3.1 Single Push

When the user wants to push a single Tiny Object from a Client to a Server, the following scenario can be followed.

Client	**Server**
	The user sets the device into **Server Mode** if it is not already.
The user of the Client selects a single Tiny Object to send.	
The user points the IR port of the Client device at the IR port of the Server device.	
The user selects the **Single Push Function** to send the selected object.	
	When received, the Server stores the object in a generic **Inbox** with the original name of the object. It is possible that the user may be asked to accept or reject the object.

It is recommended that the user be notified of the result of the operation.	

3.3.3.2 Read-all/Write-all

When the user wants to read all Tiny Objects from a Server to a Client or write all Tiny Objects from a Client to a Server, the following scenario can be followed.

Client	**Server**
	The user sets the device into **Server Mode** if it is not already.
The user of the Client selects a kind of Tiny Object to read-all or write-all.	
The user points the IR port of the Client device at the IR port of the Server device.	
The user selects the **Read-all Function or Write-all Function** to read-all or write-all the selected objects. It is recommended that a progress bar show the progress of the operation.	
	It is possible that the user may be asked to accept or reject the object.
It is recommended that the user be notified of the result of the operation.	

3.4 Application Layer

3.4.1 Feature Overview

A device following this profile must be a Client, Server or both and must support one of the content types listed in the next section.

3.4.2 Content Types

To achieve application level interoperability, content formats are defined for Tiny Object Exchange. The content formats are dependent on key application classes.

3.4.2.1 Single Push Object

The following is the content types used in the Single Push.

3.4.2.1.1 Time Log Object

Time log applications must support data exchange using the bWatch 1.0 content format specified in [IrWW].

Some wristwatches can measure not only time but also a log of various kinds of sensing information. Following are examples of advanced watches.
Running Wristwatch:
This wristwatch is for a marathon runner, and it can record Split time with the measurement point.
Trekking Wristwatch:
This wristwatch is for an alpinist and a hiker at including the altitude and temperature sensor, and it can record altitude and a temperature every constant automatically.

Diving Wristwatch:
This wristwatch is for a diver at including the pressure and temperature sensor, and it can record depth and a temperature every constant automatically.
Pulse Wristwatch:
This wristwatch is for training at including the pulse sensor, and it can record change of pulse in moving automatically.

Information captured and stored in the watches should be transferred to other device in which these data can be edited or printed out. Therefore, these informations are treated as bWatch object named Time Log (T-log).

3.4.2.1.1.1 T-Log object Definition

Figure 2 shows the schematic of T-log object structure. Main part of the T-log object is a set of records that consist of time and measurement data. There are varieties of measurement data existing in accordance with difference of watch types. And it is irrelevant to assign binary code to each measurement data. Therefore, special expression rule for the measurement data, that is independent of type or numbers of data, is required. To cope with this requirement, RECORD property is introduced. In the RECORD property, time data and measurement data are placed as values. But there is no definition for the measured data in the record property. To define values of RECORD property, DATA property is defined. Since usage of both DATA and RECORD properties are different from common properties, Block structure is introduced to distinguish these properties from common ones.

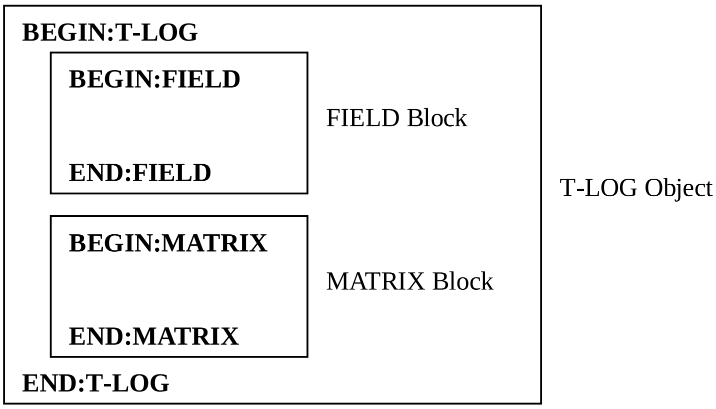


Figure 2 T-LOG object structure

There are two Blocks prepared; FIELD Block for DATA property and MATRIX Block for RECORD property. Each Block starts with "BEGIN: BLOCK" identifier and ends with "END: BLOCK" identifier. Both identifiers belong to fixed-common-property type. Table 2 shows BLOCK Identifier expression rule.

IrWW Block Identifier	PIC + SOV	VALUE VFIX+VL+0xff+SID+OID	EOB
BEGIN : FIELD	0x08 0x01	0x20 0x03 0xff 0x80 0x81	0x00
END : FIELD	0x09 0x01	0x20 0x03 0xff 0x80 0x81	0x00
BEGIN : MATRIX	0x08 0x01	0x20 0x03 0xff 0x80 0x82	0x00
END : MATRIX	0x09 0x01	0x20 0x03 0xff 0x80 0x82	0x00

Table 2 Block Identifier expression rule

FIELD Block

In FIELD Block, property stands in order of item of record data between BEGIN:FIELD and END:FIELD.

Name, unit and format of item of record data are described in this property. One property corresponds to one item of record data.

FIELD Block Property

The FIELD Block Property defined as:

```
  <Field-Property> ::= <Property-name>
      { “;” <Property-Parameter> <Attribute> } [ “:=” <Value-Format> ]
      <CRLF>
```

In the Table 3, properties that are used in FIELD Block are listed.

Property		Parameters		Value Format
Property Name	PIC	Parameter Name	PPPn+ PPC	
		Attribute Name	AC	
TIME **PTIME**	0x20 0x21	**TYPE=** (mandatory)	0x10 0x30	NONE
		VTHM **VTHMS** **VTHMSL** **VTRHM** **VTRMS** **VTRL**	0x01 0x02 0x03 0x04 0x05 0x06	
		ITEMNAME= (option)	0x10 0x40	
		String Data	0xff + AL + AV	
DATE **PDATE**	0x28 0x29	**TYPE=** (mandatory)	0x10 x30	NONE
		VDYMD **VDYMDW**	0x08 0x09	
		ITEMNAME= (option)	0x10 0x40	
		String Data	0xff + AL + AV	
UTCOFFSET	0x24			NONE
DATETIME	0x2a	**TYPE=** (mandatory)	0x10 0x30	NONE

		VTDL(LOCAL)
		VTDU(UTC)
			0x0d
			0x0e	

		ITEMNAME= (option)	0x10 0x40	
		String Data	0xff + AL + AV	
DATA	0x5C	**TYPE=** (mandatory)	0x10 0x30	VBSn(0x3x)+VL+VD
		NUMERIC **STRING**	0x01 0x02	
		ITEMNAME= (option)	0x10 0x40	
		String Data	0xff + AL + AV	
		UNIT= (option)	0x10 0x41	
		String Data	0xff + AL + AV	

Table 3 Properties used in the FIELD BLOCK

TIME / DATE / DATETIME Property

TIME / DATE / DATETIME Property is Property which means a time / a date / a date and a time.
There are TYPE= and ITEMNAME= in Parameter. For representing format of an item the TYPE= is used, and for representing a name of an item the ITEMNAME= is used.
So that TYPE= represents format, the property does not have Value Format.

Example:

The item represents the date, which kept a record.

```
DATE;TYPE=VDYMD; ITEMNAME=record date
```

PTIME / PDATE Property

PTIME / PDATE Property is Property which means a period of the time / a period of the date.
There are TYPE= and ITEMNAME= in Parameter.
So that TYPE= represents format, the property does not have Value Format.

Example:

The item represents time from measurement beginning.

```
PTIME;TYPE=VTHM;ITEMNAME=split time
```

UTCOFFSET Property

UTCOFFSET Property is Property which means an offset from UTC to local time.

There are no parameters.

DATA Property

For defining name, unit and format of item, which cannot represent in above Property, DATA Property is used.
There are TYPE=, ITEMNAME= and UNIT= in Parameter. For representing a unit of an item in character string, UNIT= is used.
Value Format is same as PICTURE format of COBOL language in character string.

Example:

The item represents a temperature (degrees centigrade).

```
DATA;TYPE=NUMERIC;ITEMNAME= temperature;UNIT= deg. C.;:=S999.99
```

MATRIX Block

In MATRIX Block, RECORD property stands in a recorded order from BEGIN:MATRIX to END:MATRIX.

RECORD property

Several measurement data that were recorded at the same time, and the time or period of the measurement are put in one record.

The RECORD Property defined as:

```
<Record-Property> ::= “RECORD” “:” <Value> { “;” <Value> } <CRLF>
```

RECORD Property Identification code is set to 0x5D as shown in Table 4.

Property Name	Property Identification Code
RECORD	0x5D

Table 4 RECORD Property Identification Code

Types of values that are expected to appear in the RECORD property are listed in the Table 5.

TYPE=	Value
VTHM VTHMS VTHMSL VTRHM VTRMS VTRL	VTHM + VL + HH;MM VTHMS + VL + HH;MM;SS VTHMSL + VL + HH;MM;SS;LLLLLL VTRHM + VL + HHHH;MM VTRMS + VL + MMMM;SS VTRL + VL + LLLLLL
VDYMD VDYMDW	VDYMD + VL + YYYY;MM;DD VDYMDW + VL +YYYY;MM;DD;WW

VTUTC	**VTUTC + VL +(S)HH;MM**
VTDL **VTDU**	**VTDL + VL + YYYY;MM;DD;HH;MM;SS** **VDYMDW + VL +YYYY;MM;DD;WW;HH;MM;SS**
NUMERIC **STRING**	**VNUM + VL + BCDs** **VBS + VL + Byte Stream**

Table 5 The Value list of RECORD Property

For example in the marathon race, time data is recorded every 5km or 10km interval.
In this case, distance from start point and split times are defined as values in the RECORD property.
Expression examples of both FIELD block and MATRIX block are shown below.

```
BEGIN:FIELD
  DATA;TYPE=NUMERIC;INTEMNAME=distance;UNIT=Km:=99.999
  PTIME;TYPE=VTHMS;ITEMNAME=split time
END:FILED
BEGIN:MATRIX
  RECORD:00.000; 00:00:00
  RECORD:10.000; 00:28:01
  RECORD:20.000; 00:57:09
  RECORD:30.000; 01:28:31
  RECORD:40.000; 01:59:06
  RECORD:42.195; 02:10:55
END:MATRIX
```

3.4.2.1.1.2 T-Log object example

Following is an example of single T-log object of text format.

```
BEGIN:BWATCH
  VERSION:1.0                                   ; The Version is 1.0.
  BEGIN:T-LOG
    TITLE:Trekking                              ; Title is "Trekking".
    NOTE:max-altitude:1125.23                   ; "Max altitude is 1125.23" as a note.
    BEGIN:FIELD                                 ; Each record has following 3 fields in order.
      DATETIME;TYPE=VTDU:                       ; DATETIME, which type is VTDU.
      DATA;TYPE=NUMERIC;ITEMNAME=temperature;UNIT=deg.C.:=S999.99
                               ; DATA, which type is NUMERIC, ITEMNAME is temperature,
                                 UNIT is deg.C. and format is S999.99.
      DATA;TYPE=NUMERIC;INTEMNAME=altitude;UNIT=m:=9999.99
                               ; DATA, which type is NUMERIC, ITEMNAME is altitude,
                                 UNIT is m and format is 9999.99.
    END:FIELD
    BEGIN:MATRIX                        ; 2 records are listed.
      RECORD:199905240608;1.05;1090.01
                               ; DATETIME is 199905240608, temperature is 1.05 deg.C.
                                 and altitude is 1090.01 m.
      RECORD:199905240815;-2.60;1125.23
                               ; DATETIME is 199905240815, temperature is -2.60 deg.C.
                                 and altitude is 1125.23 m.
    END:MATRIX
  END:T-LOG
END:BWATCH
```

And the object is translated into binary format. In Table 6, text format and corresponding bWatch format are listed.

Time Log Object	Binary Time Log Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:BWATCH	0x08		0x01	0x20 0x01 0x80	0x00
VERSION:1.0	0x0A		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:T-LOG	0x08		0x01	0x20 0x03 0xff 0x80 0x05	0x00
TITLE:Trekking	0x30		0x01	0x30 0x08 'Trekking'	0x00
NOTE:max-altitude:1125.23	0x33		0x01	0x30 0x14 'max-altitude:1125.23'	0x00
BEGIN:FIELD	0x08		0x01	0x20 0x03 0xff 0x80 0x81	0x00
DATETIME;TYPE=LOCAL	0x2a	0x10 0x30 0x0D	0x01		0x00
DATA;TYPE=NUMERIC ;ITEMNAME= temperature; UNIT= deg.C. :=S999.99	0x5c	0x10 0x30 0x01 0x10 0x40 0xFF 0x0B 'temperature' 0x10 0x41 0xFF 0x06 'deg.C.'	0x01	0x30 0x07 'S999.99'	0x00
DATA;TYPE=NUMERIC ;ITEMNAME=altitude ; UNIT=m :=9999.99	0x5c	0x10 0x30 0x01 0x10 0x40 0xFF 0x08 'altitude' 0x10 0x41 0xFF 0x01 'm'	0x01	0x30 0x07 '9999.99'	0x00
END:FIELD	0x09		0x01	0x20 0x03 0xff 0x80 0x81	0x00
BEGIN:MATRIX	0x08		0x01	0x20 0x03 0xff 0x80 0x82	0x00
RECORD:199905240608 ;1.05;1090.01	0x5D		0x01	0x8D 0x06 0x199905240608 0x40 0x02 0x1A05 0x40 0x04 0x1090A01F	0x00
RECORD:199905240815 ;-2.6;1125.23	0x5D		0x01	0x8D 0x06 0x199905240815 0x40 0x02 0xE2A6 0x40 0x04 0x1125A23F	0x00
END:MATRIX	0x09		0x01	0x20 0x03 0xff 0x80 0x82	0x00
END:T-LOG	0x09		0x01	0x20 0x03 0xff 0x80 0x05	0x00
END:BWATCH	0x01		0x01	0x20 0x01 0x80	0x00

Table 6 Example of Time Log Object

3.4.2.1.2 Business Card Object

Business Card applications must support data exchange using the Binary vCard 2.1 content format specified in [IrWW], Chapters 4 of this specification, and [VCARD].

Following is an example of single Business Card object of vCard format.

BEGIN:VCARD
VERSION:2.1 ; The Version is 2.1.
N:Kenichiro Shimokura ; The Name is Kenichiro Shimokura.
TEL;PREF;WORK;VOICE:+81-3-3795-7601 ; The Preference Work Voice
Phone Number is +81-3-3795-7601.
EMAIL:simokura@mbd.mbc.ntt.co.jp ; The Email is simokura@mbd.mbc.ntt.co.jp.
END:VCARD

VERSION, **N**, **TEL** and **EMAIL** are assumed to be a minimum set of single Business Card object. So, these four are defined **mandatory**.
And the object is translated into binary format. In Table 7, vCard format and corresponding Binary vCard format are listed.

vCard Object	Binary vCard Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:VCARD	0x08		0x01	0x20 0x01 0x01	0x00
VERSION:2.1	0x0A		0x01	0x40 0x02 0x2a1f	0x00
N:Kenichiro Shimokura	0x21		0x01	0x30 0x13 'Kenichiro Shimokura'	0x00
TEL;PREF;WORK;VOICE:+81-3-3795-7601	0x30	0x13 0x01 0x02 0x04	0x01	0x40 0x08 0xd81e3e3795e7601f	0x00

EMAIL:simokura@mbd.mbc.ntt.co.jp	0x31		0x01	0x30 0x1a ‘simokura@mbd.mbc.ntt.co.jp’	0x00
END:VCARD	0x09		0x01	0x20 0x01 0x01	0x00

Table 7 Example of Business Card Object

3.4.2.1.3 Schedule Object

Schedule applications must support data exchange using VEVENT component of the Binary vCalendar 1.0 content format specified in [IrWW], Chapters 4 of this specification, and [VCAL].

Following is an example of single Schedule object of vCalendar format.

```
BEGIN:VCALENDAR
  VERSION:1.0                          ; The Version is 1.0.
  BEGIN:VEVENT
    DTSTART:19990827T130000            ; The Start Date/Time is 1999/8/27 13:00 00.
    DTEND:19990827T170000              ; The End Date/Time is 1999/8/27 17:00 00.
    DESCRIPTION:IrWW SIG meeting       ; The Description is 'IrWW SIG meeting'.
  END:VEVENT
END:VCALENDAR
```

Single Schedule object must contain only one VEVENT component. **VERSION**, **DTSTART**, **DTEND** and **DESCRIPTION** are assumed to be a minimum set of single Schedule object. So, these four are defined **mandatory**.

And the object is translated into binary format. In Table 8, vCalendar format and corresponding Binary vCalendar format are listed.

vCalendar Object	Binary vCalendar Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:VCALENDAR	0x08		0x01	0x02	0x00
VERSION:1.0	0x0A		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:VEVENT	0x08		0x01	0xff 0x02 0x01	0x00
DTSTART:19990827T130000	0x20		0x01	0x8d 0x07 0x19990827130000	0x00
DTEND:19990827T170000	0x22		0x01	0x8d 0x07 0x 19990827170000	0x00
DESCRIPTION:IrWW SIG meeting	0x41		0x01	0x30 0x10 ‘IrWW SIG meeting’	0x00
END:VEVENT	0x09		0x01	0xff 0x02 0x01	0x00
END:VCALENDAR	0x09		0x01	0x02	0x00

Table 8 Example of Schedule Object

3.4.2.1.4 To do Object

To do applications must support data exchange using VTODO component of the Binary vCalendar 1.0 content format specified in [IrWW], Chapters 4 of this specification, and [VCAL].

Following is an example of single To do object of vCalendar format.

```
BEGIN:VCALENDAR
  VERSION:1.0                          ; The Version is 1.0.
  BEGIN:VTODO
    DUE:19990831T150000                ; The Due Date/Time is 1999/8/31 15:00 00.
    STATUS:NEEDS ACTION                ; The Status is NEEDS ACTION.
    DESCRIPTION:pay the rent.          ; The Description is 'pay the rent.'.
  END:VTODO
END:VCALENDAR
```

Single To do object must contain only one VTODO component. **VERSION**, **DUE**, **STATUS** and **DESCRIPTION** are assumed to be a minimum set of single To do object. So, these four are defined **mandatory**.

And the object is translated into binary format. In Table 9, vCalendar format and corresponding Binary vCalendar format are listed.

vCalendar Object	Binary vCalendar Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:VCALENDAR	0x08		0x01	0x02	0x00
VERSION:1.0	0x0A		0x01	0x40 0x02 0x1a0f	0x00
BEGIN:VTODO	0x08		0x01	0xff 0x02 0x02	0x00
DUE:19990831T150000	0x21		0x01	0x8d 0x07 0x19990831150000	0x00
STATUS:NEEDS ACTION	0x50		0x01	0x01	0x00
DESCRIPTION:pay the rent.	0x41		0x01	0x30 0x0D ‘pay the rent.’	0x00
END:VEVENT	0x09		0x01	0xff 0x02 0x01	0x00
END:VCALENDAR	0x09		0x01	0x02	0x00

Table 9 Example of Todo Object

3.4.2.1.5 Notes Object

Notes applications must support data exchange using the Binary vNote content format specified in [IrWW], Chapters 4 of this specification, and Chapter 10 of [IrMC].

Following is an example of single Notes object of vNote format.

BEGIN:VNOTE
VERSION:1.1 ; The Version is 1.1.
BODY:The last train of Hamura station is 23:30. ; The Description is
'The last train of Hamura station is 23:30'.
END:VNOTE

VERSION and **BODY** are assumed to be a minimum set of single Notes object. So, these two are defined **mandatory**.

And the object is translated into GBO format. In Table 10, vNote format and corresponding Binary vNote format are listed.

vNote Object	Binary vNote Object				
	PIC	Parameter	SOV	Value	EOP
BEGIN:VNOTE	0x08		0x01	0x04	0x00
VERSION:1.0	0x0a		0x01	0x40 0x02 0x1a0f	0x00
BODY:The last train of Hamura station is 23:30.	0x21		0x01	0x30 0x2A ‘The last train of Hamura station is 23:30.’	0x00
END:VNOTE	0x09		0x01	0x04	0x00

Table 10 Example of Notes Object

3.4.2.2 Read-all/Write-all Object

The following is the content types used in the Read-all/Write-all.

3.4.2.2.1 Device Information

The device information must be supported by all devices that support r Read-all/Write-all.
Device information follows the information described in Section 2.5 of [GBO].

3.4.2.2.2 Time Log Object

Time Log applications must support data exchange using the bWatch 1.0 content format specified in [IrWW] and Section 3.4.2.1.1.

Following is an example of Time log objects of bWatch format.

```
BEGIN:BWATCH
  VERSION:1.0
  BEGIN:T-LOG
          ⋮
  END:T-LOG
  BEGIN:T-LOG
          ⋮
  END:T-LOG
                  ⋮
  BEGIN:T-LOG
          ⋮
  END:T-LOG
END:BWATCH
```

Table 11 Example of Time log objects of bWatch format.

3.4.2.2.3 Time Log's Information Log Object

Time Log applications must support data exchange using the bWatch 1.0 Information log object content format specified in [IrWW].

The Client can send a request to get Information Log Object that contains information of properties existing in the Time Log Object. When the client device request GET /watch/t-log/info.log file, the server should respond to the request and send Information Object to the client.

There is a special writing manner in T-log's Information Log.
X-IRWW-Fields of T-log's Information Log is the following.

X-IRWW-Fields:
<Begin>
{ <x- irww property> }
BEGIN:FIELD
Any properties defined in the field block
END:FIELD
BEGIN:MATRIX
X-RECORD-FIELDS: Numeric
END:MATRIX
<End>

Any properties defined in the T-log's field block are described between **BEGIN:FIELD** and **END:FIELD.**
X-RECORD-FIELDS indicates maximum number of record in the T-log's matrix block.
An example of Time Log’s Information Log Object is as follows;

```
IEL:2                    ; The device supports Read-all/Write-all.
X-IRWW-FIELDS:
<Begin>
Version:                     ; The device supports a Version Number field.
TITLE:=20                    ; The device supports TITLE, which maximum size is 20 bytes.
NOTE:=30                     ; The device supports NOTE, which maximum size is 30 bytes.
BEGIN:FIELD
DATETIME;TYPE=VTDU            ; The device supports Local DATETIME type VTDU
                                (YYYY;MM;DD;HH;MM;SS).
DATA;TYPE=NUMERIC;ITEMNAME=temperature;UNIT=deg.C.:=S999.99
                              ; The device supports DATA, which type is numeric,
                                ITEMNAME is "temperature" and UNIT is "deg.C".
DATA;TYPE=NUMERIC;ITEMNAME=altitude; UNIT=m :=9999.99
                              ; The device supports DATA which type is numeric,
                                ITEMNAME is "altitude" and UNIT is "m".
END:FIELD
BEGIN:MATRIX
X-RECORD-FIELDS:10           ; The device supports up to 10 RECORDs.
END:MATRIX
<END>
Total-Records:2              ; The Total Number of T-log object is 2.
Maximum-Records:100          ; The Maximum Number of T-log objects that can be stored is 100.
```

And the object is translated into binary format. In Table 12, Time Log’s Information Log Object and corresponding bWatch's Information Log Object are listed.

Time Log’s Information Log Object	Binary Time Log’s Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x40 0x01 0x2f	0x00
X-IRWW-FIELDS:	0x80		0x01		0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
TITLE:=20	0x30		0x01	0x40 0x01 0x20	0x00
NOTE:=30	0x33		0x01	0x40 0x01 0x30	0x00
BEGIN:FIELD	0x08		0x01	0x20 0x03 0xff 0x80 0x81	0x00
DATETIME;TYPE=VTDU	0x2a	0x10 0x30 0x0E			0x00
DATA;TYPE=NUMERIC ;ITEMNAME= temperature; UNIT= deg.C. :=S999.99	0x5C	0x10 0x30 0x01 0x10 0x40 0xFF 0x0B ‘temperature’ 0x10 0x41 0xFF 0x06 ‘deg.C.’	0x01	0x30 0x07 ‘S999.99’	0x00
DATA;TYPE=NUMERIC ;ITEMNAME=altitude ; UNIT=m :=9999.99	0x5C	0x10 0x30 0x01 0x10 0x40 0xFF 0x08 ‘altitude’ 0x10 0x41 0xFF 0x01 ‘m’	0x01	0x30 0x07 ‘9999.99’	0x00
END:FIELD	0x09		0x01	0x20 0x03 0xff 0x80 0x81	0x00
BEGIN:MATRIX	0x08		0x01	0x20 0x03 0xff 0x80 0x82	0x00
X-RECORD-FIELDS:10	0xDD		0x01	0x40 0x01 0x10	0x00
END:MATRIX	0x09		0x01	0x20 0x03 0xff 0x80 0x82	0x00
<END>	0x89				0x00
Total-Records:2	0xF4		0x01	0x40 0x01 0x2f	0x00
Maximum-Records:100	0xF6		0x01	0x40 0x02 0x100f	0x00

Table 12 Example of Time Log’s Information Log Object

3.4.2.2.4 Owner's Business Card Object

Business Card applications must support data exchange using the Binary vCard 2.1 content format specified in [IrWW], Section 3.4.2.1.2, and [VCARD].

3.4.2.2.5 Business Card Object

Business Card applications must support data exchange using the Binary vCard 2.1 content format specified in [IrWW], Section 3.4.2.1.2, and [VCARD].

