FIT FILE TYPES

Description

Copyright Information and Usage Notice

Revision History

Revision	Effective Date	Description
1.0	May 2010	Initial release
1.1	February 2010	Added File Types: Device, Course, Sport Settings, Goals and Totals Added Sections: Bike and SDM profiles (Settings File)
1.2	November 2011	Fixed error in Sport File description Added appendix with HRV description
1.3	January 2012	Added Schedule, Activity Summary, Monitoring and Daily Monitoring File Added further description to Activity File
1.4	April 2012	Moved hrv data description to Activity File description. Described activity file messages in more detail.
1.5	February 2013	Amended usage agreement Added Daily Monitoring File Broke out general considerations Clarified file_id field requirements Added new zones to Sport Settings File
1.6	May 2014	Added MonitoringB
1.7	May 2015	Add Segments and SegmentList
2.0	May 2016	Release for FIT 2.0
2.1	June 2016	Virbx plugin
2.2	May 2017	Swim Workout Details

Table of Contents

List of Figures

List of Tables

1 FIT File Type Overview

Different applications of FIT files lead to a natural grouping of message based on purpose. This document describes FIT File Types, which consist of common message groupings and methods for best practice. Table 1-1 outlines the FIT file types covered in this document.

Table 1-1. Common FIT File Types

File Number	FIT File Type	Purpose
1	Device	Describes a devices file structure and capabilities
2	Settings	Describes a user's parameters such as Age & Weight as well as device settings
3	Sport Settings	Describes a user's desired sport/zone settings
4	Activity	Records sensor data and events from active sessions
5	Workout	Describes a structured activity that can be designed on a computer and transferred to a display device to guide a user through the activity
6	Course	Uses data from an activity to recreate a course
7	Schedule	Provides scheduling of workouts and courses
9	Weight	Records weight scale data
10	Totals	Summarizes a user's total activity, characterized by sport
11	Goals	Describes a user's exercise/health goals
14	Blood Pressure	Records blood pressure data
15	MonitoringA	Records detailed monitoring data (i.e. logging interval < 24 Hr)
20	Activity Summary	Similar to Activity file, contains summary information only
28	Daily Monitoring	Records daily summary monitoring data (i.e. logging interval = 24 hour)
32	MonitoringB	Records detailed monitoring data (i.e. logging interval < 24 Hr)
34	Segment	Describes timing data for virtual races
35	Segment List	Describes available segments

1.1 General Considerations

The file types described in this document are ordinary FIT files and follow the FIT Protocol. They may contain valid FIT messages and fields other than the minimum or recommended practice described in this document. Additional fields may be added to the described messages from time to time and may be used if desired, **always refer to the FIT profile for the most recent content**. The FIT file header and CRC are requirements for all FIT files and must always be included. Use of the 12 byte header is considered legacy and is not recommended.

It is permitted that the quantity and position of message definitions be different than the recommended sequence presented in this document so long as they still conform to the FIT Protocol. For example all definitions could be written at the beginning of the file or a definition message could be provided for every data message etc.

Unless otherwise specified, messages without timestamps are permitted in any order and messages with timestamps must occur in increasing order. Refer to individual file definitions for further details.

Where possible it is recommended messages be defined using a single local message type (i.e. local message type = 0). This may be accomplished by defining message "A" using local message type 0, and then writing necessary data messages. Local message 0 would then be redefined for message "B" and necessary data messages written etc. This ensures simple processors can handle all file data. See the FIT Protocol for further details on local message definitions.

2 Related Documents

The following supplementary documentation and files are also provided in the SDK:

- Flexible & Interoperable Data Transfer (FIT) Protocol document
- FIT Global Messages and Fields (Profile.xls)
- FIT code generator
- FIT to CSV Conversion Tool
- Reference code examples
- Example FIT files

Many FIT applications will involve the ANT-FS protocol to facilitate the wireless transfer of FIT files. For further information regarding ANT-FS and related details for transferring FIT files specifically, refer to the following documents:

- ANT File Share (ANT-FS) Technology
- ANT-FS Reference Design and User Manual

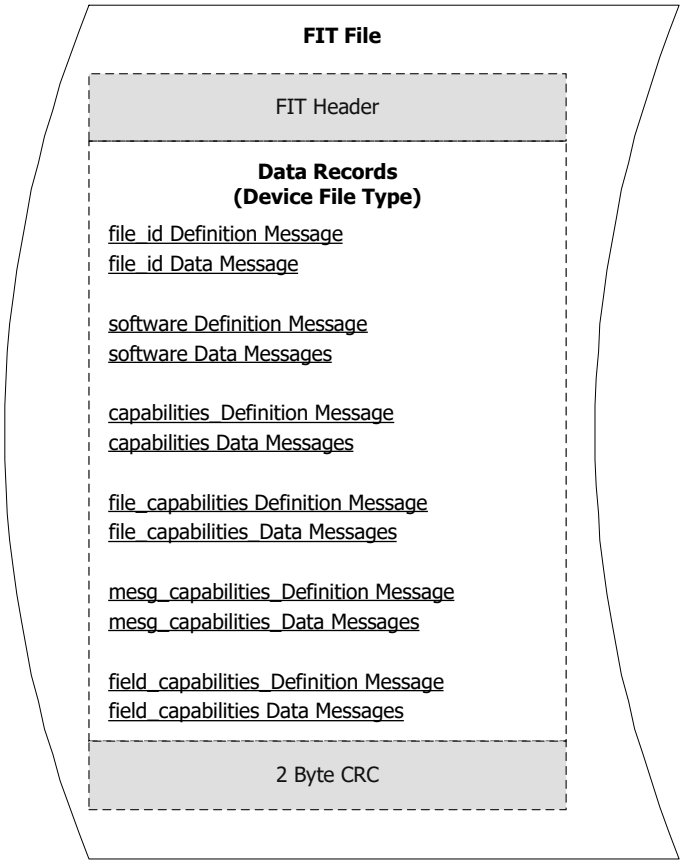
Some ANT+ Device Profiles include a FIT component. Refer to the relevant ANT+ Device Profile for details.

3 Device File

The device file contains data records that provide information on a device's file structure/capabilities. The records provide details on the types of files a device supports, and restrictions/capabilities (if applicable) of the messages and fields contained within each file type (Figure 3-1).

See section 1.1 for other general guidelines.

Figure 3-1. Device File



3.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 1** for a device file. The following FIT messages can also be included in a device file:

Table 3-1. FIT Messages Contained in Device File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Device File (=1)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
software	message_index	N	UINT16	Provides an index such that other FIT messages can be related to this message
	version	N	UINT16	SW Version, scaled by 100
	part_number	N	string	SW Identifier, managed by manufacturer
capabilities	languages	N	UINT8z	Array of languages supported (refer to language enum in profile.xls)
	workouts_supported	N	workout_capabilities (UINT32z)	Bit field describing the device's workout capabilities
file_capabilities	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	type	N	file (enum)	Refer to profile.xls
	flags	N	file_flags (UINT8z)	Refer to profile.xls
	directory	N	string	The directory name where the file is stored
	max_count	N	UINT16	Maximum number of files that may be stored in the directory
	max_size	N	UINT32	Maximum size of the associated file type.
mesg_capabilities	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	file	N	file (enum)	Refer to profile.xls
	mesg_num	N	mesg_num (UINT16)	Refer to profile.xls
	count_type	N	mesg_count (enum)	Refer to profile.xls
	count	N	UINT16	Dynamic field representing the message count. The value in this field depends on the count_type (refer to Table 3-2)

FIT Message	FIT Fields	Required	Type	Description
field_capabilities	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	File	N	file (enum)	Refer to profile.xls
	mesg_num	N	mesg_num (UINT16)	Refer to profile.xls
	field_num	N	UINT8	Refer to profile.xls
	count	N	UINT16	Supported number of times the field may appear in the associated message

3.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

3.1.2 "software" Message

The software message describes the device's software version and part number. If different software components are to be tracked, multiple software messages could be recorded with differing message indices.

3.1.3 "capabilities" Message

The capabilities message is used to communicate what languages a device supports and which workout functionalities are supported (if at all).

3.1.4 "file_capabilities" Message

The file_capabilities message can be used to indicate the device's directory structure (if applicable), the files stored within each directory, and may describe the content of the file. For example, this message may indicate a device has a "sports" directory that may contain up to 3 readable and writeable FIT sport files.

If a device does not have a directory structure, the device file shall be stored in the root directory.

3.1.5 "mesg_capabilities" Message

The mesg_capabilities message can be used to indicate the supported FIT messages within a specified FIT file. For example, this message may indicate a device has a settings file that may only contain a single user profile message.

If a specific FIT message is not described in the capabilities field for a supported file type, no assumptions can be made

3.1.5.1 "count" Dynamic Field

The count field is a dynamic field that is dependent on the value of the count_type field as described in Table 3-2.

Table 3-2. List of count_type and count Dynamic Field Values

count_type	Count value (dynamic field value)
num_per_file	num_per_file
max_per_file	max_per_file
max_per_file_type	max_per_file_type

3.1.6 "field_capabilities" Message

Most FIT fields appear once within a single FIT message; however, some FIT messages may support multiple appearances of the same field within a single message. If the latter is supported, the field capabilities message is used to indicate how many times the specified field may appear.

3.2 Device File Example

Figure 3-2 shows an example device file. The file begins with file_id definition and data messages, indicating the file is a device file (file_id.type = 1), the manufacturer is 'dynastream' (file_id.manufacturer =15), and the product is '1' with serial number '123456'.

The device file then contains software definition and data messages indicating the device is operating software revision 1.01 (software.version = 101); and capabilities definition and data messages indicating interval workouts are supported provided a distance source exists (capabilities.workout_supported=0x00000201).

The file_capabilities message defines the device's file structure. In this case, the device has a /Settings directory that may contain a single, readable and writeable, settings file. An /Activities directory is also supported and may contain an unspecified number of readable activities files. Note that the activities directory is not writeable. Finally, the device has a /Weight directory that may contain a single, read only, weight file. There is no directory for the device file as it shall always be stored in the root directory.

In this example file, message capabilities are only defined for the settings file. The first mesg_capabilities message indicates a settings file shall only contain a single user_profile message per file. The following messages indicate a settings file may contain single hrm_profile and sdm_profile messages, and up to 2 bike_profile messages.

Finally, the field_capabilities messages indicates the settings file may not contain a user_profile.weight field (i.e. count = 0).

file_id Definition Message (local message type=0, fields: type, mfg, product, serial_number)				
F	1	15	1	123456

software Definition Message (local message type=0, fields: version)	
S	101

capabilities Definition Message (local message type=0, fields: workouts_supported)	
C	0x00000201

file_capabilities Definition Message (local message type=0, fields: directory, type, max_count, flags)				
FC	"Settings"	2	1	6
FC	"Activities"	4	0xFFFF	2
FC	"Weight"	9	1	2

mesg_capabilities Definition Message (local message type=0, fields: file, mesg_num, count_type, count)				
MC	2	3	0	1
MC	2	4	0	1
MC	2	5	0	1
MC	2	6	0	2

field_capabilities Definition Message (local message type=0, fields: file, mesg_num, field_num, count)				
DC	2	3	4	0

HEADER BYTE
(F: file_id, S: software C: capabilities FC: file_capabilities MC: mesg_capabilities DC: field_capabilities)

Figure 3-2. Device File Example

4 Settings File

The settings file contains data records that provide user and device information in the form of profiles. Each profile is grouped into either user, bike, or specific device profiles (such as HRMs, SDMs and activity monitors). The profiles provide information about the user, bicycle, sensors that a device may pair to, and user interface preferences (Figure 4-1).

Currently settings files contain single user information. This file type may be extended in the future to allow for multi-user profiles.

See section 1.1 for other general guidelines.

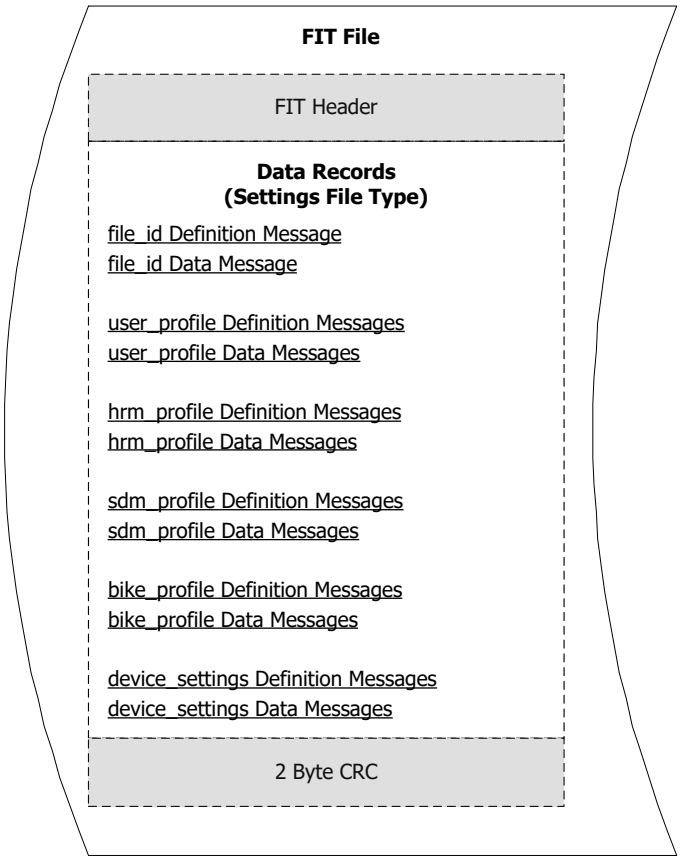


Figure 4-1. Settings File

4.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 2** for a settings file. The file_id message, and at least one of the listed messages below, are the only required FIT messages in a settings file. Messages/fields are included on an "as needed" basis.

Table 4-1. FIT Messages Contained in Settings File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Settings File (=2)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
user_profile hrm_profile sdm_profile bike_profile device_settings	Application Specific, refer to FIT profile for message details			

4.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

4.1.2 "user_profile" Message

The user_profile message provides information about the user such that workout parameters can be properly set, and to allow for measurements dependent on user data (e.g. weight). Although most devices are single user, some devices such as weight scales and blood pressure monitors may support multiple users.

4.1.3 "hrm_profile" Message

The hrm_profile message is used in devices that interact with fitness equipment. It contains the device identification of the user's heart rate monitor that may already be paired with a device such as a watch. In this example, when the watch pairs with fitness equipment, a settings file containing the hrm_profile message is transferred to the fitness equipment allowing the fitness equipment to search for the user's specific heart rate monitor.

4.1.4 "sdm_profile" Message

Similar to the hrm_profile, the sdm_profile message contains the device identification of the user's stride based speed and distance monitor that may already be paired with a device such as a watch.

4.1.5 "bike_profile" Message

The bike_profile message provides information about the user's bicycle(s), and their associated devices such as speed, distance and power sensors. This allows related parameters to be properly set, and for measurements that are dependent on bicycle information (e.g. wheel size). Multiple bike_profiles may be contained within a single settings file.

4.1.6 "device_settings" Message

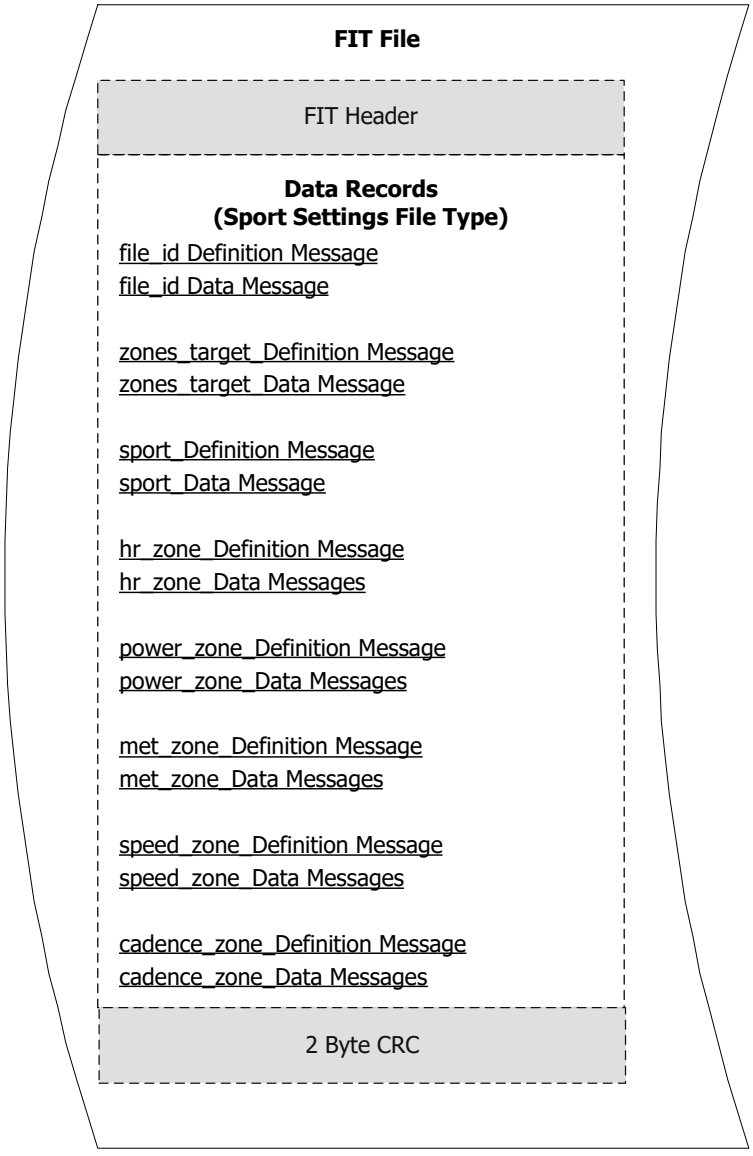
The device_settings message currently contains only the UTC offset which can be used for converting between system time and UTC time.

5 Sport Settings File

The sports settings file contains information about the user's desired target zones. The records provide details on the types of zones supported (such as heart rate or power), and the desired target levels. The sports settings file allows these values to be grouped by sport (Figure 5-1). There should only be one sport message per file.

See section 1.1 for other general guidelines.

Figure 5-1. Sport Settings File



5.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 3** for a sport settings file. The following FIT messages can also be included in a sport file:

Table 5-1. FIT Messages Contained in Sport Settings File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Sport Settings File (=3)
	manufacturer	Y	Manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
zones_target	max_heart_rate	N	UINT8	User parameters for calculating sport zone targets
	threshold_heart_rate	N	UINT8	User parameters for calculating sport zone targets
	functional_threshold_power	N	UINT16	User parameters for calculating sport zone targets
	hr_calc_type	N	hr_zone_calc (enum)	Refer to profile.xls
	pwr_calc_type	N	pwr_zone_calc (enum)	Refer to profile.xls
sport	sport	N	sport (enum)	Refer to profile.xls
	sub_sport	N	sub_sport (enum)	Refer to profile.xls
	name	N	string	Zone identifier
hr_zone	message_index	N	UINT16	Provides an index such that other FIT messages can be related to this message
	high_bpm	N	UINT8	Upper hr boundary for zone (bpm)
	name	N	string	Zone identifier
power_zone	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	high_value	N	UINT16	Upper power boundary for zone (watts)
	name	N	string	Zone identifier
met_zone	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	high_bpm	N	UINT8	
	calories	N	UINT16	Indicates the kcal/min to apply for metabolic calculation
	fat_calories	N	UINT8	Indicates the fat kcal/min to apply for metabolic calculation

FIT Message	FIT Fields	Required	Type	Description
speed_zone	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	high_value	N	UINT16	Upper speed boundary for zone (m/s)
	name	N	string	Zone identifier
cadence_zone	message_index	N	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	high_value	N	UINT16	Upper cadence boundary for zone (rpm)
	name	N	string	Zone identifier

5.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

5.1.2 "zones_target" Message

Some sport zone target values are calculated according to user parameters such as maximum or threshold heart rate or power values. The zones_target message is used to define these parameters, and shall only require one data message per file.

5.1.3 "sport" Message

The sport message indicates which sport, and/or sub sport, the zones are applicable for. There shall only be one sport data message per file. Refer to the FIT SDK for the list of available sports.

5.1.4 "hr_zone" Message

The hr_zone message is used to define the user's desired heart rate zones. Only the target maximum value is required to define a zone, and the minimum value will be set to the maximum of the previous zone. For example, if heart rate zone 1 high_bpm is set to 80 bpm, then zone 1 is defined as 0 to 80 bpm. Heart rate zone 2 may have high_bpm set to 110, resulting in a target zone 2 of 80 to 110 bpm. The user may also define a name for each zone, such as "warm up" or "cool down."

5.1.5 "power_zone" Message

The power_zone message is used to define the user's desired power zones. Similar to heart rate, only the target maximum value is required to define a zone, and the minimum value will be set to the maximum of the previous zone. The user may also define a name for each zone, such as "warm up" or "cool down."

5.1.6 "met_zone" Message

The met_zone message is used to define the user's desired metabolic zones. This allows the user to define targets based on heart rate, and the calories and/or fat calories per min calculation to apply when calculating metabolic burn.

5.1.7 "speed_zone" Message

The speed_zone message is used to define the user's desired speed zones. Similar to heart rate, only the target maximum value is required to define a zone, and the minimum value will be set to the maximum of the previous zone. The user may also define a name for each zone, such as "warm up" or "cool down."

5.1.8 "cadence_zone" Message

The cadence_zone message is used to define the user's desired cadence zones. Similar to heart rate, only the target maximum value is required to define a zone, and the minimum value will be set to the maximum of the previous zone. The user may also define a name for each zone, such as "warm up" or "cool down."

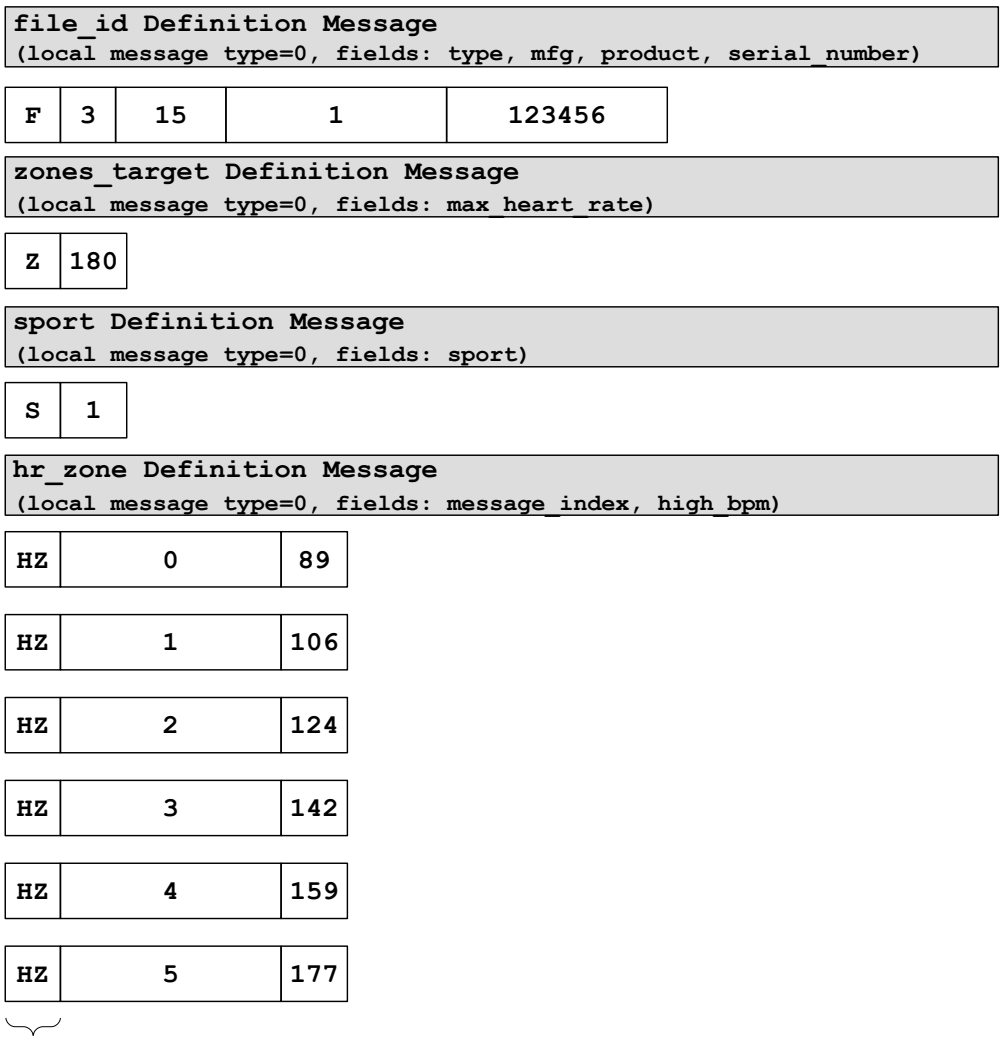
5.2 Sport Settings File Example

Figure 5-2 shows an example sport settings file. The file begins with file_id definition and data messages, indicating the file is a sport settings file (file_id.type = 3), the manufacturer is dynastream (file_id.manufacturer =15), and the product is "1" with serial number "123456."