Following is an example of Business Card objects of vCard format.

```
BEGIN:VCARD
   VERSION:2.1
   N:Name 1
   TEL; VOICE: Phone number 1
   EMAIL:Email 1
END:VCARD

BEGIN:VCARD
   VERSION:2.1
   N:Name 2
   TEL;VOICE: Phone number 2
   EMAIL:Email 2
END:VCARD

                          ⋮

BEGIN:VCARD
   VERSION:2.1
   N:Name N
   TEL;VOICE:Phone number N
   EMAIL:Email N
END:VCARD
```

Table 13 Example of Business Card objects of vCard format.

3.4.2.2.6 Business Card's Information Log Object

Business Card applications must support data exchange using the Binary Business Card's Information log object content format specified in [IrWW], Section 3.4.2.1.2, and [IrMC].

The Client can send a request to get Information Log Object that contains information of properties existing in the Business Card Object. When the client device request GET /watch/pim/bc/info.log file, the server should respond to the request and send Information Object to the client. An example of Business Card's Information Log Object is as follows;

IEL:2	; The device supports Read-all/Write-all.
X-IRWW-FIELDS:	
<Begin>	
Version:	; The device supports a Version Number field.
N:	; The device supports N.
X-TEL-FIELDS:4	; The device supports up to four Phone numbers.
TEL:TYPE=WORK;VOICE	; The device supports Work Voice Phone Number.
TEL:TYPE=WORK;FAX	; The device supports Work Fax Phone Number.
TEL:TYPE=HOME;VOICE	; The device supports Home Voice Phone Number.
TEL:TYPE=HOME;FAX	; The device supports Home Fax Phone Number.
EMAIL:	; The device supports EMAIL.
<End>	
Total-Records:2	; The Total Number of Records in the Business Card is 2.
Maximum-Records:30	; The Maximum Number of Records that can be stored is 30.

And the object is translated into binary format. In Table 14, Business Card's Information Log Object and corresponding Binary Business Card's Information Log Object are listed.

Business Card's Information Log	Binary Business Card's Information Log Object

Object	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x40 0x01 0x2f	0x00
X-IRWW-FIELDS:	0x80		0x01		0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
N:	0x21		0x01		0x00
X-TEL-FIELDS:4	0xb0		0x01	0x40 0x01 0x3f	0x00
TEL;TYPE=WORK;VOICE:	0x30	0x11 0x30 0x02 0x30 0x04	0x01		0x00
TEL;TYPE=WORK;FAX:	0x30	0x11 0x30 0x02 0x30 0x05	0x01		0x00
TEL;TYPE=HOME;VOICE:	0x30	0x11 0x30 0x03 0x30 0x04	0x01		0x00
TEL;TYPE=HOME;FAX:	0x30	0x11 0x30 0x03 0x30 0x05	0x01		0x00
EMAIL:	0x31		0x01		0x00
<END>	0x89				0x00
Total-Records:2	0xf4		0x01	0x40 0x01 0x2f	0x00
Maximum-Records:30	0xf6		0x01	0x40 0x01 0x30	0x00

Table 14 Example of Business Card's Information Log Object

3.4.2.2.7 Schedule Object

Schedule applications must support data exchange using VEVENT component of the Binary vCalendar 1.0 content format specified in [IrWW], Section 3.4.2.1.3, and [VCAL].

Following is an example of Schedule objects of vCalendar format.

```
BEGIN:VCALENDAR
  VERSION:1.0
  BEGIN:VEVENT
   DTSTART:Start Date/Time 1
   DTEND:End Date/Time 1
   DESCRIPTION:Description 1
  END:VEVENT
  BEGIN:VEVENT
   DTSTART:Start Date/Time 2
   DTEND:End Date/Time 2
   DESCRIPTION:Description 2
  END:VEVENT
                 ⋮
  BEGIN:VEVENT
   DTSTART: Start Date/Time N
   DTEND: End Date/Time N
   DESCRIPTION:Description N
  END:VEVENT
END:VCALENDAR
```

Table 15 Example of Schedule objects of vCalendar format

3.4.2.2.8 Schedule's Information Log Object

Schedule applications must support data exchange using the Binary Calendar's Information log object content format specified in [IrWW], Section 3.4.2.1.3, and [IrMC].

The Client can send a request to get Information Log Object that contains information of properties existing in the Schedule Object. When the client device request GET /watch/pim/sd/info.log file, the server should respond to the request and send Information Object to the client. An example of Schedule's Information Log Object is as follows;

IEL:2	; The device supports Read-all/Write-all.
X-IRWW-FIELDS:	
<Begin>	
Version:	; The device supports a Version Number field.
DTSTART:	; The device supports DTSTART.
DTEND:	; The device supports DTEND.
DESCRIPTION:	; The device supports DESCRIPTION.
<End>	
Total-Records:2	; The Total Number of Records in the Schedule is 2.
Maximum-Records:30	; The Maximum Number of Records that can be stored is 30.

And the object is translated into binary format. In Table 16, Schedule's Information Log Object and corresponding Binary Schedule's Information Log Object are listed.

Schedule's Information Log Object	Binary Schedule's Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x40 0x01 0x2f	0x00
X-IRWW-FIELDS:	0x80		0x01		0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
DTSTART:	0x20		0x01		0x00
DTEND:	0x22		0x01		0x00
DESCRIPTION:	0x41		0x01		0x00
<END>	0x89				0x00
Total-Records:2	0xf4		0x01	0x40 0x01 0x2f	0x00
Maximum-Records:30	0xf6		0x01	0x40 0x01 0x30	0x00

Table 16 Example of Schedule Card's Information Log Object

3.4.2.2.9 To do Object

To do applications must support data exchange using TODO component of the Binary vCalendar 1.0 content format specified in [IrWW], Section 3.4.2.1.4, and [VCAL].

Following is an example of To do objects of vCalendar format.

```
BEGIN:VCALENDAR
  VERSION:1.0
  BEGIN:VTODO
   DUE:Due Date/Time 1
   STATUS:Status 1
   DESCRIPTION:Description 1
  END:VTODO

  BEGIN:VTODO
   DUE:Due Date/Time 2
   STATUS:Status 2
   DESCRIPTION:Description 2
  END:VTODO
                  :
  BEGIN:VTODO
   DUE:Due Date/Time N
   STATUS:Status N
   DESCRIPTION:Description N
  END:VTODO
END:VCALENDAR
```

Table 17 Example of To do objects of vCalendar format

3.4.2.2.10 To do's Information Log Object

To do applications must support data exchange using the Binary Calendar's Information log object content format specified in [IrWW], Section 3.4.2.1.4, and [IrMC].

The Client can send a request to get Information Log Object that contains information of properties existing in the To do Object. When the client device request GET /watch/pim/td/info.log file, the server should respond to the request and send Information Object to the client. An example of To do's Information Log Object is as follows;

IEL:2	; The device supports Read-all/Write-all.
X-IRWW-FIELDS:	
<Begin>	
Version:	; The device supports a Version Number field.
DUE:	; The device supports DUE.
STATUS:	; The device supports STATUS.
DESCRIPTION:	; The device supports DESCRIPTION.
<End>	
Total-Records:2	; The Total Number of Records in the To do is 2.
Maximum-Records:30	; The Maximum Number of Records that can be stored is 30.

And the object is translated into binary format. In Table 18, Schedule's Information Log Object and corresponding Binary To do's Information Log Object are listed.

To do's Information Log Object	Binary To do's Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x40 0x01 0x2f	0x00
X-IRWW-FIELDS:	0x80		0x01		0x00
<BEGIN>	0x88				0x00
VERSION:	0x0a		0x01		0x00
DUE:	0x21		0x01		0x00
STATUS:	0x50		0x01		0x00
DESCRIPTION:	0x41		0x01		0x00
<END>	0x89				0x00
Total-Records:2	0xf4		0x01	0x40 0x01 0x2f	0x00
Maximum-Records:30	0xf6		0x01	0x40 0x01 0x30	0x00

Table 18 Example of To do's Information Log Object

3.4.2.2.11 Notes Object

Notes applications must support data exchange using the Binary vNote content format specified in [IrWW], Section 3.4.2.1.5, and [IrMC].

Following is an example of Notes objects of vNote format.

```
BEGIN:VNOTE
  VERSION:1.1
  BODY:Text 1
END: VNOTE
```

```
BEGIN:VNOTE
  VERSION:1.1
  BODY:Text 2
END: VNOTE
```

⋮

```
BEGIN:VNOTE
  VERSION:1.1
  BODY:Text N
END: VNOTE
```

Table 19 Example of Notes objects of vNote format

3.4.2.2.12 Notes Information Log Object

Notes applications must support data exchange using the Binary Notes Information log object content format specified in [IrWW], Section 3.4.2.1.5, and [IrMC].

The Client can send a request to get Information Log Object that contains information of properties existing in the Note Object. When the client device request GET /watch/pim/nt/info.log file, the server should respond to the request and send Information Object to the client. An example of Notes Information Log Object is as follows;

IEL:2	; The device supports Read-all/Write-all.
X-IRWW-FIELDS:	
<Begin>	
Version:	; The device supports a Version Number field.
BODY:	; The device supports BODY.
<End>	
Total-Records:2	; The Total Number of Records in the To do is 2.
Maximum-Records:30	; The Maximum Number of Records that can be stored is 30.

And the object is translated into binary format. In Table 20, Notes Information Log Object and corresponding Binary Notes Information Log Object are listed.

Notes Information Log Object	Binary Notes Information Log Object				
	PIC	Parameter	SOV	Value	EOP
IEL:2	0xf0		0x01	0x40 0x01 0x2f	0x00
X-IRWW-FIELDS:	0x80		0x01		0x00
<BEGIN>	0x88				0x00
VERSION:	0x0A		0x01		0x00
BODY:	0x21		0x01		0x00
<END>	0x89				0x00
Total-Records:2	0xf4		0x01	0x40 0x01 0x2f	0x00
Maximum-Records:30	0xf6		0x01	0x40 0x01 0x30	0x00

Table 20 Example of Notes Information Log Object

3.4.3 Application Procedure

Both Client and Server are built on top of the OBEX application framework.

3.4.3.1 Application procedure for Single Push object

A Client uses OBEX_PUT to push single object to the inbox of a Server.

Table 21 shows an example of the application procedure required by the Client to push an object to a Server.

Client	**Details**
OBEX CONNECT.	
Push the object using OBEX GET. OBEX DISCONNECT.	

Table 21 Application procedure for Push object

In Tiny Object Exchange, following file names which distinguish from Tiny Object Exchange are used.

Object	**File name**
Time Log Object	xxx.wlg
Business Card Object	xxx.bcf
Schedule Object	xxx.bet
To do Object	xxx.btd
Notes Object	xxx.bnt

Note "xxx" is a user defined name.

Table 22 File name assignment rule of Tiny Object Exchange

3.4.3.2 Application procedure for Read-all objects

A Client uses OBEX PUT to write-all objects to the store of a Server. And a Client uses OBEX GET to read-all objects from the store of a Server. Transferring files requires a single PUT or GET operation per file. The Client connects to the Server's storage and transfers all objects. Note that inbox isn't used in Read-all/Write-all objects.

Information Log contains general information about specific Object Stores. Before Client read or write Tiny Object from or to Server, It becomes help of processing of information by reading Information Log.

Table 23 shows an example of the application procedure required by the Client to read all objects from a Server.

Client	**Details**
OBEX CONNECT.	
Read the Information Log using OBEX GET.	In accordance with read Information Log, memory is prepared in which the data are stored.
Read all objects using OBEX GET.	
OBEX DISCONNECT.	

Table 23 Application procedure for Read-all objects

It is not necessary for a client to always use Information Log.

Table 24 shows an example of the application procedure required by the Client to write all objects to a Server.

Client	**Details**
OBEX CONNECT.	
Read the Information Log using OBEX GET.	
Write all objects using OBEX PUT.	In accordance with read Information Log, only necessary data are sent.
OBEX DISCONNECT.	

Table 24 Application procedure for Write-all objects

It is not necessary for a client to always use Information Log.

In Tiny Object Exchange, following file names which distinguish from Tiny Object Exchange are used.

Object	**File name**
Device Information	/watch/devinfo.bin
Time Log object (common log)	/watch/t-log.wlg
Time Log Information Log object (common log)	/watch/t-log/info.log
Time Log object (vender log)	/watch/vlog/xxx/yyy.wlg
Time Log's Information Log object (vender log)	/watch/vlog/xxx/yyy/info.log
Owner's Business Card Object	/watch/pim/ob.bcf
Business Card Object	/watch/pim/bc.bcf
Business Card’s Information Log Object	/watch/pim/bc/info.log
Schedule Object	/watch/pim/sd.bet
Schedule’s Information Log Object	/watch/pim/sd/info.log
To do Object	/watch/pim/td.bdd
To do’s Information Log Object	/watch/pim/td/info.log
Notes Object	/watch/pim/nt.bnt
Notes Information Log Object	/watch/pim/nt/info.log

Note "xxx" is a vender name. "yyy" is a log name.

Table 25 File name assignment rule of Tiny Object Exchange

If there are different kinds of logs existing in the watch, only one type of log is registered as Common Log. Owner should select and change log type registered in the Common log field. Common Log is necessary for inter-vender data exchange. But in inter-vender communication, it is possible to access directly the specific log and extract it. To allow this direct access, Vender log object is defined. File name assignment rule example for Vender object is shown in the Table 26.

stream-log-object-name	Information-log-name
watch/vlog/vender/dive.wlg	/watch/vlog/vender/dive/info.log
/watch/vlog/vender/altitude.wlg	/watch/vlog/vender/altitude/info.log

Table 26 Vender object file name assignment rule example

3.5 OBEX

3.5.1 OBEX Operations

OBEX operations follows the information described in [IrWW].

The table below shows the OBEX operations, which are used in the Tiny Object Exchange profile.

Operation no.	OBEX Operation	Client	Server
1	Connect	M	M
2	Disconnect	O	M
3	Put	M	M
4	Get	N/A in case of Single push support. M in case of Read-all / Write-all Support.	N/A in case of Single push support. M in case of Read-all / Write-all Support.
5	Abort	M	M
6	SetPath	N/A	N/A

Table 12 OBEX Operations

3.5.2 OBEX Headers

OBEX Headers follows the information described in [IrWW].

3.5.3 Establishing an OBEX session

Establishing an OBEX session follows the procedure described in [IrWW].

3.5.4 Pushing Object

Pushing Object follows the procedure described in [IrWW].

3.5.5 Reading Objects

Reading Object follows the procedure described in [IrWW].

3.5.6 Writing Objects

Writing Object follows the procedure described in [IrWW].

3.5.7 Disconnecting an OBEX session

Disconnecting an OBEX session follows the procedure described in [IrWW].

3.6 Tiny TP

Tiny TP follows the information described in [IrWW].

3.7 IrLMP

IrLMP follows the information described in [IrWW].

3.8 IAS

IAS follows the information described in [IrWW].

In Table 27, the Parameters of Tiny Object is listed.

PI	PI name	PL	PV data type	PV Description	Status
0x04	T-log Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Single Pull Support	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x05	Business Card Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Undefined	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x06	Schedule Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Undefined	0

			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x07	To Do Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Undefined	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	
				0x00 is undefined version	
0x08	Notes Support	2		The first octet =	
				Support Level	
				(bit mask)	
			bit 0	Single Push Support	0 or 1
			bit 1	Undefined	0
			bit 2	Write-all Support	0 or 1
			bit 3	Read-all Support	0 or 1
				The second octet :	
				Upper side nibble(4bit)	0001
				Major version number	
				Lower side nibble(4bit)	0000
				Minor version number	

				0x00 is undefined version	

Table 27 The Parameters of Tiny Object

3.9 Service Hint Bit

Service Hint Bit follows the information described in [IrWW].

3.10 IrLAP

IrLAP follows the information described in [IrWW].

3.11 Physical Layer

Physical Layer follows the information described in [IrWW].

4 Appendix

4.1 Unique code assignment tables of the Binary vCard

Binary vCARD 2.1 Unique Property Code

	0x20	0x30	0x40	0x50	0x60
0x00	FN	TEL	TITLE		REV
0x01	N	EMAIL	ROLE		UID
0x02	PHOTO	MAILER	LOGO		KEY
0x03	BDAY		AGENT		
0x04			ORG		
0x05					
0x06					
0x07					
0x08	ADR	TZ	NOTE		
0x09	LABEL	GEO			
0x0A					
0x0B					
0x0C			SOUND		
0x0D			URL		
0x0E					
0x0F					
	PC for Object Field				

Assignment of Binary vCARD 2.1 Property Parameter Attribute Code (PAC)

Related Property	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10
PHOTO(0x22)	GIF	CGM	WMF	BMP	MET	PMB	DIB	PICT	TIFF	PS	PDF	JPEG	MPEG	MPEG2	AVI	QTIME
ADR(0x28)	DOM	INTL	POSTAL	PARCEL	HOME	WORK										
LABEL(0x09)	DOM	INTL	POSTAL	PARCEL	HOME	WORK										
TEL(0x30)	PREF	WORK	HOME	VOICE	FAX	MSG	CELL	PAGER	BSS	MODEM	CAR	ISDN	VIDEO			
LOGO(0x42)	GIF	CGM	WMF	BMP	MET	PMB	DIB	PICT	TIFF	PS	PDF	JPEG	MPEG	MPEG2	AVI	QTIME
EMAIL(0x31)	AOL	APPLE	ATT	CIS	EWORLD	INET	IBM	MCI	POWER	PRODI	TLX	X400	VIDEO			
SOUND(0x4C)	WAVE	PCM	AIFF													
KEY(0x60)	X509	PGP														

4.2 Unique code assignment tables of the Binary vCalendar

Binary vCalender 1.0 Unique Property Cod

	0x20	0x30	0x40	0x50	0x60
0x00	DTSTART	DALARM	SUMMARY	STATUS	
0x01	DUE		DESCRIPTION		
0x02	DTEND				
0x03					
0x04					
0x05					
0x06					
0x07					
0x08					
0x09					
0x0A					
0x0B					
0x0C					
0x0D					
0x0E					
0x0F					
	PC for Object Field				

Assignment of Binary vCalender 1.0 Property Parameter Attribute Code (PAC)

Related Property	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
STATUS(0x50)	NEEDS ACTION	ACCEPTED	SENT	TENTATIVE	CONFIRMED	DECLINED	COMPLETED	DELEGATED

4.3 Unique code assignment tables of the Binary vNote

Binary vNote 1.1 Unique Property Code

	0x20	0x30	0x40	0x50	0x60
0x00	SUMMARY				X-IRMC-LUID
0x01	BODY				DCREATED
0x02	CATEGORIES				LAST-MODIFIED
0x03	CLASS				
0x04					
0x05					
0x06					
0x07					
0x08					
0x09					
0x0A					
0x0B					
0x0C					
0x0D					
0x0E					
0x0F					
	PC for Object Field				

Assignment of Binary vNote 1.1
Property Parameter Attribute Code (PAC)

Related Property	0x01	0x02	0x03
CLASS(0x22)	PRIVATE	PUBLIC	CONFIDENTIAL

第4部

一般バイナリオブジェクト(GBO)仕様

Infrared Data Association

IrWW (Infrared Wrist Watch)
Generic Binary Object (GBO)
Specification

Version 1.0

December 10, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

IrWW SIG

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Ken-ichiro Shimokura k.shimokura@east.ntt.co.jp
Gontaro Kitazumi kitazumi@osw.co.jp

Significant Contributors:

History:

Version 0.5:	Infrared Data Association IrWW IrDA for Wrist Watches for Directional Vote
Version 0.8:	IrWW Generic Access Profile for Draft Vote.
Version 1.0:	Final vote version separated from IrWW Generic Access Profile for Draft Vote.

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 Introduction 1

1.1 Scope 1

1.2 Relationship of IrWW Profile 2

1.3 Symbols and Conventions 2

1.4 References 3

1.5 Content Format 3

2 Structure of Generic Binary Object (GBO) 4

2.1 Basic Structure of GBO 4

2.1.1 Syntax of Generic Binary Object 5

2.2 Base elements of GBO 6

2.2.1 Handling of Nested GBO Structure 6

2.2.2 Property Operation Code (POC) 7

2.2.3 Expression of Property Part, Group Part and Property Code (PC) 7

Expression of Property part 10

2.2.5 Element of Property Parameter Part 10

2.2.6 Common Property Parameter Code 11

2.2.6.1 Detail of Common Property Parameter Code 12

2.2.6.2 Detail of Property Parts and Syntax Flow 13

Basic coding rule for Value 15

Expression of fixed value 16

2.2.6.5 Expression of Byte stream value 17

2.2.6.6 Expression of Numerical value 18

2.2.6.7 Expression method of several values and value connection 19

2.3 Object Parth algorithm 21

2.4 Object Stream Identifier (OBJECT MAGIC CODE) 22

2.5 Device Information Object 22

3 IrWW Object (bWatch) 23

3.1 bWATCH Structure Identifier 23

3.2 bWATCH Object Identifier 23

3.3 Common IrWW PC (OUPC) 24

3.3.1 Expression of IrWW unique Time-related value 25

3.3.1.1 VTHM time-related value VC = 0x81 25

3.3.1.2 VHTMS time-related value VC = 0x82 26

3.3.1.3 VTHMSL Time-related value VC = 0x83 26

3.3.1.4 VTRHM Time-related value VC = 0x84 26

3.3.1.5 VTRMS Time-related value VC = 0x85 27

3.3.1.6 VTRL Time-related value VC = 0x86 27

3.3.1.7 VDYMD Time-related value VC = 0x88 27

3.3.1.8 VDYMDW Time-related value VC = 0x89 28

3.3.1.9 VTUTC Time-related value VC = 0x8c 28

3.3.1.10 VDTL Time-related value VC = 0x8D 29

3.3.1.11 VDTU Time-related value VC = 0x8E 29

1 Introduction

This document is described object expression guidelines and how to express Generic Binary Object. Generic Binary Object (GBO) is a method of object expression. GBO is smaller and usefully than other object expression for tiny processor as 4bits and 8bits CPU.

This specification is related to IrWW (IrDA for Wristwatches). IrWW provides time-based data communication scheme for a wristwatch by using IrDA Communication Standards. But GBO can use even other object exchange represented by IrMC.

1.1 Scope

Since Generic Binary Object (GBO) is based on format of the vCard, GBO can be applicable to other communication environment such as IrMC. In the following, detail structure as well as coding mechanism and syntax of GBO are explained.

1.2 Relationship of IrWW Profile

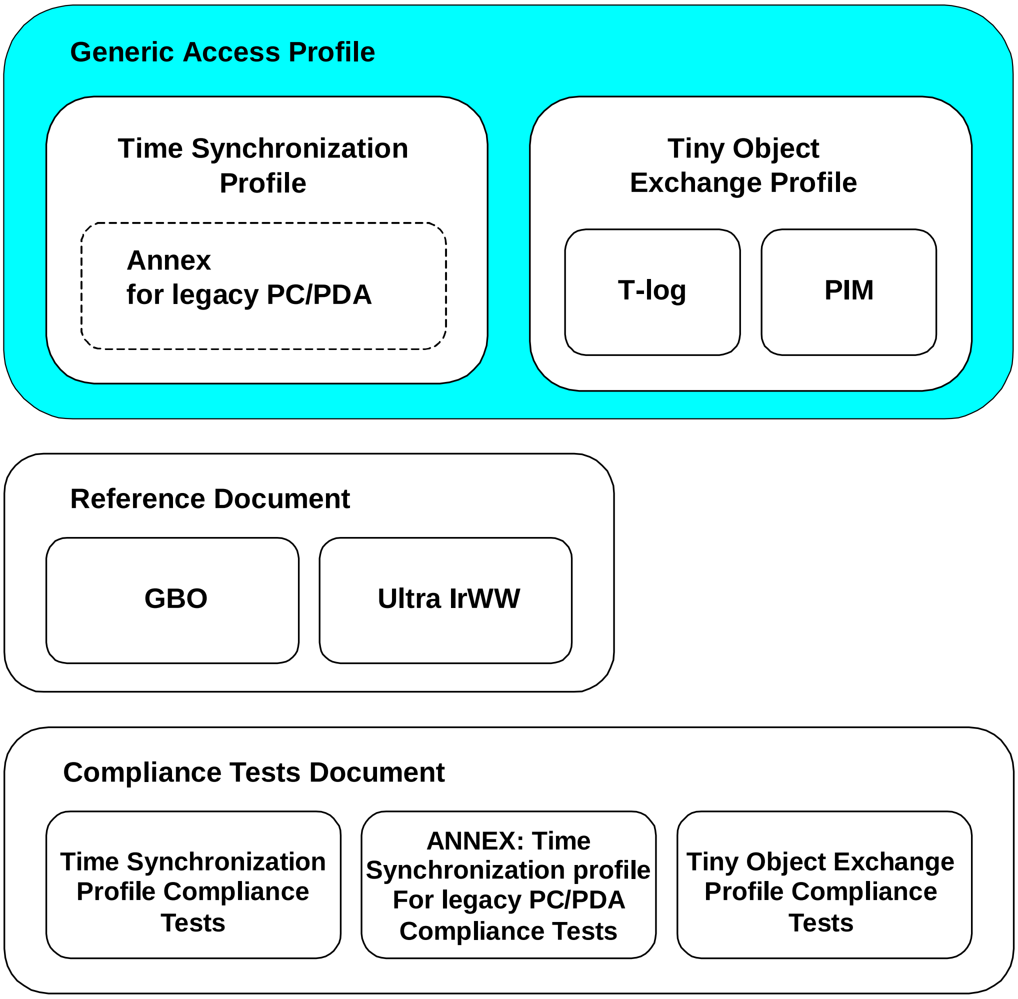

Generic Access Profile
Time Synchronization Profile
Annex for legacy PC/PDA
Tiny Object Exchange Profile
T-log
PIM
Reference Document
GBO
Ultra IrWW
Compliance Tests Document
Time Synchronization Profile Compliance Tests
ANNEX: Time Synchronization profile For legacy PC/PDA Compliance Tests
Tiny Object Exchange Profile Compliance Tests

Figure 1 Relation of this specification

1.3 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.
C	Conditional support. Refers to capabilities that shall be used in when certain other capabilities are also used.
X	Excluded. Refers to capabilities that may be supported by the device but shall never be used in this profile.

N/A	Not applicable in the given context. It is impossible to use this capability.

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.

1.4 References

[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
[TINYTP]	Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, 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.

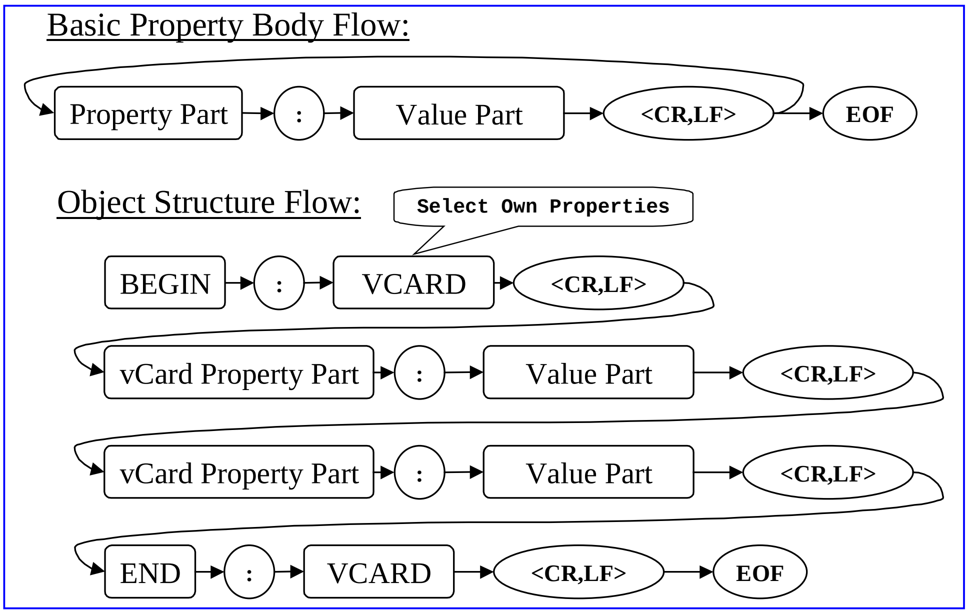
1.5 Content Format

To achieve application level interoperability, content format is defined for time-related object transfer. Since there are limitations of CPU performance and memory space, objects should be compact and easy to handle. To cope with this requirement, a format for reducing object data size without losing any important information are introduced. In the following, basic scheme of format (Generic Binary Object : GBO) and application to time-related objects (bWATCH) are described.

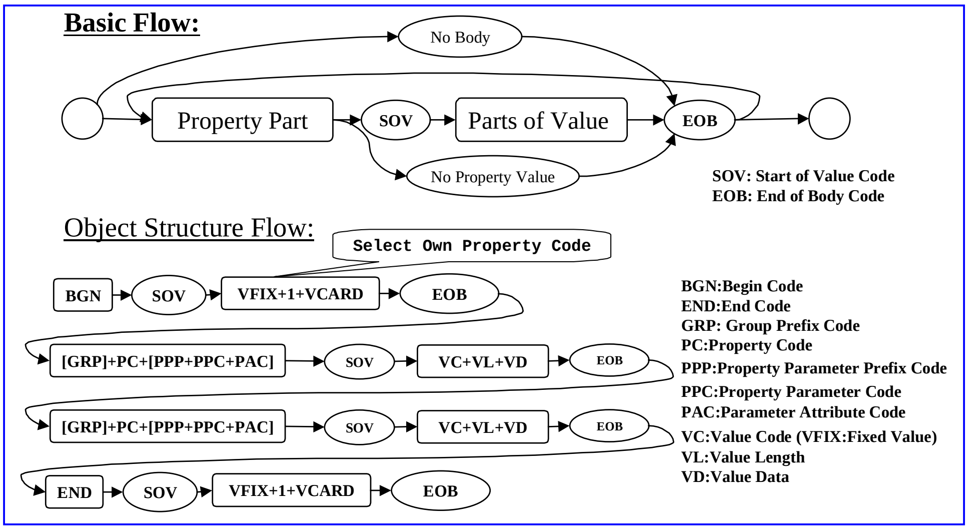
2 Structure of Generic Binary Object (GBO)

This section is described common structure of Generic Binary Object. And describe the guideline that you must keep when you use GBO.

2.1 Basic Structure of GBO



vCARD Structure



General Binary Object Structure

Figure 2-1 Basic structure comparison vCARD vs GBO

Basic structure of GBO is based on that of vCARD. But almost all data (property name, property parameter and property value) consist of pure binary-data instead of character string. In the figure 2-1, basic structure of vCARD and GBO are compared.
Each binary object consists of several properties. These properties have meaning to specify one particular object. In order to distinguish one property group specifying one particular object from another, it is necessary to introduce object identifiers. Two properties such as "BEGIN:OBJECT" (BGN) and "END:OBJECT" (END) properties are used as object identifiers. Between these two identifiers, property is described line by line. Figure 2-2 shows basic structure of the object unit.

GBO Symbolic expression:

```
BGN, SOV, VFIX, VL(1),VCARD, EOB
      VER, SOV, VNUM, VL(2), '2.','1F', EOB
      N,SOV, VBS1,VL(8)'Kitazumi',VBS1,VL(7),'Gontaro', EOB
      TEL,PPP2,TYP=,PREF,WORK,SOV,
        VNUM,VL(7),'+8','1-','37','95','-7','60','1F', EOB
END, SOV, VFIX, VL(1), VCARD, EOB
```

GBO binary expression:

```
0x08, 0x01, 0x20, 0x01, 0x01, 0x00
      0x0A, 0x01, 0x30, 0x02, 0x2A, 0x1F, 0x00
      0x21,0x01, 0x30,0x08,'Kitazumi',0x30,0x07,'Gontaro',0x00
      0x30,0x11,0x30,0x01,0x02,0x01,
            0x30,0x07,0xd8,0x1e,0x37,0x95,0xe7,0x60,0x1f,0x00
0x09, 0x01, 0x20, 0x01, 0x00
```

Figure 2-2 Basic structure of object unit

A property describing specific object exists in the inside of these identifiers. A property consists of an identification code, property parameters and values. This constitution is derived from that of vCARD.

Example of vCARD

```
BEGIN: VCARD
      VERSION: 2.1
      N: Kitazumi; Gontaro
      TEL; PREF; WORK: +81-3-3795-7601
END: VCARD
```

Figure 2-3 vCRAD object format

Therefore, a generic binary object is compatible with vCARD form in the logic level.

2.1.1 Syntax of Generic Binary Object

GOB is simple byte stream object. In the figure 2-1, arrows indicate syntax flow of GBO. According to those arrows, GBO should be interpreted and generated.

2.2 Base elements of GBO

In the world of GBO, there are three parts in property body. The first one is a property part, the Second one is a property parameter part and the last one is a value part.

The most important thing is Property Operation Code (POC). The property name (as Begin, End, N, TEL…), Delimiter / Separator (as ':', ';', CR/LF…) and some kind of stream operation control codes are assigned to POC according to each. One byte of POC indicates how to operate or analyze next byte stream. So, the GBO interpreter can analyze objects without back tracking the byte stream. Unfortunately, if you find an un-encoded property or parameter defined in vCARD or other vStuff, you can use an Un-Encoded Property Code (UES) or an Un-Encoded Property Parameter Code (UEP). So, All of vCARD or vStuff can translate to GBO field. The POC detail is described in the next section.

The Combination of property parameters and property parameter attributes are assigned to Property Parameter Code (PPC) and Parameter Attribute Code (PAC). And they are expressed by specified grammar of GBO.

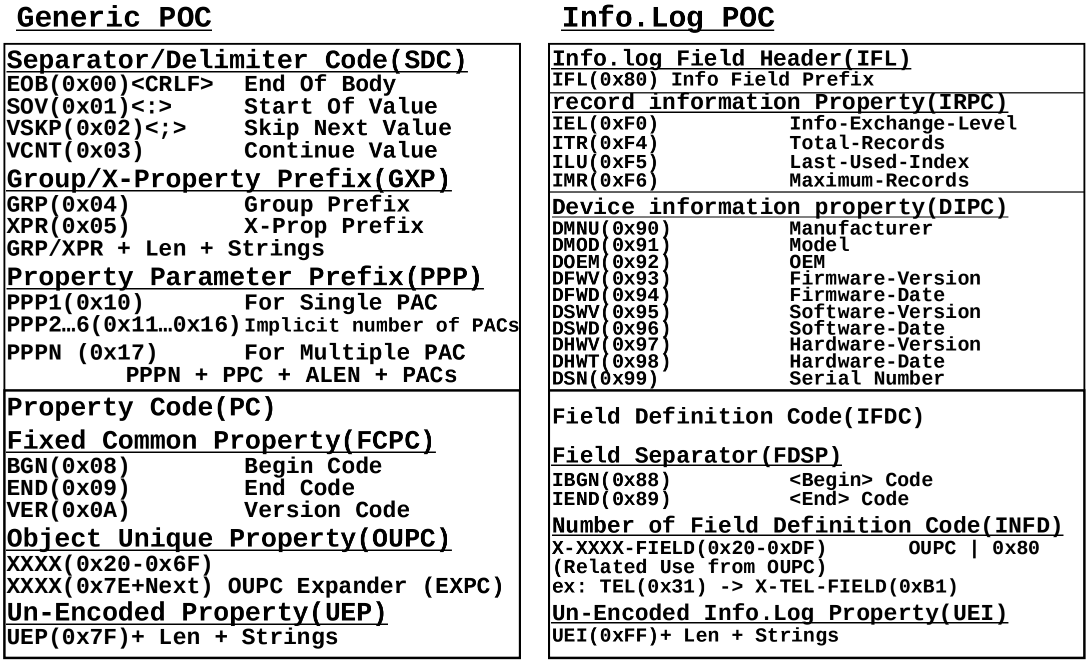
Finally the overview of value expression is as follows. Property part and Value part must be separated by Start of Value Code (SOV). SOV Code is same as vCARD's ':' delimiter. And There are three GBO value types; fixed value (VFIX) type, byte stream value (VBS) type, numerical value (VNUM) type and object unique (VOU) type. One property and value pair is finished by End of Body Code (EOB). This is basic syntax of GBO.

2.2.1 Handling of Nested GBO Structure

One GBO object has some property bodies as figure 3-3. The first property body must have BGN property code. And the last property body must have END property code as already described. If there is another object identifier appearing as a property during the preceding object is interpreted, it is necessary to switch property table for newly found object. And when END property appears, previous object's property table should be assigned again. This mechanism makes layered structure consists of main object and sub-object or nesting structure of other kind of objects available. To this mechanism possible, the END object should have same value of BGN at same nested levels.

2.2.2 Property Operation Code (POC)

Table 2-1 Property Operation Code Table



Generic POC

Separator/Delimiter Code(SDC)	
EOB(0x00)<CRLF>	End Of Body
SOV(0x01)<:>	Start Of Value
VSKP(0x02)<;>	Skip Next Value
VCNT(0x03)	Continue Value
Group/X-Property Prefix(GXP)	
GRP(0x04)	Group Prefix
XPR(0x05)	X-Prop Prefix
GRP/XPR + Len + Strings	
Property Parameter Prefix(PPP)	
PPP1(0x10)	For Single PAC
PPP2…6(0x11…0x16)	Implicit number of PACs
PPPN (0x17)	For Multiple PAC
PPPN + PPC + ALEN + PACs	
Property Code(PC)	
Fixed Common Property(FCPC)	
BGN(0x08)	Begin Code
END(0x09)	End Code
VER(0x0A)	Version Code
Object Unique Property(OUPC)	
XXXX(0x20-0x6F)	
XXXX(0x7E+Next)	OUPC Expander (EXPC)
Un-Encoded Property(UEP)	
UEP(0x7F)+ Len + Strings	

Info.Log POC

Info.log Field Header(IFL)	
IFL(0x80) Info Field Prefix	
record information Property(IRPC)	
IEL(0xF0)	Info-Exchange-Level
ITR(0xF4)	Total-Records
ILU(0xF5)	Last-Used-Index
IMR(0xF6)	Maximum-Records
Device information property(DIPC)	
DMNU(0x90)	Manufacturer
DMOD(0x91)	Model
DOEM(0x92)	OEM
DFWV(0x93)	Firmware-Version
DFWD(0x94)	Firmware-Date
DSWV(0x95)	Software-Version
DSWD(0x96)	Software-Date
DHWV(0x97)	Hardware-Version
DHWT(0x98)	Hardware-Date
DSN(0x99)	Serial Number
Field Definition Code(IFDC)	
Field Separator(FDSP)	
IBGN(0x88)	<Begin> Code
IEND(0x89)	<End> Code
Number of Field Definition Code(INFD)	
X-XXXX-FIELD(0x20-0xDF)	OUPC | 0x80
(Related Use from OUPC)	
ex: TEL(0x31) -> X-TEL-FIELD(0xB1)	
Un-Encoded Info.Log Property(UEI)	
UEI(0xFF)+ Len + Strings	

Table 2-1 shows Property Operation Codes (POC). The left-hand table shows Generic POC, these POC use at all of Objects. POC contains fixed common property (FCPC) and some special functional properties (SDC, GPX, PPP and UEP). To make binary-based object and character-based object compatible, special POCs are specified. Usage of the special POCs is explained in the next section. And a separator and delimiter code (SDC), which separates properties, parameters and values, are also included in this code space. And right-hand table shows POC for Info.Log (information log). Info.Log POC is consisting of Info.Log field header (IFL), Record information properties (IRPC), Device information properties (DIPC) and Field definition codes (IFDC).

The BGN in FCPC indicates beginning of the object unit and determine object type (vCARD, vStuff, bWatch or other object) by the value. BGN code plays another rule to assign Object Unique Property Codes (OUPC) that are spcified by the property value to property code table.

The object unique property codes (OUPC) is determined for each application. If you want to see Object related unique property code, check the specification of bWatch or bvCARD.

2.2.3 Expression of Property Part, Group Part and Property Code (PC)

As described in the previous section, there are some POCs that have special usage to keep compatibility with character-based objects. For example, grouping header of vCARD can't be transformed directly into binary space. To handle this problem, GRP code is defined. This code is accompanied with character string and each character string is described with string length and string value. Figer2-3 shows basic structure of group code usage. There are other

POCs such as XPR and UPC defined. These POC's characteristics are the same as GRP. The XPR code is used for X-prefixed non-common property code header. The UPC code is used as XPR. This UPC code is used for Un-Encoded Property header. GRP and XPR correspond to Group name or X-xxx property of vCARD. Figures 2-4 and 2-5 show examples of GRP and XPR property.

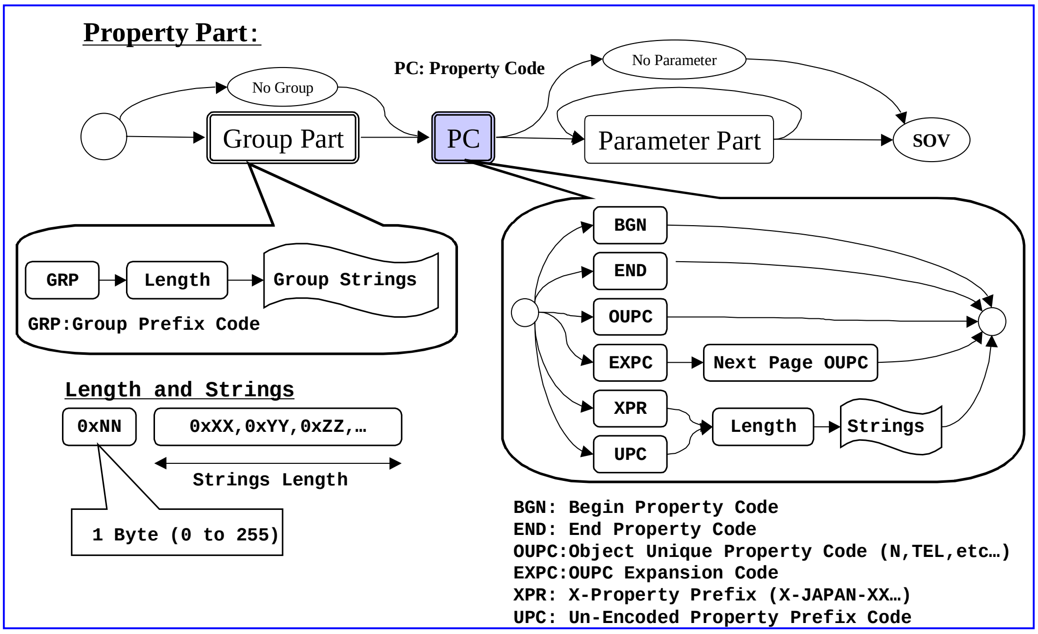


Figure 2-3 Property part expression rule

Expression of GROUP_NAME:

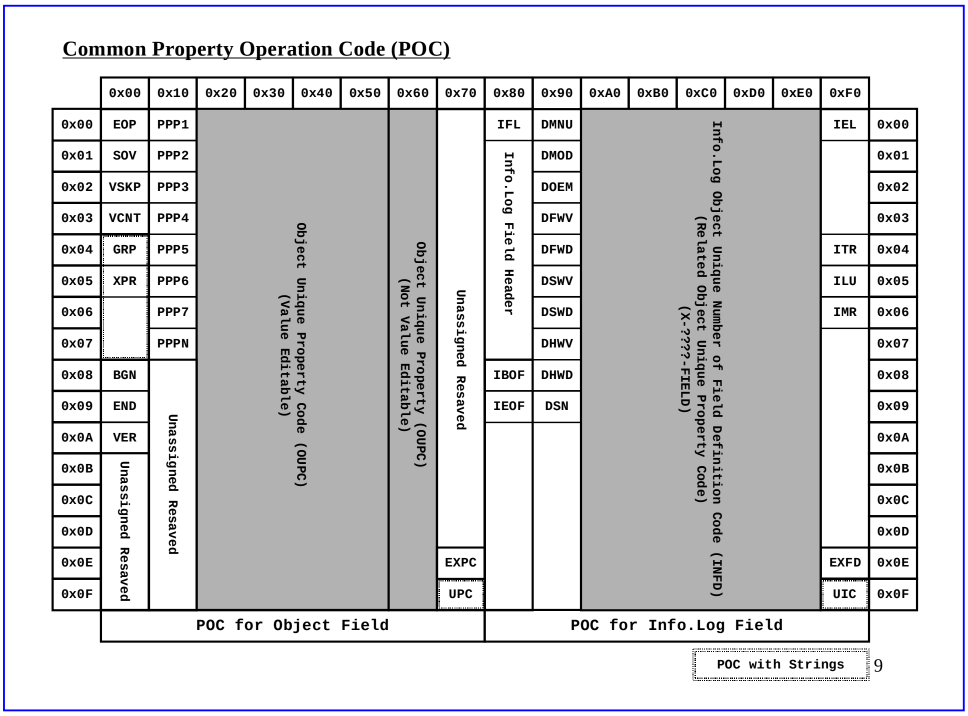
POC	Length	String data (10bytes)
0x04 (GRP)	0x0A	'G','R','O','U','P','_','N','A',M','E'

Figure 2-4 POC coding example (GRP property)

Expression of X-UNKNOWN-PROPERTY:

POC	Length	String data (15bytes)
0x05 (XPR)	0x0F	'U','N','K','N','O','W','N','-, 'P','R','O','P','E','R','T','Y'

Figure 2-5 POC coding example (XPR property)

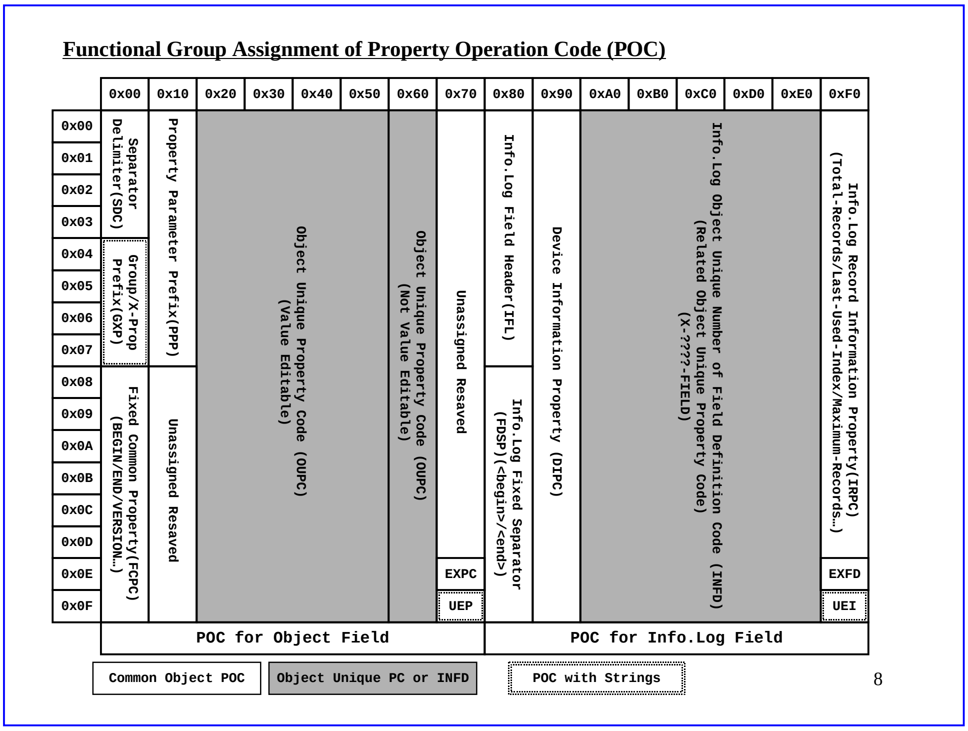


Common Property Operation Code (POC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0	
0x00	EOP	PPP1	Object Unique Property Code (OUPC) (Value Editable)				Object Unique Property (OUPC) (Not Value Editable)	Unassigned Resaved	IFL	DMNU	Info.Log Object Unique Number of Field Definition Code (INFD) (Related Object Unique Property Code) (X-????-FIELD)					IEL	0x00
0x01	SOV	PPP2							Info.Log Field Header	DMOD							0x01
0x02	VSKP	PPP3								DOEM							0x02
0x03	VCNT	PPP4								DFWV							0x03
0x04	GRP	PPP5								DFWD						ITR	0x04
0x05	XPR	PPP6								DSWV						ILU	0x05
0x06		PPP7								DSWD						IMR	0x06
0x07		PPPN								DHWV							0x07
0x08	BGN	Unassigned Resaved							IBOF	DHWD							0x08
0x09	END								IEOF	DSN							0x09
0x0A	VER																0x0A
0x0B	Unassigned Resaved																0x0B
0x0C																	0x0C
0x0D																	0x0D
0x0E								EXPC								EXFD	0x0E
0x0F								UPC								UIC	0x0F
	POC for Object Field								POC for Info.Log Field								

POC with Strings 9

Table 2-2 Common Property Code



Functional Group Assignment of Property Operation Code (POC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0
0x00	Separator Delimiter(SDC)	Property Parameter Prefix(PPP)	Object Unique Property Code (OUPC) (Value Editable)				Object Unique Property Code (OUPC) (Not Value Editable)	Unassigned Resaved	Info.Log Field Header(IFL)	Device Information Property (DIPC)	Info.Log Object Unique Number of Field Definition Code (INFD) (Related Object Unique Property Code) (X-????-FIELD)					Info.Log Record Information Property(IRPC) (Total-Records/Last-Used-Index/Maximum-Records…)
0x01																
0x02																
0x03																
0x04	Group/X-Prop Prefix(GXP)															
0x05																
0x06																
0x07																
0x08	Fixed Common Property(FCPC) (BEGIN/END/VERSION…)	Unassigned Resaved							Info.Log Fixed Separator (FDSP)(<begin>/<end>)							
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E								EXPC								EXFD
0x0F								UEP								UEI
	POC for Object Field								POC for Info.Log Field							

Common Object POC | Object Unique PC or INFD | POC with Strings 8

Table 2-3 Function Group Assignment of Property Operation

2.2.4 Expression of Property part

The Property Parameter part specifies some property options. It specifies own character set, encoding style or property related attributes. This section describes about syntax and coding rule of Property parameters. Figure 2-6 shows basic syntax flow of the property parameter part.