The sport file then contains the zones_target definition and data messages. In this case, the zones_target message specifies that the maximum heart rate for zone calculations is 180 bpm. The sport message then indicates the file is related to running activities.

Finally, the hr_zone message is used to define running heart rate based zones. The first zone is defined as heart rate below 89 bpm. The next zone is 90 to 106 bm, followed by zones 107 to 124 bpm, 125 to 142 bpm, 143 to 159 bpm and 160 to 177 bpm.

Figure 5-2. Sport Settings File Example



6 Blood Pressure File

A blood pressure file contains time-stamped discrete measurement data. Data is reported after measurement, rather than a continuous real time format of data that is recorded in other files types such as activity files. The file is organized such that all definition messages are declared first, prior to recording any blood_pressure messages. No definition messages should appear after data messages have been recorded. To link multiple data messages, they must have identical timestamps. Pairs of blood pressure and device information data messages are linked through common timestamps.

The file_id definition and data messages should be recorded first, using the local message type 0. Local message type 0 should then be redefined for the FIT user_profile message (if used). The associated user_profile data messages should immediately follow the user_profile definition message. Once all relevant users have been recorded, local message type 0 should be redefined for blood_pressure messages. Using a single local message type to record the file_id, user_profile, and blood_pressure messages will ensure simple processors can handle all BP related data.

Once blood_pressure has been defined, any other desired FIT messages that will be recorded in the remainder of the file should also be defined in this section. The BP and other data messages shall fill the remainder of the file (Figure 6-1).

See section 1.1 for other general guidelines.

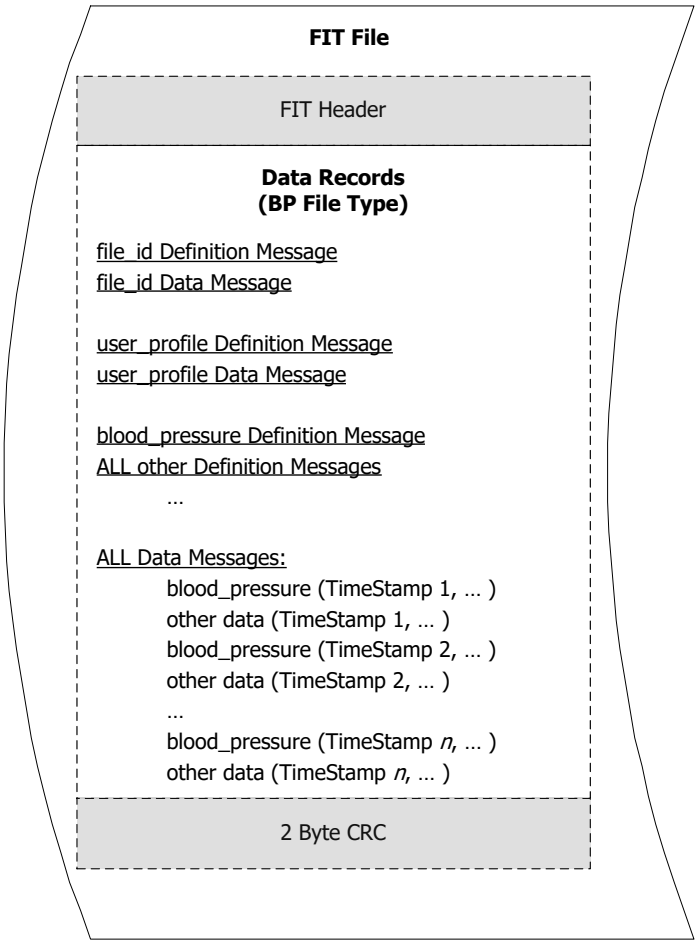


Figure 6-1. Blood Pressure File

6.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 14** for a blood pressure file. The BP file requires the file_id, and blood_pressure FIT messages. Other FIT messages, such as user_profile and device_info, may be included if desired as shown in Table 6-1.

Table 6-1. FIT Messages Contained in BP File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	BP file (= 14)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time	File creation time
	number	N*	UINT16	File identifier
user_profile	message_index	N	UINT16	Provides an index such that other FIT messages can be related to this user
	local_id	N	UINT16	BP monitor's local user ID
	friendly_name	N	string	User identifier
	gender	N	gender (enum)	Male/female
	age	N	UINT8	Years
	height	N	UINT8	1/100 m
	weight	N	UINT16	1/10 kg
	resting_heart_rate	N	UINT8	bpm
blood_pressure	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	user_profile_index	N	UINT16	Provides a link to the user_profile message. e.g. user_profile_index = 1 relates to the user_profile message with message_index = 1
	systolic_pressure	Y	UINT16	Bp data, mmHg
	diastolic_pressure	Y	UINT16	Bp data, mmHg
	mean_arterial_pressure	N	UINT16	Bp data, mmHg
	heart_rate	Y	UINT8	Bp data, bpm
	map_3_sample_mean	N	UINT16	Bp data, mmHg
	map_morning_values	N	UINT16	Bp data, mmHg
	map_evening_values	N	UINT16	Bp data, mmHg
	heart_rate_type	N	hr_type (enum)	normal, irregular
device_info	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	device_index	N	device_index (UINT8)	
	manufacturer	N	manufacturer (UINT16)	Managed by ANT+. Some legacy devices may set the MSB. Do not interpret.

FIT Message	FIT Fields	Required	Type	Description
	serial_number	N	UINT32z	Managed by manufacturer
	product	N	UINT16	Managed by manufacturer
	software_version	N	UINT16	Managed by manufacturer
	hardware_version	N	UINT8	Managed by manufacturer
	cum_operating_time	N	UINT32	s
	battery_voltage	N	UINT16	1/256 V
	battery_status	N	battery_status (enum)	new/good/ok/low/critical
	antplus_device_type	N	antplus_device_type (UINT8)	ANT+ managed device id 18 (0x12) for BP monitor
	ant_device_type	N	(UINT8z)	Manufacturer managed id

* Field is only required if the optional FIT message is recorded

6.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

6.1.2 "user_profile" Message

If the optional user_profile message is included, the file shall contain a user_profile message with a matching message_index defined for each user_profile_index used in blood_pressure messages. If this message is not used, it is implied that user ID's are not supported on any level.

6.1.3 "blood_pressure" Message

Blood_pressure message containing systolic pressure, diastolic pressure and pulse (i.e. heart_rate). If the measurement is associated with a user, link the messages by setting the user_profile_index field to the message_index in the matching user_profile message.

6.1.4 "device_info" Message

If the optional device_info message is included, it must contain the timestamp field in order to link each device_info message to its respective blood_pressure message. Only one of antplus_device_type and ant_device_type may be used.

6.2 Blood Pressure File Examples

Figure 6-2 shows an example FIT BP file. Note that the file contains the FIT header, definition and data messages for file_id, followed by the definition and data messages for user_profile, followed by the definition and data messages for blood_pressure and device_info.

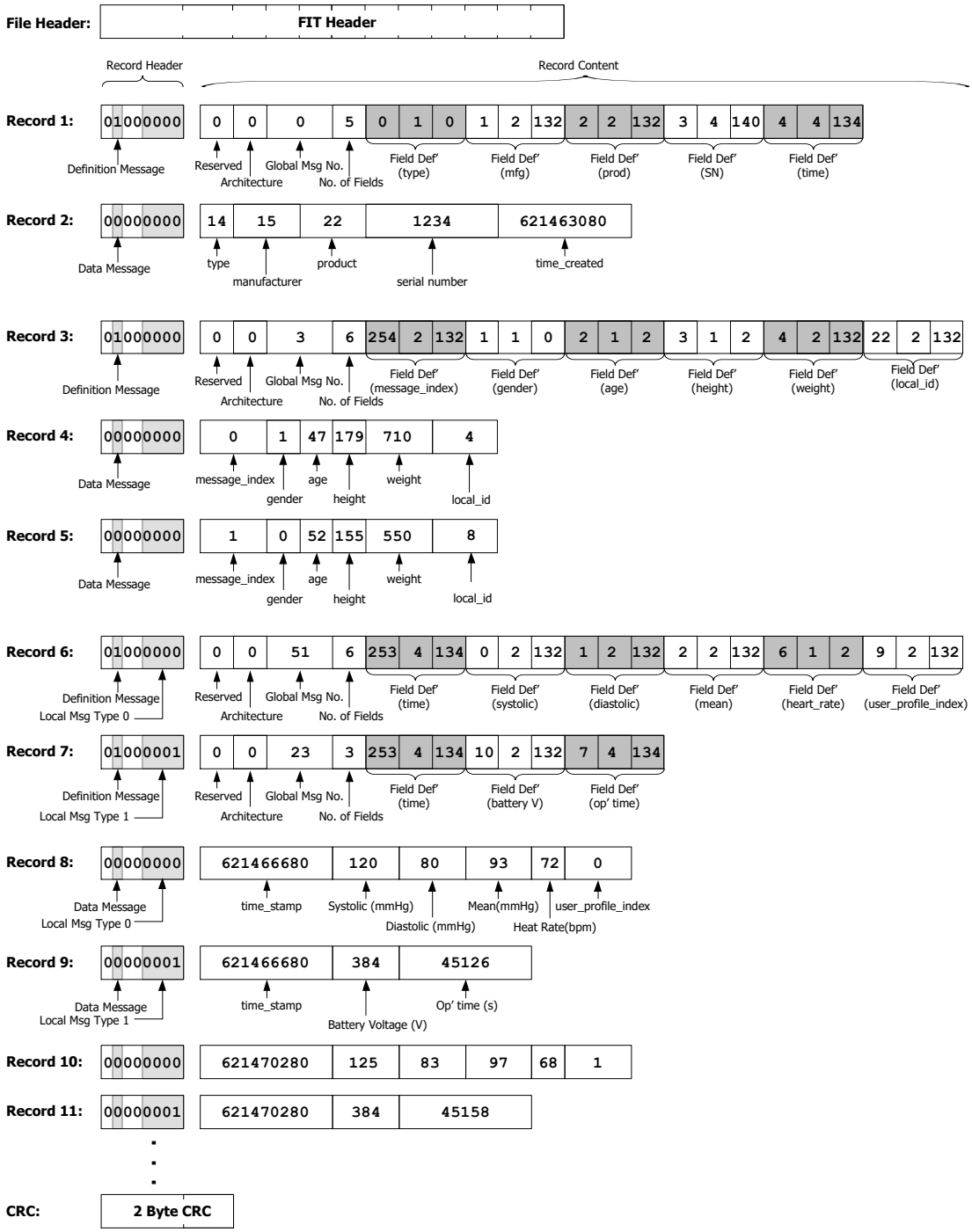


Figure 6-2. Multi-user BP File Example

The file_id, user_profile and blood_pressure messages shall all use local message type 0 in order to minimize the RAM requirements for handling BP specific data on limited processors. Any other data messages to be logged alongside blood_pressure, such as device_info, shall use a different local message type.

All user ID's must be defined prior to defining and recording measured data. The association of user information to message_index or user_profile_index may not change value within a file.

In this multi-user case, the file contains data from two users. One is a 47 year old male stored locally under user ID 4, and another is a 52 year old woman stored under local user ID 8. All of their data is recorded under their local user ID on the device, which is linked to their profile data. When the FIT file is written, the user_profile and blood_pressure data is linked through the message_index and user_profile_index fields respectively.

The local_id and message_index fields do not need to match; however, message_index and user_profile_index must match. The message_index field shall only be numbered sequentially from 0, in increments of 1. The number of local IDs a device has is dependent on the BP monitor's capabilities.

For a single user BP file, the user_profile_index does not need to be included in the blood_pressure message. Instead, the local_id can be defined once, using the user_profile message (with or without the message_index field), and all subsequent blood_pressure data records will be associated to that user. For example, in Figure 6-3, all data is associated to local_id "3". If the blood_pressure message is defined without the user_profile_index field, it is assumed that all data records that follow are associated to user_profile_index 0. Similarly, if the message_index field is not recorded and only one user_profile message exists, all blood_pressure data will be associated to that single user profile.

For simple BP monitors that do not support user ID's, the user_profile message is not required (Figure 6-4).

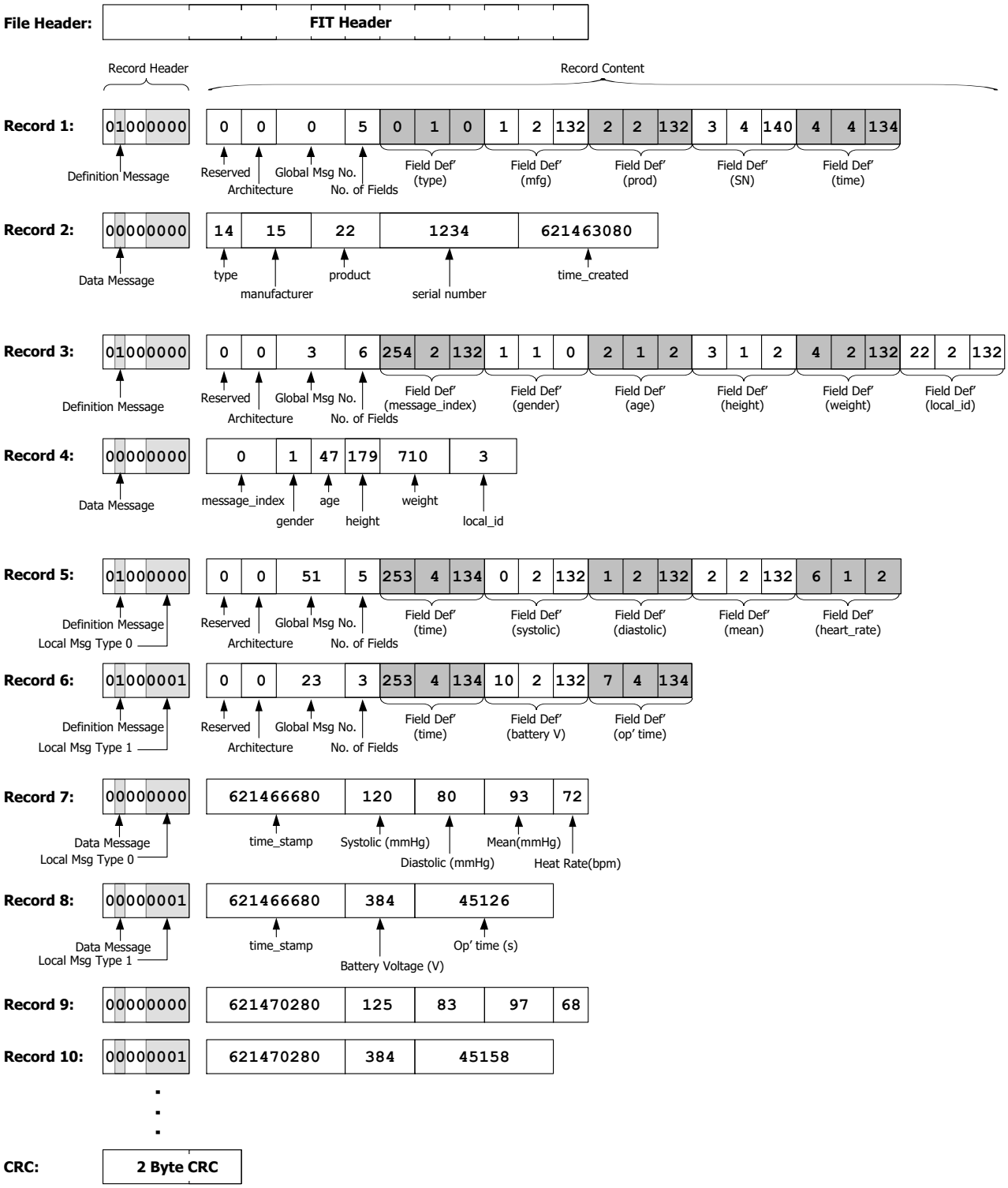


Figure 6-3. Single User BP System

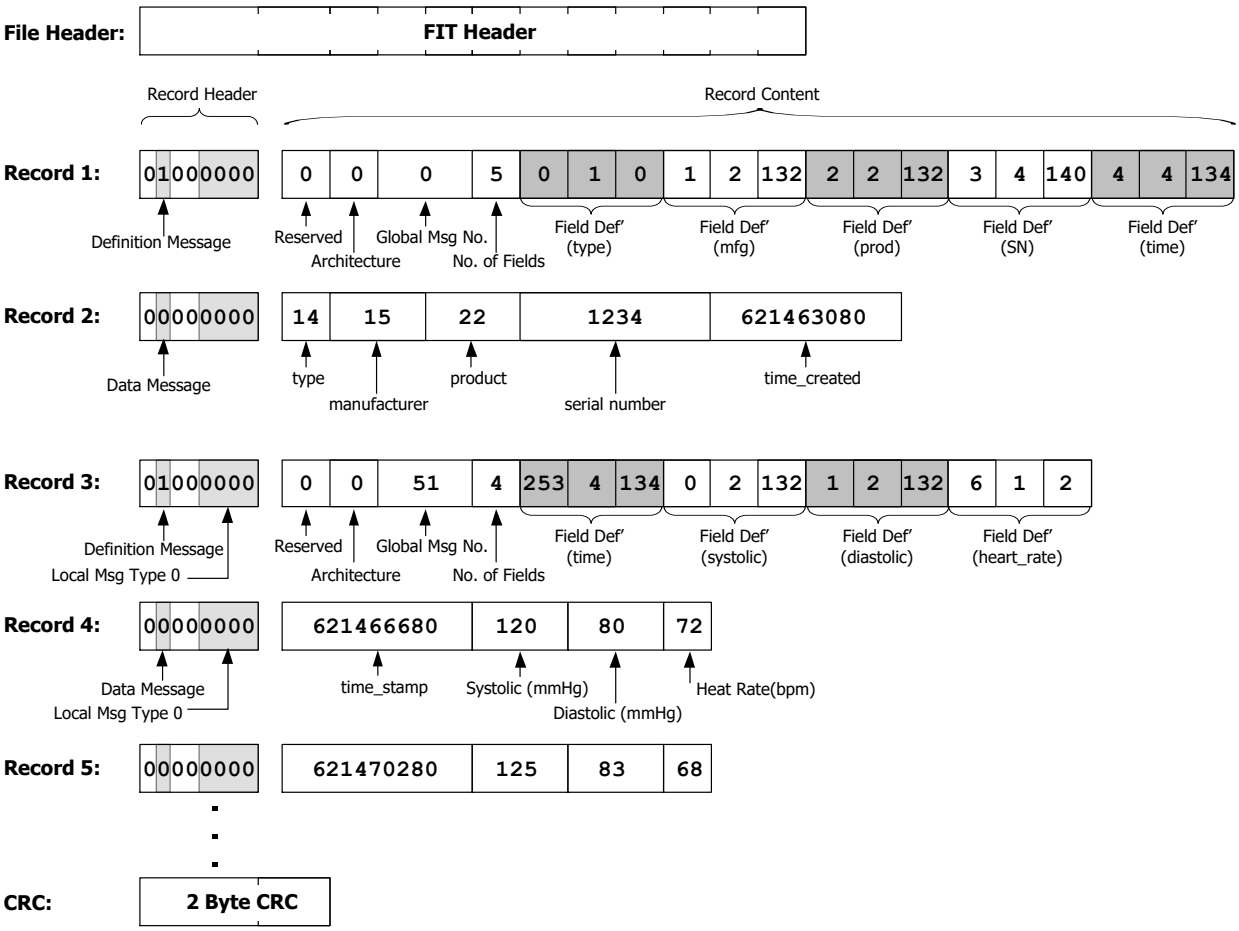


Figure 6-4. BP System without User Profile Support

7 Weight File

A weight file is similar in structure to the blood pressure file type. A weight file contains time-stamped discrete measurement data that is reported after measurement. The file is organized such that all definition messages are declared first, prior to recording any weight messages. No definition messages should appear after weight data messages have been recorded. To link multiple data messages in a weight file, they must have identical timestamps (Figure 7-1).

See section 1.1 for other general guidelines

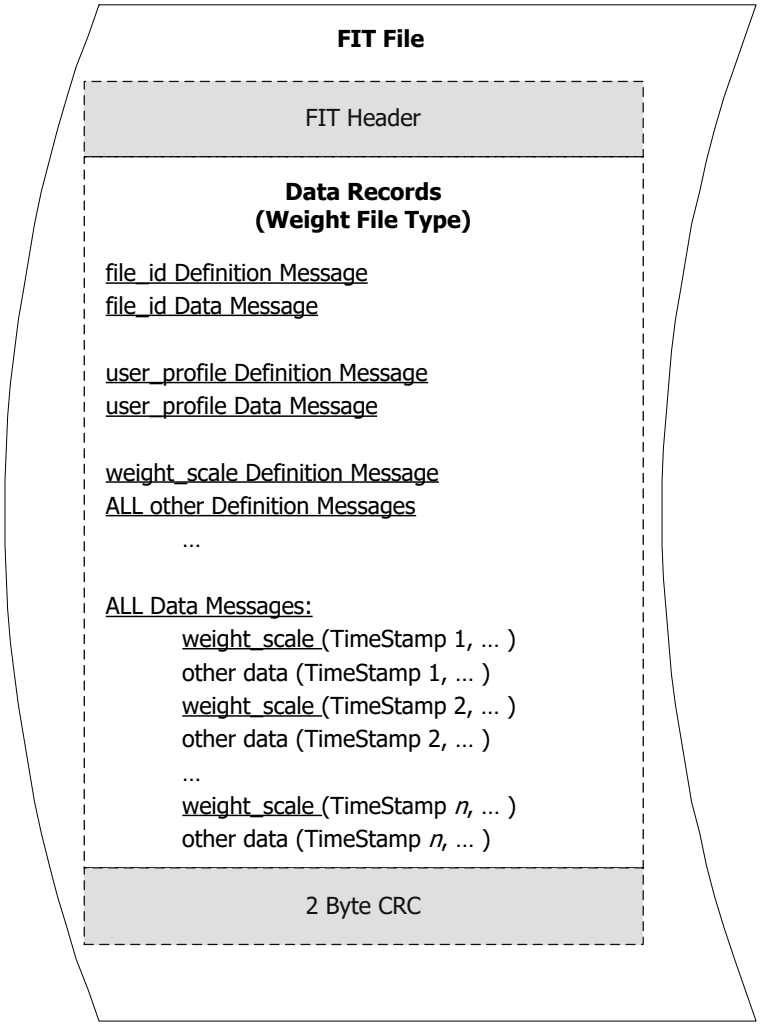


Figure 7-1. Weight File

7.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 9** for a weight file. A weight file must contain the file_id, user_profile (if user profiles supported) and weight_scale messages as described in Table 7-1. It may also, optionally, contain the device_info message.