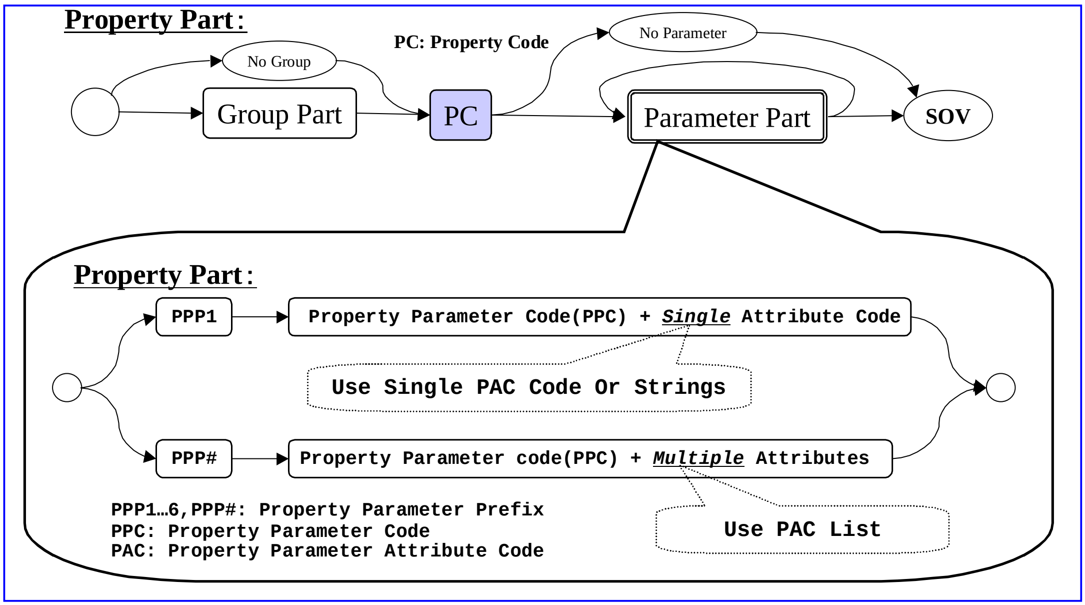

Property Part:
No Group
PC: Property Code
No Parameter
Group Part
PC
Parameter Part
SOV
Property Part:
PPP1
Property Parameter Code(PPC) + Single Attribute Code
Use Single PAC Code Or Strings
PPP#
Property Parameter code(PPC) + Multiple Attributes
PPP1…6,PPP#: Property Parameter Prefix
PPC: Property Parameter Code
PAC: Property Parameter Attribute Code
Use PAC List

Figure 2-6 Property part description rule

2.2.5 Element of Property Parameter Part

A property parameter part consists of three elements. The first element is Property Parameter Prefix (PPP). This code is start code of property part. The second element is Property Parameter Code (PPC). This code determines a kind of property parameters. The last element is Property Parameter Attributes (PAC).

A property parameter part must start with a PPP prefix because PPC is not included by POC code space. There are some types of PPP codes. And also each PPP has different number of PACs

Table 2-4 Common Property Parameter Code (PPC) table

Common PPC:		PAC:
ENC=(0x20)	ENCODING=	Default(0x00) US ASCII QUOT(0x01) QUOTED-PRINTABLE B64(0x02) BASE64 OTHR(0xFF) Other encoding
CHR=(0x21)	CHARSET=	Default(0x00) ISO1-9(0x01-0x09) ISO-8859-1…9 UNIC(0x0F) UNICODE SJIS(0x80) SHIFT_JIS OTHR(0xFF) Other character set
LNG=(0x22)	LANGUAGE=	Default(0x00) 0xFF:Use Strings
TYP=(0x30)	TYPE=(default)	0x00-0xFF: Depend upon Property Code(PC)
VAL=(0x31)	VALUE=	0xFF:Use Strings

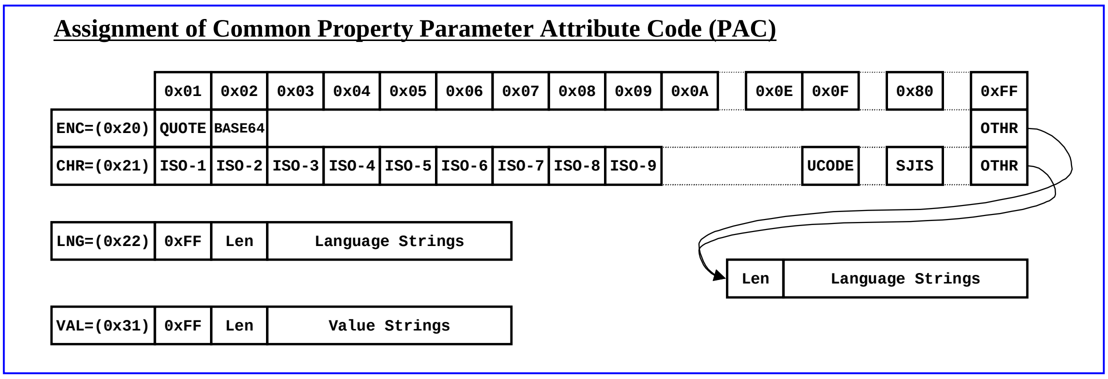

Assignment of Common Property Parameter Attribute Code (PAC)

Figure 3-7 Common property parameter attribute code

2.2.6 Common Property Parameter Code

In vStuff object, there are some commonly using property-parameters. Table 2-4 and Figure 2-7 shows Common Property Parameter Code (PPC) and Parameter attributes (PAC) related to PPC. GBO can express complete binary objects. There is some common PPCs unnecessary in general such as 'ENC='. But it becomes necessary when reverse converts GBO into vCARD or other text based objects. The TYP= code defines a default property. In vCARD version 2.1 it is possible to omit 'TYPE=' parameter. In converting to GBO, the default parameter is regarded as 'TYP=' code.

2.2.6.1 Detail of Common Property Parameter Code

These Common PPC have at least one attribute. If PPC has one attribute, PPP code is PPP1 (0x10). PPP1 code accompanies one byte attribute code. If attribute code (PAC) is 0xff, this code regarded as an un-encoded attribute. They must have a strings attribute after this code. In this case, length (AL) and value (AV) are added after this code. Figure 2-7 and tables 2-5, 2-6, 2-7 and 2-8 shows details of attribute code for PPCs which symbol are "ENC=", "CHR=", "LNG=" and "VAL=", respectively.

Table 2-5 Single attributes Property Parameter table (ENCODING=)

Parameter Name	Property Parameter Code (PPC)
ENCODING=	0x20 (ENC=)
Attribute Name	Attribute Code (PAC)
No Attribute (PLANE)	0x00
QUOTED-PRINTABLE	0x01
BASE64	0x02
Un-Encoded Name	0xFF + AL + AV

Example: N; ENCODING=QUOTED-PRINTABLE: 0x21, 0x10, 0x20, 0x01

Table 2-6 Single attributes Property Parameter table (CHARSET=)

Parameter Name	Property Parameter Code (PPC)
CHARSET=	0x21 (CHR=)
Attribute Name	Attribute Code (PAC)
No Attribute (default)	0x00 US ASCII or Current CHARASET
ISO-8859-1	0x01
ISO-8859-2	0x02
ISO-8859-3	0x03
ISO-8859-4	0x04
ISO-8859-5	0x05
ISO-8859-6	0x06
ISO-8859-7	0x07
ISO-8859-8	0x08
ISO-8859-9	0x09
UNICODE	0x0F
Shift-JIS	0x80
Un-Encoded Name	0xFF + AL + AV

Example: **FN: CHARSET=KOREAN: 0x20, 0x10, 0x21, 0xFF, 0x06 'K', 'O', 'R', 'E', 'A', 'N'**

Table 2-7 Single attributes Property Parameter table (LANGUAGE=)

Parameter Name	Property Parameter Code (PPC)
LANGUAGE=	0x22 (LNG=)
Attribute Name	Attribute Code (PAC)
Undefined Name	0xFF + AL + AV

Table 2-8 Single attributes Property Parameter table (VALUE=)

Parameter Name	Property Parameter Code (PPC)

VALUE=	0x31 (VAL=)
Attribute Name	Attribute Code (PAC)
Undefined Name	0xFF + AL + AV

2.2.6.2 Detail of Property Parts and Syntax Flow

PPPx code is Property Parameter Prefix in POC code space. The PPPx change current code table to PPC code table. And determines number of attributes for PPC. PPP1 have only one Parameter Attribute. If PPC have multiple attributes, it is general to place each attribute. Each attribute can be expressed by using the format explained above. But this method generates too long and too redundant object, as number of attribute increase. For this reason, it is necessary to avoid repeating the same format many times. To compress property parameter field, PPP2 through PPPN provide formats that can handle multiple attributes like single one. Figure 2-8 depicts expression rule for multiple attributes.

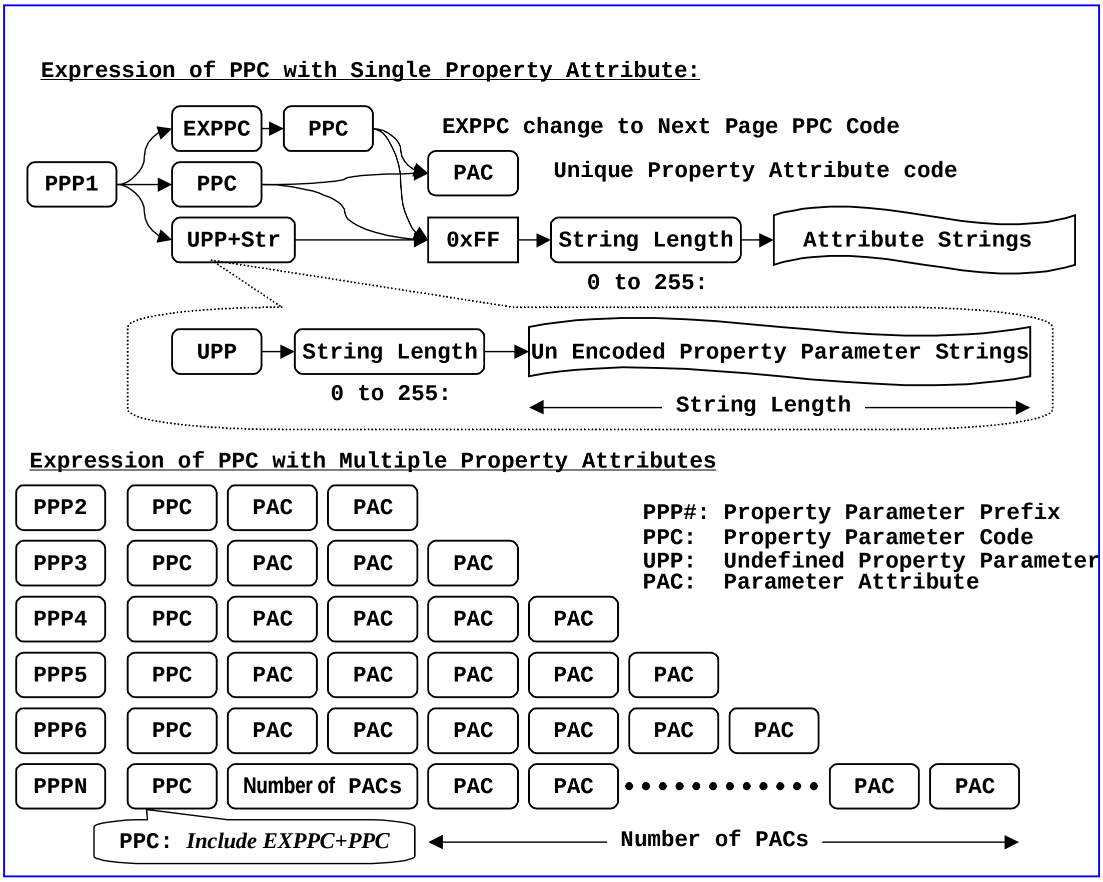

Expression of PPC with Single Property Attribute:
PPP1
EXPPC
PPC
PPC
UPP+Str
EXPPC change to Next Page PPC Code
PAC
Unique Property Attribute code
0xFF
String Length
Attribute Strings
0 to 255:
UPP
String Length
Un Encoded Property Parameter Strings
0 to 255:
String Length
Expression of PPC with Multiple Property Attributes
PPP2
PPP3
PPP4
PPP5
PPP6
PPPN
PPC
PAC
Number of PACs
PPP#: Property Parameter Prefix
PPC: Property Parameter Code
UPP: Undefined Property Parameter
PAC: Parameter Attribute
PPC: Include EXPPC+PPC
Number of PACs

Figure 2-8 **Expression rule for PPC with property attribute**

Table 2-9

Function Group Assignment of Property Parameter Code (PPC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0
0x00	Unassigned Resaved		Common Property Parameter (ENCODING=/CHRSET=/LANGUAGE=/…)	Common Property Parameter (TYPE=/VALUE=/…)	Object Unique Property Parameters											
0x01																
0x02																
0x03																
0x04																
0x05																
0x06																
0x07																
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																EXPPC
0x0F																UPP
	PPC for Common Object				PPC for Unique Object											

UPP with strings[14]

Table 2-10

Assignment of Property Parameter Code (PPC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0
0x00	Unassigned Resaved		ENC=	TYP=	Object Unique Property Parameters											
0x01			CHR=	VAL=												
0x02			LNG=													
0x03																
0x04																
0x05																
0x06																
0x07																
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																EXPPC
0x0F																UPP
	PPC for Common Object				PPC for Unique Object											

PPC with strings[15]

2.2.6.3 Basic coding rule for Value

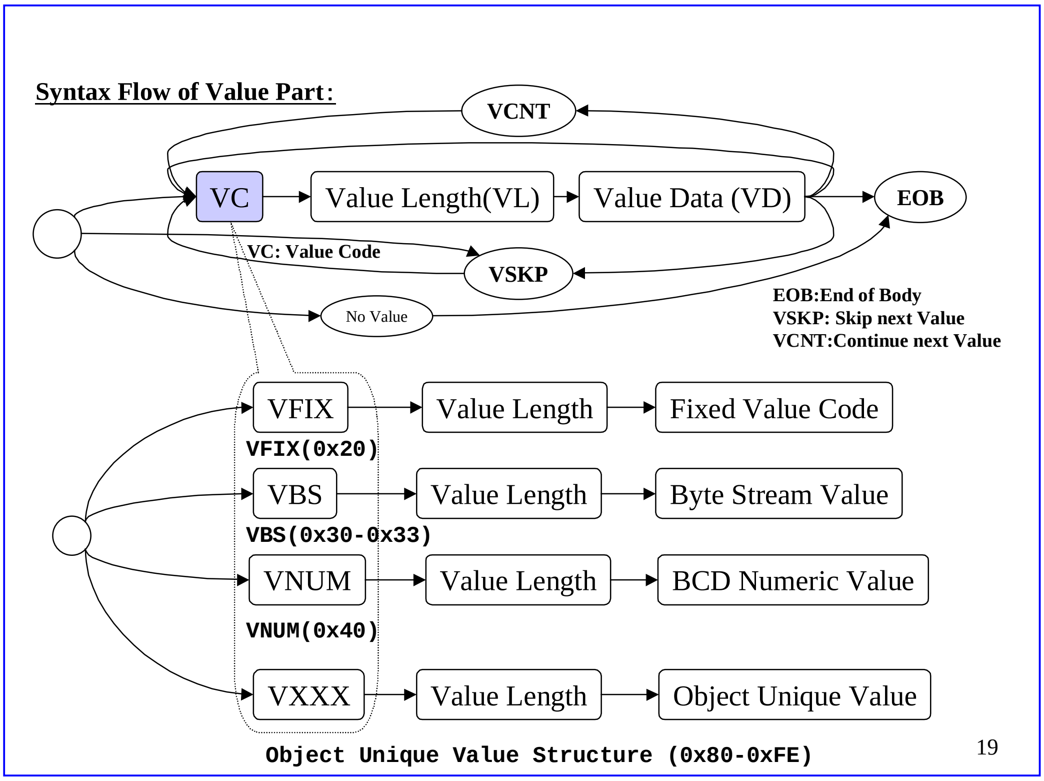


Figure 2-9 generic value coding rule

Figure 2-9 shows basic coding rule of Value. Value field starts after Start of value code (SOV). Elements of a value part consist of Value Code (VC), Value Length (VL) and Value Data (VD). And VC is categorized of three common value expression and object unique structures. Value elements list is shown in Table 2-11.

Table 2-11 Value Identification Code Table

VC	Symbol	Description	Definition of value data (VD)	Size
0x00-0x1f		Reserved		
0x20	VFIX	Fixed Value	VL + Unique for Property	1+N
0x21-0x2F		Undefined Reserved		
0x30	VBS1	Byte Stream Value	VL(1 byte) + Byte Stream	1+N
0x31	VBS2	Byte Stream Value	VL(2 bytes) + Byte Stream	2+N
0x32	VBS3	Byte Stream Value	VL(3 bytes) + Byte Stream	3+N
0x33	VBS4	Byte Stream Value	VL(4 bytes) + Byte Stream	4+N
0x34-0x3F		Reserved	Reserved for Data Compression	
0x40	VNUM	Numerical Value	VL + BCDs	1+N
0x41-0x7E		Undefined Reserved		
0x7F	UES			
0x80-0xFF		Unique Value	VL + VD	1+N

2.2.6.4 Expression of fixed value

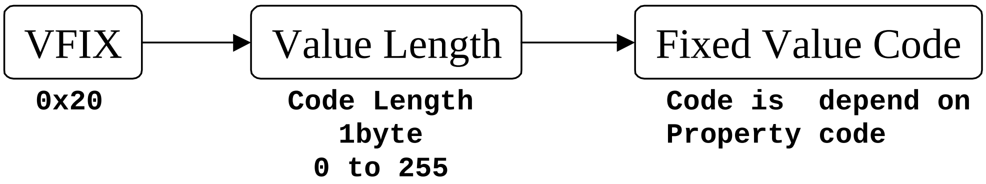
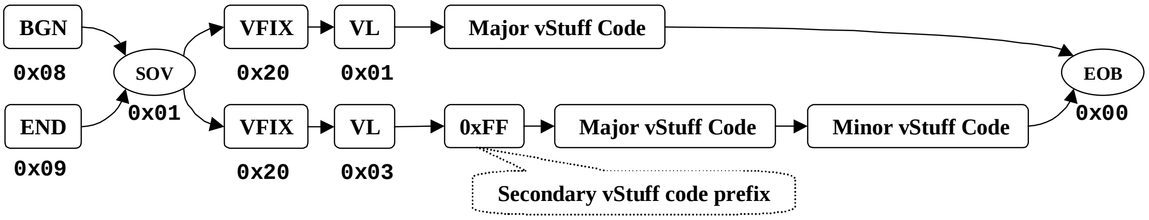


Figure 2-10 Fixed value expression rule

Usually, fixed value is used with BGN/END property. In this case, some values are reserved for object assignment. By using these fixed values, GBO assigns to major vStuff or bWatch object. If there is sub-object existing, for example vEVENT and vTODO in the vCalendar, sub-object can be specified in conjunction with main object. In this case value of three are set to VL. And 0xFF, main object identification code and sub-object identification code are set, respectively. In the figure 2-10, expression rule is shown. In other case, fixed values represent object status and condition. YES/NO and ON/OFF are good example. Fixed values are assigned to each condition or status. For example 0x01 is set to represent YES, 0x02 is set to represent NO. Fixed values are mainly used in the objects such as IrWW.



IrMC Begin/End Value

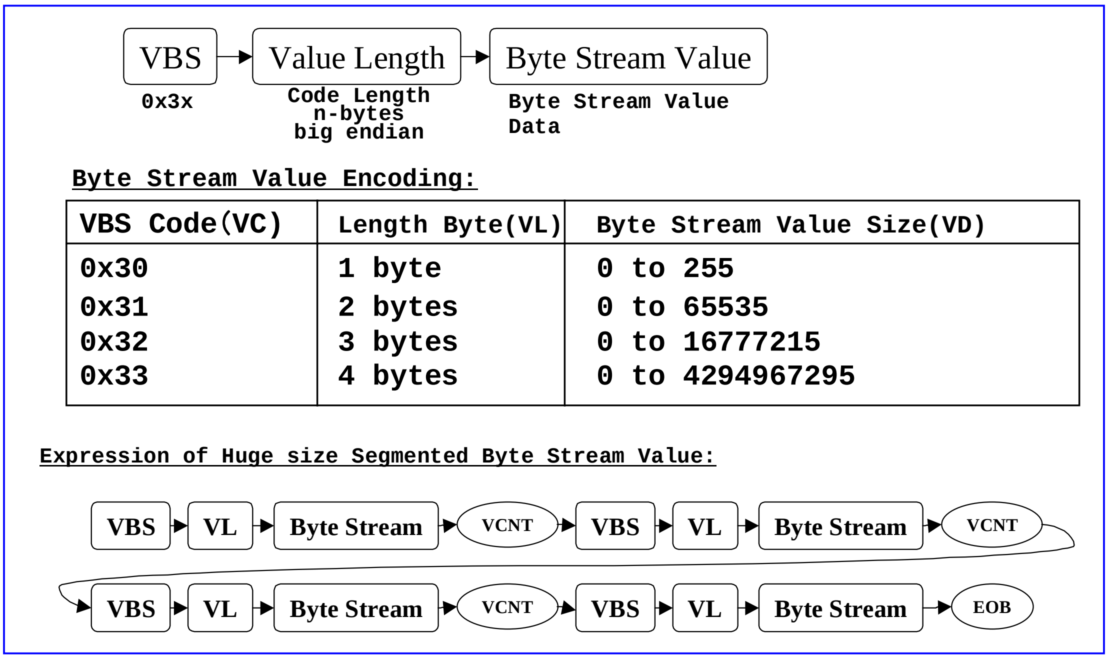
0x01	VCARD		
0x02	VCALENDAR		
0xFF	0x02	0x01	VEVENT
0xFF	0x02	0x02	VTODO
0x03	VMSG		
0xFF	0x03	0x01	VENV
0xFF	0x03	0x02	VBODY
0x04	VNOTE		

IrWW Begin/End Value

0x80	BWATCH		
0xFF	0x80	0x01	T-SYNC
0xFF	0x80	0x02	ALARM
0xFF	0x80	0x03	TIMER
0xFF	0x80	0x04	STOPWATCH
0xFF	0x80	0x05	T-LOG
0xFF	0x80	0x81	FIELD
0xFF	0x80	0x82	MATRIX

Figure 2-11 EBG/END related Fixed value

2.2.6.5 Expression of Byte stream value



VBS Code(VC)	Length Byte(VL)	Byte Stream Value Size(VD)
0x30	1 byte	0 to 255
0x31	2 bytes	0 to 65535
0x32	3 bytes	0 to 16777215
0x33	4 bytes	0 to 4294967295

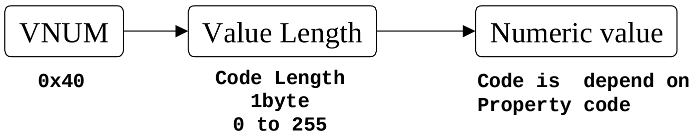
Figure 2-12 Byte Steam value expression rule

Figure 2-12 shows byte-stream value expression rule. In order to handle long stream data, one to four byte VL space are assigned to VCs of which codes are from 0x30 to 0x33, respectively. If long byte-stream appears, size should be checked and select appropriate VIC code which corresponding VD capacity exceeds data size. Single byte-stream value can hold data up to 4 G-bytes. If you want to manage more huge data, you can use VCNT prefix with next value. This code can be put together by several values.

2.2.6.6 Expression of Numerical value

Figure 3-14 shows numerical value expression rule. VC is set to 0x40. BCD (Binary Coded Decimal) is used for numerical data expression. Byte order is going USB to LSB. Two BCD data are packed in one byte (one bytes par 2 BCDs). Numerical data often include some kinds of numerical related character. For these characters unassigned BCD code (0xA to 0xF) are used.

Value of 0xF has special meaning. It describes no data or wildcard. Wildcard is used especially for time-related value. In general usage, wildcard means no data (filler).