Table 7-1. FIT Messages Contained in Weight File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Weight File (=9)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
user_profile	message_index	N	UINT16	Provides an index such that other FIT messages in the file can be related to this user
	local_id	N	user_local_id (UINT16)	Weight scale's local user ID.
	friendly_name	N	string	User identifier
	gender	N	gender (enum)	Male/female
	age	N	UINT8	Years
	height	N	UINT8	1/100 m
	activity_class	N	activity_class (enum)	level/level_max/athlete
weight_scale	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	user_profile_index	N	UINT16	Provides a link to the user_profile message. e.g. user_profile_index = 1 relates to the user_profile message with message_index = 1
	weight	Y	UINT16	1/100 kg
	percent_fat	N	UINT16	1/100 %
	percent_hydration	N	UINT16	1/100 %
	visceral_fat_mass	N	UINT16	1/100 kg
	bone_mass	N	UINT16	1/100 kg
	muscle_mass	N	UINT16	1/100 kg
	basal_met	N	UINT16	¼ kcal/day
	active_met	N	UINT16	¼ kcal/day

FIT Message	FIT Fields	Required	Type	Description
device_info	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	device_index	N	device_index (UINT8)	
	manufacturer	N	manufacturer (UINT16)	Managed by ANT+. Some legacy devices may set the MSB. Do not interpret.
	serial_number	N	UINT32z	Managed by manufacturer
	product	N	UINT16	Managed by manufacturer
	software_version	N	UINT16	Managed by manufacturer
	hardware_version	N	UINT8	Managed by manufacturer
	cum_operating_time	N	UINT32	s
	battery_voltage	N	UINT16	1/256 V
	battery_status	N	battery_status (enum)	new/good/ok/low/critical
	ant_plus_device_type	N	antplus_device_type (UINT8)	ANT+ managed device id 18 (0x12) for Weight scale
	ant_device_type	N	UINT8z	Manufacturer managed id

* Field is only required if the optional FIT message is recorded

7.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

7.1.2 "user_profile" Message

If the optional user_profile message is included, then the file shall contain a user_profile message with a matching message_index defined for each user_profile_index used. If this message is not recorded, it is implied that user ID's are not supported on any level. For ANT+ implementations refer to the device profile for further details about local_id usage.

7.1.3 "weight_scale" Message

The weight_scale message containing weight

7.1.4 "device_info" Message

If the optional device_info message is included, it must contain the timestamp field in order to link each device_info message to its respective weight_scale message. Only one of antplus_device_type and ant_device_type may be used.

7.2 Weight File Examples

Figure 7-2 shows an example FIT weight file. Note that the file contains the FIT header, definition and data messages for file_id, followed by the definition and data messages for user_profile, followed by the definition and data messages for weight_scale and device_info.

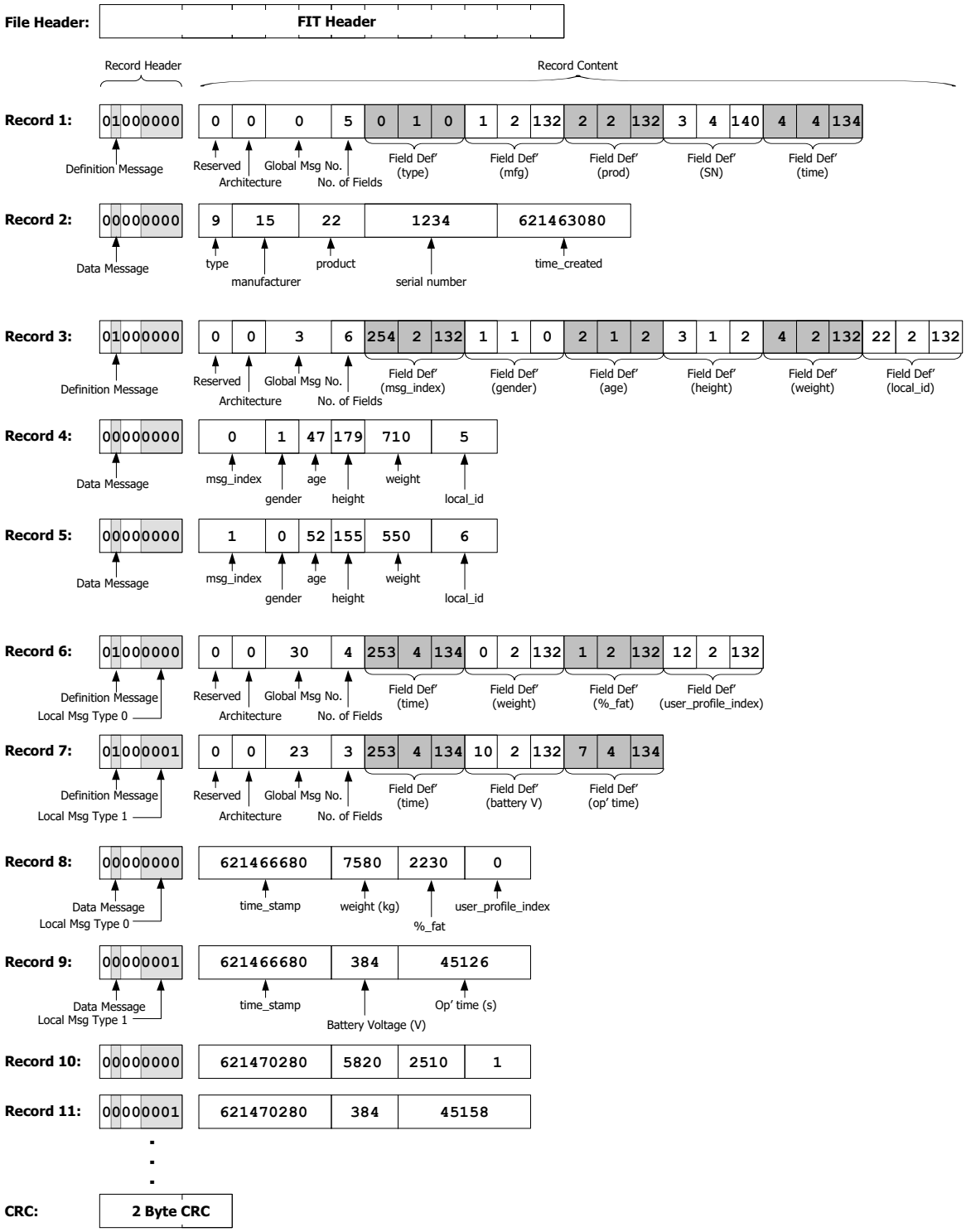


Figure 7-2. Multi-user Weight File Example

The file_id, user_profile and weight_scale messages shall all use local message type 0 in order to minimize the RAM requirements for handling weight scale data on limited processors. Any other data messages, such as device_info, shall use a different local message type as desired.

Note:

- All user ID's must be defined prior to defining and recording measured data. The association of user information to message_index or user_profile_index may not change value within a file.
- FIT files cannot be created/edited during a weight scale measurement

In the example shown in Figure 7-2, the file contains data from two users. One is a 47 year old male stored locally under user ID 5, and another is a 52 year old woman stored under local user ID 6; which is indexed within the file to message_index 0 and 1 respectively. All of their data is recorded on the device under the local ID which is linked to their profile data. When the FIT file is written, the user_profile and weight_scale data is linked through the message_index and user_profile_index fields respectively

The local_id and message_index fields do not need to match; however, message_index and user_profile_index must match. The message_index field shall only be numbered sequentially from 0, in increments of 1. The number of local user ID's will be dependent on the weight scale devices capabilities (i.e. user profile ID).

For a single user weight file, the user_profile_index does not need to be included in the weight_scale message. Instead, the user information can be defined once, using the user_profile message (with or without the message_index and/or local_id fields), and all subsequent weight_scale data records will be associated to that user. For example, in Figure 7-3, all data is associated to the user information recorded in message_index "0". If the weight_scale message is defined without the user_profile_index field, it is assumed that all data records that follow are associated to user_profile_index 0. Similarly, if the message_index field is not recorded and only one user_profile message exists, all weight_scale data will be associated to that single user profile.

For simple weight scales that do not support user ID's, the user_profile message is not required

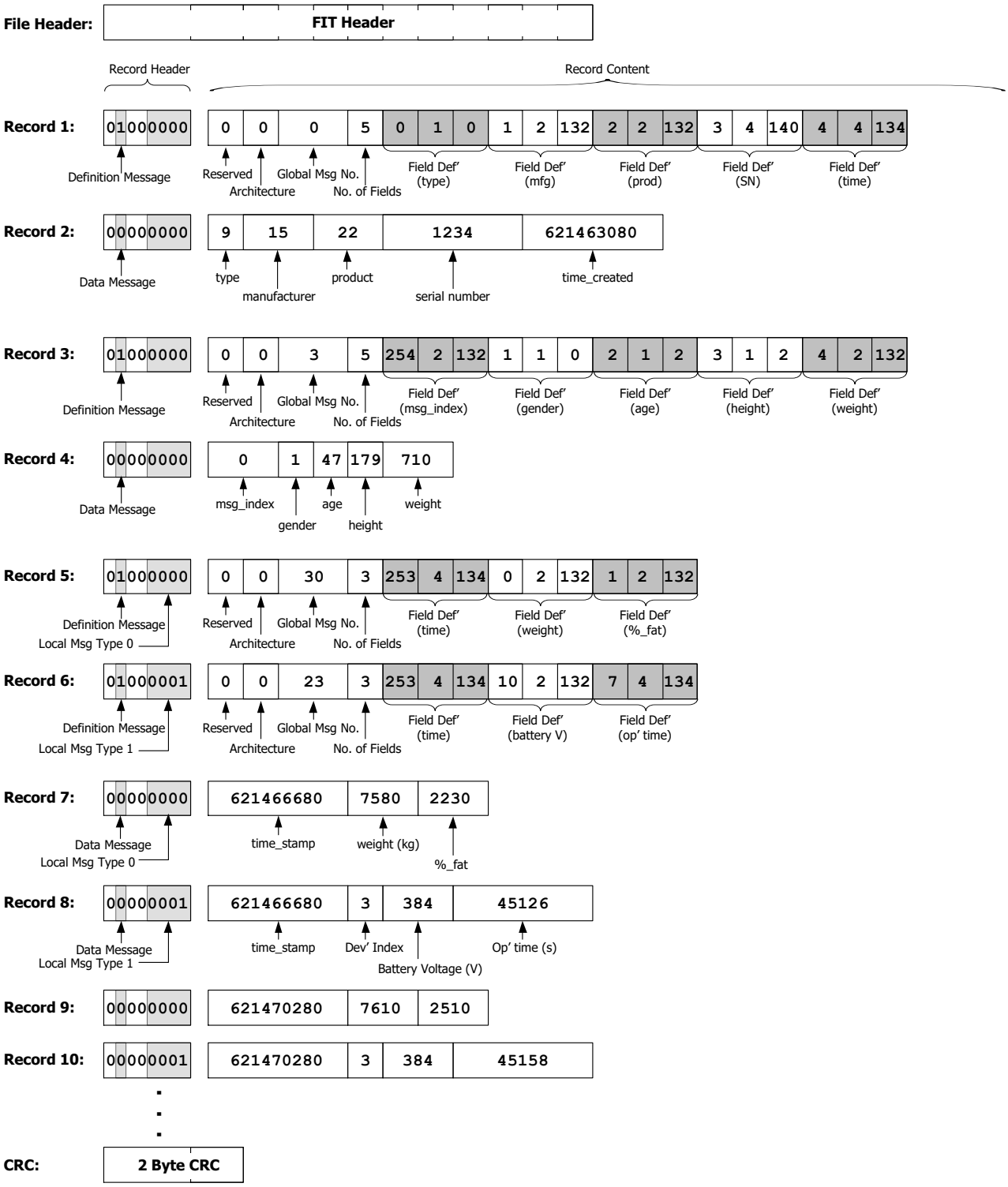


Figure 7-3. Single or Unidentified User Systems

8 Workout File

A workout file describes a structured activity and guides a user through the activity. It can be designed on a computer and transferred to a display device or generated on the device itself. The workout file must, at a minimum, contain the file_id, workout and at least one workout_step FIT message (Figure 8-1). The file_id, and workout messages need only be recorded once, at the start of the workout file. The rest of the workout file will consist of multiple workout_step messages.

See section 1.1 for other general guidelines

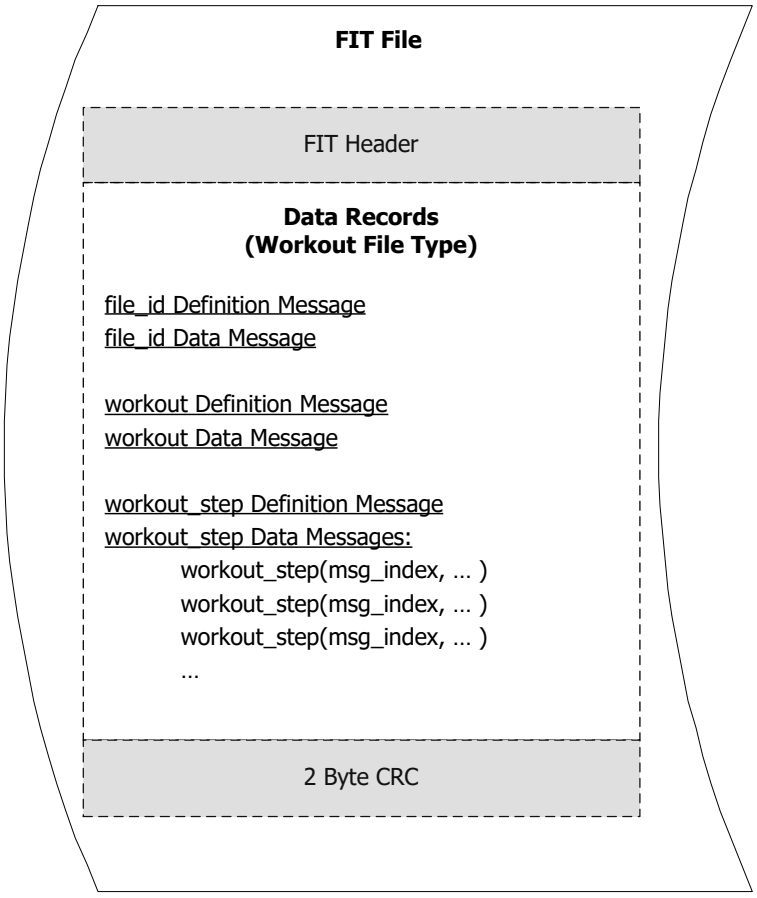


Figure 8-1. Workout File

Workouts are described as a series of steps. Each step is used to define a target effort for a set duration (Figure 8-2, step A and B), or to define a repetition pattern (Figure 8-2, step C).

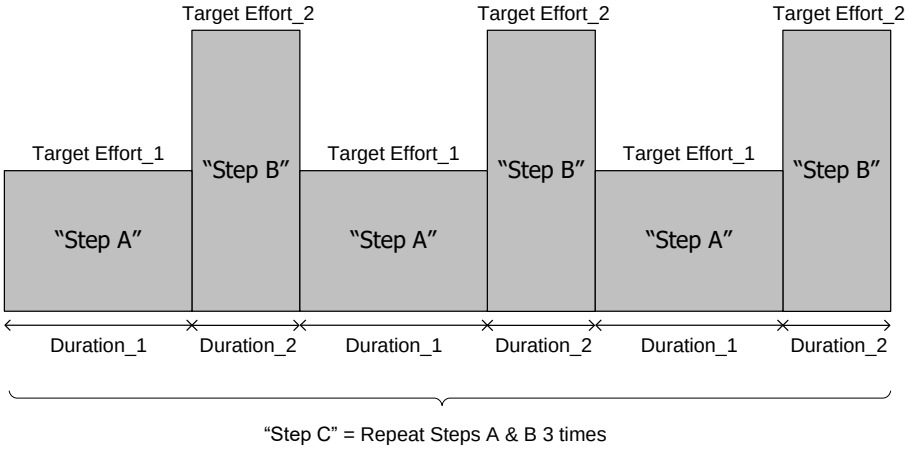


Figure 8-2. Defining Workout Steps

The following sections will describe the FIT messages of a workout file.

8.1 FIT Messages

The general message structure for both the workout and workout_step messages are show below in Figure 8-3.

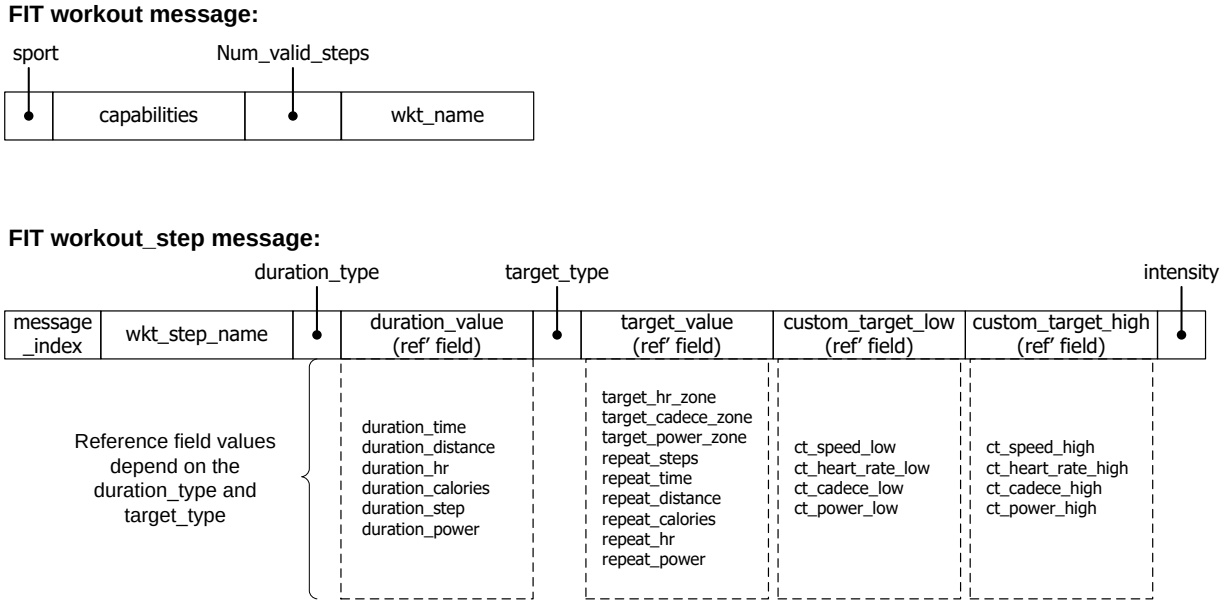


Figure 8-3. FIT workout and workout_step Message Structure

All FIT files must start with a file_id message. The FIT **file_id.type = 5** for a workout file. The full list of FIT messages contained in a workout file are outlined in Table 8-1. Note that not all fields are required.

Table 8-1. FIT Messages Contained in Workout File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Workout file (=5)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
workout	sport	N	sport (enum)	Indicates type of sport workout
	capabilities	N	UINT32z	Bitfield describing workout capabilities. Refer to SDK
	num_valid_steps	Y	UINT16	Indicates the number of valid steps (i.e. Count of workout_step messages) contained in the file
	wkt_name	N	string	User friendly string identifying name of workout
	pool_length	N	UINT16	Total length of the pool in meters
	pool_length_unit	N	display_measure (enum)	Unit used to display to the user
workout_step	message_index	Y	UINT16	Provides an index for each step such that a repeat step can refer back to a specific workout step
	wkt_step_name	N	string	User friendly string identifying name of the workout step
	duration_type	Y	wkt_step_duration (enum)	Indicates the type of parameter that will define the workout step's duration.
	duration_value	N	UINT32, workout_hr, or workout_power	Dynamic field representing the value of the duration. The value in this field depends on the duration_type (Table 8-2)
	target_type	Y	wkt_step_target (enum)	Indicates the type of parameter that will define the workout step's target range/zone.
	target_value	N	UINT32, workout_hr, workout_power, target_stroketype	Dynamic field representing the value of the target. The value in this field depends on either duration_type or target type as outlined in Table 8-3.

	custom_target_value_low	N	UINT32, workout_hr, or workout_power	If the workout target uses a custom range, rather than a defined zone, this field is used to specify the lower boundary. Dynamic field dependent on target_type (Table 8-3)
	custom_target_value_high	N	UINT32, workout_hr, or workout_power	If the workout target uses a custom range, rather than a defined zone, this field is used to specify the upper boundary. Dynamic field dependent on target_type (Table 8-3)
	intensity	N	intensity (enum)	Represents the workout steps intensity level (Table 8-4)
	equipment	N	workout_equipment (enum)	Represents the equipment that should be used when performing the step

8.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. Since a device may contain multiple workout files it is important that the combination of type, manufacturer, product and serial_number is unique so that they can be differentiated by schedule files for example.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

8.1.2 "workout" Message

The workout message is recorded once, at the start of the file and provides a summary of the workout information contained in the file. It describes the sport the workout is related too, workout capabilities, and the number of defined workout steps contained in the file. Using the Figure 8-2 example, the number of defined steps is 3 (i.e. steps A, B and C).

8.1.3 "workout_step" Message

The workout_step message is used to define each workout step. For defining a single step, this message describes:

- Duration type: e.g. time, distance, etc
- Duration value: e.g. 1min, 100m, etc
- target type: e.g. heart rate, speed, etc
- target value: this may be a preconfigured zone (e.g. heart rate zone '1' or '2') or a custom value (e.g. 65% to 75% max heart rate)

For defining a repetition step, this message describes:

- Duration type: repeat a sequence of workout_steps
- Duration value: the step to start repetitions from (i.e. step A in Figure 8-2)
- target value: number of repeats, time limit of repeats, etc

8.1.3.1 "message_index" Field

The message_index provides a 0 based index for each workout step that may be used by repeat steps to reference a specific workout step. The message_index must be unique for each workout_step and is in the range 0-workout.num_valid_steps (since there must be workout.num_valid_steps workout_step messages in the file).

8.1.3.2 "duration_type" Dynamic Fields

The duration_value and target_value fields are dynamic fields that are dependent on the value of the duration_type field as described in Table 8-2.

Table 8-2. List of duration_types and Relevant Dynamic Field Values

duration_type	duration_value (dynamic field value)	target_value (dynamic field value)
Time	duration_time	
Distance	duration_distance	
hr_less_than	duration_hr	
hr_greater_than	duration_hr	
calories	duration_calories	
open	duration_value	
repeat_until_steps_cmplt	duration_step	repeat_steps
repeat_until_time	duration_step	repeat_time
repeat_until_distance	duration_step	repeat_distance
repeat_until_calories	duration_step	repeat_calories
repeat_until_hr_less_than	duration_step	repeat_hr
repeat_until_hr_greater_than	duration_step	repeat_hr
repeat_until_power_less_than	duration_step	repeat_power
repeat_until_power_greater_than	duration_step	repeat_power
power_less_than	duration_power	
power_greater_than	duration_power	
repetition_time	duration_time	

8.1.3.3 "target_type" Dynamic Fields

The target_value, and custom_target_low/high fields are dynamic fields that are dependent on the value of the target _type field as described below in Table 8-3.

Table 8-3. List of target_types and Relevant Dynamic Field Values

target_type	target_value (dynamic field value)	custom_target_value_low (dynamic field value)	custom_target_value_high (dynamic field value)
speed	target_speed_zone	custom_target_speed_low	custom_target_speed_high
heart_rate	target_hr_zone	custom_target_heart_rate_low	custom_target_heart_rate_high
open	target_value	custom_target_value_low	custom_target_value_high
cadence	target_cadence_zone	custom_target_cadence_low	custom_target_cadence_high
power	target_power_zone	custom_target_power_low	custom_target_power_high
stroke_type	target_swim_stroke	0	0

8.1.3.4 Target values vs Custom target values

Unless defining repeat steps, the target_value dynamic field typically refers to setting a target zone. These target zones represent target limits that have already been established through other means; such as: predefined on fitness equipment, in a settings file, or through a user interface. If zones are predefined their numbering should start at 1 (since 0 is reserved

to indicate a custom zone). The workout_step can then be used to set a target heart rate, power or other zone value. In this case, the custom_target_value_high and custom_target_value_low fields should be set to 0.

If a specific target range is desired, the custom_target_low and custom_target_high fields may be used to set the upper and lower boundaries of the desired target range. The target_value would be set to 0. Refer to the FIT SDK for specific field/zone values. When creating swim workouts, the target is used to indicate the swim stroke for the step.

8.1.3.5 Workout Intensity

The workout_steps intensity field differentiates between sets that are designated for warm up, recovery, active and cool down. The intensity field does not affect target or duration values, but tracking the intensity field allows the program designer to calculate the total amount of active time within a workout.

Table 8-4. Workout Intensity Values

Intensity Value	Intensity Description
0	Active
1	Rest
2	Warmup
3	Cooldown

8.1.3.6 Setting Power and Heart Rate Values

Power and heart rate values can be set as specific or relative values. Specific values are set in integer values representing beats per minute (bpm) for heart rate, or watts for power. Relative values are set as an integer value ranging from 0 to 100% of the user's maximum heart rate or 0 – 1000% functional threshold power (ftp).

As the integers 0 to 100 (heart rate) and 0 to 1000 (power) range are reserved for relative values, specific heart rate and power values must be incremented by 100 bpm or 1000 watts respectively. Examples are provided below.

Table 8-5. Expressing Heart Rate and Power in Specific and Relative Values

Desired Heart Rate	Value in HR Field	Desired Power	Value in Power Field
125 bpm	225	325 Watts	1325
85% user's max HR	85	275%	275

8.2 Workout File Examples

The following examples illustrate how to correctly define workout steps, from setting individual steps to repeating steps, and setting custom target values.

8.2.1 Individual Workout Steps

Figure 8-4 shows an example of four workout_steps records used to define a workout that has a warmup step ("A"), two active steps ("B1" and "B2"), and a cooldown step ("C").

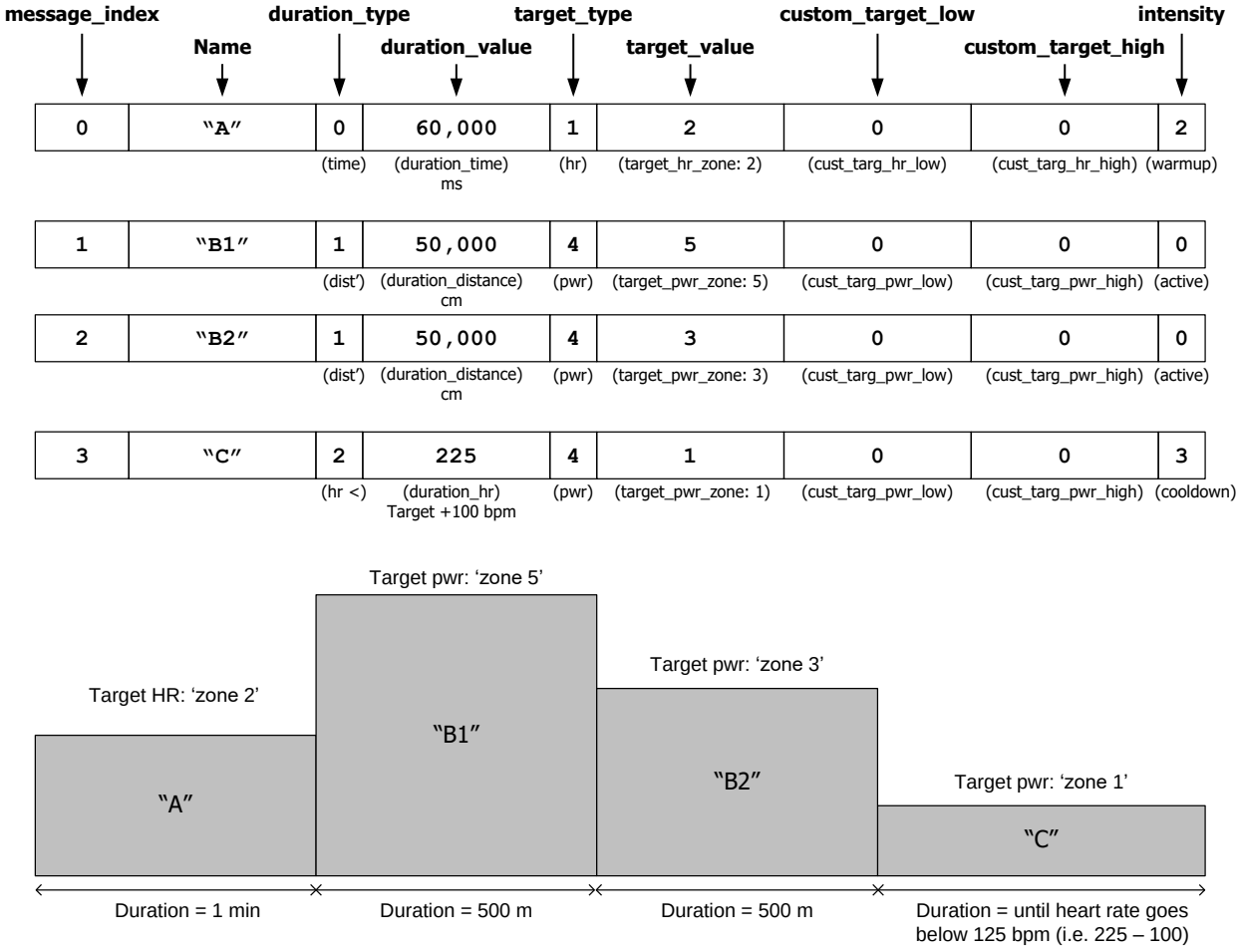


Figure 8-4. Example Workout_Steps

Message_index values always start at 0, and increment with each workout_step message. As such, the first workout_step message uses message_index 0. The duration_type is set to 0 (i.e. time), which means the duration_value dynamic field will contain duration_time data, which is a time value in units of milliseconds. Similarly, the target_type is set to 1 (i.e. heart rate) and the target_value field will refer to target_hr_zone data, which is an integer value representing the pre-defined zone. As the target zone is defined, no custom values are required and shall be set to 0. The intensity field is set to 2, indicating the step is a warmup step. In this case, the duration is set to 60 seconds of activity to be performed in heat rate zone 2.

Workout_steps "B1" and "B2" are indexed at message_index 1 and 2 respectively. For both steps, the duration_type is set to 1 (i.e. distance), which means the duration_value dynamic field will contain duration_distance data, which is a distance value in units of centimeters. Similarly, the target_type is set to 4 (i.e. power) and the target_value field will refer to

target_power_zone data, which is an integer value representing the pre-defined zone. As the target zones are defined, no custom values are required and shall be set to 0. The intensity field is set to 0, indicating these are active steps. In this case, the duration is set to 500 meters seconds of activity each to be performed in power zone 5, and then 3.

The final workout_step "C" is at message_index 3, the duration_type is set to 3, indicating the duration_type is "hr_less_than" and the duration_value will refer to duration_hr data. This means that the step will be performed for as long as it takes the user's heart rate to drop below that of the specified hr value (in duration_hr). The target_type is set to 4 (i.e. power) and the target_value field will refer to target_power_zone data, which is an integer value representing the pre-defined zone. As the target zones are defined, no custom values are required and shall be set to 0. The intensity field is set to 3, indicating this is a cooldown step. In this case, the user will perform the activity in power zone 1, until the user's heart rate is below 125 bpm. NB that the duration_hr value is the target value + 100 (i.e. 125 + 100 bpm), refer to section 8.1.3.6 for details on setting heart rate or power values.

8.2.2 Repeat Steps Example

Figure 8-5 uses the same steps from the example in Figure 8-4, however another step ("Rep") is added to repeat the active steps ("B1" and "B2"). Note that the added step has changed the message_index value for step "C" from 3 to 4. This is because **the message_index field must be sequential**.

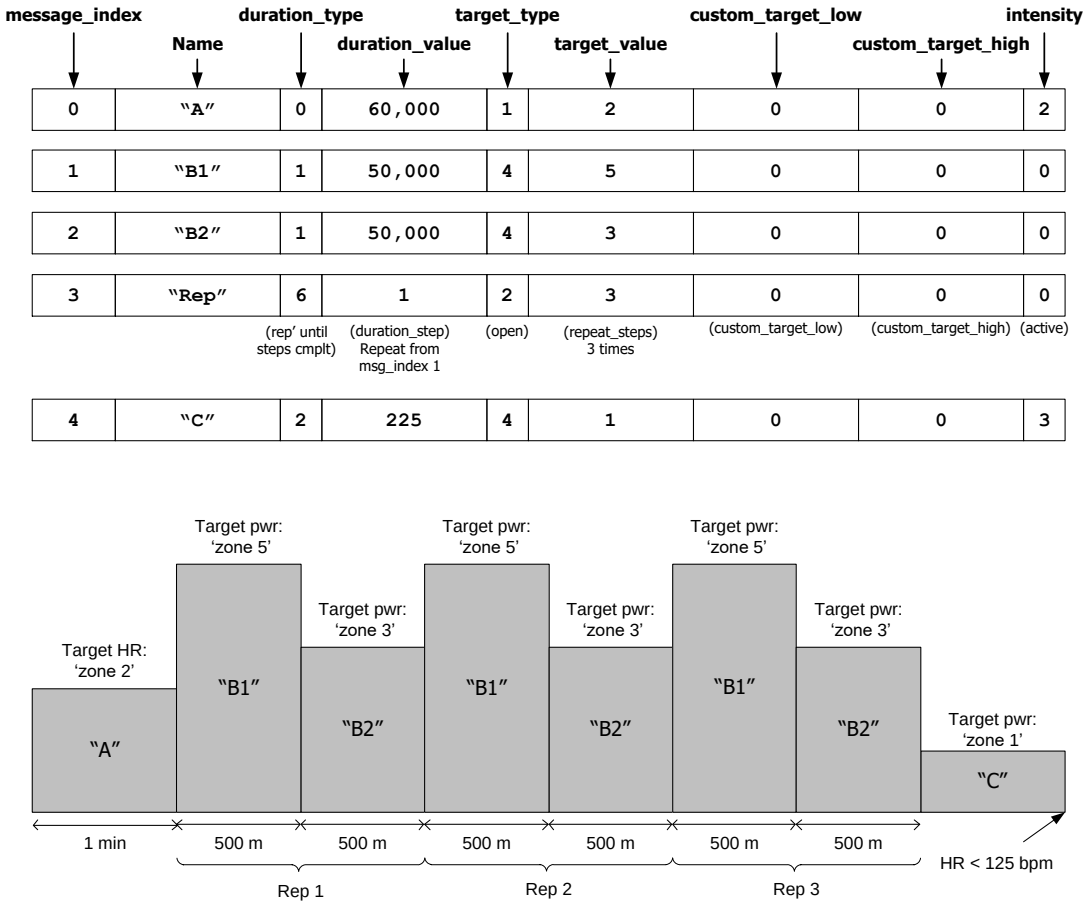


Figure 8-5. Workout_Steps for Repeating Steps

The repeat workout step ("Rep") has a duration_type value of 6, meaning "repeat_until_steps_completed", and the duration_value will be of type duration_step, and will contain the message_index of the step to start the repetitions from. In other words, setting the duration_step field to a value of 1, will indicate that the repetition will start from the workout_step with a message_index = 1 (i.e. step "B1"), and follow through all subsequent steps up until the repeat step. In this case, this means steps "B1" and "B2" will be repeated. For repeat steps, the duration_type also determines the value in the target_value dynamic field, and indicates this field will contain repeat_steps data, which is an integer value representing the number of times the sequence shall be repeated before progressing onto the next step (i.e. "C").

8.2.3 Repeat Until Example

For repeat steps that use duration_types containing "repeat_until_[type]_greater than" or "repeat_until_[type]_less_than", the sequence will repeat until the specified value is met then drop out of the current step and immediately drop into the next step. This scenario is illustrated in Figure 8-6.

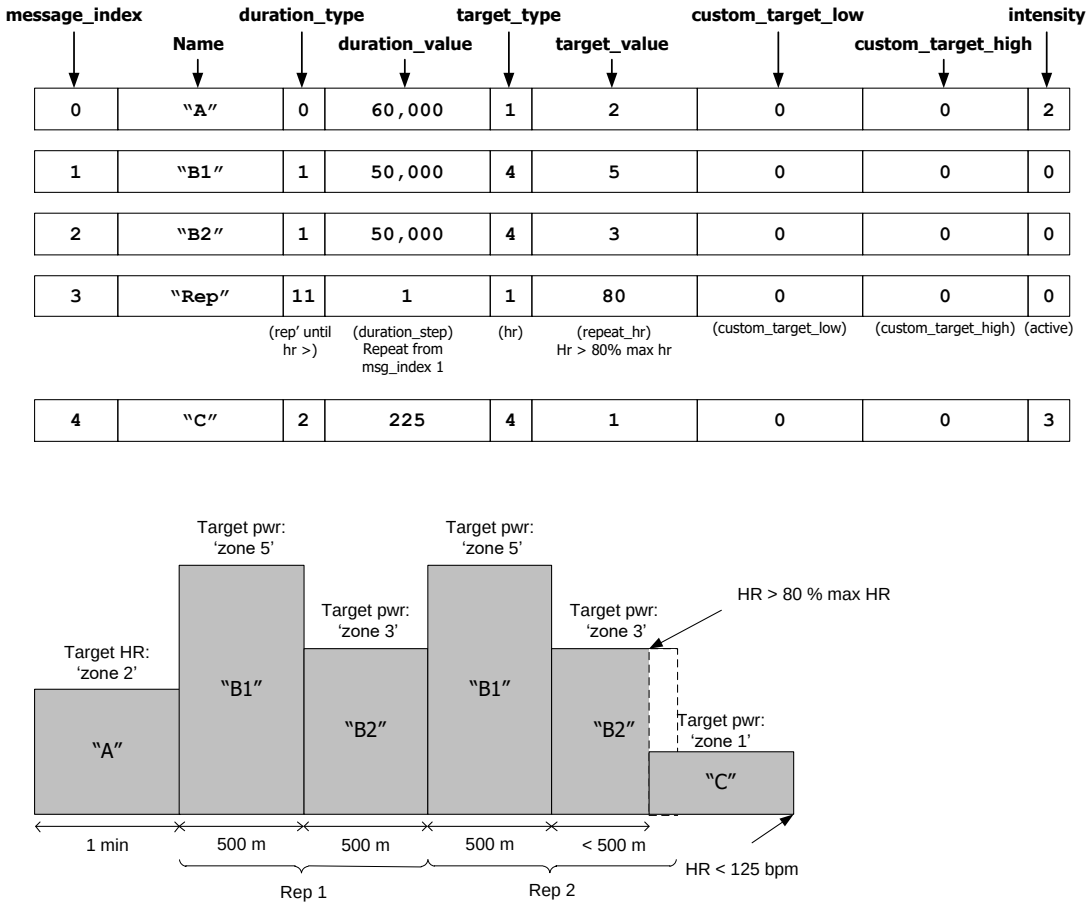


Figure 8-6. Repeat Steps Using "greater than" or "less than" Duration Types.

Step "Rep" has now been defined using duration_type value of 11, meaning "repeat_until_hr_greater_than", and the duration_value will be of type duration_step, and will again contain the message_index of the step to start the repetitions from. The duration_step field is again set to a value of 1, indicating the repetition will include steps "B1" and "B2". For repeat steps, the duration_type also determines the value in the target_value dynamic field, and indicates this field will contain repeat_hr data, refer to section 8.1.3.6 for details on setting heart rate or power values. In this case, repeat_hr is set to 80, indicating that the steps will be repeated until the user's heart rate is greater than 80% of their maximum heart

capacity. Once the this heart rate has been exceeded, the workout jumps out of the current step (i.e. "B2") and commences the next step (i.e. "C").

8.2.4 Using Custom Target Values

If predefined target zones are unavailable or undesired, custom target values may be used instead. Figure 8-7 below uses the same workout steps from the example in Figure 8-4, however custom target values are used instead of target zones.

If custom targets are used, the relevant target_value field (i.e. target_hr zone and target_power_zone in the example below) shall be set to 0, indicating that custom values will be used. The data type of the custom values is dependent on the target_type as described in Table 8-3.

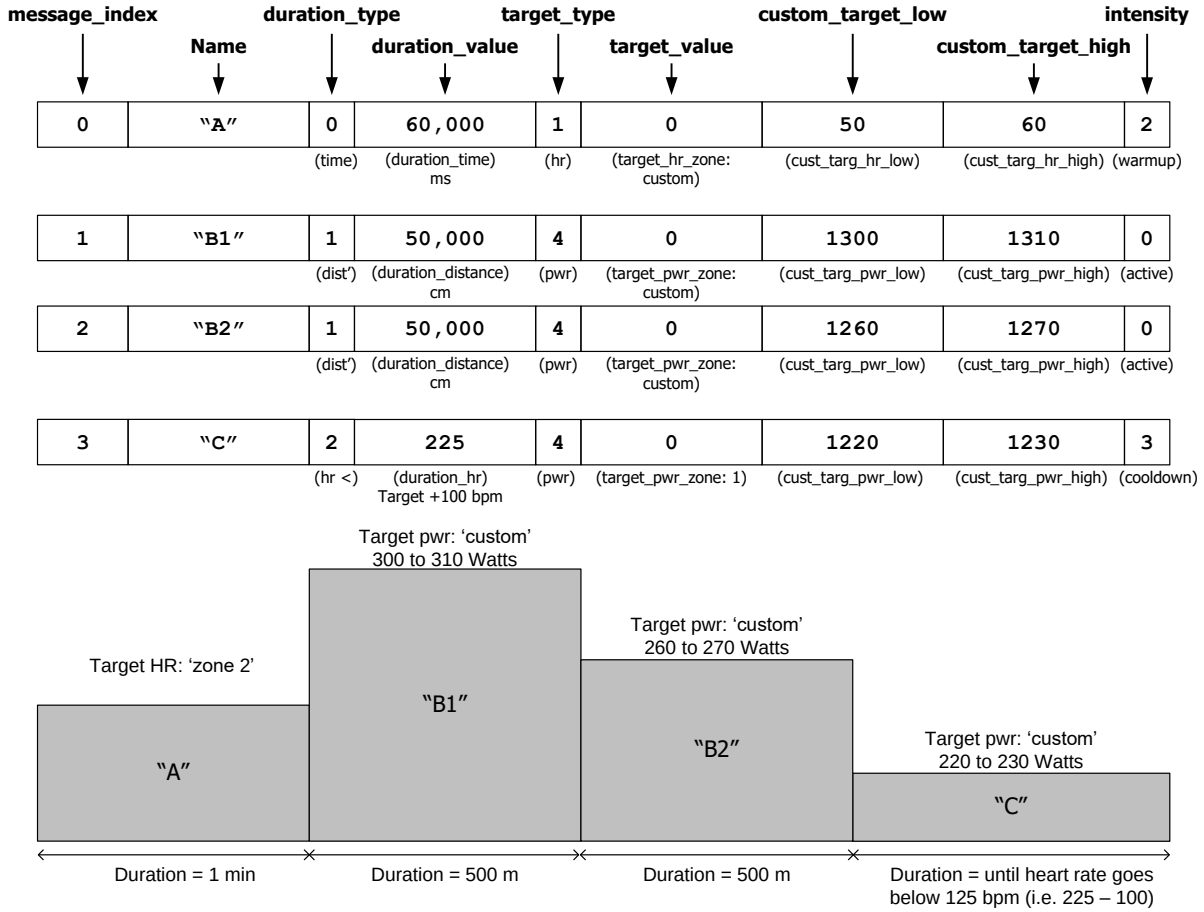


Figure 8-7. Example Workout_Steps Using Custom Target Values

In this example, step "A" target_type is set to 1 (i.e. heart rate) and the target_value field is set to zero indicating custom values will be used. The custom_value_low and custom_value_high fields will be of custom_heart_rate_low and custom_heart_rate_high data types respectfully, setting a target heart rate range of 50-60% of the user's maximum heart rate.). Refer to section 8.1.3.6 for details on setting heart rate or power values.

Similarly, workout_steps "B1", "B2" and "C" the target_type is set to 4 (i.e. power) and the target_value field set to 0 for custom target values. The custom_value_low and custom_value_high fields will be of custom_power_low and custom_power_high data types respectfully, setting a target speed range of 300 to 310 Watts for "B1", 260 to 270 Watts for "B2" and 220 to 230 Watts for step "C".

8.2.5 Swim Workout Example

Swim workouts take advantage of the target to specify the stroke type desired for a given interval, and distances as the duration to specify the desired length of each step. Using a repetition_time duration in a rest after at the end of a repeat can be used to create a "repeat on" workout.

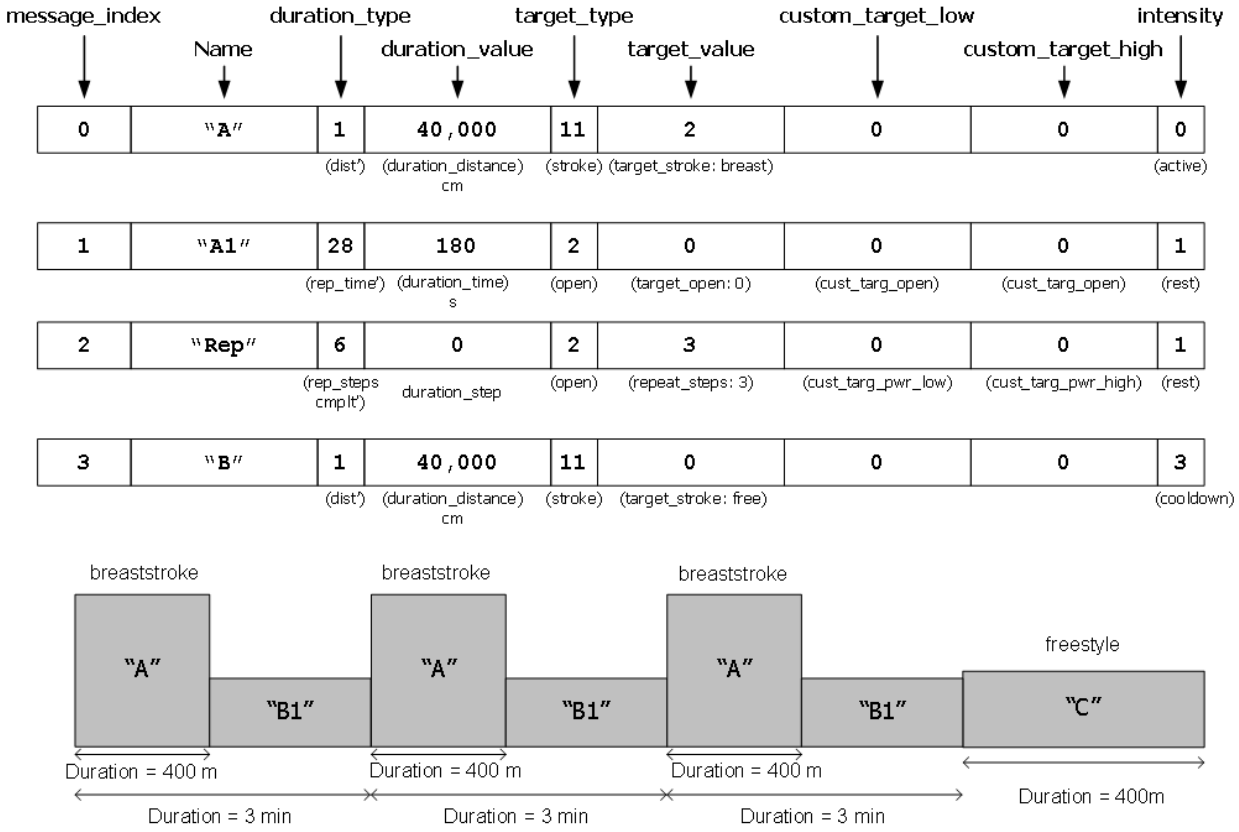


Figure 8-8. Swim Workout Example

9 Activity File

Activity files are used to record sensor data and events from an active session. The basic structure of an activity file is shown in (Figure 9-1).