BCD(binary Coded Decimal)Encoding:

Code	symbol	Contents	Code	Symbol	Contents
0x0	0	Numeric '0'	0x8	8	Numeric '8'
0x1	1	Numeric '1'	0x9	9	Numeric '9'
0x2	2	Numeric '2'	0xA	.	Decimal Point '.'
0x3	3	Numeric '3'	0xB	,	Comma ','
0x4	4	Numeric '4'	0xC		Not use
0x5	5	Numeric '5'	0xD	+	Sign '+'
0x6	6	Numeric '6'	0xE	-	Sign '-'
0x7	7	Numeric '7'	0xF	F/W	Filler / Wildcard

Packed BCDs in a Byte:

7	6	5	4	3	2	1	0
USD				LSD			

USD: Upper side Digit
LSD: Lower side Digit

Example:

```
123:      0x20,0x02,0x12,0x3F
1234:     0x20,0x02,0x12,0x34
1.34:     0x20,0x02,0x1A,0x34
1,234.5: 0x20,0x04,0x1B,0x23,0x4A,0x5F
-1.25:    0x20,0x03,0xE1,0xA2,0x5F
+1234     0x20,0x03,0xD1,0x23,0x4F
```

Figure 2-13 Numerical value expression rule

2.2.6.7 Expression method of several values and value connection

Some properties have several values. For example, the name property (N) in vCARD has five value fields. The value is a concatenation of the Family name (first field), Given Name (second field), Additional Names (third field), Name Prefix (fourth field), and Name Suffix (fifth field) strings. In this case, you can use VSKP combinations. The following is examples of GBO expression in this case.

vCARD sample:

```
N: Kitazumi; Gontaro;; Mr.
```

GBO sample (symbolic expression)

```
N,SOV, VBS1,VL(8),’Gontaro’, VBS1,VL(7),‘Kitazumi’, VSKP, VBS1,VL(3),‘Mr.’, EOB
```

vCARD sample:

```
N: ; Kitazumi;; Mr.
```

GBO sample (symbolic expression)

```
N, SOV, VSKP, VBS1,VL(8),’Kitazumi’, EOB
```

Some time, You want to separate long one value to short values, In this case you can use VCNT combinations. This code uses for huge binary stream data usually.

vCARD sample:

```
N: Long-Long-Long-So-Long-Name-Jugemu-Jugemu-JugemuNo-Furingai
```

GBO sample (symbolic expression)

```
        N, SOV,
        VBS1,VL(28),’Long-Long-Long-So-Long-Name-’,
        VCNT,
        VBS1,VL(30),’Jugem-Jugem-JugemuNo-Furingai’,
        EOB
```

Table 2-12

Assignment of Value Code (VC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0
0x00	Delimiter (Same as VIC)	Unassigned Resaved	Fixed Value(VGF)	Byte Stream Value(VGB)	Numeric Value(VGN)	Unassigned Resaved			Object Unique VC (VC+VL+VD)							
0x01																
0x02																
0x03																
0x04	Unassigned Reserved			Byte Stream Value Use Data compression Method (T.B.D.)												
0x05																
0x06																
0x07																
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																
0x0F								UES								UES
	Common Object VC								Unique Object VC							

Table 2-13

Assignment of Value Code (VC)

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70	0x80	0x90	0xA0	0xB0	0xC0	0xD0	0xE0	0xF0
0x00	EOP	Unassigned Resaved	VFIX	VBS1	VNUM	Unassigned Resaved			Object Unique VC (VC+VL+VD)							
0x01	SOV		Fixed Value(VGF)	VBS2	Numeric Value(VGN)											
0x02	SKP			VBS3												
0x03				VBS4												
0x04	Unassigned Reserved			Byte Stream Value Use Data compression Method (T.B.D.)												
0x05																
0x06																
0x07																
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																
0x0F																UES
	Common Object VC								Unique Object VC							

2.3 Object Parth algorithm

This section gives sample algorithm for GBO syntax analyzer. This flowchart is helpful to understand all of GBO syntax flow.

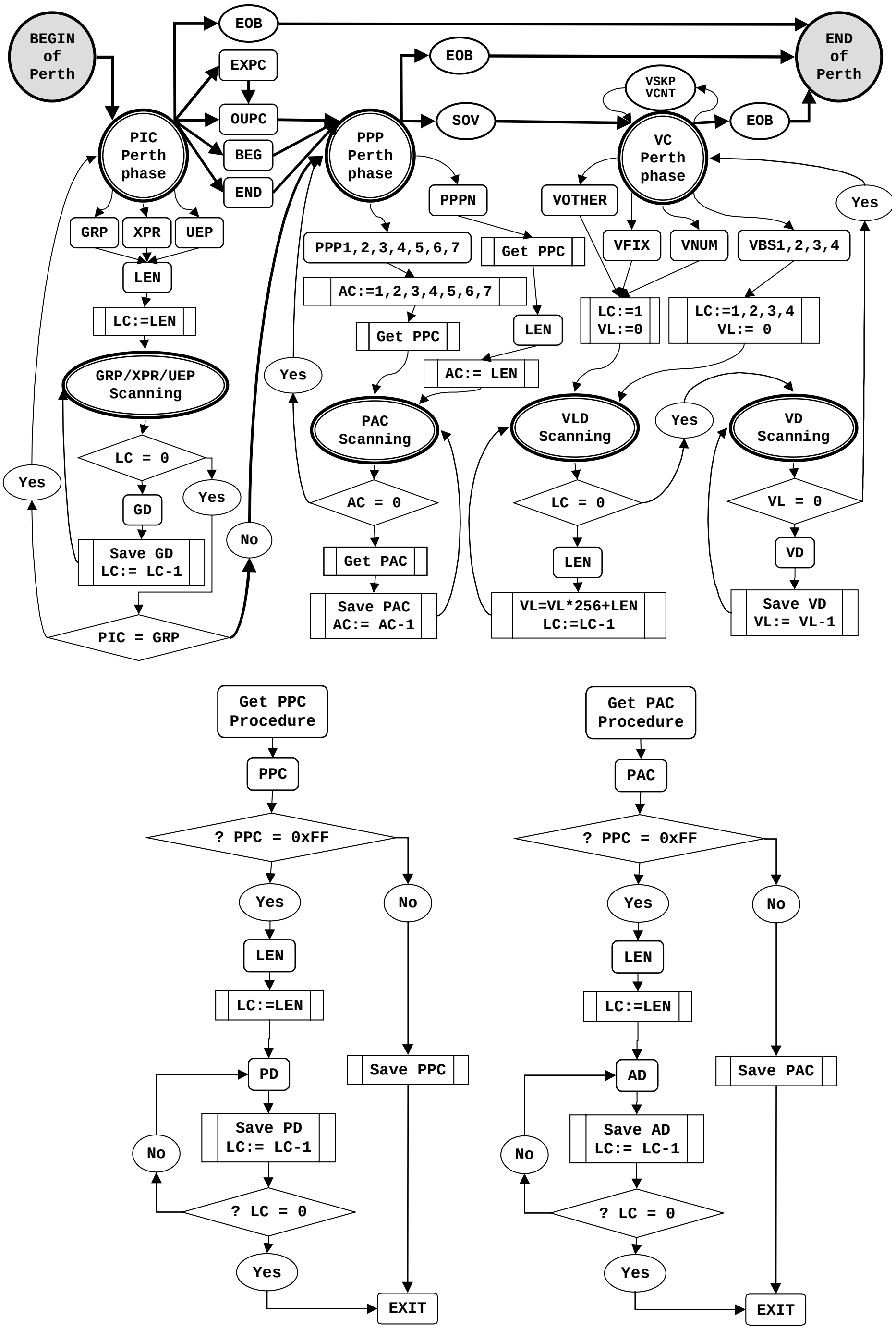


Figure 2-14 Sample Algorithm for GBO

2.4 Object Stream Identifier (OBJECT MAGIC CODE)

In order to identify the beginning of the object stream, following four bytes data are added. This is popular identification method and widely used as a header of binary data such as JPEG and UPF.

Char1	Char2	Char3	FCS

char1～3: character “GBO” expressed in ASCII code
FCS: char1 ex-or char2 ex-or char3

2.5 Device Information Object

In order to specify IrWW device implementation, device must have a device information object.

IrWW Device Information Object Name	/DevInfo.bin

Table 2-14 Device Information object expression

Device information Property Name	PC + SOV	VALUE	EOB
		VBS + VL + Value	
DMNU (Manufacturer)	0x90 0x01	0x30+VL+ Strings	0x00
DMOD (Model)	0x91 0x01	0x30+VL+ Strings	0x00
DOEM (OEM)	0x92 0x01	0x30+VL+ Strings	0x00
DFWV(Firmware-Version	0x93 0x01	0x30+VL+ Strings	0x00
DFWD(Firmware-Date)	0x94 0x01	0x30+VL+ Strings	0x00
DHWV(Software-Version)	0x95 0x01	0x30+VL+ Strings	0x00
DHWD(Software-Date)	0x96 0x01	0x30+VL+ Strings	0x00
DSWV(Hardware-Version	0x97 0x01	0x30+VL+ Strings	0x00
DSWD(Hardware-Date)	0x98 0x01	0x30+VL+ Strings	0x00
DSN(Serial Number)	0x99 0x01	0x30+VL+ Strings	0x00

3 IrWW Object (bWatch)

Time-related information is common and most important data in the wristwatch. A time data transferred from a sender can be used as a reference of time-related functions of a receiver, such as a time-syncronizeation, an alarm, a stopwatch, and a timer. Moreover, there are many advanced wristwatches in which various kinds of sensors are implemented. For example, watches with GPS, pulse, temperature and pressure (air or water) sensors are commercialized. These watches can record time-history of sensing information. In the IrWW communication environment, time-related information and time-history information mentioned above are treated as objects. bWATCH is used as a category name in which above objects are categorized.

3.1 bWATCH Structure Identifier

To distinguish bWATCH category objects from others, Structure Identifiers are introduced. Structure Identifiers belong to fixed-common-property type. Expression rule is shown in the Table 3-1.

Table 3-1 bWATCH Structure Identifier expression rule

Structure Identifier	PC + SOV	VALUE	EOB
		VFIX+VL+SID	
BEGIN : BWATCH	0x08 0x01	0x20 0x01 0x80	0x00
END : BWATCH	0x09 0x01	0x20 0x01 0x80	0x00

3.2 bWATCH Object Identifier

IrWW object has placed inside the Category Identifiers. Each object starts with “BEGIN OBJECT” identifier and ends with “END OBJECT” identifier. Both identifiers belong to fixed-common-property type. In binary-based object, both Structure ID and Object ID are combined in the value field. Table 3-2 shows Object identifier expression rule.

Table 3-2 bWATCH Object Identifier expression rule

IrWW Object Identifier	PC + SOV	VALUE	EOB
		VFIX+VL+0xFF+SID+OID	
BEGIN : T-SYNC	0x08 0x01	0x20 0x03 0xFF 0x80 0x01	0x00
END : T-SYNC	0x09 0x01	0x20 0x03 0xFF 0x80 0x01	0x00
BEGIN : ALARM	0x08 0x01	0x20 0x03 0xFF 0x80 0x02	0x00
END : ALARM	0x09 0x01	0x20 0x03 0xFF 0x80 0x02	0x00
BEGIN : TIMER	0x08 0x01	0x20 0x03 0xFF 0x80 0x03	0x00
END : TIMER	0x09 0x01	0x20 0x03 0xFF 0x80 0x03	0x00
BEGIN : STOPWATCH	0x08 0x01	0x20 0x03 0xFF 0x80 0x04	0x00
END : STOPWATCH	0x09 0x01	0x20 0x03 0xFF 0x80 0x04	0x00
BEGIN : T-LOG	0x08 0x01	0x20 0x03 0xFF 0x80 0x05	0x00
END : T-LOG	0x09 0x01	0x20 0x03 0xFF 0x80 0x05	0x00

3.3 Common IrWW PC (OUPC)

Table 3-3 defines data, parameter, and the flag, which generally is used with bWATCH Object as Common Property. And Common Property Parameters

Table 3-13 IrWW Property Codes

Property	PC	Property Parameters		Value type			
TIME PTIME	**0x20** **0x21**	TYPE= 0x10 0x30		Time-related value			
		0x01 **0x02** **0x03** **0x04** **0x05** **0x06**	HH:MM HH:MM:SS HH:MM:SS:LLLLLL HHHH:MM MMMM:SS LLLLLL	VTHM(0x81) + 0x02 + 2bytes VTHMS(0x82) + 0x03 + 3bytes VTHMSL(0x83) + 0x06 + 6bytes VTRHM(0x84) + 0x03 + 3bytes VTRMS(0x85) + 0x03 + 3bytes VTRL(0x86) + 0x03 + 3bytes			
UTCOFFSET	**0x24**			VTUTC(0x8C) + 0x02 + 2bytes			
DATE PDATE	**0x28** **0x29**	**TYPE=** 0x10 0x30		Time-related value			
		0x08 **0x09**	YYYY.MM.DD YYYY.MM.DD.WW	VDYMD(0x88) + 0x04 + 4bytes VDYMDW(0x89) + 0x05 + 5bytes			
DATETIME	**0x2a**	**TYPE=** 0x10 0x30		Time-related value			

		0x0D
		0x0E
			VDTL(LOCAL)
			VDTU(UTC)
				VDTL(0x8D) + 0x07 + 7bytes
				VDTU(0x8E) + 0x07 + 7bytes			

TITLE/SUBTITLE DESCRIPTION NOTE	**0x30** **0x31** **0x32**			String data 0x3x + NN + MM bytes			
REFERENCE	**0x40**			Fixed Value 0x20 0x01 VD 0x20 0x01 VD			
				RTIME		RTYPE	
				YES NO	**0x01** **0x02**	ATOMIC GPS RADIO TCXO QUARTZ UNDEFINED	**0x01** **0x02** **0x03** **0x04** **0x05** **0xff**
CATEGORY	**0x48**			Fixed Value 0x20 0x01			
				WAKE CALL MEETING APPOINT OTHER	**0x01** **0x02** **0x03** **0x04** **0xff**		
ACTIVE STATE REPEAT	**0x50** **0x51** **0x52** **0x53**			Fixed Value 0x20 0x01			
				ON OFF	**0x01** **0x02**		
ACTION	**0x58**	**TYPE=** 0x10 0x30		User defined values			

		0x01 **0x02** **0x03**	SOUND VIBRATION DISPLAY	0x3x + NN + MM bytes
DATA	**0x5C**	**TYPE=** 0x10 0x30		
		0x01 **0x02**	NUMERIC STRING	0x40 + NN+ MM bytes 0x3x + NN + MM bytes
		ITEMNAME= 0x10 0x40		
		String	0xFF + AL + AV	
		UNIT= 0x10 0x41		
		String	0xFF + AL + AV	
RECORD	**0x5D**			Allow definition in the FILED Block

3.3.1 Expression of IrWW unique Time-related value

Figure 3-4 shows time-related value expression rule. All number and other characters in objects use BCD (same as numerical data).

SOV	Value Identifier Code (VC)	Symbol	Value Length (VL)	Value Data expression
0x01	0x81 HH:MM	VTHM	2 bytes	See 3.3.1.1
0x01	0x82 HH:MM:SS	VTHMS	3 bytes	See 3.3.1.2
0x01	0x83 HH:MM:SS:LLLLLL	VTHMSL	6 bytes	See 3.3.1.3
0x01	0x84 HHHH:MM	VTRHM	3 bytes	See 3.3.1.4
0x01	0x85 MMMM:SS	VTRMS	3 bytes	See 3.3.1.5
0x01	0x86 LLLLLL	VTRL	3 bytes	See 3.3.1.6
0x01	0x88 YYYYMMDD	VDYMD	4 bytes	See 3.3.1.7
0x01	0x89 YYYYMMDDWW	VDYMDW	5 bytes	See 3.3.1.8
0x01	0x8C (S)HH:MM	VTUTC	2 bytes	See 3.3.1.9
0x01	0x8D YYYY:MM:DD:HH:MM:SS	VDTL	7 bytes	See 3.3.1.10
0x01	0x8E YYYY:MM:DD:HH:MM:SS	VDTU	7 bytes	See 3.3.1.11

Figure 3-4 Time-related value expressions

Structure of each VD is explained in the following section which section number is shown in the figure. one BCD code use 4bit. Note: These figure is expressed 4bit (Nibble) coding.

3.3.1.1 VTHM time-related value VC = 0x81

This value explains data in hour/minute manner. The value covers from 00:00 to 99:59

Time data [HH:MM]			
H	H	M	M

HH: Hours of the time.
MM: Minutes of the time

Example:

```
TIME;TYPE=VTHM:12;59
0x20 0x10 0x30 0x01 0x01 0x81 0x02 0x12 0x59 0x00
```

3.3.1.2 VHTMS time-related value VC = 0x82

This value explains data in hour/minute/second manner. The value covers from 00:00:00 to 99:59:59

Time data [HH:MM:SS]					
H	H	M	M	S	S

HH: Hours of the time.
MM: Minutes of the time
SS: Second of the time

Example:

```
TIME;TYPE=VTHMS:18;45;32
0x20 0x10 0x30 0x02 0x01 0x82 0x03 0x18 0x45 0x32 0x00
```

3.3.1.3 VTHMSL Time-related value VC = 0x83

This value explains data in hour/minute/second/under second manner. The value covers from 00:00:00:000000 to 99:59:59:999999

Time data [HH:MM:SS:LLLLLL]											
H	H	M	M	S	S	L	L	L	L	L	L

HH: Hours of the time.
MM: Minutes of the time
SS: Second of the time
LLLLLL: Time less than microsecond unit explains a second.

Example:

```
TIME;TYPE=VTHMSL:7:45:23:455677
0x20 0x10 0x30 0x03 0x01 0x83 0x06 0x07 0x45 0x23 0x45 0x56 0x77 0x00
```

3.3.1.4 VTRHM Time-related value VC = 0x84

This value explains data in hour/minute manner. The value covers from 0000:00 to 9999:59

Time data [HHHH:MM]					
H	H	H	H	M	M

HHHH: Hours of the time.
MM: Minute of the time

Example:

```
TIME;TYPE=VTRHM:56:19
0x20 0x10 0x30 0x04 0x01 0x84 0x03 0x00 0x56 0x19 0x00
```

3.3.1.5 VTRMS Time-related value VC = 0x85

This value explains data in minute/second manner. The value covers from 0000:00 to 9999:59

Time data [MMMM:SS]					
M	M	M	M	S	S

MMMM : Minute of the time
SS : Second of the time

Example:

```
TIME;TYPE=VTRMS:57:50
0x20 0x10 0x30 0x05 0x01 0x85 0x03 0x00 0x57 0x50 0x00
```

3.3.1.6 VTRL Time-related value VC = 0x86

This value explains under-second data by micro-seconds order. The value covers from 000000 to 999999

Time data [LLLLLL]					
L	L	L	L	L	L

LLLLLL: Time of less than a second is measured by microsecond

Example:

```
PTIME;TYPE=VTRL:776655
0x21 0x10 0x30 0x06 0x01 0x86 0x03 0x77 0x66 0x55 0x00
```

3.3.1.7 VDYMD Time-related value VC = 0x88

This value explains data in year/month/day manner. The value covers from 0000:01:01 to 9999:12:31

Date data [YYYY:MM:DD]							
Y	Y	Y	Y	M	M	D	D

YYYY:	Year of the date (Age of the Christian era).
MM:	month of the date
DD:	day of the date

Example:

```
TIME;TYPE=VDYMD:1999;06;04
0x20 0x10 0x30 0x08 0x01 0x88 0x04 0x19 0x99 0x06 0x04 0x00
```

3.3.1.8 VDYMDW Time-related value VC = 0x89

This value explains data in year/month/day/week manner. Information about Day Light Saving (DST) time is involved in week data.

Date data with week [YYYY.MM.DD.WW]									
Y	Y	Y	Y	M	M	D	D	W	W

YYYY: Year of the date (Age of the Christian era).
MM: month of the date
DD: day of the date
WW

W	W

DST	0	0	0	0	x	X	x

Lowest 3bits in the lower nibble 0-6 Sunday to Saturday
xxx day of the week (Sunday = 0...Saturday = 6)
DST: Daylight saving time ‘1’ / Out of Day light saving time ‘0’.

Example :
TIME;TYPE=VDYMDW:1999;06;04;Friday;DST=ON
0x20 0x10 0x30 0x09 0x01 0x89 0x19 0x99 0x06 0x04 0x85 0x00

3.3.1.9 VTUTC Time-related value VC = 0x8c

This value explains Local time offset from UTC in hour/minute manner. Sign (+ or -) information is involved in the higher nibble of hour data.

Offset from UTC [(S)HH:MM]			
H	H	M	M

S	0	0	X

HH: Hours of offset
S: Sign of offset 0 = ‘+’ / 1 = ‘-’.
MM: Minute of offset.

Example:

```
TIME;TYPE=VTUTC:+9;00
0x20 0x10 0x30 0x0C 0x01 0x8C 0x02 0x09 0x00 0x00
```

3.3.1.10 VDTL Time-related value VC = 0x8D

This value explains local date and time in year/month/day/hour/minute/second manner. The value covers from 0000:01:01:00:00:00 to 9999:12:31:23:59:59.

DATETIME (LOCAL)[YYYY:MM:DDHH:MM:SS]													
Y	Y	Y	Y	M	M	D	D	H	H	M	M	S	S

YYYY:	Year of the date (Age of the Christian era).
MM:	month of the date
DD:	day of the date
HH:	Hours of the time.
MM:	Minutes of the time
SS:	Second of the time

Example:

```
TIME;TYPE=VDTL:1999;06;07;07;45;55
0x20 0x10 0x30 0x0D 0x01 0x8D 0x07 0x19 0x99 0x06 0x07 0x07 0x45 0x55 0x00
```

3.3.1.11 VDTU Time-related value VC = 0x8E

This value explains UTC date and time in year/month/day/hour/minute/second manner. The value covers from 0000:01:01:00:00:00 to 9999:12:31:23:59:59.

Expression of this value is the same as that of Local DATETIME. Only the VC can distinguish data types.

DATETIME (UTC)[YYYY:MM:DDHH:MM:SS]													
Y	Y	Y	Y	M	M	D	D	H	H	M	M	S	S

YYYY:	Year of the date (Age of the Christian era).
MM:	month of the date
DD:	day of the date
HH:	Hours of the time.
MM:	Minutes of the time
SS:	Second of the time

```
TIME;TYPE=VTDU:1999;06;07;07;45;55
0x20 0x10 0x30 0x0E 0x01 0x8E 0x07 0x19 0x99 0x06 0x07 0x07 0x45 0x55 0x00
```

第5部

Ultra IrWW プロトコル仕様

Infrared Data Association

Ultra IrWW

Time Synchronization and Time Data Exchange using *Ultra* Protocol

Version 1.0

December 10, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

IrWW SIG

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Ken-ichiro Shimokura	k.shimokura@east.ntt.co.jp
Gontaro Kitazumi	kitazumi@osw.co.jp

Significant Contributors:

History:

Version 0.5:	Infrared Data Association IrWW IrDA for Wrist Watches for Directional Vote
Version 0.8:	IrWW Connectionless-Based Profile - IrDA Time-Adjust Object Push Profile - for Draft Vote.
Version 1.0:	Final vote version separated from IrWW Connectionless-Based Profile - IrDA Time-Adjust Object Push Profile -

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 Introduction 1

1.1 Scope 1

Relationship of IrWW Profile 1

1.3 Symbols and Conventions 2

1.4 References 2

2 Details of Ultra IrWW 3

2.1 Protocol Stack 3

2.1.1 Requirement of IrLMP 3

2.1.2 Requirement of IrLAP 4

2.1.3 Requirement of Physical Layer 4

2.2 Frame Structure of Ultra IrWW 4

2.2.1 Frame Format 4

2.2.2 Frame Structure 5

2.2.3 IrWW Payload Data Structure 5

2.3 Communication Procedure 6

1 Introduction

Ultra IrWW protocol provides simple connection-less object communications. If there is a wristwatch that has IrDA physical layer, Ultra and Ultra IrWW is minimum set of communication procedure that makes data exchange available. Also, Ultra IrWW is designed to preferentially exchange time-related information. Therefore, the protocol makes the communication between a quartz clock with analog indicator and that with digital one.

1.1 Scope

It prescribes time-related data exchange between wristwatches that implemented IrDA physical layer.

1.2 Relationship of IrWW Profile

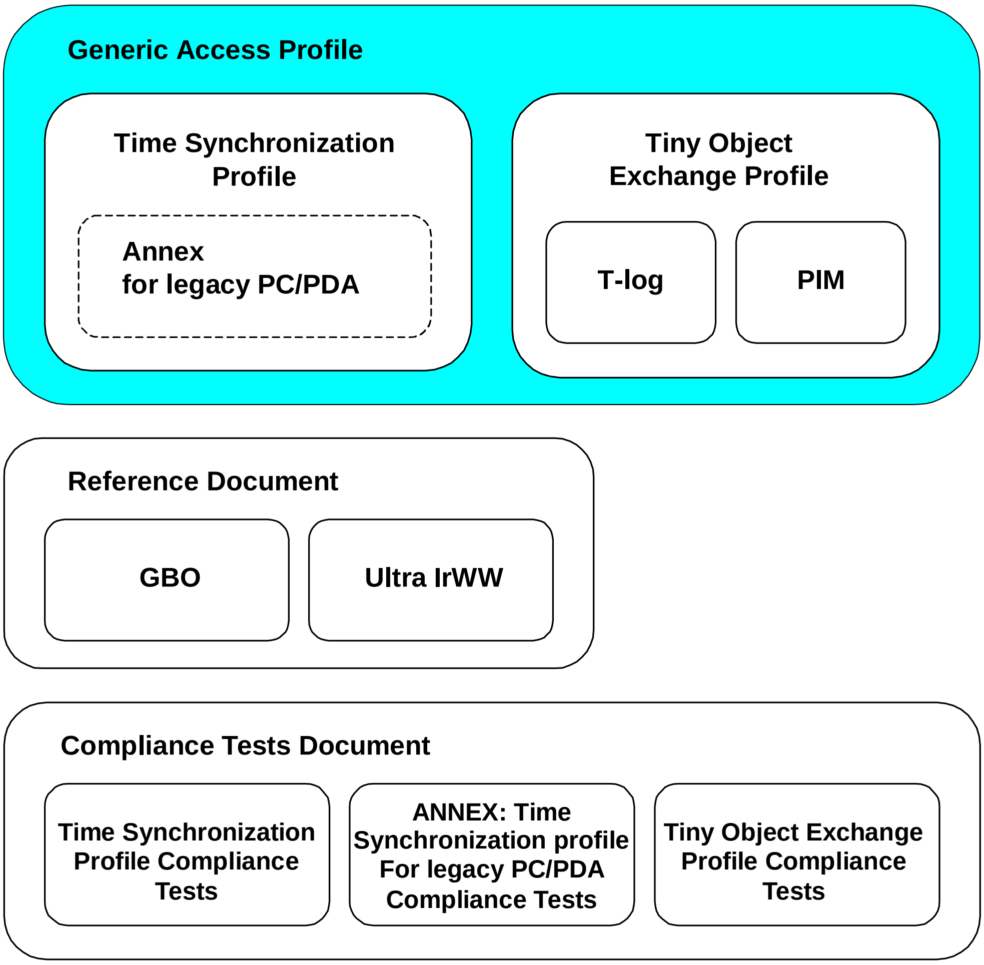

Generic Access Profile
Time Synchronization Profile
Annex for legacy PC/PDA
Tiny Object Exchange Profile
T-log
PIM
Reference Document
GBO
Ultra IrWW
Compliance Tests Document
Time Synchronization Profile Compliance Tests
ANNEX: Time Synchronization profile For legacy PC/PDA Compliance Tests
Tiny Object Exchange Profile Compliance Tests

Figure 1 Relation of this specification

1.3 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.
C	Conditional support. Refers to capabilities that shall be used in when certain other capabilities are also used.
X	Excluded. Refers to capabilities that may be supported by the device but shall never be used in this profile.
N/A	Not applicable in the given context. It is impossible to use this capability.