See section 1.1 for other general guidelines

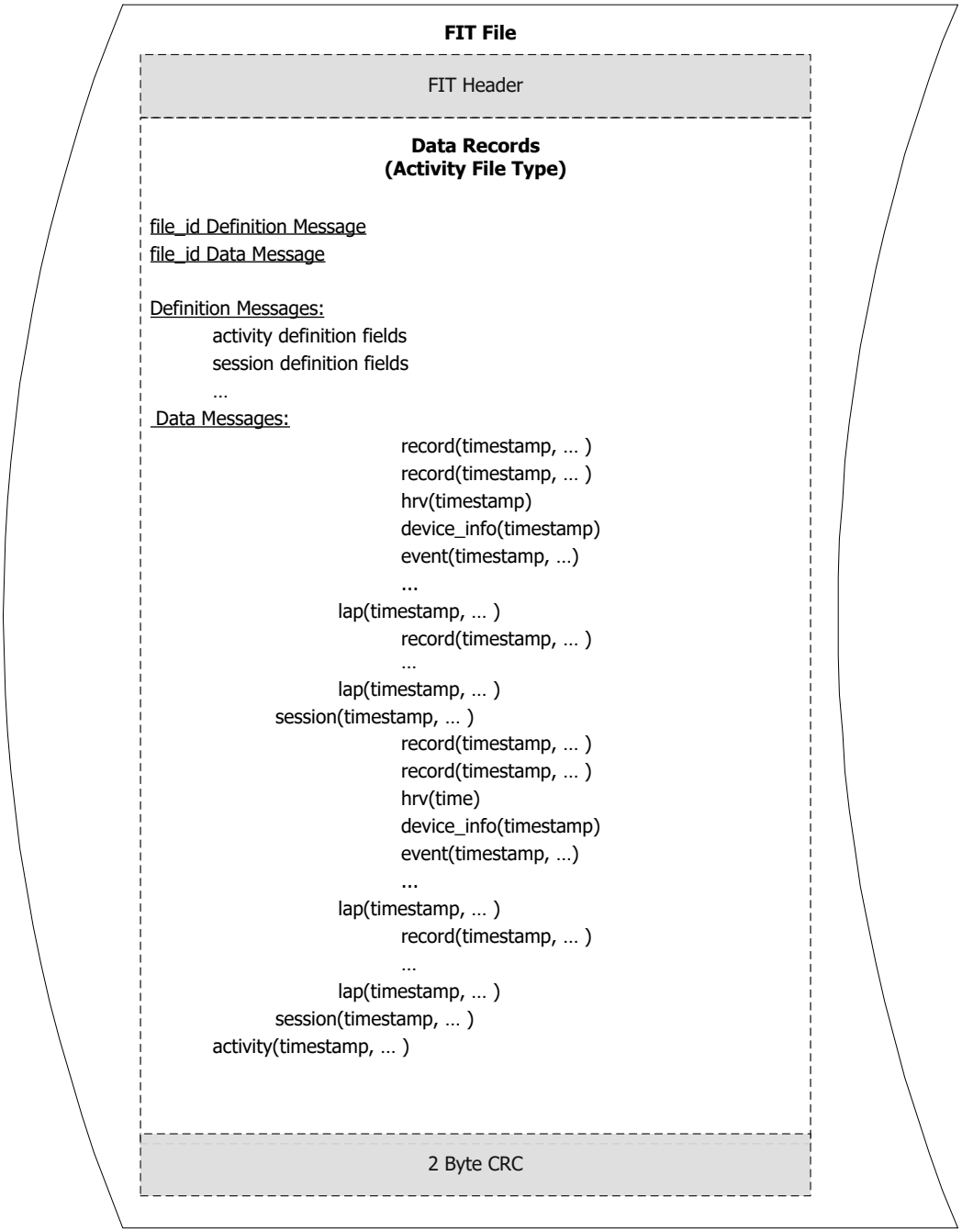


Figure 9-1. Activity File

An activity file shall contain file_id, activity, session, and lap messages. The file may also contain record, event, length and/or hrv messages. All data messages in an activity file (other than hrv) are related by a timestamp.

9.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 4** for an activity file. A partial list of FIT messages and fields contained in an activity file are outlined in Table 9-1. For a full list of messages and fields refer to the FIT SDK.

Table 9-1. FIT Messages Contained in an Activity File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Activity file (=4)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
activity	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	local_timestamp	N*	date_time (UINT32)	Timestamp epoch expressed in local time. Used to determine the time_zone or convert timestamps to local time
	num_sessions	Y	UINT16	Indicates total number of sessions included in the activity file
	type	Y	activity (enum)	refer to SDK
	event	Y	event (enum)	refer to SDK
	event_type	Y	event_type (enum)	refer to SDK
	...			
session	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	start_time	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	total_elapsed_time	Y	UINT32	Total number of msec since timer started (includes pauses)
	sport	Y	sport (enum)	refer to SDK
	event	Y	event (enum)	refer to SDK
	event_type	Y	event_type (enum)	refer to SDK
	…	…	…	…
lap	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	event	Y	event (enum)	refer to SDK
	event_type	Y	event_type (enum)	refer to SDK
	…	…	…	…

FIT Message	FIT Fields	Required	Type	Description
length	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	event	Y*	event (enum)	refer to SDK
	event_type	Y*	event_type (enum)	refer to SDK
	…	…	…	…
record	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	…	…	…	…
event	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	event	Y*	event (enum)	refer to SDK
	event_type	Y*	event_type (enum)	refer to SDK
	…	…	…	…
device_info	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	…	…	…	…
hrv	time	Y*	UINT16	Refer to section 9.1.2.5

* Only required if optional message included

Note that the activity file makes use of dynamic fields, meaning the interpretation of some message fields will depend on the value of another field. For example, Field Definition #10 of the Session message is Total_Cycles. However, if the sport is Running, Total_Cycles should be interpreted at Total_Strides where it would be interpreted as Total_Strokes if the sport is rowing.

9.1.1 Activity File Structure

The activity file shall adhere to the following rules:

- An activity file contains one activity message.
- Multiple Session messages may be recorded for a single activity.
- Multiple Laps may be recorded for a single session.
- Multiple Lengths may be recorded for a single lap.
- The activity, session, lap, lengths summary messages may be grouped together at the start of the file (Figure 9-2) or interleaved with record, event, hrv messages (Figure 9-1). In either case these summary messages must be in chronological order.
- Record, event and hrv messages must be in chronological order.

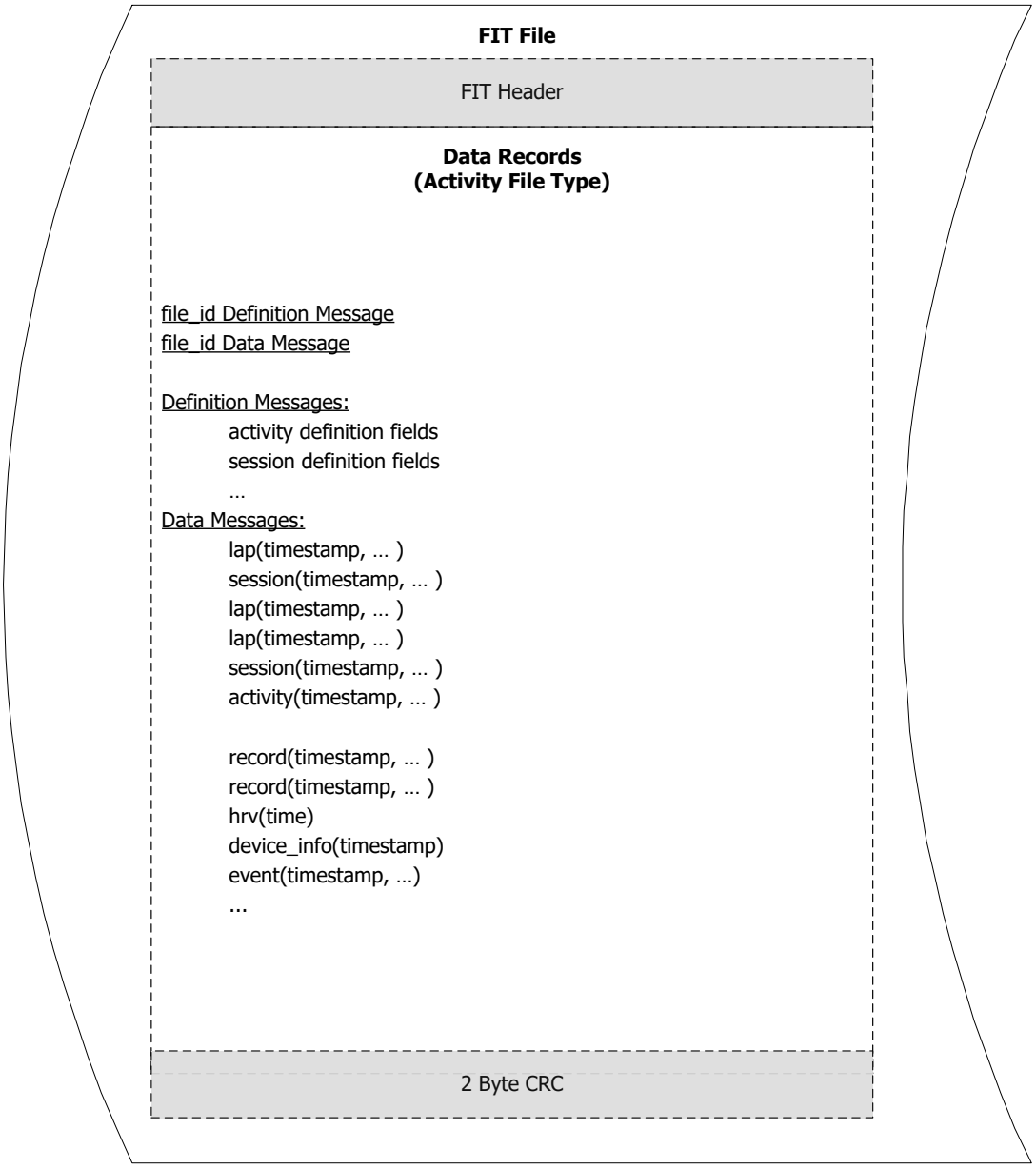


Figure 9-2. Alternate Activity File Structure

9.1.2 Activity File Message Description

An overview of the FIT activity file messages and fields is provided in Table 9-2.

Table 9-2. Activity File Message Descriptions

FIT Message	Description
activity	Provides a high level description of the overall activity file. This includes overall time, number of sessions and the type of each session.
session	Provides more summary detail including totals and averages over the entire session.
lap	Provide summary detail over the duration of a single lap. A lap breaks a session into segments of interest and could be based on distance, time, user action (i.e. button press), even landmarks or waypoints.
length	Provide summary detail over the duration of a single length. A length is a set distance such as the length of a pool, the length of a track/circuit.
record	Provide relatively high resolution, time-stamped data about the activity. This message carries instantaneous data such as speed, position, heart rate and bicycle power. Record messages must be in chronological order.
event	Used to record events within an activity including starting and stopping the timer. The event message can also record alerts. Event messages must be in chronological order.
hrv	Used to record heart rate variability data. The hrv data messages contain an array of RR intervals and are interleaved with record and event messages in chronological order.
device_info	Used to record information about the device such as battery life and operating time.

9.1.2.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

9.1.2.2 "record" Messages

The majority of activity data is stored using the record message. The record message allows information such as position (latitude, longitude), speed, heart rate, etc to be stored.

Accumulating fields shall NOT be reset to zero at any point within an activity file. These are fields such as distance, compressed_speed_distance, accumulating_power and compressed_accumulating_power. Some legacy devices have implemented a reset to zero in the record.distance field at the end of an individual session. In order to maintain reverse compatibility, when decoding the record.distance field, session boundaries shall be checked. If the record.distance field is set to zero at a session boundary, the decoder shall interpret this as a reset to 0 event and NOT a rollover event.

All records shall be timestamped (using either the timestamp field, or compressed timestamp headers).

9.1.2.3 "event" Messages

Event messages may be present throughout an activity file to indicate the presence of events such as timer start/stop, battery status, course points, alerts, etc. Most events are recorded using the FIT event message; however, Session, lap, and length messages are a special type of event message as described in the next sub section.

All events shall be timestamped and include both the event and event_type fields. Certain events shall have additional data options or requirements. For example, the recovery_hr event shall only occur after a session stop. The duration of the recovery is determined by the manufacturer.

9.1.2.4 Summary Messages

The session, lap, and length messages provide summaries of specific sections of an activity as described in Table 9-2. These summary messages also serve as session, lap and length event messages.

9.1.2.5 HRV Data

Heart Rate Variability (HRV) data may be recorded in an activity file. The hrv data message is an array of RR interval values. All hrv messages contained within the activity file shall be concatenated together into a single, large array of data. Note that hrv data is not timestamped, and shall be synchronized by checking successive RR intervals as they occur between timestamp activity "record" data.

Some devices require the HRV functionality to be enabled by the user. This can be achieved using a settings file, by including the hrm_profile message and setting the log_hrv field to true.

9.2 VIRB FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 4** for an activity file. A partial list of VIRB FIT messages and fields contained in an activity file are outlined in Table 9-12. For a full list of messages and fields refer to the FIT SDK.

Table 9-2. VIRB FIT Messages Contained in an Activity File

FIT Message	FIT Fields	Required	Type	Description
three_d_sensor_calibration	timestamp	Y	date_time (UINT32)	Whole second part of the timestamp
	sensor_type	Y	sensor_type	Indicates which sensor the calibration is for
	calibration_factor	Y	UNIT32	Calibration factor used to convert from raw ADC value to degrees, g, etc.
	calibration_divisor	Y	UINT32	Calibration factor divisor
	level_shift	Y	UNIT32	Level shift value used to shift the ADC value back into range
	offset_cal	Y	SINT32	Internal calibration factors, one of each: xy, yx, zx
	orientation_matrix	Y	SINT32	A 3x3 rotation matrix (row major)
	…	…	…	…
accelerometer_data	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	timestamp_ms	Y	UINT16	Millisecond part of the timestamp
	sample_time_offset	Y	UINT16	Each time in the array describes the time at which the accelerometer sample with the corresponding index was taken.
	accel_x	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	accel_y	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	accel_z	Y	UINT	These are the raw ADC reading. A conversion will need to be done on this data once read.
	calibrated_accel_x	N	FLOAT32	Calibrated accelerometer reading.

FIT Message	FIT Fields	Required	Type	Description
	calibrated_accel_y	N	FLOAT32	Calibrated accelerometer reading.
	calibrated_accel_z	N	FLOAT32	Calibrated accelerometer reading.
gyroscope_data	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	timestamp_ms	Y	UNIT16	Millisecond part of the timestamp
	sample_time_offset	Y	UINT16	Each time in the array describes the time at which the accelerometer sample with the corresponding index was taken
	gyro_x	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	gyro_y	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	gyro_z	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	calibrated_gyro_x	N	FLOAT32	Calibrated gyroscope reading
	calibrated_gyro_y	N	FLOAT32	Calibrated gyroscope reading
	calibrated_gyro_z	N	FLOAT32	Calibrated gyroscope reading
magnetometer_data	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	timestamp_ms	Y	UINT16	Millisecond part of the timestamp
	sample_time_offset	Y	UINT16	Each time in the array describes the time at which the accelerometer sample with the corresponding index was taken
	mag_x	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	mag_y	Y	UINT16	These are the raw ADC reading. A conversion will

FIT Message	FIT Fields	Required	Type	Description
				need to be done on this data once read..
	mag_z	Y	UINT16	These are the raw ADC reading. A conversion will need to be done on this data once read.
	calibrated_mag_x	N	UINT16	Calibrated magnetometer reading
	calibrated_mag_y	N	UINT16	Calibrated magnetometer reading
	calibrated_mag_z	N	UINT16	Calibrated magnetometer reading

Note that the activity file makes use of dynamic fields, meaning the interpretation of some message fields will depend on the value of another field. For example, sensor_type indicates which sensor the calibration message is for and as such asserts different units to calibration_factor.

9.2.1 "three_d_sensor_calibration" Messages

The three_d_sensor_calibration message identifies the calibration values needed to convert the data recorded by the sensors to the desired units for the end user. The calibration message can currently be used for values collected by accelerometers, gyroscopes, and magnetometer (compass) into g, deg/s, and G's respectively. The calibration values help to tare the sensor, adjust for positioning, and keep the data points within the correct range.

9.2.2 "accelerometer_data" Messages

The accelerometer sensor collects samples that consist of x, y, z points and a sample offset time that indicates the milliseconds between when the sample was taken and the timestamp in the message. The samples may span between multiple seconds but the message is limited to 30 samples and each sample point is limited to 16-bits. The x, y, z points are the raw ACD values with some mild processing done on them by the accelerometer. Calibrated x, y, and z fields exist to allow processed x, y, z points to be added to the message.

9.2.3 "gyroscope_data" Messages

The gyroscope sensor collects samples that consist of x, y, z points and a sample offset time that indicates the milliseconds between when the sample was taken and the timestamp in the message. The samples may span between multiple seconds but the message is limited to 30 samples and each sample point is limited to 16-bits. The x, y, z points are the raw ACD values with some mild processing done on them by the gyroscope. Calibrated x, y, and z fields exist to allow processed x, y, z points to be added to the message.

9.2.4 "magnetometer_data" Messages

The magnetometer sensor collects samples that consist of x, y, z points and a sample offset time that indicates the milliseconds between when the sample was taken and the timestamp in the message. The samples may span between multiple seconds but the message is limited to 30 samples and each sample point is limited to 16-bits. The x, y, z points are the raw ACD values with some mild processing done on them by the magnetometer. Calibrated x, y, and z fields exist to allow processed x, y, z points to be added to the message.

10 Activity Summary File

Activity summary files are a compact version of the activity file and contain only activity, session and lap messages (**Error! Reference source not found.**).

See section 1.1 for other general guidelines

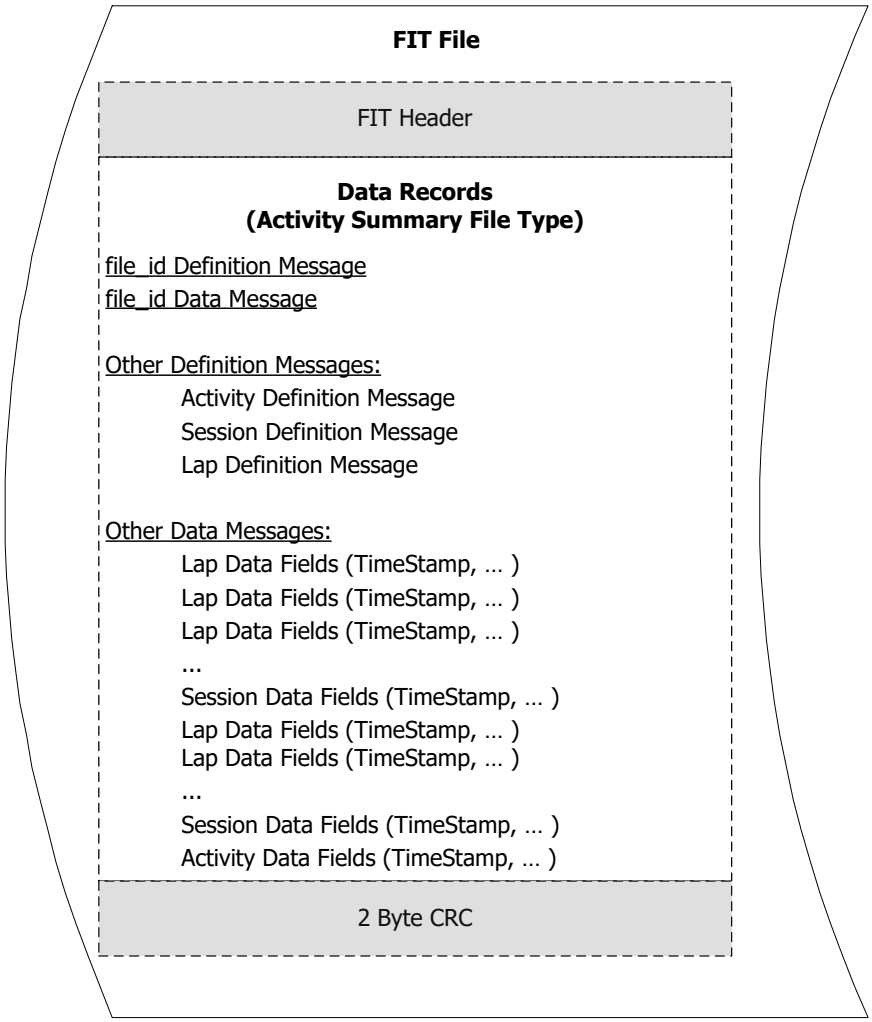


Figure 10-1. Activity Summary File

10.1 FIT Messages

All FIT files must start with a file_id message. The FIT file_id.type = 20 (0x14) for an activity summary file.

The activity summary file follows similar requirements as the activity file:

- Summary messages are logged when the interval concludes
- Only a single activity message shall be recorded.
- Multiple session messages may be recorded for a single activity message.
- Multiple lap messages may be recorded for each session message.

11 Course File

A course file contains data from a recorded activity that can be transferred to a display device to guide a user through the same activity. All FIT files must start with a file_id message. The FIT **file_id.type = 6** for a course file.

The course file should, at a minimum, contain the file_id, lap, record, and course FIT messages; and may optionally contain the course_point message (Figure 11-1).

The file_id, course, lap, and optional course_point messages shall be defined and recorded sequentially, using only local message type (i.e. 0). The file_id, and course messages need only be recorded once, at the start of the course file. At least one lap message will be recorded in each course file; however multiple lap messages may be recorded if desired. Redefining local message type 0 for all of these messages will ensure simple processors can handle all course data. The rest of the course file will consist of multiple record messages detailing the course (Figure 11-1).