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.

1.4 References

[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

[Ultra] Guidelines for Ultra Protocols, Version 1.0, Infrared Data Association

2 Details of Ultra IrWW

2.1 Protocol Stack

Ultra IrWW is placed on the upper layer IrDA Ultra Protocol. Ultra IrWW are necessary for exchanging a time-based data between wristwatches. Figure 1 depicts IrWW positioning on the currently available IrDA standards. In the figure, newly developed applications are indicated with slanted lines. Technical details of Ultra IrWW are shown in the later part of this document.

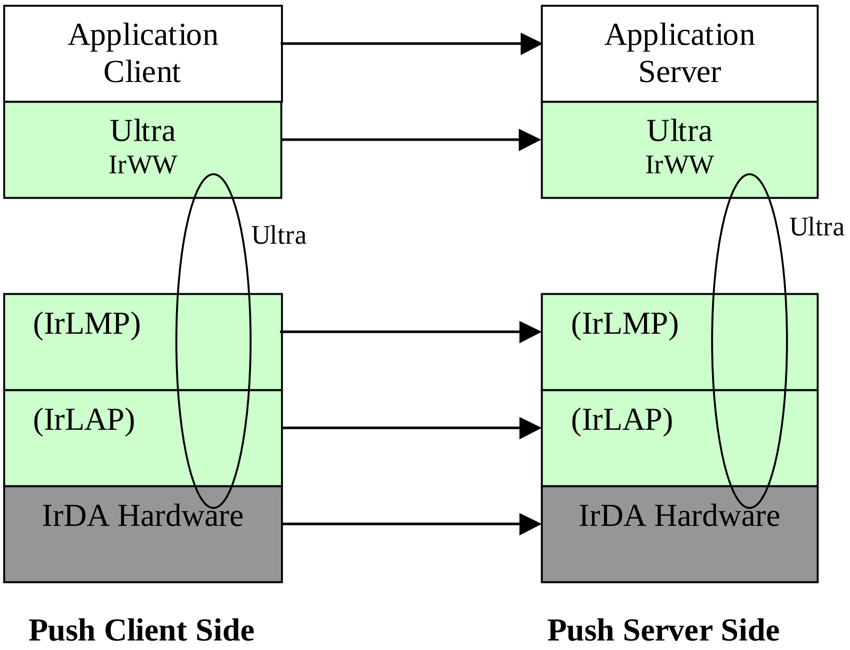


Figure 2 Usege of IrDA stack

IrDA Hardware is governed by the [IrPHY]

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

IrLMP is a multiplexing layer specified in [IrLMP]

Ultra is connectionless protocol specified in [Ultra]

Ultra IrWW is placed on the upper layer Ultra protocol. Specified in [Ultra]

2.1.1 Requirement of IrLMP

A device must support connectionless LSAP (both DLSAP and SLSAP are 0x70), and the data packet must be delivered to the upper layer transparently.

2.1.2 Requirement of IrLAP

The data packet must be delivered to IrLMP or upper layer.

2.1.3 Requirement of Physical Layer

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

2.2 Frame Structure of Ultra IrWW

Ultra IrWW is based on Ultra Protocol specified in [Ultra].
IrDA IrWW Devices must follow the recommendation in the Ultra specification that limits the maximum service data field size to 62 octets in length. Connectionless service is thus limited to use frames with maximum payload size of 60 octets.

2.2.1 Frame Format

Figure 3 shows schematic of the Ultra protocol frame.

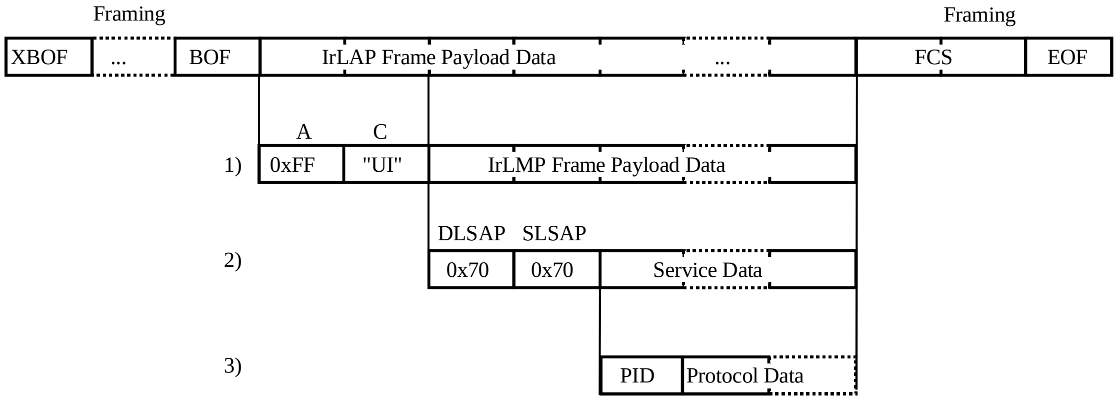


Figure 3 The framing and headers

PID is protocol ID of the high rank layer using ultra protocol. 0x01 is reserved for Ultra-OBEX currently. And 0x02 is reserved for Ultra-IrWW.

PID Octet	
PID No	**Function**
0x00	Reserved
0x01	For Ultra OBEX
0x02	**For Ultra IrWW**
0x03 – 0x7f	TBD
MSB(bit 7)	For extention

Note: All PID values are reserved and assigned by IrDA.

Table 1 The PID Octet Encoding

2.2.2 Frame Structure

When the protocol identifier has a value of 02h, IrWW objects may be placed in the service data field. The Ultra IrWW frame structure is shown in figure 4.

BOF	ADR	CMD	IrLMP		PID	SAR	Ultra Payloads	FCS	EOF
0xC0	0xFF	“UI”	0x70	0x70	0x02	0x00	IrWW payloads	FCS	0xC1

Figure 4 Ultra IrWW frame structure

In the Ultra IrWW, object data size is less than 60bytes so that communication procedure should be completed with one UI frame. Therefore, SAR is always set to 00h, and Ultra IrWW Frame consists of only one frame.

2.2.3 IrWW Payload Data Structure

Figure 5 shows IrWW payload data structure.

IrWW Payloads	
OID	IrWW Object Value

Figure 5 Ultra IrWW Object Payloads

OID: IrWW Object ID (One byte)

OID specifies IrWW objects that are transferred with Ultra protocol. Unique number is given to each object.

Object	OID
Time-Adjust	0x00
Alarm	0x10
Timer	0x20
Stopwatch	0x30

Table 2 IrWW Object

2.3 Communication Procedure

Figure 6 shows an example of time sequence of Ultra IrWW communication. The advantage of the Ultra IrWW is that an IrWW object generated in the DEVICE 1 UI level can be sent to the DEVICE 2 UI level almost directly. This characteristic is effective for transmitting time-based information with little time-delay.

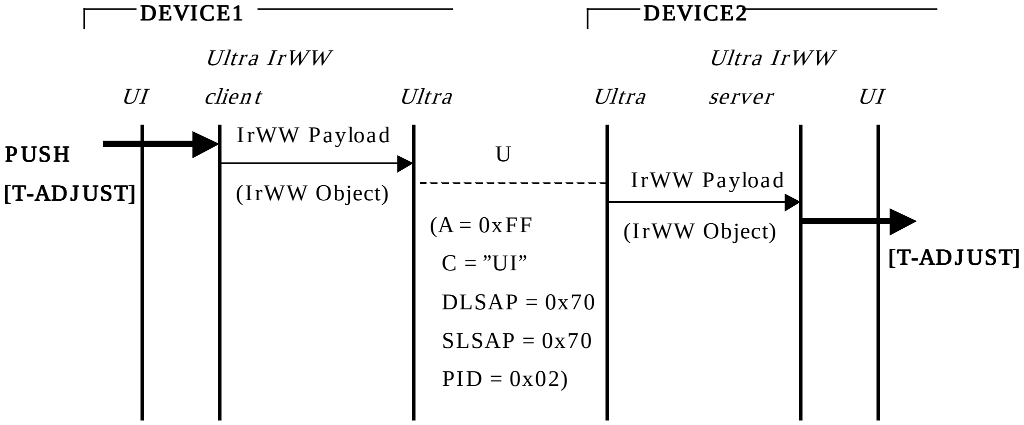


Figure 6 Example of Ultra IrWW time sequence

第6部

時刻同期プロファイル相互接続性試験

Infrared Data Association

IrWW (Infrared Wrist Watch)
Time Synchronization profile
Compliance Tests
for Ultra IrWW

Version 1.0

December 10, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Tomozumi Saruwatari, saruwatari@sii.co.jp

Significant Contributors:

History:
Version 1.0: IrWW Time Synchronize Profile Compliance Tests.

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 INTRODUCTION 1

1.1 Purpose 1
1.2 Scope 1
1.3 References 1

2 ENVIRONMENT 2

2.1 Hardware 2
2.2 Software 2

3 OVERVIEW COMPLIANCE TESTS 2

3.1 What Tests Apply to my Product ? 2
3.1.1 Client/Server 2
3.1.2 Ultra IrWW Transfer Tests 2
3.1.3 Objects 2
3.1.4 Miscellaneous Tests 3
3.2 General Test Steps 3

4 ULTRA IRWW TRANSFER TESTS 3

4.1 Client 3
4.2 Server 3

5 ERROR RESPONSES TESTS 4

5.1 Client 4
5.2 Server 4

6 MISCELLANEOUS TESTS 4

6.1 Contents 4
6.1.1 Client 4
6.1.2 Server 5

7 TEST RESULT TEMPLATE 5

1 Introduction

1.1 Purpose

This document specifies the test cases used to verify an IrWW device which implements Time Synchronization Profile. The goal of this document is to ensure interoperability between such IrWW devices and products.

The purpose of this document is to define the minimum behaviors that should be required of a device, in order to be declared IrWW compliant.

1.2 Scope

The tests described in this document do not aim to verify the lower layers of the IrDA stack. The tests target the new application layers like Time-Sync object. This specification does not require that the device has already passed the IrPHY, IrLAP and IrLMP compliance tests.

The applications supported by an IrWW device include at lease mandatory object or more of the following:

Time-Sync object (mandatory)
Alarm object (option)
Timer object (option)
Stopwatch object (option)

1.3 References

[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
[Ultra]	Guidelines for Ultra Protocols, Version 1.0, Infrared Data Association.
[IrWW]	IrWW Generic Access Profile, Version 0.8, Infrared Data Association.
[GBO]	Generic Binary Object, Version 0.8, Infrared Data Association
[Ultra IrWW]	Ultra Protocols for IrWW, Version 0.8, Infrared Data Association

2 Environment

2.1 Hardware

The following list includes examples of the type of hardware that is required:
- Device Under Test (DUT)
- PC or another device to generate test frames, with built-in IR port or an IR dongle.
- IR Probe for tracing low level IR frames (this is optional).

2.2 Software

The following list includes examples of the type of software that is required:
- Source of OBEX Connectionless Data (IrDA Ultra). Note, Win9X does not support Ultra frames. Ultra frames can be generated manually, and transmitted using a simple serial communication package. Care must be taken to adhere to the Ultra timing constraints. For example, consecutive frames should be sent with a gap of between 500ms and 2s. PicoRay from Phoenix supports connectionless data.
- Software package which can generate and check Time-Sync object etc, using GBO for Ultra IrWW.

3 Overview Compliance Tests

The test scenarios assume that the device under test (DUT) is a Wristwatch device, and that the Tester is software running on a PC.

3.1 What Tests Apply to my Product ?

The IrWW specification defines various applications. Most of these applications are optional. The tests that your product must pass, depends on the applications that your product offers.

3.1.1 Client/Server

The tests have been named according to whether the DUT is acting as a client or a server.

IrWW_C_x_x_x_x	Client (initiates the data transfer)
IrWW_S_x_x_x_x	Server (responds to requests for data transfer)

3.1.2 Ultra IrWW Transfer Tests

The connectionless-based transfer tests verify the basic IrWW support that is called Ultra IrWW. All IrWW devices must pass these tests. The tests are named as
IrWW_x_U_x_x_x.

3.1.3 Objects

The tests have been named according to the object they are accessing.

IrWW_x_x_TIME_x_x	Time-Sync
IrWW_x_x_ALARM_x_x	Alarm
IrWW_x_x_TIMER_x_x	Timer
IrWW_x_x_STOPW_x_x	Stopwatch

IrWW_x_x_BXX_x_x	Independent of the object

If a device supports one of the object types, then it must pass the compliance tests for that object type. A device will fail the compliance tests if it fails the tests for one object type, even if it passes all the others. For example, a device which supports both Time-Sync and Alarm, must pass all of the Time-Sync and Alarm tests before the device can pass.

3.1.4 Miscellaneous Tests

The Miscellaneous tests verify that the general behavior of the DUT. The tests are named as **IrWW_x_M_x_x**.

3.2 General Test Steps

The majority of the tests involve transferring OBEX objects between the DUT under test and the PC. The following steps should be performed.
1. Switch on the DUT under test, and activate the IR port if appropriate.
2. Switch on the PC, and enable the IR if appropriate.
3. Point the IrDA ports at each other. Ensure that they are within the operating distances defined for the devices. i.e. some devices only have a 20cm operating range.
4. Carry out the transmission.
5. Verify that the data was transferred without any problems.

4 Ultra IrWW Transfer Tests

Time Synchronization Profile V0.95 requires that Ultra IrWW is the mandatory support. In the Ultra IrWW, object data size is less than 60bytes so that communication procedure should be completed with one UI frame. Therefore, SAR is always set to 00h, and Ultra IrWW Frame consists of only one frame.

4.1 Client

IrWW_C_U_BXX_GBO_1 (Mandatory)
1. General Test Steps 1-3
2. Make the DUT transmit an object (Time-Sync etc) using Ultra IrWW. The IrWW object size should be less than 60 bytes including OID byte.
3. Verify that the Tester received the object correctly

4.2 Server

IrWW_S_U_BXX_GBO_1 (Mandatory)
1. General Test Steps 1-3
2. Make the Tester transmit an object (Time-Sync etc) using Ultra IrWW. The IrWW object size should be less than 60 bytes including OID byte.
3. Verify that the DUT received the object correctly. In general, DUT will set the data, such as time, alarm, timer and stopwatch, using the received object automatically. As an option, the user may be prompted whether to accept or reject the object.

5 Error Responses Tests

Ultra IrWW is based on connectionless protocol, therefore the error codes are not decided. However the following error conditions might be considered.

Inappropriate headers or IDs
Longer object size
Lack of mandatory properties
Inappropriate order of mandatory properties

When these errors might happen, the server must ignore received object without setting the data. It is recommended that the server gives a user the notice of failure of transmission.

5.1 Client

No Tests.

5.2 Server

IrWW_S_U_BXX_RSP_1 (Option)
1. General Test Steps 1-3
2. Make the Tester transmit an object (Time-Sync etc) with inappropriate headers or IDs using Ultra IrWW. The IrWW object size should be less than 60 bytes including OID byte.
3. The DUT must ignore the received object without setting the data and gives the notice of failure.

IrWW_S_U_BXX_RSP_2 (Option)
1. General Test Steps 1-3
2. Make the Tester transmit a longer object than 60 bytes (Time-Sync etc) using Ultra IrWW.
3. The DUT must ignore the received object without setting the data and gives the notice of failure.

IrWW_S_U_BXX_RSP_3 (Option)
1. General Test Steps 1-3
2. Make the Tester transmit an object (Time-Sync etc) without some mandatory properties using Ultra IrWW. The IrWW object size should be less than 60 bytes including OID byte.
3. The DUT must ignore the received object without setting the data and gives the notice of failure.

IrWW_S_U_BXX_RSP_4 (Option)
1. General Test Steps 1-3
2. Make the Tester transmit an object (Time-Sync etc) with inappropriate order of mandatory properties using Ultra IrWW. The IrWW object size should be less than 60 bytes including OID byte.
3. The DUT must ignore the received object without setting the data and gives the notice of failure.

6 Miscellaneous Tests

6.1 Contents

6.1.1 Client

IrWW_C_M_TIME_CONTENTS_1 (Mandatory)

1. General Test Steps 1-3
2. Make the DUT transmit the Time-Sync object using Ultra IrWW.
3. Verify that the Tester received the object correctly.
4. Check that Object Identifier (OID) is 0x00.
5. Check that all mandatory properties are set in order.

IrWW_C_M_ALARM_CONTENTS_1 (Mandatory if the DUT supports Alarm)
1. General Test Steps 1-3
2. Make the DUT transmit the Alarm object using Ultra IrWW.
3. Verify that the Tester received the object correctly.
4. Check that Object Identifier (OID) is 0x10.
5. Check that all mandatory properties are set in order.

IrWW_C_M_TIMER_CONTENTS_1 (Mandatory if the DUT supports Timer)
1. General Test Steps 1-3
2. Make the DUT transmit the Timer object using Ultra IrWW.
3. Verify that the Tester received the object correctly.
4. Check that Object Identifier (OID) is 0x20.
5. Check that all mandatory properties are set in order.

IrWW2_S_M_STOPW_CONTENTS_1 (Mandatory if the DUT supports Stopwatch)
1. General Test Steps 1-3
2. Make the DUT transmit the Stopwatch object using Ultra IrWW.
3. Verify that the Tester received the object correctly.
4. Check that Object Identifier (OID) is 0x30.
5. Check that all mandatory properties are set in order.

6.1.2 Server

No Tests.

7 Test Result Template

Test Name	Pass/Fail	Comments
IrWW_C_U_BXX_GBO_1		
IrWW_S_U_BXX_GBO_1		
IrWW_S_U_BXX_RSP_1		
IrWW_S_U_BXX_RSP_2		
IrWW_S_U_BXX_RSP_3		
IrWW_S_U_BXX_RSP_4		
IrWW_C_M_TIME_CONTENTS_1		
IrWW_C_M_ALARM_CONTENTS_1		
IrWW_C_M_TIMER_CONTENTS_1		
IrWW_C_M_STOPW_CONTENTS_1		

第7部

付記 既存 PC/PDA における時刻同期プロファイル 相互接続性試験

Infrared Data Association

IrWW (Infrared Wrist Watch)
ANNEX: Time Synchronization profile
for legacy PC/PDA
Compliance Tests

Version 1.0

December 21, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Michihiro Enokida enokida@citizen.co.jp

Significant Contributors:

History:

Version 0.10: Time Synchronization Profile for legacy PC/PDA Compliance Tests.

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 INTRODUCTION 1

1.1 Purpose 1
1.2 Scope 1
1.3 Definitions and Acronyms 1
1.4 References 1

2 ENVIRONMENT 3

2.1 Hardware 3
2.2 Software 3

3 OVERVIEW COMPLIANCE TESTS 4

3.1 What Tests Apply to my Product ? 4
3.1.1 Client/Server 4
3.1.2 Protocol Tests 4
3.1.3 Read-all/Write-all Tests 4
3.1.4 Single Push Tests 4
3.1.5 Single Pull Tests 4
3.1.6 Objects 4
3.1.7 Miscellaneous Tests 5
3.2 General Test Steps 5

4 PROTOCOL TESTS 6

4.1 Discovery Hint Bits 6
4.1.1 Client 6
4.1.2 Server 6
4.2 IAS Entries 6
4.2.1 Client 6
4.2.2 Server 7
4.3 Devinfo 7
4.3.1 Client 7
4.3.2 Server 8
4.4 Info Logs 8
4.4.1 Client 8
4.4.2 Server 8

5 READ-ALL/WRITE-ALL TESTS 9

5.1 Client 9
5.2 Server 9

6 SINGLE PUSH TESTS 11

6.1 Client 11
6.2 Server 11

7 SINGLE PULL TESTS 12

7.1 Client 12
7.2 Server 12

8 ERROR RESPONSES TESTS 13

8.1 Client 13
8.2 Server 13

9 MISCELLANEOUS TESTS 14

9.1 Contents 14
9.1.1 Client..... 14
9.1.2 Server 14
10 TEST RESULT TEMPLATE 16

1 Introduction

1.1 Purpose

This document specifies the test cases used to verify whether an IrWW device complies Time Synchronization profile for legacy PC/PDA or not. The goal of this document is to ensure interoperability between IrWW devices and products.
The purpose of this document is to define the minimum behaviors that should be required of a device, in order to be declared IrWW compliant.

1.2 Scope

The tests described in this document do not aim to verify the lower layers of the IrDA stack. The tests target the new application layers like Time-Sync. This specification does not require that the device has already passed the IrPHY, IrLAP and IrLMP compliance tests.

The applications supported by an IrWW device include one or more of the following:
Time-Sync (bWatch)
Alarm (bWatch)
Timer (bWatch)
Stopwatch (bWatch)

1.3 Definitions and Acronyms

Alarm In a wristwatch, normally it means daily alarm function.
Daily alarm notifies the same setting time everyday.
One shot alarm notifies the setting time just once.

Timer Function to count down to the presetting period of time.

Stopwatch Function to count up with a second resolution or under a second besides displaying current time

Split Time Function to show elapsed time from the start point one by one while keeping count up.

1.4 References

[IrPHY] Serial Infrared Physical Layer Link Specification, IrPHY, Version 1.3, Infrared Data Association

[IrLAP] Serial Infrared Link Access Protocol, IrLAP, Version 1.1, Infrared Data Association

[IrLMP] Link Management Protocol, IrLMP, Version 1.1, Infrared Data Association

[IAS] Infrared Data Association, “IrLMP Service Hint Bit Assignments and Known IAS Definitions”, Version 1.0a 13th July1996

[TINYTP] Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, 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.

[IrWW1] IrWW Generic Access Profile, Version 0.98, Infrared Data Association.

[IrWW2] IrWW Time Synchronization Profile, Version 0.98, Infrared Data Association.

[IrWW3] IrWW Tiny Object Exchange Profile, Version 0.98, Infrared Data Association.

[GBO] Generic Binary Object for IrWW, Version 0.98, Infrared Data Association.

[Ultra IrWW] Ultra IrWW, Version 0.98, Infrared Data Association

2 Environment

2.1 Hardware

The following list includes examples of the type of hardware that is required:

- Device Under Test (DUT)
- PC or another device to generate test frames, with built-in IR port or an IR dongle
  e.g Actisys or Extended Systems
- IR Probe for tracing low level IR frames (this is optional, if everything works first time !)

2.2 Software

The following list includes examples of the type of software that is required:

- Source of OBEX Connection Oriented frames.
- IrWW Test program (PC) or built in Test Programs (IrDA Test Equipment).

3 Overview Compliance Tests

The test scenarios assume that the device under test (DUT) is a IrWW device, and that the Tester is software running on a PC.

3.1 What Tests Apply to my Product ?

Time Synchronization profile for legacy PC/PDA defines various applications and functions. These applications are optional. The tests that your product must pass, depends on the level of support that your product offers.

Application	Format	Single Pull	Single Push	Read-all/Write-all
Time-Sync	bWatch	Option	-	-
Alarm	bWatch	-	Option	Option
Timer	bWatch	-	Option	Option
Stopwatch	bWatch	-	Option	Option

3.1.1 Client/Server

The tests have been named according to whether the DUT is acting as a client or a server.

IrWW2_C_x_x_x_x	Client (initiates the data transfer)
IrWW2_S_x_x_x_x	Server (responds to requests for data transfer)

3.1.2 Protocol Tests

The protocol tests verify the basic IrWW support. All devices must pass these tests. The tests are named as **IrWW2_x_P_x_x_x**.

3.1.3 Read-all/Write-all Tests

The Read-all/Write-all tests verify that the DUT can put objects and get objects. The tests are named as **IrWW2_x_A_x_x_x**

3.1.4 Single Push Tests

The Single Push tests verify that the DUT correctly presents received objects to the user. It is recommended that devices which have a user interface, should pass these tests. The tests are named as **IrWW2_x_S_x_x_x**.

3.1.5 Single Pull Tests

The Single Pull tests verify that the DUT can get an object. It is recommended that devices which have a user interface, should pass these tests. The tests are named as **IrWW2_x_S_x_x_x**.

3.1.6 Objects

Time Synchronization Profile for legacy PC/PDA does not require that the device support any of the Time-Sync, Alarm, Timer or Stopwatch or objects.

The tests have been named according to the object they are accessing.

IrWW2_x_x_TSYNC_x_x	Time-Sync
IrWW2_x_x_ALARM_x_x	Alarm
IrWW2_x_x_TIMER_x_x	Timer
IrWW2_x_x_STOPW_x_x	Stopwatch

IrWW2_x_x_BXX_x_x	Independent of the object

If a device supports one of the object types, then it must pass the compliance tests for that object type. A device will fail the compliance tests if it fails the tests for one object type, even if it passes all the others. For example, a device which supports both Time-Sync and Alarm, must pass all of the Time-Sync and Alarm tests before the device can pass.