See section 1.1 for other general guidelines

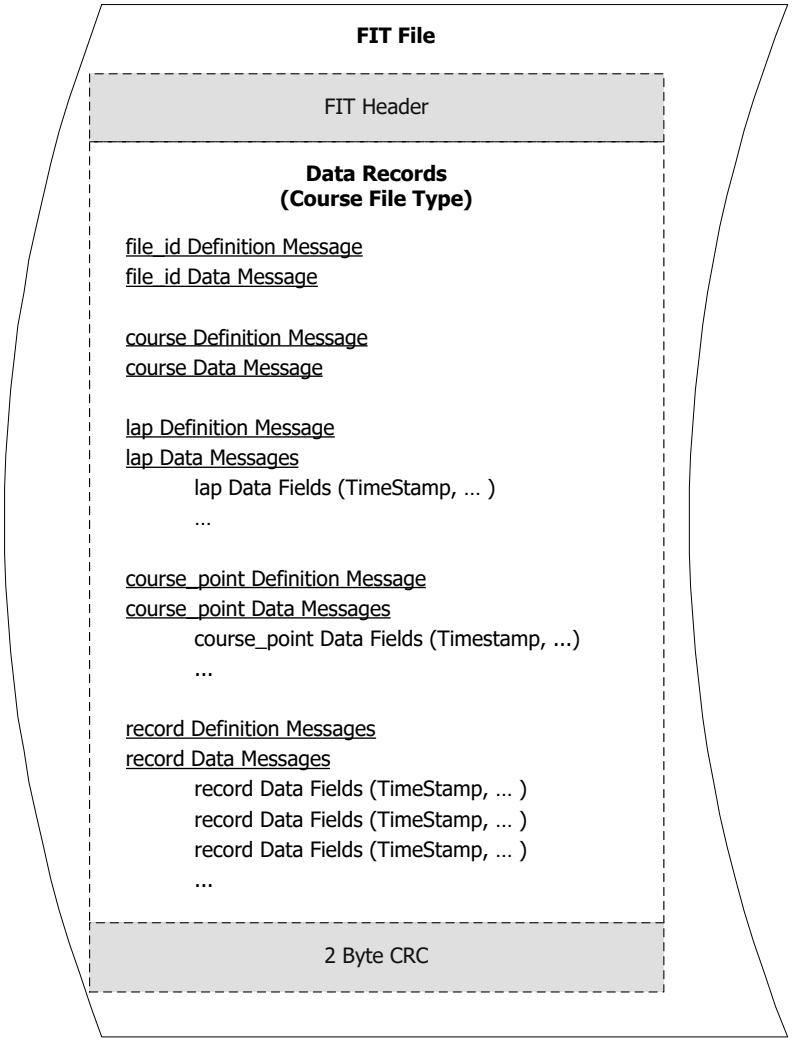


Figure 11-1. Course File

Course files contain a series of activity record data messages which can be used by a display, or fitness console, to recreate the activity for the same, or different, users to repeat (Figure 11-2). Record messages may contain positional information such as latitude, longitude and altitude; user information such as speed, heart rate and power; as well as information such as current distance and temperature. Each record is used to create a point along the course.

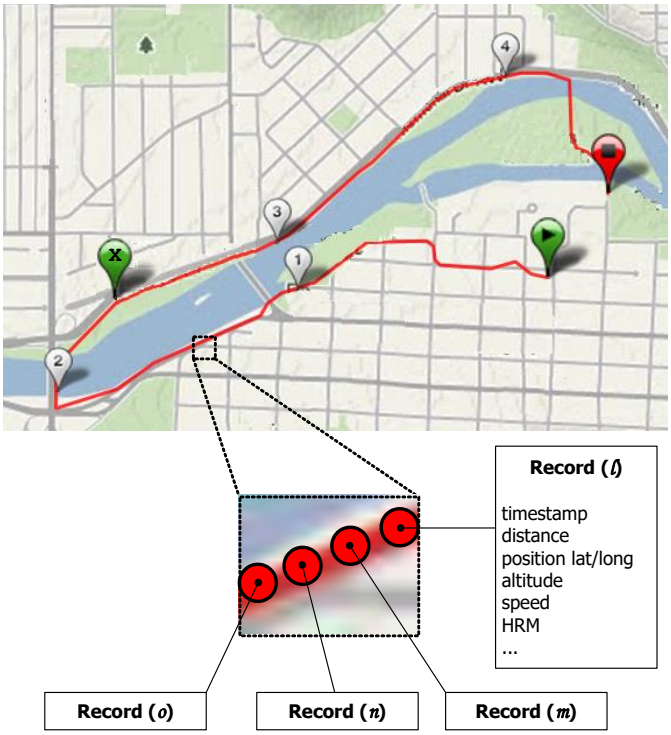


Figure 11-2. Activity Record Messages Used to Create a "River Run" Course

Course files also contain lap and course_point messages to provide summary activity data, and key course milestones and/or landmarks. Figure 11-3 shows the example "River Run" course file with lap and course_point messages.

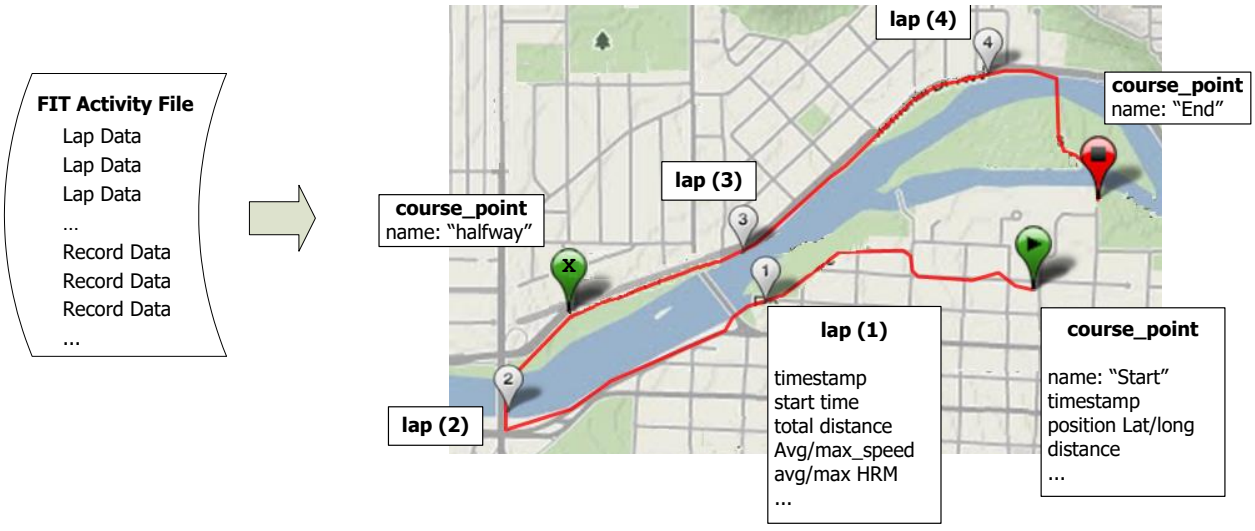


Figure 11-3. "River Run" Course File with Laps and Course_points

11.1 FIT Messages

The list of FIT messages and fields contained in a course file are outlined in Table 11-1. Note that not all fields are required.

Table 11-1. FIT Messages Contained in Course File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Course File (=6)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	number	N	UINT16	File identifier
course	sport	N	sport (enum)	Type of sport course relates to
	name	Y	string	Name of course
	capabilities	N	course_capabilities (enum)	Indicates content of course file
course_point	message_index	N	message_index (UINT16)	Provides an index for each course point
	timestamp	Y*	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	position_lat	N	SINT32	Semicircles
	position_long	N	SINT32	Semicircles
	distance	N	UINT32	1/100 m
	type	N	course_point (enum)	Refer to FIT SDK for course types
	name	N	string	
lap**	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	start_time	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	total_distance	N		
record**	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	distance	Y	UINT32	Distance incm
	position_lat	N	SINT32	Latitude in semicircles
	position_long	N	SINT32	Longitude in semicircles

* Only required if optional message included

** Fields for FIT message only partially listed. Refer to FIT SDK for full listing.

11.2 Course File Example

Figure 11-4 shows a 9.5 km running activity that shall be used to create the example course file. Each point along the course is represented by one of the activity file's "record" messages, which consists of positional, distance and user information.

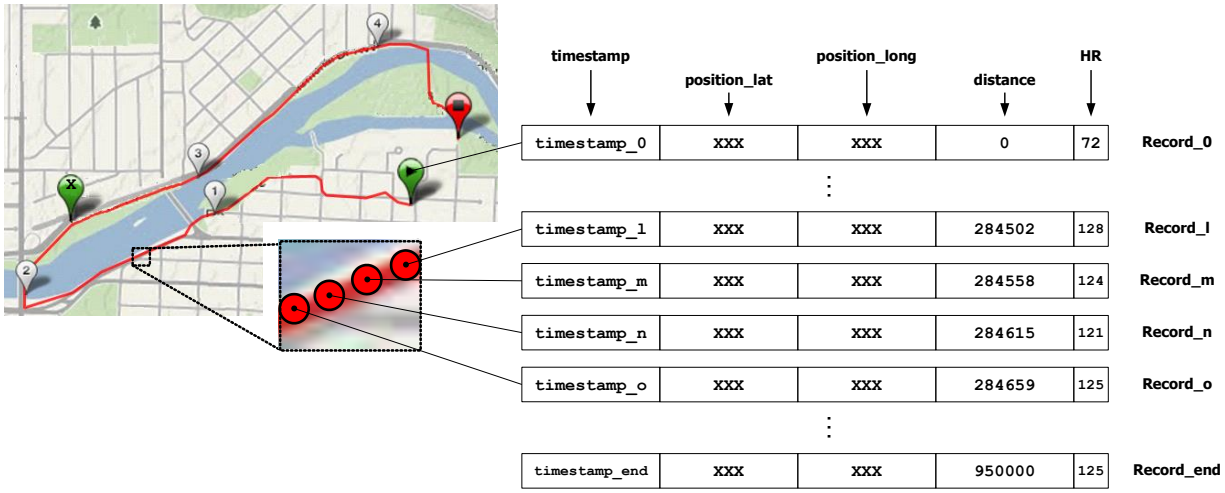
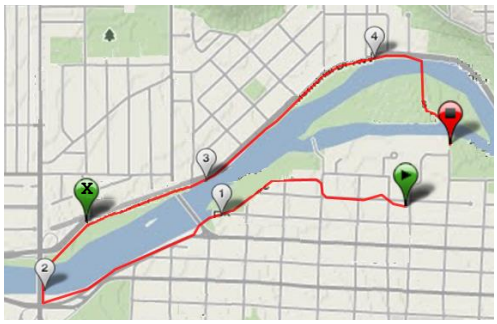


Figure 11-4. Example record Data Messages of a Course File

Each record has an associated timestamp, latitude and longitude position, distance run, and the user's heart rate. For simplicity the actual lat/long coordinates are represented by "XXX".

In this case, four laps events were also recorded, each representing a 2km distance completed (Figure 11-5). Each lap message contains the lap start/end times, and the user's maximum and average heart rate. This example also utilses the FIT course_point message to represent the start, halfway and end points of the course.



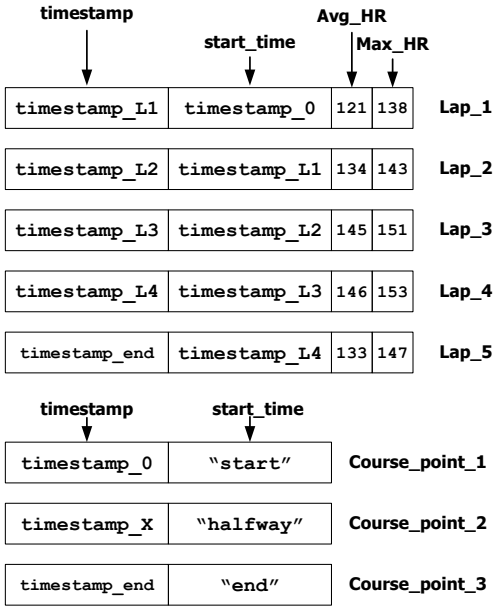
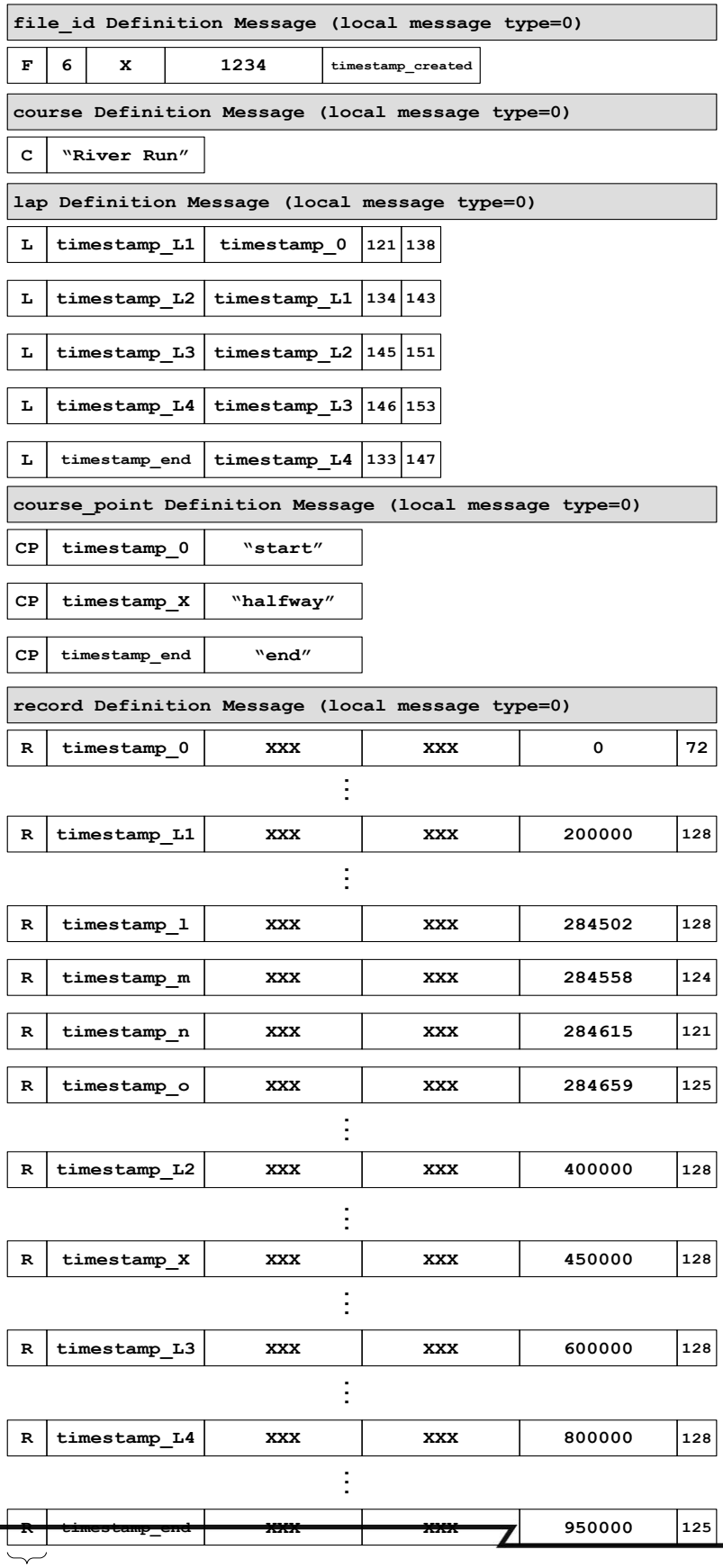


Figure 11-5. Example lap and course_point Data Messages of a Course File

Figure 11-6. Example Course File



The resultant course file is formatted as shown in Figure 11-6.

In this example, each lap and course_point message can be associated to a record message through a matching timestamp. As such, each lap or course_point message does not need to contain any positional, distance, or heart rate data as this can be obtained from the matching record.

Every definition and data message in this example uses local message type 0, ensuring simple processes can handle all course data

Record messages are stored in chronological order

12 Goals File

Goals files allow a user to communicate their exercise/health goals. Goals may be set for a variety of activities, over specific periods of time, and with desired targets set according to total duration, calories consumed, distance travelled, number of steps taken and/or frequency of activity (Figure 12-1). Multiple goals may be set, and grouped according to sport.

See section 1.1 for other general guidelines

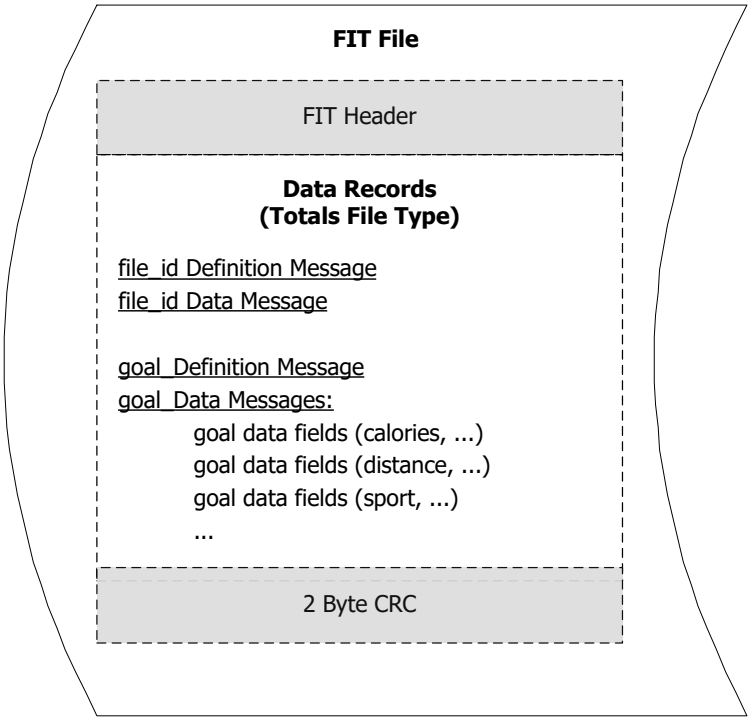


Figure 12-1. Goals File

12.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 11** for a goals file. The following FIT messages can also be included in a goals file.

Table 12-1. FIT Messages Contained in Goals File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Goals File (=11)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time	File creation time
	number	N*	UINT16	File identifier
goal	message_index	N	UINT16	Provides an index such that other FIT messages can be related to this message
	sport	N	sport (enum)	Type of sport relating to reported goals
	sub_sport	N	sub_sport (enum)	Type of sub sport relating to reported goals
	start_date	N	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	end_date	N	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	type	N	goal (enum)	Refer to profile.xls
	value	N	UINT32	
	repeat	N	bool	
	target_value	N	UINT32	
	recurrence	N	goal_recurrance (enum)	Refer to profile.xls
	recurrence_value	N	UINT16	Total calories consumed during recorded activity
	enabled	N	bool	

Goal messages indicate the user's goal for a specific sport (if applicable) and subsport (if applicable)

13 Totals File

Totals files are used to summarize a user's activities and may contain multiple totals messages each representing summaries of a different activity type/sport (Figure 13-1).

See section 1.1 for other general guidelines

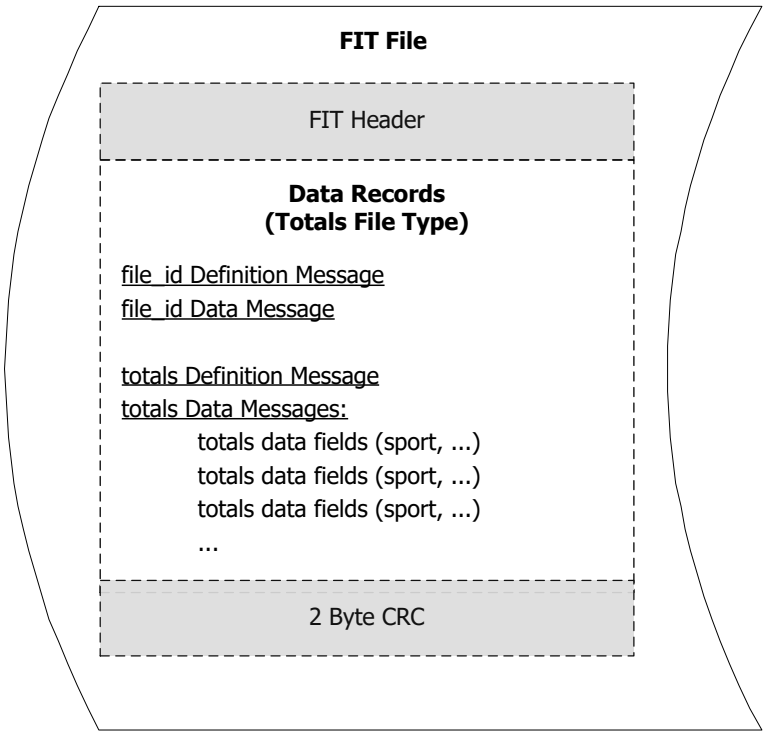


Figure 13-1. Totals File

13.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 10** for a totals file. The following FIT messages can also be included in a totals file.

Table 13-1. FIT Messages Contained in Totals File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Totals File (=10)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
totals	message_index	N	UINT16	Provides an index such that other FIT messages can be related to this message
	timestamp	Y	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	timer_time	N	UINT32	Total time of activities, excluding pauses/rests
	distance	N	UINT32	Total distance covered during recorded activities
	calories	N	UINT32	Total calories consumed during recorded activities
	sport	Y	sport (enum)	See profile.xls for details

Totals messages indicate the user's total amount of recorded distance, calories, and/or active time and may be grouped by sport type, such as running, cycling fitness equipment, etc.

13.2 Totals File Example

Figure 13-2 shows an example totals file. The file begins with file_id definition and data messages, indicating the file is a totals file (file_id.type = 10), the manufacturer is dynastream (file_id.manufacturer =15), and the product is "1" with serial number "123456."

The totals file then contains the totals definition and data messages. In this case, the totals message includes message_index, timestamp, timer time, distance and sport. The file indicates totals data is available for three sports: generic (sport=0), running (sport=1) and cycling (sport=2).

The user has not performed any generic or cycling activity with the totals fields indicating 0 total timer time and distance. However, the data shows the user has performed 1167 seconds worth of running, covering 4669m.

file_id Definition Message
(local message type=0, fields: type, mfg, product, serial_number)

F	10	15	1	123456

totals Definition Message
(local message type=0, fields: msg_index, timestamp, timer_time, distance, sport)

T	0	timestamp_X	0	0	0
T	1	timestamp_X	1167	4669	1
T	2	Timestamp_X	0	0	2

HEADER BYTE
(F: file_id, T: totals)

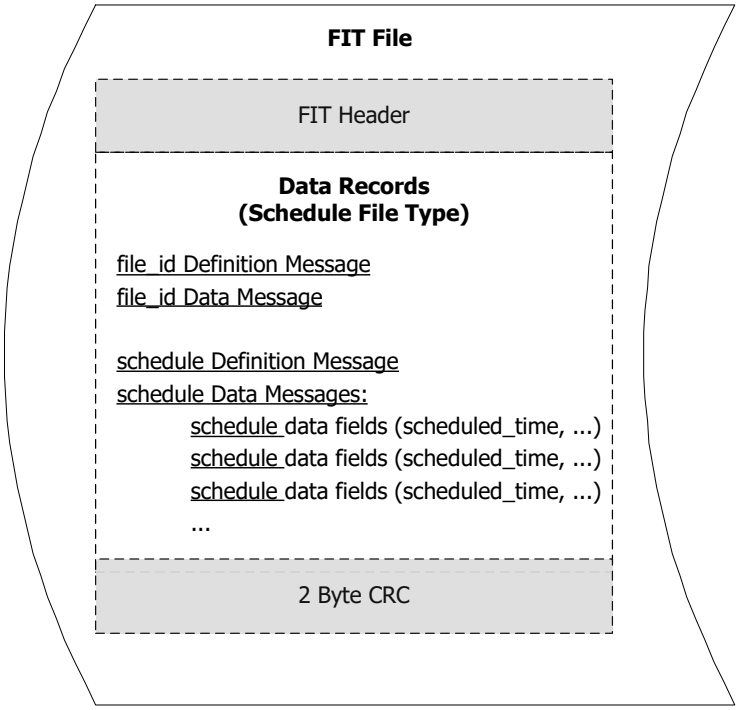
Figure 13-2. Totals File Example

14 Schedule File

Schedule files are used to schedule a user's workouts and may contain multiple schedule messages each representing the start time of a workout (Figure 14-1). A single workout, or multiple workouts, may be scheduled multiple times within a schedule file. Multiple courses or workouts may also be scheduled at the same time.

See section 1.1 for other general guidelines

Figure 14-1. Schedule File



Schedule messages do not have to be recorded chronologically by scheduled time; however, this is a recommended best practice.

14.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 7** for a schedule file. The following FIT messages can also be included in a schedule file.

Table 14-1. FIT Messages Contained in Schedule File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Schedule File (= 7)
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	N*	date_time (UINT32)	File creation time
	number	N*	UINT16	File identifier
schedule	manufacturer	Y	manufacturer (UINT16)	Corresponds to file_id.manufacturer of scheduled workout/course
	product	Y	UINT16	Corresponds to file_id.product of scheduled workout/course
	serial_number	Y	UINT32z	Corresponds to file_id.serial_number of scheduled workout/course
	time_created	Y	date_time (UINT32)	Corresponds to file_id.time_created of scheduled workout/course
	completed	N	Bool	Indicates if the scheduled item has been completed
	type	Y	schedule (enum)	Indicates if schedule is for a workout or course file
	scheduled_time	Y	local_date_time (UINT32)	Seconds since 00:00 Dec 31 1989 local time

Schedule messages indicate the time at which a specific workout or course should commence. The schedule message references the file_id of the course/workout that is to be performed.

14.2 Schedule File Example

For this example, assume the device has three FIT workout files, with file_id messages as shown in Figure 14-2.

Workout File A

file_id.manufacturer = 15
file_id.product = 1
file_id.serial_number = 12345
File_id.time_created = Timestamp_X1

Workout File B

file_id.manufacturer = 15
file_id.product = 1
file_id.serial_number = 12345
File_id.time_created = Timestamp_X2

Workout File C

file_id.manufacturer = 15
file_id.product = 1
file_id.serial_number = 12345
File_id.time_created = Timestamp_X3

Figure 14-2. Available Workout Files (file_id details)

Figure 14-3 shows an example schedule file. The file begins with file_id definition and data messages, indicating the file is a schedule file (file_id.type = 7), the manufacturer is dynastream (file_id.manufacturer =15), and the product is "1" with serial number "5555"

The schedule file then contains the schedule definition and data messages. In this case, the schedule message includes manufacturer, product, serial_number, time_created, type, completed, and scheduled_time fields.

The file indicates that there are four scheduled workouts (schedule.type=0). The combination of schedule.manufacturer, schedule.product, schedule.serial_number and schedule.time_created indicate the first scheduled workout is for "Workout File A". This combination of fields in the second schedule message indicates the scheduled workout is again for "Workout File A". The third schedule message indicates "Workout File C", and the fourth "Workout File A". The schedule.scheduled_time field indicates the time at which the workouts should occur.

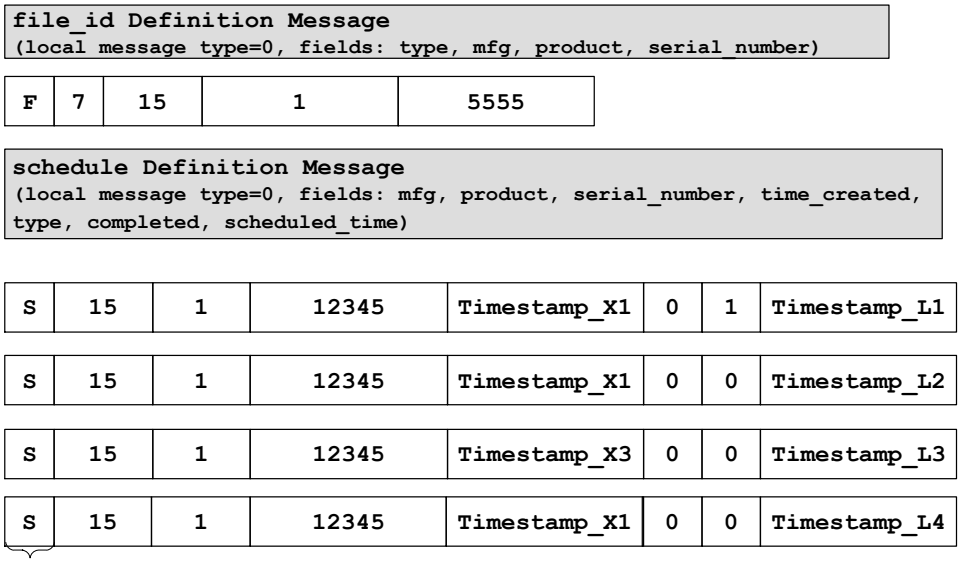


Figure 14-3. Schedule File Example

If Timestamp_L1 & L2 < current time < Timestampe_L3 & L4, this file also indicates that the first two workouts should have occurred, and the last two workouts are still to happen. The schedule.completed field can be used to determine whether the user actually performed the workout or not.

15 Monitoring A&B File

MonitoringA and MonitoringB files are used to store data that is logged over varying time intervals. The two monitoring file formats are identical apart from supporting different conventions for file_id.number and the start of accumulating data values. Monitoring B defines that accumulating data values start at the beginning of the day in the local time. This allows for example a monitoring file that logs steps for an hour in a day to also indicate the total steps accumulated in the day without reading other files. The actual data that is stored may vary depending on use case (Figure 15-1). Additional information (such as battery_status) may also be stored by use of the device_info message. No assumptions should be made about the frequency of records found in the file.

See section 1.1 for other general guidelines

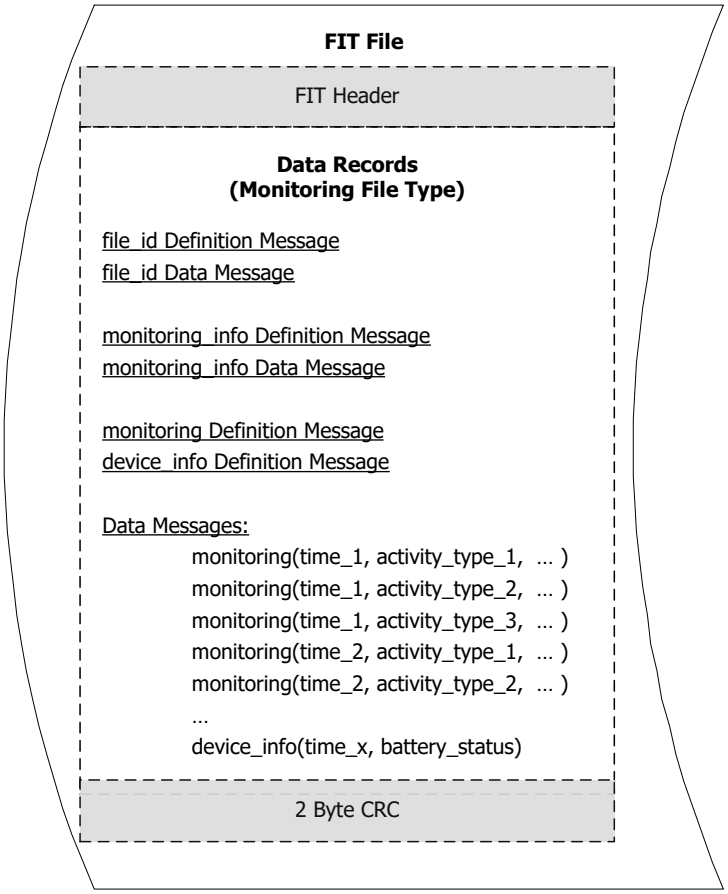


Figure 15-1. Monitoring File

15.1 FIT Messages

Activity monitoring data is stored in a FIT monitoring file. The FIT **file_id.type = 15** for a **monitoring_a** file and the FIT **file_id.type = 32** for a **monitoring_b** file. All monitoring files must contain the FIT file_id and monitoring messages as described below.

Table 15-1. FIT Messages Contained in Monitoring File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Monitoring_a file = 15 Monitoring_b file = 32
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time (UINT32)	Time file was created. Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	number	Y	UINT16	For monitoring_a indicates the type of monitoring file. Refer to 15.1.1. For monitoring_b incrementing file number. Higher numbers indicate newer files.
monitoring_info	timestamp	N	date_time (UINT32)	Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	local_timestamp	N	local_date_time (UINT32)	Seconds since 00:00 Dec 31 1989 Local Time
monitoring	timestamp	Y	date_time (UINT32)	Time of measurement recording
	device_index	N	device_index (UINT8)	Associates data to a specific device, not required for a file associated with a single device.
	calories	N	UINT16	Accumulated calories
	distance	N	UINT32	Accumulated distance
	cycles	N	UINT32	Accumulated cycles
	active_time	N	UINT32	Accumulated active time
	activity_type	N	activity_type (enum)	Refer to profile.xls
	activity_subtype	N	activity_subtype (enum)	Refer to profile.xls
	activity_level	N	activity_level (enum)	Measure of activity. Refer to profile.xls
	distance_16	N	UINT16	Compressed accumulated distance
	cycles_16	N	UINT16	Compressed accumulated cycles
	active_time_16	N	UINT16	Compressed accumulated active time

	local_timestamp	N	local_date_time (UINT32)	Timestamp expressed in local time. May be populated by monitoring reader during decode post processing.
	temperature	N	SINT16	Average temperature in 0.01°C during the logging interval
	temperature_min	N	SINT16	Minimum temperature in 0.01°C during the logging interval
	temperature_max	N	SINT16	Maximum temperature in 0.01°C during the logging interval
device_info	timestamp	N	date_time (UINT32)	Time of message generation
	battery_status	N	battery_status (UINT8)	Battery condition

15.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

For monitoring_a the file_id.number field is used as described in Table 15-2. For monitoring_b, file_id.number is an incrementing number.

Table 15-2. FIT Monitoring File file_id.number Format

FIT Field	Bits	Description	Value/Units
file_id.number	6:15	file_subtype	0: Generic Monitoring File
			1: Activity Monitoring Data 2: Temperature Data
			3:1023 - Reserved
	0:5	file_duration	0: Generic/All data
			1: New Data (since last download or power up)
			2: Detailed Daily (24 hours)
			3:63 – reserved

15.1.2 "monitoring_info" Message

The monitoring_info message is optional. This message allows the local time zone offset to be established. If the monitoring_info message is used, it need only be recorded once and both timestamp and local_timestamp shall be included.

15.1.3 "monitoring" Message

As indicated in the "Required" column, not all of the listed fields must be included in the monitoring file. In addition to the timestamp at least one of: calories, distance, cycles, active_time, distance_16, cycles_16, active_time_16 etc. should be populated.

The activity_type, activity_subtype and activity_level fields are used to classify the recorded data. These fields are optional. All data is grouped by activity type if specified. For example, steps values in the monitoring message are accumulated separately for walking and running.

15.1.4 "device_info" Message

The device_info message may be included to capture useful information about the monitoring sensor.

15.2 Accumulated Values

Monitoring data is logged in intervals and some fields are stored using accumulated values. The timestamp represents the conclusion of the interval. This means that data for a specific logging interval is calculated from the *difference* between two timestamps. For example if data is logged hourly as shown below:

Record1: Timestamp = Monday 15 August 2011 8:00am; cycles = 200; activity_type = walking

Record2: Timestamp = Monday 15 August 2011 9:00am; cycles = 1200; activity_type = walking

These two records correspond to a single 1 hour logging interval. The number of steps taken in that logging interval is simply calculated as:

Record2 – Record1 = 1000 walking steps

Monitoring files that use accumulated values, shall always log a starting point. The starting point shall be zero, or the last known recorded value. Using accumulated values allows data to be easily converted into longer intervals by sub sampling or shorter intervals by interpolating.

Refer to sections 15.3 and 15.5 for more detailed examples.

15.3 Monitoring Reader

Since the FIT profile does not define distinct fields for distance, calories, cycles etc. for each activity_type, additional decode logic is required to implement the monitoring file examples as described in section 15.4. The *Monitoring Reader* class extends the base FIT SDK and offers additional processing for monitoring files. See the FIT SDK Introductory Guide for further details.

The Monitoring Reader is only required if multiple activity_types are logged and must be differentiated.

15.4 Monitoring File Example

The FIT monitoring file includes the file_id, monitoring and optional messages (e.g. monitoring_info, device_info). These messages are structured as shown in Figure 15-2.

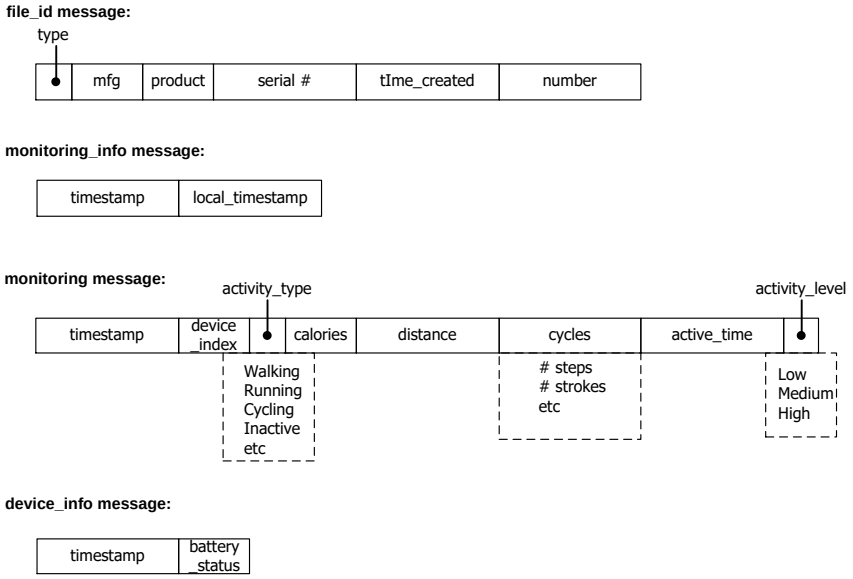


Figure 15-2. FIT Monitoring File Messages & Fields

Data from multiple activity types may be linked by timestamp. Figure 15-3 shows example data from an activity monitor that records running, cycling and 'generic' activity. Data is logged hourly for each activity type.

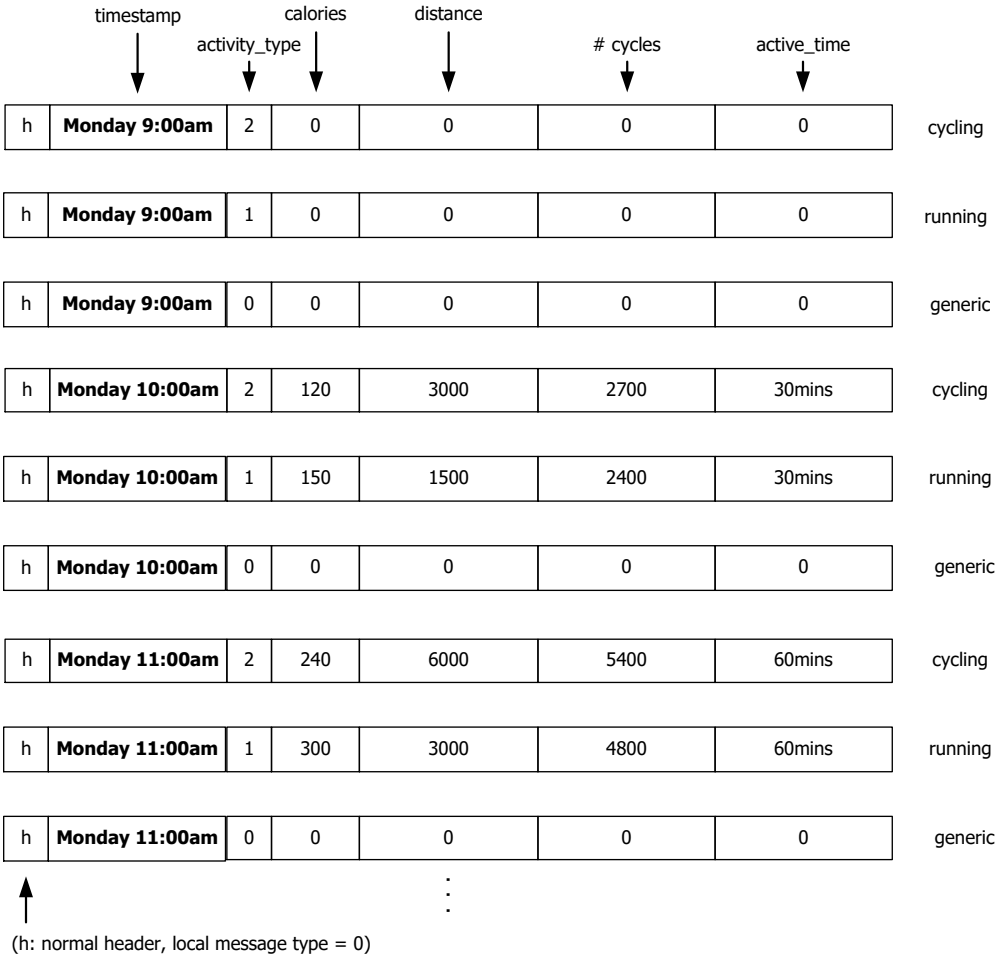


Figure 15-3. Example FIT Monitoring File Data Records

In this example, the following data may be calculated:

- Between 9 am and 10 am:
    - Cycling: 120 calories, 3 km, and 2700 "cycles" occurred over 30 minutes.
    - Running: 150 calories, 1.5 km, and 2400 steps occurred over 30 minutes.
    - No generic activity was recorded
- Between 10 am and 11 am:
    - Cycling: 120 calories, 3 km, and 2700 "cycles" occurred over 30 minutes.
    - Running: 150 calories, 1.5 km, and 2400 steps occurred over 30 minutes.
    - No generic activity was recorded

15.5 File Optimisation Options

File optimisation techniques may be used to further reduce the FIT monitoring file size. These optional optimisation techniques are discussed in the following sections.

15.5.1 Compressed Timestamp Headers

For activity monitors that detect more than one activity type, multiple data messages may be recorded for a single timestamp value (as shown in earlier example of Figure 15-3). In this case, the sport and/or activity_level field shall be included to differentiate the activities, and data for the same logging interval is linked using identical timestamps. Compressed timestamp headers may be used to reduce file size by eliminating the need of a four byte timestamp for *every* data record. Instead, a single 4 byte timestamp may be used for the first data message of a specific point in time, using a normal record header, and compressed timestamp headers may be used for identically timestamped data of different activity types. Refer to Figure 15-4 for an example.

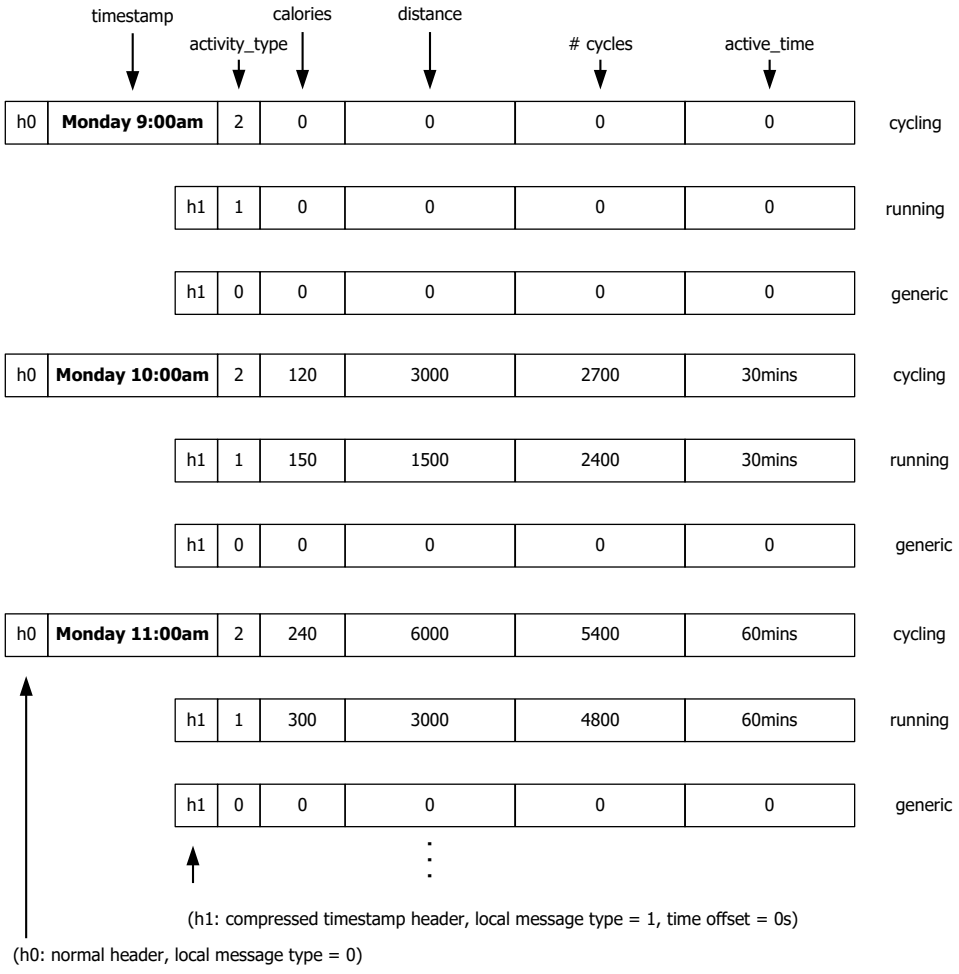


Figure 15-4. Example FIT Monitoring File Data Records with Compressed Timestamp Headers

In the given example, activity_type = 0 indicates generic activity, activity_type = 1 is running, and activity_type = 2 indicates cycling. Each activity type is recorded at a 1 hr logging interval.

The timestamp fields are used to link records. The header byte indicates if the FIT message is recorded using a normal header (h0), or a compressed timestamp header (h1).

Using a compressed timestamp header (h1) will allow linked data to be recorded without a 4 byte timestamp; instead, the header includes a time offset from the last recorded timestamp. In this case, the first record uses a normal header (h0) and

sets the time at Monday 9:00am. The two subsequent compressed timestamp headers (h1) indicate an offset of 0 seconds. This links the first 3 messages at Monday 9:00am.

Similarly, the next record uses a normal header (h0) and sets the time at Monday 10:00am, and the two subsequent compressed timestamp headers (h1) indicate an offset of 0 seconds, linking the next 3 messages at Monday 10:00am.

Note, compressed time headers are only used/necessary if multiple activity_types, activity_subtypes or activity_levels are recorded.

If using compressed timestamp headers, two local message types will be required:

- one local message type defined to include the timestamp field, and used with a normal record header. The data message will serve as the time reference for linked records (h0 in the example).
- one local message type shall be defined not including the timestamp field and used with the time compressed header showing 0 seconds offset (h1 in the example).

Calories, cycles, distance, and active time are accumulated values. This means that the actual values for a set logging interval are calculated as the difference between the current and previous records. **As a result, a starting point shall always be logged.** The starting point shall be zero, or the last known recorded value.