3.1.7 Miscellaneous Tests

The Miscellaneous tests verify that the general behavior of the DUT. The tests are named as **IrWW2_x_M_x_x**.

3.2 General Test Steps

The majority of the tests involve transferring OBEX objects between the DUT under test and the PC. The following steps should be performed.

1. Switch on the DUT under test, and activate the IR port if appropriate.
2. Switch on the PC, and enable the IR if appropriate.
3. Point the IrDA ports at each other. Ensure that they are within the operating distances defined for the devices. i.e. some devices only have a 20cm operating range.
4. The DUT should make an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l. If the test is a server test, then the Tester should make the request.
5. The DUT should make a connection to the LSAP returned in the response from the Get Value By Class request. If the test is a server test, then the Tester should make the connection.
6. Generate the OBEX GET/PUT requests from the PC package or DUT.
7. Verify that the data was transferred without any problems.

4 Protocol Tests

These tests are mandatory for all devices.

4.1 Discovery Hint Bits

The discovery hint bits should be supported.

4.1.1 Client

IrWW2_C_P_BXX_HINT_1 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the Watch Hint Bit is set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW2_C_P_BXX_HINT_2 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the Watch Hint Bit is not set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW2_C_P_BXX_HINT_3 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the OBEX Hint Bit is set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW2_C_P_BXX_HINT_4 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the OBEX Hint Bit is not set
2. General Test Steps 1-3
3. Make the DUT perform device discovery.
4. Verify that the DUT correctly decodes the hint bit.

4.1.2 Server

IrWW2_S_P_BXX_HINT_1 (Mandatory if Connection OBEX)
1. General Test Steps 1-3
2. Make the Tester perform device discovery
3. Verify that the Watch Hint Bit is set in the hint bits returned by the DUT (see the IAS specification [IAS].)

IrWW2_S_P_BXX_HINT_2 (Mandatory if Connection OBEX)

1. General Test Steps 1-3

4. Make the Tester perform device discovery

2. Verify that the OBEX Hint Bit is set in the hint bits returned by the DUT (see the IAS specification [IAS].)

4.2 IAS Entries

The IAS entries should be supported.

4.2.1 Client

IrWW2_C_P_BXX_IAS_1 (Mandatory if Connection)

1. General Test Steps 1-3
2. Make the DUT establish an IAS connection to the Tester.
3. Make the DUT generate an IAS *Get Value By Class* request, for the *IrDA:WATCH* class, attribute *Parameter*s and *DeviceInfo*.
4. Verify that the DUT correctly decodes the PI/PL/PV triples for the supported applications. Note, the IAS values define the behavior of the Tester as a server.
5. This is the minimum level of support required by IAS. If the Tester supports any of the optional requests, then these must be verified too.

IrWW2_C_P_BXX_IAS_2 (Mandatory if Connection OBEX as client)

1. General Test Steps 1-3
2. Make the DUT establish an IAS connection to the Tester.
3. Make the DUT generate an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l. Note, the class OBEX should be guaranteed, as this was in the original OBEX specification. However, in practice some applications use the class IrDA:OBEX which was specified in the OBEX errata, and OBEX:IrXfer is used by Microsoft. It is implementation specific if classes other than OBEX are requested.
4. Verify that the DUT correctly decodes the integer value returned for the LSAP of the OBEX connection on the Tester.

4.2.2 Server

IrWW2_S_P_BXX_IAS_1 (Mandatory if Connection)

1. General Test Steps 1-3
2. Make the Tester establish an IAS connection to the DUT.
3. Make the Tester generate an IAS *Get Value By Class* request, for the *IrDA:WATCH* class, attribute *Parameter*s and *DeviceInfo*.
4. Verify that the DUT correctly generates the PI/PL/PV triples for the supported applications. Note, the IAS values define the behavior of the DUT as a server.
5. This is the minimum level of support required by IAS. If the DUT supports any of the optional requests, then these must be verified too.

IrWW2_S_P_BXX_IAS_2 (Mandatory if Connection OBEX)

1. General Test Steps 1-3
2. Make the Tester establish an IAS connection to the DUT.
3. Make the Tester generate an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l.
4. Verify that the IAS data received contains the integer value for the LSAP of the OBEX connection within the DUT.
5. If the DUT supports other classes for the OBEX connection, these should be verified as well.

4.3 Devinfo

The devinfo.bin object should be supported.

4.3.1 Client

IrWW2_C_P_BXX_DEVINFO_1 (Optional)

1. General Test Steps 1-5
2. Make the DUT request the /watch/devinfo.bin object from the Tester.
3. Verify that object is received correctly on the DUT. Verify that the DUT correctly parses the fields that it supports.

4.3.2 Server

IrWW2_S_P_BXX_DEVINFO_1 (Mandatory if Connection OBEX)

1. General Test Steps 1-5
2. Make the Tester request the /watch/devinfo.bin object from the DUT.
3. Verify that object is received correctly on the Tester.
4. Verify that the mandatory fields for the manufacturer and model are present.

4.4 Info Logs

The Information Log objects should be supported.

4.4.1 Client

IrWW2_C_P_BXX_INFOLOG_1 (Optional)

1. General Test Steps 1-5
2. Make the DUT request the /watch/alarm/info.log object from the Tester
3. Make the DUT request the /watch/timer/info.log object from the Tester
4. Make the DUT request the /watch/stop/info.log object from the Tester
5. Verify that the appropriate info logs are received correctly on the DUT. Verify that the DUT correctly parses the fields that it supports.

4.4.2 Server

IrWW2_S_P_BXX_INFOLOG_1 (Optional)

1. General Test Steps 1-5
2. Make the Tester request the /watch/alarm/info.log object from the DUT.
3. Make the Tester request the /watch/timer/info.log object from the DUT.
4. Make the Tester request the /watch/stop/info.log object from the DUT.
5. Verify that the appropriate info logs are received correctly on the Tester.

5 Read-all/Write-all Tests

The OBEX PUT should be with an OBEX filename like */watch/alarm.wal*.

5.1 Client

IrWW2_C_A_BXX_PUT_1 (Mandatory if the DUT supports Write-all as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a stream of objects (alarms etc) using OBEX PUT.
3. Verify that the Tester received the stream of objects correctly

IrWW2_C_A_BXX_PUT_2 (Mandatory if the DUT supports Write-all as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a stream of objects (alarms etc) using OBEX PUT. The source should be full, so the maximum sized stream is created. For example, if the DUT can hold 100 alarms, then the stream of objects should be filled with 100 alarms for this test.
3. Verify that the Tester received the stream of objects correctly

IrWW2_C_A_BXX_GET_1 (Mandatory if the DUT supports Read-all/Write-all GET as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a request for an object which contains a stream of entries (alarm etc using /watch/alarm.wal) using OBEX GET.
3. Verify that the DUT received the object correctly from the Tester.

IrWW2_C_P_BXX_PUT_3 (Optional as delete mechanism not required as a client)

1. General Test Steps 1-5
2. Make the DUT transmit an object (alarm etc) using OBEX PUT at Write-all. The OBEX PUT should have BODY which includes only an object stream identifier.
3. Verify that the Tester deleted the entire data store.

5.2 Server

IrWW2_S_A_BXX_PUT_1 (Mandatory if the DUT supports Write-all as a Server)

1. General Test Steps 1-5
2. Enter some data into the data store of the DUT (create a few alarms)
3. Make the Tester transmit a stream of objects (alarms etc) using OBEX PUT.
4. Verify that the DUT received, and stored, the stream of objects correctly. Verify that the original information was overwritten.

IrWW2_S_A_BXX_PUT_2 (Mandatory if the DUT supports Write-all as a Server)

1. General Test Steps 1-5
2. Make the Tester transmit a stream of objects (alarms etc) using OBEX PUT. The stream of objects should be as large as the DUT can receive.
3. Verify that the DUT received, and stored, the stream of objects correctly.

IrWW2_S_A_BXX_GET_1 (Mandatory if the DUT supports Read-all/Write-all as a Server)

1. General Test Steps 1-5
2. Enter some data into the data store of the DUT (create a few alarms).
3. Make the Tester transmit a read request for a stream of objects (alarms etc) using OBEX GET.
4. Verify that the Tester received the stream of objects correctly.

IrWW2_S_P_BXX_PUT_3 (Mandatory if the DUT supports Write-all)

1. General Test Steps 1-5
2. Make the Tester transmit an object (alarm etc) using OBEX PUT from the Tester at Write-all. The

OBEX PUT should have BODY which includes only an object stream identifier.

3. Verify that the DUT deleted the entire data store. It is implementation specific whether a zero size file is created.

6 Single Push Tests

Any DUT that has a user interface should pass these tests.
The OBEX PUT should be with an OBEX filename like *abcde.wal*

6.1 Client

IrWW2_C_S_BXX_PUT_1 (Mandatory if the DUT supports Single Push as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit an object (alarm, timer or stopwatch) using OBEX PUT.
3. Verify that the Tester received the object correctly.

6.2 Server

IrWW2_S_S_BXX_PUT_1 (Mandatory if the DUT has a user interface)

1. General Test Steps 1-5
2. Make the Tester transmit a single object (alarm, timer or stopwatch) using OBEX PUT.
3. Verify that the DUT received the object correctly, and if appropriate, that the user was prompted whether to accept or reject the object.

7 Single Pull Tests

Any DUT that has a user interface should pass these tests.
The OBEX GET should be with an OBEX filename like *abcde.wsy*

7.1 Client

IrWW2_C_S_BXX_GET_1 (Mandatory if the DUT supports Single Pull as a Client)

1. General Test Steps 1-5
2. Make the DUT receive an object (Time-Sync) using OBEX GET.
3. Verify that the DUT received the object correctly.

7.2 Server

IrWW2_S_S_BXX_GET_1 (Mandatory if the DUT has a user interface)

1. General Test Steps 1-5
2. Make the Tester receive a single object (Time-Sync) using OBEX GET.
3. Verify that the Tester received the object correctly, and if appropriate, that the user was prompted whether to accept or reject the object.

8 Error Responses Tests

The error conditions should be generated, and the return codes verified.
Some devices may not support all of the error codes. For example, the *Restricted Access* code is only appropriate for devices, which support limited access to some of the entries.

8.1 Client

No Tests.

8.2 Server

IrWW2_S_P_BXX_RSP_1 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (alarm etc) using OBEX PUT from the Tester to a restricted access location.
3. Verify that the DUT returns the “Unauthorised” response code 0xC1

IrWW2_S_P_BXX_RSP_2 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (alarm etc) using OBEX PUT from the Tester to a non-existence IrWW object.
3. Verify that the DUT returns the “Not Found” response code 0xC4

IrWW2_S_P_BXX_RSP_3 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (alarm etc) using OBEX PUT from the Tester to a locked object store.
3. Verify that the DUT returns the “Database Locked” response code 0xE1

IrWW2_S_P_BXX_RSP_4 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (alarm etc) using OBEX PUT from the Tester to an object store which is out of space.
3. Verify that the DUT returns the “Database Full” response code 0xE0

9 Miscellaneous Tests

9.1 Contents

9.1.1 Client

IrWW2_C_M_ALARM_CONTENTS_1 (Mandatory if the DUT supports Alarm)
1. General Test Steps 1-5
2. Make the DUT transmit an alarm object request (bWatch) using OBEX GET at read-all.
3. Verify that the DUT received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:ALARM and END: ALARM.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_C_M_TIMER_CONTENTS_1 (Mandatory if the DUT supports Timer)
1. General Test Steps 1-5
2. Make the DUT transmit an timer object request (bWatch) using OBEX GET at read-all.
3. Verify that the DUT received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:TIMER and END: TIMER.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_C_M_STOPW_CONTENTS_1 (Mandatory if the DUT supports Stopwatch)
1. General Test Steps 1-5
2. Make the DUT transmit an stopwatch object request (bWatch) using OBEX GET at read-all.
3. Verify that the DUT received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:STOPWATCH and END: STOPWATCH.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_C_M_TSYNC_CONTENTS_1 (Mandatory if the DUT supports Time-Sync)
1. General Test Steps 1-5
2. Make the DUT transmit an Time-Sync object request (bWatch) using OBEX GET at single pull.
3. Verify that the DUT received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:T-SYNC and END: T-SYNC.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

9.1.2 Server

IrWW2_S_M_ALARM_CONTENTS_1 (Mandatory if the DUT supports Alarm)
1. General Test Steps 1-5
2. Make the Tester transmit an alarm object request (bWatch) using OBEX GET at read-all.
3. Verify that the Tester received the bWatch correctly.

4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:ALARM and END: ALARM.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_S_M_TIMER_CONTENTS_1 (Mandatory if the DUT supports Timer)
1. General Test Steps 1-5
2. Make the Tester transmit an timer object request (bWatch) using OBEX GET at read-all.
3. Verify that the Tester received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:TIMER and END: TIMER.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_S_M_STOPW_CONTENTS_1 (Mandatory if the DUT supports Stopwatch)
1. General Test Steps 1-5
2. Make the Tester transmit an stopwatch object request (bWatch) using OBEX GET at read-all.
3. Verify that the Tester received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:STOPWATCH and END: STOPWATCH.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

IrWW2_S_M_TSYNC_CONTENTS_1 (Mandatory if the DUT supports Time-Sync)
1. General Test Steps 1-5
2. Make the Tester transmit an Time-Sync object request (bWatch) using OBEX GET at single pull.
3. Verify that the Tester received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:T-SYNC and END: T-SYNC.
8. Check that the bWatch contains the mandatory fields specified in [IrWW2].

10 Test Result Template

Testing DATE:	Testing Environments
DUT(Device Under Test)	Test Tools or System:
Product Company:	
Product Model:	Hardware:
Product Revision:	Software:
Product Sample Serial No.	
Remarks:	

Test Name	Pass/Fail	Comments
IrWW2_C_P_BXX_HINT_1		
IrWW2_C_P_BXX_HINT_2		
IrWW2_C_P_BXX_HINT_3		
IrWW2_C_P_BXX_HINT_4		
IrWW2_S_P_BXX_HINT_1		
IrWW2_S_P_BXX_HINT_2		
IrWW2_C_P_BXX_IAS_1		
IrWW2_C_P_BXX_IAS_2		
IrWW2_S_P_BXX_IAS_1		
IrWW2_S_P_BXX_IAS_2		
IrWW2_C_P_BXX_DEVINFO_1		
IrWW2_S_P_BXX_DEVINFO_1		
IrWW2_C_P_BXX_INFOLOG_1		
IrWW2_S_P_BXX_INFOLOG_1		
IrWW2_C_A_BXX_PUT_1		
IrWW2_C_A_BXX_PUT_2		
IrWW2_C_A_BXX_GET_1		
IrWW2_S_A_BXX_PUT_1		
IrWW2_S_A_BXX_PUT_2		
IrWW2_S_A_BXX_GET_1		
IrWW2_C_S_BXX_PUT_1		
IrWW2_S_S_BXX_PUT_1		
IrWW2_C_S_BXX_GET_1		
IrWW2_S_S_BXX_GET_1		
IrWW2_S_P_BXX_RSP_1		
IrWW2_S_P_BXX_RSP_2		
IrWW2_S_P_BXX_RSP_3		
IrWW2_S_P_BXX_RSP_4		
IrWW2_C_P_BXX_DEL_1		
IrWW2_S_P_BXX_DEL_1		
IrWW2_S_P_BXX_DEL_2		
IrWW2_C_M_ALARM_CONTENTS_1		
IrWW2_C_M_TIMER_CONTENTS_1		
IrWW2_C_M_STOPW_CONTENTS_1		

IrWW2_C_M_TSYNC_CONTENTS_1		
IrWW2_S_M_ALARM_CONTENTS_1		
IrWW2_S_M_TIMER_CONTENTS_1		
IrWW2_S_M_STOPW_CONTENTS_1		
IrWW2_S_M_TSYNC_CONTENTS_1		

第8部

簡易オブジェクト交換プロファイル相互接続性試験

Infrared Data Association

IrWW (Infrared Wrist Watch)

Tiny Object Exchange profile Compliance Tests

Version 1.0

December 26, 1999

NTT
Okaya Systemware
CASIO
CITIZEN
SII

Document Status

Document Owner:

Ken-ichiro Shimokura, k.shimokura@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Yuichiro “Tuck” Takagawa, y.takagawa@east.ntt.co.jp
(NTT Nippon Telegraph and Telephone Corp)
Gontaro Kitazumi, kitazumi@osw.co.jp (Okaya Systemware Co., Ltd.)
Satomi Suzuki, suzuki@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Hiroshi Morohoshi, h-moroho@tmgw.rd.casio.co.jp (Casio Computer Co., Ltd.)
Michihiro Enokida, enokida@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroyuki Kihara, kihara@citizen.co.jp (Citizen Watch Co., Ltd.)
Hiroshi Odagiri, hiroshi.odagiri@sii.co.jp (Seiko Instruments Inc)
Tomozumi Saruwatari, saruwatari@sii.co.jp (Seiko Instruments Inc)

Document Editors:

Hiroshi Morohoshi h-moroho@tmgw.rd.casio.co.jp

Significant Contributors:

History:

Version 1.0: IrWW Tiny Object Exchange Profile Compliance Tests for Final Vote.

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 INTRODUCTION 3

1.1 Purpose 3
1.2 Scope 3
1.3 Background & Motivation for the Document 3
1.4 References 3

2 ENVIRONMENT 5

2.1 Hardware 5
2.2 Software 5

3 OVERVIEW COMPLIANCE TESTS 6

3.1 What Tests Apply to my Product ? 6
3.1.1 Client/Server 6
3.1.2 Protocol Tests 6
3.1.3 Read-all/Write-all Tests 6
3.1.4 Single Push Tests 6
3.1.5 Objects 6
3.1.6 Miscellaneous Tests 7
3.2 General Test Steps 7

4 PROTOCOL TESTS 8

4.1 Discovery Hint Bits 8
4.1.1 Client 8
4.1.2 Server 8
4.2 IAS Entries 8
4.2.1 Client 8
4.2.2 Server 9
4.3 Devinfo 9
4.3.1 Client 9
4.3.2 Server 10
4.4 Info Logs 10
4.4.1 Client 10
4.4.2 Server 10

5 READ-ALL/WRITE-ALL TESTS 11

5.1 Client 11
5.2 Server 11

6 SINGLE PUSH TESTS 13

6.1 Client 13
6.2 Server 13

7 ERROR RESPONSES TESTS 14

7.1 Client 14
7.2 Server 14

8 MISCELLANEOUS TESTS 15

8.1 Contents 15
8.1.1 Client 15
8.1.2 Server 15

9 TEST RESULT TEMPLATE 17

1 Introduction

1.1 Purpose

This document specifies the test cases used to verify an IrWW device. The goal of this document is to ensure interoperability between IrWW devices and products.
The purpose of this document is to define the minimum behaviors that should be required of a device, in order to be declared IrWW compliant.

1.2 Scope

The tests described in this document do not aim to verify the lower layers of the IrDA stack. The tests target the new application layers like T-log. This specification does not require that the device has already passed the IrPHY, IrLAP and IrLMP compliance tests.

The applications supported by an IrWW device include one or more of the following:
Time-Log (bWatch)
Business Card (bvCard)
Schedule (bvCalendar)
To do (bvCalendar)
Notes (bvNote)

1.3 Background & Motivation for the Document

There is a big difference between supporting IrWW Tiny Object Exchange Profile, and being interoperable. IrWW Tiny Object Exchange Profile and the specifications that it references, have a large number of optional fields. Even the mandatory fields are not always supported in commercially available products.

1.4 References

[IrPHY] Serial Infrared Physical Layer Link Specification, IrPHY, Version 1.3, Infrared Data Association

[IrLAP] Serial Infrared Link Access Protocol, IrLAP, Version 1.1, Infrared Data Association

[IrLMP] Link Management Protocol, IrLMP, Version 1.1, Infrared Data Association

[IAS] Infrared Data Association, “IrLMP Service Hint Bit Assignments and Known IAS Definitions”, Version 1.0a 13th July1996

[TINYTP] Tiny TP: A Flow Control Mechanism for use with IrLMP, Version 1.1, 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.

[IrWW1] IrWW Generic Access Profile, Version 0.98, Infrared Data Association.

[IrWW2] IrWW Time Synchronization Profile, Version 0.98, Infrared Data Association.

[IrWW3] IrWW Tiny Object Exchange Profile, Version 0.98, Infrared Data Association.

[GBO] Generic Binary Object for IrWW, Version 0.98, Infrared Data Association.

[Ultra IrWW] Ultra Protocols for IrWW, Version 0.98, Infrared Data Association

[IrWWTEST2] IrWW Time Synchronization Profile Compliance Tests, Version 0.98, Infrared Data Association

2 Environment

2.1 Hardware

The following list includes examples of the type of hardware that is required:

- Device Under Test (DUT)
- PC or another device to generate test frames, with built-in IR port or an IR dongle
  e.g Actisys or Extended Systems
- IR Probe for tracing low level IR frames (this is optional, if everything works first time !)

2.2 Software

The following list includes examples of the type of software that is required:

- Source of OBEX Connection Oriented frames.
- IrWW Test program (PC) or Built in Test programs (IrDA Test Equipment).

3 Overview Compliance Tests

The test scenarios assume that the device under test (DUT) is a mobile device, and that the Tester is software running on a PC.

3.1 What Tests Apply to my Product ?

All IrWW devices must support Time-Sync application of [IrWW2]. About the Time Synchronization compliance tests, please refer to [IrWWTEST1].

IrWW Tiny Object Exchange defines various applications and functions. These applications are optional. The tests that your product must pass, depends on the level of support that your product offers.

Application	Format	Single Push Function	Read-all/Write-all Function
Time-Log	bWatch	Option	Option
Business Card	bvCard	Option	Option
Schedule	bvCalendar	Option	Option
To do	bvCalendar	Option	Option
Notes	bvNote	Option	Option

3.1.1 Client/Server

The tests have been named according to whether the DUT is acting as a client or a server.

IrWW3_C_x_x_x_x	Client (initiates the data transfer)
IrWW3_S_x_x_x_x	Server (responds to requests for data transfer)

3.1.2 Protocol Tests

The protocol tests verify the basic IrWW support. All devices must pass these tests. The tests are named as **IrWW3_x_P_x_x_x**.

3.1.3 Read-all/Write-all Tests

The Read-all/Write-all tests verify that the DUT can put objects and get objects. The tests are named as **IrWW3_x_A_x_x_x**

3.1.4 Single Push Tests

The Single Push tests verify that the DUT correctly presents received objects to the user. It is recommended that devices which have a user interface, should pass these tests. The tests are named as **IrWW3_x_U_x_x_x**.

3.1.5 Objects

IrWW Tiny Object Profile does not require that the device support any of the Time-Log, Business Card, Schedule, To do or Notes objects.

The tests have been named according to the object they are accessing.

IrWW3_x_x_TLOG_x_x	Time-Log
IrWW3_x_x_OWNER_x_x	Owner's Business Card
IrWW3_x_x_CARD_x_x	Business Card
IrWW3_x_x_SCHE_x_x	Schedule
IrWW3_x_x_TODO_x_x To do	
IrWW3_x_x_NOTE_x_x	Notes
IrWW3_x_x_BXX_x_x	Independent of the object

If a device supports one of the object types, then it must pass the compliance tests for that object type. A device will fail the compliance tests if it fails the tests for one object type, even if it passes all the others. For example, a device which supports both Business Card and Schedule, must pass all of the Business Card and Schedule tests before the device can pass.

3.1.6 Miscellaneous Tests

The Miscellaneous tests verify that the general behavior of the DUT. The tests are named as **IrWW3_x_M_x_x**.

3.2 General Test Steps

The majority of the tests involve transferring OBEX objects between the DUT under test and the PC. The following steps should be performed.

1. Switch on the DUT under test, and activate the IR port if appropriate.
2. Switch on the PC, and enable the IR if appropriate.
3. Point the IrDA ports at each other. Ensure that they are within the operating distances defined for the devices. i.e. some devices only have a 20cm operating range.
4. The DUT should make an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l. If the test is a server test, then the Tester should make the request.
5. The DUT should make a connection to the LSAP returned in the response from the Get Value By Class request. If the test is a server test, then the Tester should make the connection.
6. Generate the OBEX GET/PUT requests from the PC package or DUT.
7. Verify that the data was transferred without any problems.

4 Protocol Tests

These tests are mandatory for all devices.

4.1 Discovery Hint Bits

The discovery hint bits should be supported.

4.1.1 Client

IrWW3_C_P_BXX_HINT_1 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the Watch Hint Bit is set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW3_C_P_BXX_HINT_2 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the Watch Hint Bit is not set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW3_C_P_BXX_HINT_3 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the OBEX Hint Bit is set
2. General Test Steps 1-3
3. Make the DUT perform device discovery
4. Verify that the DUT correctly decodes the hint bit.

IrWW3_C_P_BXX_HINT_4 (Mandatory if the DUT checks the Hint Bits)
1. Configure the Tester so that the OBEX Hint Bit is not set
2. General Test Steps 1-3
3. Make the DUT perform device discovery.
4. Verify that the DUT correctly decodes the hint bit.

4.1.2 Server

IrWW3_S_P_BXX_HINT_1 (Mandatory if Connection OBEX)
1. General Test Steps 1-3
2. Make the Tester perform device discovery
3. Verify that the Watch Hint Bit is set in the hint bits returned by the DUT (see the IAS specification [IAS].)

IrWW3_S_P_BXX_HINT_2 (Mandatory if Connection OBEX)

1. General Test Steps 1-3
4. Make the Tester perform device discovery
2. Verify that the OBEX Hint Bit is set in the hint bits returned by the DUT (see the IAS specification [IAS].)

4.2 IAS Entries

The IAS entries should be supported.

4.2.1 Client

IrWW3_C_P_BXX_IAS_1 (Mandatory if Connection)

1. General Test Steps 1-3
2. Make the DUT establish an IAS connection to the Tester.
3. Make the DUT generate an IAS *Get Value By Class* request, for the *IrDA:WATCH* class, attribute *Parameter*s and *DeviceInfo*.
4. Verify that the DUT correctly decodes the PI/PL/PV triples for the supported applications. Note, the IAS values define the behavior of the DUT as a server.
5. This is the minimum level of support required by IAS. If the DUT supports any of the optional requests, then these must be verified too.

IrWW3_C_P_BXX_IAS_2 (Mandatory if Connection OBEX as client)

1. General Test Steps 1-3
2. Make the DUT establish an IAS connection to the Tester.
3. Make the DUT generate an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l. Note, the class OBEX should be guaranteed, as this was in the original OBEX specification. However, in practice some applications use the class IrDA:OBEX which was specified in the OBEX errata, and OBEX:IrXfer is used by Microsoft. It is implementation specific if classes other than OBEX are requested.
4. Verify that the DUT correctly decodes the integer value returned for the LSAP of the OBEX connection on the Tester.

4.2.2 Server

IrWW3_S_P_BXX_IAS_1 (Mandatory if Connection)

1. General Test Steps 1-3
2. Make the Tester establish an IAS connection to the DUT.
3. Make the Tester generate an IAS *Get Value By Class* request, for the *IrDA:WATCH* class, attribute *Parameter*s and *DeviceInfo*.
4. Verify that the DUT correctly generates the PI/PL/PV triples for the supported applications. Note, the IAS values define the behavior of the DUT as a server.
5. This is the minimum level of support required by IAS. If the DUT supports any of the optional requests, then these must be verified too.

IrWW3_S_P_BXX_IAS_2 (Mandatory if Connection OBEX)

1. General Test Steps 1-3
2. Make the Tester establish an IAS connection to the DUT.
3. Make the Tester generate an IAS *Get Value By Class* request, for the *OBEX* class, and attribute *IrDA:TinyTP:LsapSe*l.
4. Verify that the IAS data received contains the integer value for the LSAP of the OBEX connection within the DUT.
5. If the DUT supports other classes for the OBEX connection, these should be verified as well.

4.3 Devinfo

The devinfo.bin object should be supported.

4.3.1 Client

IrWW3_C_P_BXX_DEVINFO_1 (Optional)

1. General Test Steps 1-5
2. Make the DUT request the /watch/devinfo.bin object from the Tester.
3. Verify that object is received correctly on the DUT. Verify that the DUT correctly parses the fields that it supports.

4.3.2 Server

IrWW3_S_P_BXX_DEVINFO_1 (Mandatory if Connection OBEX)

1. General Test Steps 1-5
2. Make the Tester request the /watch/devinfo.bin object from the DUT.
3. Verify that object is received correctly on the Tester.
4. Verify that the mandatory fields for the manufacturer and model are present.

4.4 Info Logs

The Information Log objects should be supported.

4.4.1 Client

IrWW3_C_P_BXX_INFOLOG_1 (Optional)

1. General Test Steps 1-5
2. Make the DUT request the /watch/t-log/info.log object from the Tester
3. Make the DUT request the /watch/vlog/xxx/yyy/info.log object from the Tester
   "xxx" is a vender name. "yyy" is a log name.
4. Make the DUT request the /watch/pim/bc/info.log object from the Tester
5. Make the DUT request the /watch/pim/sd/info.log object from the Tester
6. Make the DUT request the /watch/pim/td/info.log object from the Tester
7. Make the DUT request the /watch/pim/nt/info.log object from the Tester
8. Verify that the appropriate info logs are received correctly on the DUT. Verify that the DUT correctly parses the fields that it supports.

4.4.2 Server

IrWW3_S_P_BXX_INFOLOG_1 (Optional)

1. General Test Steps 1-5
2. Make the Tester request the /watch/t-log/info.log object from the DUT.
3. Make the Tester request the /watch/vlog/xxx/yyy/info.log object from the DUT. "xxx" is a vender name. "yyy" is a log name.
4. Make the Tester request the /watch/pim/bc/info.log object from the DUT.
5. Make the Tester request the /watch/pim/sd/info.log object from the DUT.
6. Make the Tester request the /watch/pim/td/info.log object from the DUT.
7. Make the Tester request the /watch/pim/nt/info.log object from the DUT.
8. Verify that the appropriate info logs are received correctly on the Tester. Verify the fields returned.

5 Read-all/Write-all Tests

The OBEX PUT should be with an OBEX filename like */watch/pim/bc.bcf*.

5.1 Client

IrWW3_C_A_BXX_PUT_1 (Mandatory if the DUT supports Write-all as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a stream of objects (bvCards etc) using OBEX PUT. If possible, the stream should include empty entries.
3. Verify that the Tester received the stream of objects correctly

IrWW3_C_A_BXX_PUT_2 (Mandatory if the DUT supports Write-all as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a stream of objects (bvCards etc) using OBEX PUT. The stream should be empty.
3. Verify that the Tester received the stream of objects correctly

IrWW3_C_A_BXX_PUT_3 (Mandatory if the DUT supports Write-all as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a stream of objects (bvCards etc) using OBEX PUT. The source should be full, so the maximum sized stream is created. For example, if the business card can hold 100 entries, then the business card should be filled with 100 entries for this test. If possible, the stream should include empty entries.
3. Verify that the Tester received the stream of objects correctly

IrWW3_C_A_BXX_GET_1 (Mandatory if the DUT supports Read-all/Write-all GET as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit a request for an object which contains a stream of entries (bvCard etc using /watch/pim/bc.bcf) using OBEX GET.
3. Verify that the DUT received the object correctly from the Tester. Verify that all locations are correct, including any empty entries if they were present on the Tester.

5.2 Server

IrWW3_S_A_BXX_PUT_1 (Mandatory if the DUT supports Write-all as a Server)

1. General Test Steps 1-5
2. Enter some data into the data store of the DUT (create a few bvCards)
3. Make the Tester transmit a stream of objects (bvCards etc) using OBEX PUT. If possible, the stream should include empty entries.
4. Verify that the DUT received, and stored, the stream of objects correctly. Verify that the original information was overwritten.

IrWW3_S_A_BXX_PUT_2 (Mandatory if the DUT supports Write-all as a Server)

1. General Test Steps 1-5
2. Make the Tester transmit a stream of objects (bvCards etc) using OBEX PUT. The stream should be empty.
3. Verify that the DUT received, and stored, the stream of objects correctly.

IrWW3_S_A_BXX_PUT_3 (Mandatory if the DUT supports Write-all as a Server)

1. General Test Steps 1-5
2. Make the Tester transmit a stream of objects (bvCards etc) using OBEX PUT. The stream of objects should be as large as the DUT can receive. If possible, the stream should include empty entries.
3. Verify that the DUT received, and stored, the stream of objects correctly.

IrWW3_S_A_BXX_GET_1 (Mandatory if the DUT supports Read-all/Write-all as a Server)

1. General Test Steps 1-5
2. Enter some data into the data store of the DUT (create a few bvCards). If possible, the data store should include empty entries.
3. Make the Tester transmit a read request for a stream of objects (bvCards etc) using OBEX GET.
4. Verify that the Tester received the stream of objects correctly. If the DUT supports Single Push, the stream should contain empty entries for the entries which are blank. Otherwise, the DUT can just send the non-blank entries

6 Single Push Tests

Any DUT that has a user interface should pass these tests.
The OBEX PUT should be with an OBEX filename like *james.bcf*

6.1 Client

IrWW3_C_U_BXX_PUT_1 (Mandatory if the DUT supports Single Push as a Client)

1. General Test Steps 1-5
2. Make the DUT transmit an object (bvCard etc) using OBEX PUT.
3. Verify that the Tester received the object correctly.

6.2 Server

IrWW3_S_U_BXX_PUT_1 (Mandatory if the DUT has a user interface)

1. General Test Steps 1-5
2. Make the Tester transmit a single object (bvCard etc) using OBEX PUT.
3. Verify that the DUT received the object correctly, and if appropriate, that the user was prompted whether to accept or reject the object.

7 Error Responses Tests

The error conditions should be generated, and the return codes verified.
Some devices may not support all of the error codes. For example, the *Restricted Access* code is only appropriate for devices, which support limited access to some of the entries.

7.1 Client

No Tests.

7.2 Server

IrWW3_S_P_BXX_RSP_1 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (bvCard etc) using OBEX PUT from the Tester to a restricted access location.
3. Verify that the DUT returns the “Unauthorised” response code 0xC1

IrWW3_S_P_BXX_RSP_2 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (bvCard etc) using OBEX PUT from the Tester to a non-existence IrWW object.
3. Verify that the DUT returns the “Not Found” response code 0xC4

IrWW3_S_P_BXX_RSP_3 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (bvCard etc) using OBEX PUT from the Tester to a locked object store.
3. Verify that the DUT returns the “Database Locked” response code 0xE1

IrWW3_S_P_BXX_RSP_4 (Optional)

1. General Test Steps 1-5
2. Make the Tester transmit an object (bvCard etc) using OBEX PUT from the Tester to an object store which is out of space.
3. Verify that the DUT returns the “Database Full” response code 0xE0

8 Miscellaneous Tests

8.1 Contents

8.1.1 Client

IrWW3_C_M_TLOG_CONTENTS_1 (Mandatory if the DUT supports T-log)
1. General Test Steps 1-5
2. Make the DUT transmit an object request (bWatch) using OBEX PUT at write-all.
3. Verify that the Tester received the bWatch correctly.

IrWW3_C_M_OWNER_ CONTENTS_1 (Mandatory if the DUT supports Owner's Business Card)
1. General Test Steps 1-5
2. Make the DUT transmit the Owner's Business Card (bvCard), by selecting the appropriate menu/button. The vCard should be sent by OBEX PUT to the remote InBox.
3. Verify that the Tester received the bvCard correctly.

IrWW3_C_M_CARD_CONTENTS_1 (Mandatory if the DUT supports Business Card)
1. General Test Steps 1-5
2. Make the DUT transmit an object request (bvCard) using OBEX PUT at write-all.
3. Verify that the Tester received the bvCard correctly.

IrWW3_C_M_SCHE_CONTENTS_1 (Mandatory if the DUT supports Schedule)
1. General Test Steps 1-5
2. Make the DUT transmit an object request (bvCalendar) using OBEX PUT at write-all.
3. Verify that the Tester received the bvCalendar correctly.

IrWW3_C_M_TODO_CONTENTS_1 (Mandatory if the DUT supports To do)
1. General Test Steps 1-5
2. Make the DUT transmit an object request (bvCalendar) using OBEX PUT at write-all.
3. Verify that the Tester received the bvCalendar correctly.

IrWW3_C_M_NOTE_CONTENTS_1 (Mandatory if the DUT supports Notes)
1. General Test Steps 1-5
2. Make the DUT transmit an object request (bvNote) using OBEX PUT at write-all.
3. Verify that the Tester received the bvNote correctly.

8.1.2 Server

IrWW3_S_M_TLOG_CONTENTS_1 (Mandatory if the DUT supports T-log)
1. General Test Steps 1-5
2. Make the Tester transmit an object request (bWatch) using OBEX GET at read-all.
3. Verify that the Tester received the bWatch correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bWatch is encapsulated with BEGIN:BWATCH and END: BWATCH.
6. Check that there is the mandatory field VERSION after BEGIN: BWATCH.
7. Check that the information is encapsulated with BEGIN:T-LOG and END: T-LOG.
8. Check that the bWatch contains the mandatory fields specified in [IrWW3].
9. Check that the bvCalendars are received with an overall BEGIN: BWATCH and END: BWATCH.

IrWW3_S_M_OWNER_CONTENTS_1 (Mandatory if the DUT supports Owner's Business Card)

1. General Test Steps 1-5
2. Make the Tester transmit an object request (bvCard) using OBEX GET at read-all.
3. Verify that the Tester received the bvCard correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bvCard is encapsulated with BEGIN:VCARD and END:VCARD.
6. Check that there is the mandatory field VERSION.
7. Check that the vCard contains the mandatory fields specified in [IrWW3].

IrWW3_S_M_CARD_CONTENTS_1 (Mandatory if the DUT supports Business Card)

1. General Test Steps 1-5
2. Make the Tester transmit an object request (bvCard) using OBEX GET at read-all.
3. Verify that the Tester received the bvCard correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bvCard is encapsulated with BEGIN:VCARD and END:VCARD.
6. Check that there is the mandatory field VERSION.
7. Check that the vCard contains the mandatory fields specified in [IrWW3].
8. Check that the vCards are received as separate objects.

IrWW3_S_M_SCHE_CONTENTS_1 (Mandatory if the DUT supports Schedule)

1. General Test Steps 1-5
2. Make the Tester transmit an object request (bvCalendar) using OBEX GET at read-all.
3. Verify that the Tester received the bvCalendar correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bvCalendar is encapsulated with BEGIN:VCALENDAR and END:VCALENDAR.
6. Check that there is the mandatory field VERSION.
7. Check that the information is encapsulated with BEGIN:VEVENT and END:VEVENT.
8. Check that the bvCalendar contains the mandatory fields specified in [IrWW3].
9. Check that the bvCalendars are received with an overall BEGIN:VCALENDAR and END:VCALENDAR.

IrWW3_S_M_TODO_CONTENTS_1 (Mandatory if the DUT supports To do)

1. General Test Steps 1-5
2. Make the Tester transmit an object request (bvCalendar) using OBEX GET at read-all.
3. Verify that the Tester received the bvCalendar correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bvCalendar is encapsulated with BEGIN:VCALENDAR and END:VCALENDAR.
6. Check that there is the mandatory field VERSION.
7. Check that the information is encapsulated with BEGIN:VTODO and END:VTODO.
8. Check that the bvCalendar contains the mandatory fields specified in [IrWW3].
9. Check that the bvCalendars are received with an overall BEGIN:VCALENDAR and END:VCALENDAR.

IrWW3_S_M_NOTE_CONTENTS_1 (Mandatory if the DUT supports Notes)

1. General Test Steps 1-5
2. Make the Tester transmit an object request (bvNote) using OBEX GET at read-all.
3. Verify that the Tester received the bvNote correctly.
4. Check that there is object stream identifier in the beginning.
5. Check that the bvNote is encapsulated with BEGIN:VNOTE and END:VNOTE
6. Check that there is the mandatory field VERSION after BEGIN:VNOTE.
7. Check that the information begins with the tag BODY
8. Check that the bvNotes are received as separate objects.

9 Test Result Template

Testing DATE:	**Testing Environments**
DUT(Device Under Test)	**Test Tools or System:**
Product Company:	**Hard Ware:**
Product Model:	
Product Revision:	**Software:**
Product Sample Serial No.	
Remarks:	

Test Name	**Pass/Fail**	**Comments**
IrWW3_C_P_BXX_HINT_1		
IrWW3_C_P_BXX_HINT_2		
IrWW3_C_P_BXX_HINT_3		
IrWW3_C_P_BXX_HINT_4		
IrWW3_S_P_BXX_HINT_1		
IrWW3_S_P_BXX_HINT_2		
IrWW3_C_P_BXX_IAS_1		
IrWW3_C_P_BXX_IAS_2		
IrWW3_S_P_BXX_IAS_1		
IrWW3_S_P_BXX_IAS_2		
IrWW3_C_P_BXX_DEVINFO_1		
IrWW3_S_P_BXX_DEVINFO_1		
IrWW3_C_P_BXX_INFOLOG_1		
IrWW3_S_P_BXX_INFOLOG_1		
IrWW3_C_A_BXX_PUT_1		
IrWW3_C_A_BXX_PUT_2		
IrWW3_C_A_BXX_PUT_3		
IrWW3_C_A_BXX_GET_1		
IrWW3_S_A_BXX_PUT_1		
IrWW3_S_A_BXX_PUT_2		
IrWW3_S_A_BXX_PUT_3		
IrWW3_S_A_BXX_GET_1		
IrWW3_C_U_BXX_PUT_1		
IrWW3_S_U_BXX_PUT_1		
IrWW3_S_P_BXX_RSP_1		
IrWW3_S_P_BXX_RSP_2		
IrWW3_S_P_BXX_RSP_3		
IrWW3_S_P_BXX_RSP_4		
IrWW3_C_M_TLOG_CONTENTS_1		
IrWW3_C_M_OWNER_CONTENTS_1		

IrWW3_C_M_CARD_CONTENTS_1		
IrWW3_C_M_SCHE_CONTENTS_1		
IrWW3_C_M_TODO_CONTENTS_1		
IrWW3_C_M_NOTE_CONTENTS_1		
IrWW3_S_M_TLOG_CONTENTS_1		
IrWW3_S_M_OWNER_CONTENTS_1		
IrWW3_S_M_CARD_CONTENTS_1		
IrWW3_S_M_SCHE_CONTENTS_1		
IrWW3_S_M_TODO_CONTENTS_1		
IrWW3_S_M_NOTE_CONTENTS_1		

＜付録＞

腕時計型端末用赤外線通信仕様 (IrWW)の概要紹介

腕時計型端末用赤外線通信仕様(IrWW)は、赤外線通信機能を持つ腕時計（以下、腕時計型端末と呼ぶ）間で時刻を合わせたり、名刺やスケジュールなどの PIM(Personal Information Manager)情報を簡単に交換できる通信仕様である。IrWW の規格書は、プロファイル、参照規格(Reference Document)、相互接続試験仕様書(Compliance Tests Document)の 3 つから構成される。

プロファイルには、IrDA の既存のプロトコルスタックおよび IrWW に新たに規定するプロトコルスタックのうちどのプロトコルを使って、どのような手順で、どのようなオブジェクトをやりとりするかを記述している。IrWW では時刻同期、簡易オブジェクト交換の 2 種類のプロファイルを規定している。時刻同期、簡易オブジェクト交換の各プロファイルにおいて対象となるプロトコルスタック、利用モデル(Usage Model)、対応オブジェクトの一覧を以下に示す。

プロファイル名	時刻同期プロファイル			
利用モデル	クライアント(腕時計型端末)からサーバ(腕時計型端末)へのオブシェクトのプッシュ			
プロトコル スタック	Ultra IrWW (必須)		IrLAP+IrLMP+TinyTP+IAS+IrOBEX (PC(パソコン)との通信の場合のみ使用)	
対応オブシェクト	データ形式	サポートのレベル	データ形式	サポートのレベル
Time-Sync	GBO for Ultra IrWW	必須	BWatch1.0(GBO)	オプション
Alarm	GBO for Ultra IrWW	オプション	BWatch1.0(GBO)	オプション
Timer	GBO for Ultra IrWW	オプション	BWatch1.0(GBO)	オプション
Stopwatch	GBO for Ultra IrWW	オプション	BWatch1.0(GBO)	オプション

プロファイル名	簡易オブジェクト交換プロファイル	
利用モデル	クライアント(腕時計型端末)からサーバ(腕時計型端末、PC)へのオブシェクトのプッシュ、クライアント(PC)からサーバ(腕時計型端末)への全データの読出し、書込み	
プロトコル スタック	IrLAP+IrLMP+TinyTP+IAS+IrOBEX	
対応オブシェクト	データ形式	サポートのレベル
Time-Log	BWatch 1.0 (GBO)	オプション
Business Card	BvCard 2.1 (GBO)	オプション
Schedule	BvCalender 1.0 (GBO)	オプション
To do	BvCalender 1.0 (GBO)	オプション
Notes	bvNote1.1 (GBO)	オプション

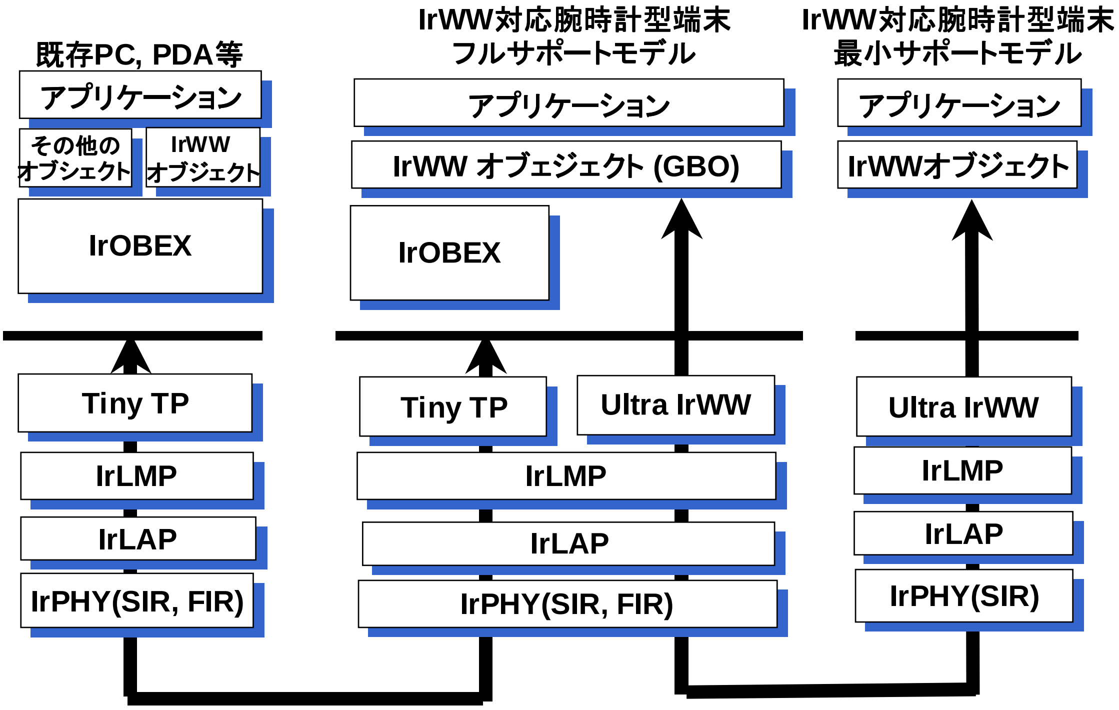
時刻同期プロファイルは、時刻、アラーム、タイマー、ストップウォッチといった時刻合わせや時計の設定に関するオブシェクトを扱うものであり、実装が限られる腕時計型端末への適用を前提とした最小実装が可能なプロファイルとなっている。使用プロトコルとしては、後述の Ultra IrWW を必須としている。利用モデルはオブジェクトをプッシュするだけの単純なもので、時刻に関する基本的なオブジェクトのみを扱っている。また、オブシェクトは、後述の GBO(Gemreric Binary Object)仕様によって圧縮したコードを扱っている。また、時刻生成タイミングから IrWW フレームの生成、送信にかかる遅延時間(⊿T)を時刻合わせのメカニズムに導入することにより、おおむね 10ms 以内の精度での腕時計型端末間の時刻同期が可能な仕様となっている。

一方、簡易オブジェクト交換プロファイルでは、プロトコルスタックとして IrOBEX を用いており、主に時計と PC(パソコン)あるいは PDA(携帯情報端末)との間で名刺やスケジュール等、単位オブジェクトのプッシュやオブジェクト全体の読出し、書込みが可能となっている。簡易オブジェクト交換プロファイルで扱うオブジェクトは、インターネット等で標準的に使用されている vCard や vCalender を GBO によってコード化したものを使用している。

参照規格(Reference Document)は、IrWW において新たに規定するプロトコル仕様であり、前述の GBO と Ultra IrWW の 2 つを規定している。GBO は IMC(Internet Mail Consortium)方式のデータと互換性を保ったまま、データをバイナリ化してデータ量を圧縮することで通信時間の短縮を可能とし、腕時計型端末の消費電力の低減も図ることを目的とした規格である。Ultra IrWW は、IrDA 唯一のコネクションレス形簡易プロトコルである Ultra プロトコルを IrWW で利用するために必要な識別用のパラメータやオブジェクトのデータ構造を規定している。

相互接続性試験仕様書(Compliance Tests Docmunet)は、前記の 2 種のプロファイルならびに PC や PDA との接続に関する計 3 種の相互接続性試験を定めている。

IrWW でサポートする腕時計型端末は、プロファイルの選択によって、下図のプロトコル構成に示す 2 種類が存在することになり、最小サポート・モデルでは、Ultra IrWW のみを実装して、腕時計型端末同士の時刻同期のみをサポートすることとなる。一方、フルサポートモデルでは、Ultra IrWW と IrOBEX の両方を実装するので、IrWW の 2 つのプロファイルいずれにも対応し、腕時計型端末だけでなく PC や PDA との接続が可能である。


既存PC, PDA等
アプリケーション
その他の オブジェクト
IrWW オブジェクト
IrOBEX
Tiny TP
IrLMP
IrLAP
IrPHY(SIR, FIR)
IrWW対応腕時計型端末 フルサポートモデル
アプリケーション
IrWW オブェジェクト (GBO)
IrOBEX
Tiny TP
Ultra IrWW
IrLMP
IrLAP
IrPHY(SIR, FIR)
IrWW対応腕時計型端末 最小サポートモデル
アプリケーション
IrWWオブジェクト
Ultra IrWW
IrLMP
IrLAP
IrPHY(SIR)