In the provided example, there are two intervals recorded. The starting point is a zero point and activity data is calculated for each activity_type and logging interval as shown below:

1st logging interval (9-10am):

- no generic activity
- 30 mins of running, 150 cals, 1.5km, 2400 steps
- 30 mins of cycling, 120 cals, 3km, 2700 cycles

2nd logging interval (10-11am):

- no generic activity
- 30 mins of running, 150 cals, 1.5km, 2400 steps
- 30 mins of cycling, 120 cals, 3km, 2700 cycles

15.5.2 Data Record Compression

Unchanging monitoring data does not have to be logged for each time logging interval. In the example in Figure 15-4, the activity monitor did not record any "generic" activity between 9 and 11 am. The file could further be compressed by omitting the "generic" monitoring messages until actual "generic" activity data is recorded for that activity type (Figure 15-5).

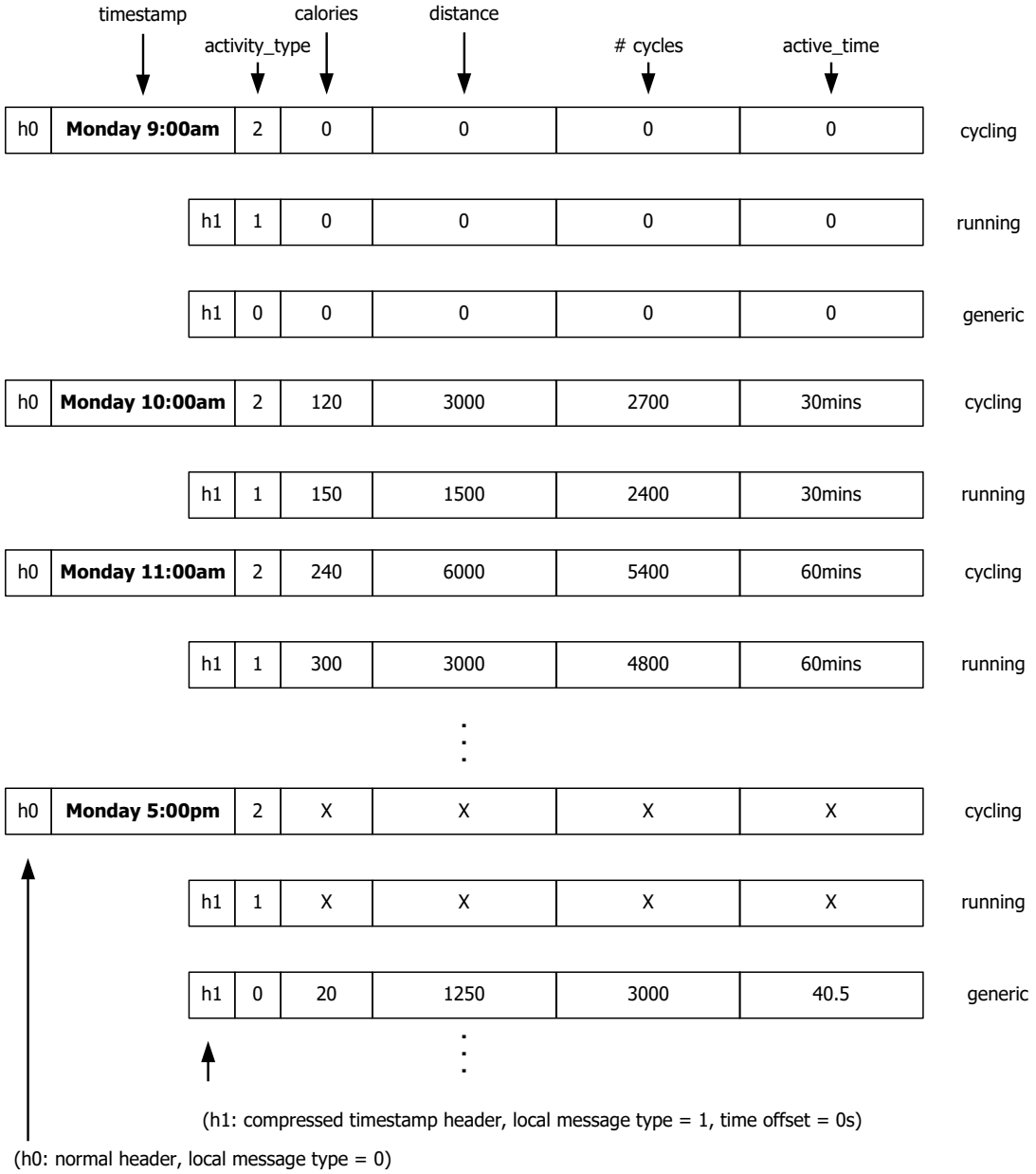


Figure 15-5. Example 1: Compressed Timestamp Headers & Data Record Compression

It is possible using this method that some time resolution may be lost; in Figure 15-5, for example, 40 minutes and 30 seconds of generic activity was recorded sometime between 9:00am and 5:00pm, but it's not known exactly when in that 8 hours the activity occurred. Although the records appear to be logged every hour, no assumptions about the message frequency should be made.

This loss of resolution can be prevented by recording a starting reference point as shown in Figure 15-6. Including the 'generic' activity data record at 4:00pm indicates specifically that no generic activity was recorded between 9:00 am and 4 pm. Furthermore, the 40 minutes of generic activity is now known to have occurred between 4:00pm and 5:00pm.

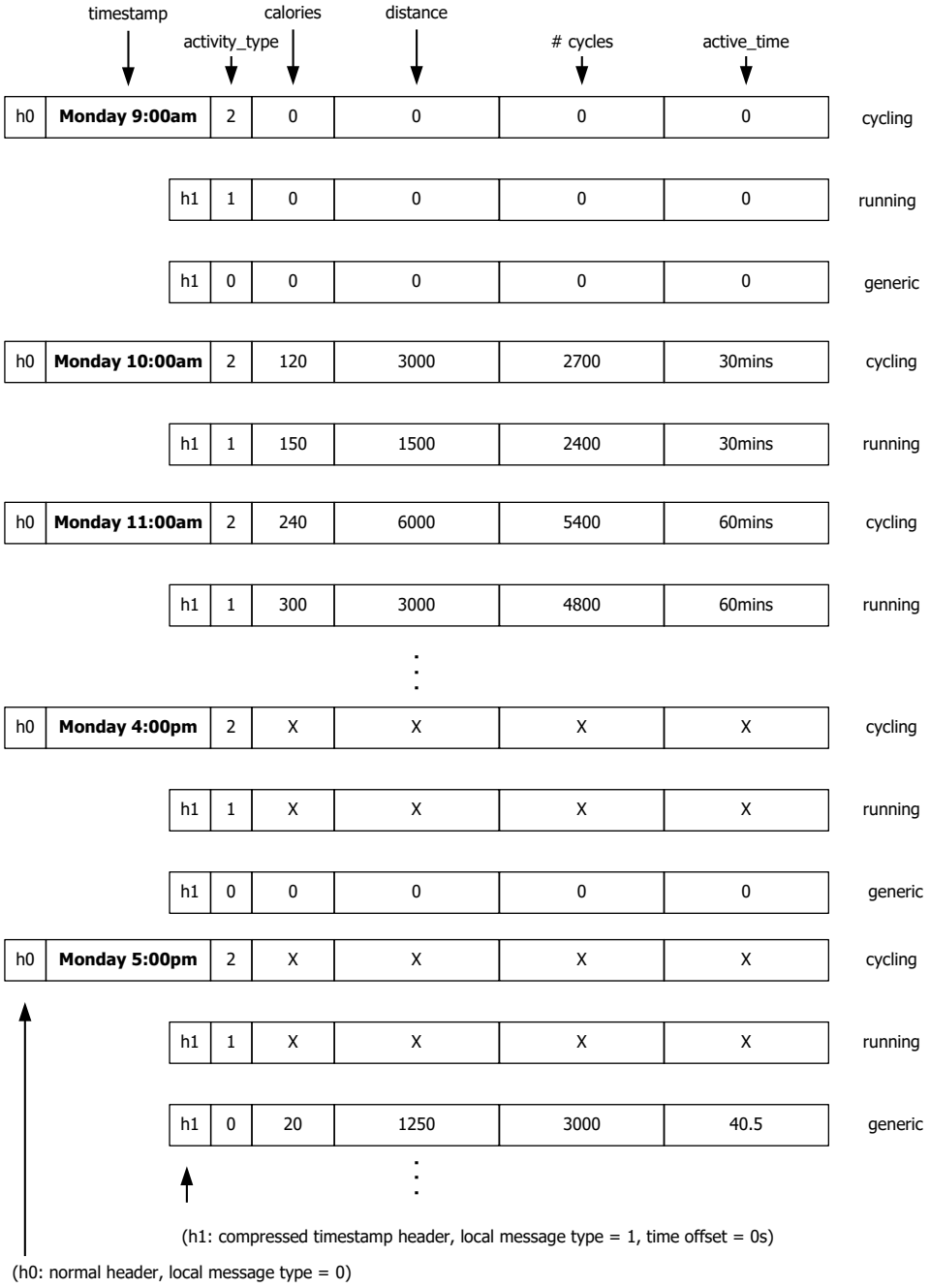


Figure 15-6. Example 2: Compressed Timestamp Headers & Further Data Compression

NOTE: The developer has full control over balancing compression and time resolution.

16 Daily Monitoring File

Daily monitoring files follow the same format as monitoring files, however data is logged at 24 hour time intervals (Figure 16-1). This allows summary information to be stored separately in a small file for rapid download. The FIT **file_id.type = 28 (0x1C)** for a daily monitoring file.

See section 1.1 for other general guidelines

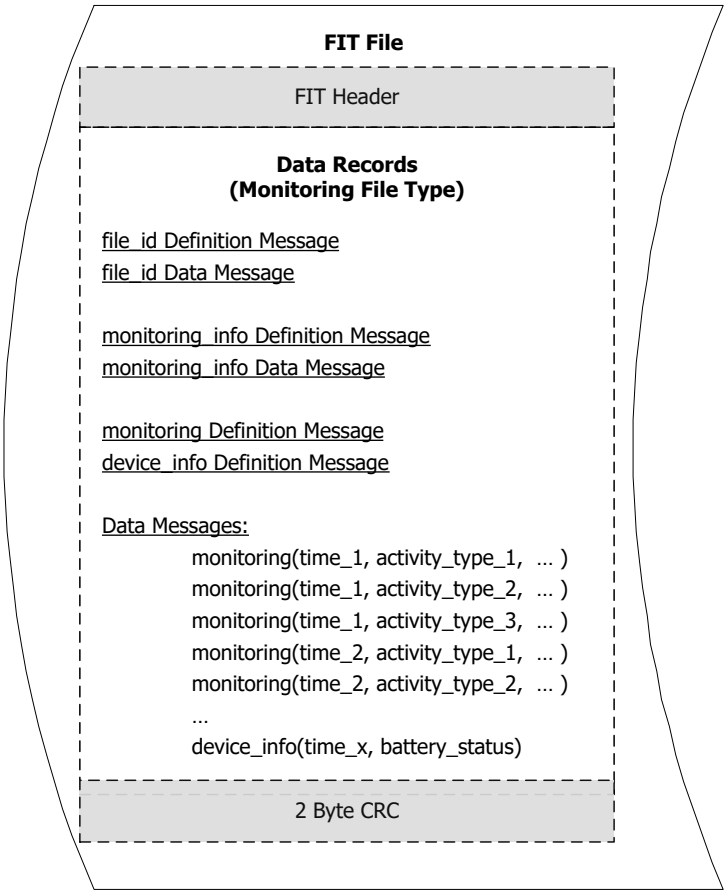


Figure 16-1. Daily Monitoring File

Daily monitoring files follow the same requirements as for a monitoring file:

- The same required fields as stated in Table 15-1.
- The same file structure.
- The same file optimization and compression techniques.

Different requirements:

- The data shall be recorded at a logging_interval of 24 hours (i.e. daily values).
- The file_id.file_number structure is similar as for a monitoring file, however, the file_duration field is as defined in FIT Messages
- Table 16-1.

16.1 FIT Messages

Table 16-1. FIT Daily Monitoring File file_id.file_number Format

FIT Field	Bits	Description	Description
file_id.number	6:15	File_subtype	0: Generic Monitoring File
			1: Activity Monitoring Data
			2: Temperature Data
			3:1023 - Reserved
	0:5	File_duration	0: Generic/All data
			1: New Data (since last download or power up)
			2: monthly
			3:63 - reserved

17 Segment File

Segment files contain data defining a route and timing information to gauge progress against previous performances or other users. Segment files contain a segment_id, segment_leaderboard_entry and segment_lap message to define and summarize the segment and a number of segment_point messages to define the route and target time. Other messages such as file_creator are optional and may be used to provide additional information about the segment.

See section 1.1 for other general guidelines

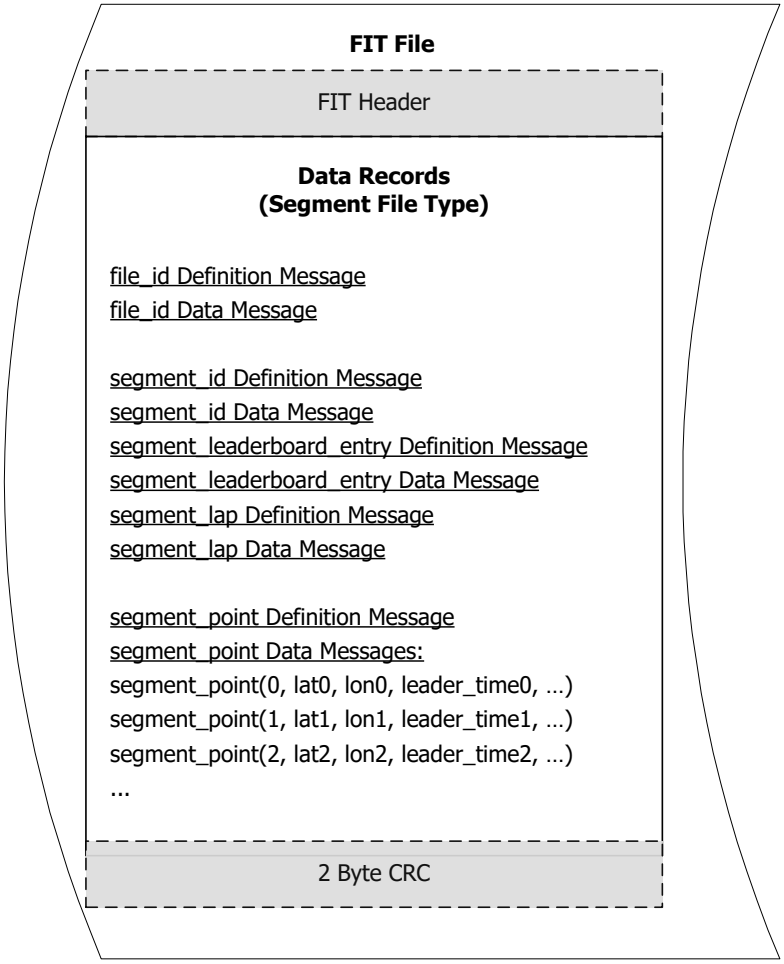


Figure 17-1. Segment File

17.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 34** for a Segment file. The segment file must contain the FIT file_id and messages as described below.

Table 17-1. FIT Messages Contained in a Segment File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Segment File = 34
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time	Time file was created. Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	number	N	UINT16	File identifier
segment_id	name	Y	STRING	Friendly name assigned to segment
	uuid	Y	STRING	UUID of the segment
	sport	Y	sport (enum)	Sport associated with the segment
	enabled	Y	BOOL	Enabled state of the segment file
	user_profile_primary_key	Y	UINT32	Unique identifier for the user that created the segment file
	default_race_leader	Y	UINT8	Message Index for the Leader Board entry selected as the default race participant
	delete_status	N	segment_delete_status	Status used to determine if any segments need to be deleted
segment_leaderboard_entry	message_index	Y	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	name	N	STRING	Friendly name assigned to leader
	type	Y	segment_leaderboard_type (enum)	Leader classification (segment leader, personal best, etc.)
	group_primary_key	N	UINT32	Unique ID of a group. Needed when leader type is 'group'

	activity_id	Y	UINT32	ID of the activity associated with this leader time
	segment_time	Y	UINT32	Total time to complete the segment for this leader time
segment_lap	uuid	Y	STRING	UUID of the segment
	total_ascent	Y	UINT16	Total ascent over the segment
	total_descent	Y	UINT16	Total descent over the segment
	swc_lat	Y	SINT32	South west corner latitude
	swc_long	Y	SINT32	South west corner longitude
	nec_lat	Y	SINT32	North east corner latitude
	nec _long	Y	SINT32	North east corner longitude
	message_index	Y	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	start_position_lat	Y	SINT32	Segment starting position
	start_position_long	Y	SINT32	Segment starting position
	end_position_lat	Y	SINT32	Segment end position
	end_position_long	Y	SINT32	Segment end position
	total_distance	Y	UINT32	Length of entire segment
	sport	Y	sport (enum)	Sport associated with the segment
segment_point	message_index	Y	message_index (UINT16)	Sequence number of segment_point message.
	position_lat	Y	SINT32	Latitude of segment point
	position_long	Y	SINT32	Longitude of segment point
	distance	Y	SINT32	Accumulated distance along the segment at the described point
	altitude	Y	UINT16	Altitude of segment point
	leader_time	Y	UINT32[N]	Time for the various leaders to reach the segment point. Indexed by segment_leaderboard message_index

17.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

17.1.2 "segment_id" Message

The segment_id message uniquely identifies a segment and gives basic information about it. It may be related to a segment_lap message by the uuid.

17.1.2.1 "default_race_leader" Field

Set to the message_index of the segment_leaderboard_entry message to indicate the desired leader the user is competing with.

17.1.2.2 "delete_status" Field

Can be used to flag segments for removal from existing segment and segment_list files.

17.1.3 "segment_leaderboard_entry" Message

These messages define which leader types timing information is available for. There must be at least as many segment_leaderboard_entry messages as there are elements in the segment_point.leader_time field.

17.1.4 "segment_lap" Message

The segment_lap message contains details about the segment such total ascent and descent, segment length and the location of the segment start and end points. Optionally a segment lap message may be included in an activity file to summarize data collected during a segment.

17.1.5 "segment_point" Message

A series of connected segment_point messages define the segment route. Each segment point includes the elapsed time for the leader to reach this position. Timing for leaders in different categories can be included by using a different element of leader_time for each category. Leader_time[N] may be indexed using the corresponding segment_leaderboard_entry message.

segment_leaderboard_entry mesgs:

index 0	"Johnny"	challenger
index 1	"anon12"	group
index 2	"Lance"	overall

segment_point mesgs:

index 0	lat_0	lon_0	dist_0	[0.0][0.0][0.0]
...				
index k	lat_k	lon_k	dist_k	[54.1][47.4][43.0]

Figure 17-2. Segment Point Example

In the example in Figure 17-2, the time for the overall segment leader 'Lance' to reach segment_point k is 43.0s.

18 Segment List File

Segment List files maintain a list of available segments on the device. Segment List files contain segment_file messages. Other messages such as file_creator are optional and may be used to provide additional information.

See section **1.1** for other general guidelines

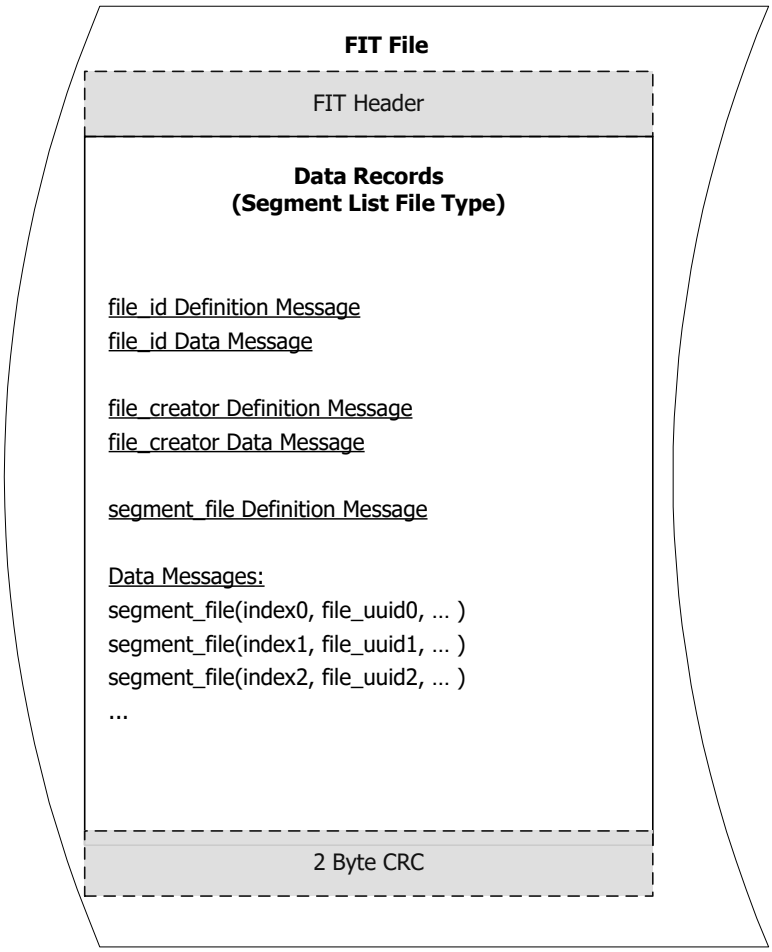


Figure 18-1. Segment List File

18.1 FIT Messages

The FIT **file_id.type = 35** for a segment list file. The segment list file must contain the FIT file_id and segment_file messages as described below.

Table 18-1. FIT Messages Contained in Segment List File

FIT Message	FIT Fields	Required	Type	Description
file_id	type	Y	file (enum)	Segment List File = 35
	manufacturer	Y	manufacturer (UINT16)	ANT+ managed. Contact antalliance@thisisant.com for details
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
	time_created	Y	date_time	Time file was created. Seconds since UTC 00:00 Dec 31 1989 If <0x10000000 = system time
	number	N	UINT16	File identifier
segment_file	message_index	Y	message_index (UINT16)	Provides an index such that other FIT messages can be related to this message
	file_uuid	Y	String	UUID of the segment file
	enabled	Y	BOOL	Enabled state of the segment file
	user_profile_primary_key	Y	UINT32	Primary key of the user that created the segment file
	leader_type	Y	segment_leaderboard_type (enum)	Leader type of each leader available in the segment file
	leader_group_primary_key	N	UINT32	Group primary key of each leader in the segment file
	leader_activity_id	Y	UINT32	Activity ID of each leader in the segment file
file_creator	software_version	Y*	UINT16	
	hardware_version	N	UINT8	

* Field is only required if the optional FIT message is recorded

18.1.1 "file_id" Message

The file_id message identifies the format/content of the FIT file (type field) and the combination of fields provides a globally unique identifier for the file. Each FIT file should contain one and only one file_id message. If the combination of type, manufacturer, product and serial_number is insufficient, for example on a device supporting multiple device files, the time_created or number fields must be populated to differentiate the files.

If the file is created offline (for example using a PC application) the file_id fields could be set as per the creating application or to the values of the destination device, if known. If the file is in an intermediate state, only type need be set, so long as the other fields are later updated by the target device.

18.1.2 "segment_file" Message

The segment_file message holds all of the relevant information necessary to determine if updates are required for any of the segments on the target device. The user_profile_primary_key is used to identify the user sending the list. The leader_activity_id holds the activity IDs used to create the data in that segment file. These IDs are used to compare to the current ID for each leader type in the file and if any of them are different from what is in the list, an update is required.

The file uuid is used to identify each individual segment on the target device. The leader_type is used to identify which type of leader each entry in the leader_activity_id field is associated with.