FIT FILE TYPES

Description

ANT+ Managed Network Document
D00001309 Rev 1.01

P +1 403.932.9292 F +1 403.932.6521

228 River Avenue, Cochrane, Alberta, Canada T4C 2C1 thisisant.com

Copyright Information and Usage Notice

This information disclosed herein is the exclusive property of Dynastream Innovations Inc. The recipient of this document must be a Licensee under the terms of the ANT+ Managed Network Product Development License Agreement and/or the ANT+ Managed Network Commercial Product License Agreement and must use the information in this document according to the terms and conditions of these agreements which include, but are not limited to the following:

a) Licensee agrees that it will not distribute, transfer or otherwise provide ANT+ Managed Network Key(s) or information from ANT+ Managed Network Documents to any person or entity other than employees of Licensee who need to have access to the ANT+ Managed Network Key(s) or information from ANT+ Managed Network Documents.

b) Licensee agrees that it will comply with all "ANT+ Managed Network Documents", will not deviate from Device Type configurations in "ANT+ Managed Network Documents" and will only use Device Types that are defined in "ANT+ Managed Network Documents".

c) DYNASTREAM MAKES NO CONDITIONS, WARRANTIES OR REPRESENTATIONS ABOUT THE SUITABILITY, RELIABILITY, USABILITY, SECURITY, QUALITY, CAPACITY, PERFORMANCE, AVAILABILITY, TIMELINESS OR ACCURACY OF THE ANT NETWORK KEY, ANT NETWORK, ANT DEVICE TYPE DEFINITIONS AND MESSAGE FORMATS OR ANY OTHER PRODUCTS OR SERVICES SUPPLIED UNDER THIS AGREEMENT OR THE NETWORKS OF THIRD PARTIES. DYNASTREAM EXPRESSLY DISCLAIMS ALL CONDITIONS, WARRANTIES AND REPRESENTATIONS, EXPRESS, IMPLIED OR STATUTORY INCLUDING, BUT NOT LIMITED TO, IMPLIED CONDITIONS OR WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, DURABILITY, TITLE AND NON-INFRINGEMENT, WHETHER ARISING BY USAGE OF TRADE, COURSE OF DEALING, COURSE OF PERFORMANCE OR OTHERWISE.

d) Licensee agrees to indemnify and hold harmless Dynastream for claims, whether arising in tort or contract, against Dynastream, including legal fees, expenses, settlement amounts, and costs, arising out of the application, use or sale of Licensee's designs and/or products that are related to Licensee's use of ANT, ANT+ Managed Network Keys or ANT+ Managed Network Documents.

If you are not a Licensee under the ANT+ Managed Network Product Development License Agreement and / or the ANT+ Managed Network Commercial Product License Agreement, you must destroy this document immediately. You must further confirm that you have destroyed this document and have not distributed information from this document. Please consult your legal department if you are unsure of your status as a Licensee.

The information contained in this document is subject to change without notice and should not be construed as a commitment by Dynastream Innovations Inc. unless such commitment is expressly given in a covering document.

Products (including information provided under the terms of ANT+ Managed Network Product Development License Agreement and/or the ANT+ Managed Network Commercial Product License Agreement) sold by DYNASTREAM are not designed for use in life support and/or safety equipment where malfunction of the Product can reasonably be expected to result in personal injury or death. You use or sell such products for use in life support and/or safety applications at your own risk and agree to defend, indemnify and hold harmless DYNASTREAM from any and all damages, claims, suits or expense resulting from such use.

Revision History

Revision	Effective Date	Description
1.0	May 2010	Initial release
1.01	May 2010	Added: Course files; more description to settings files; File Type Numbers

Table of Contents

List of Figures

List of Tables

1 Overview of the FIT File Protocol

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. Common FIT File Types

FIT File Type	Purpose
Settings	Describes a user's parameters such as Age & Weight as well as device settings
Blood Pressure	Records blood pressure data
Weight	Records weight scale data
Activity	Records data and events from active sessions
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

2 Related Documents

The following supplementary documentation and files are 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

3 Settings File

The settings file contains data records that provide user and device information. The most common application for a settings file is the user profile, but it can also contain information about sensors that the device is paired with, as well as user interface preferences (Figure 3-1).

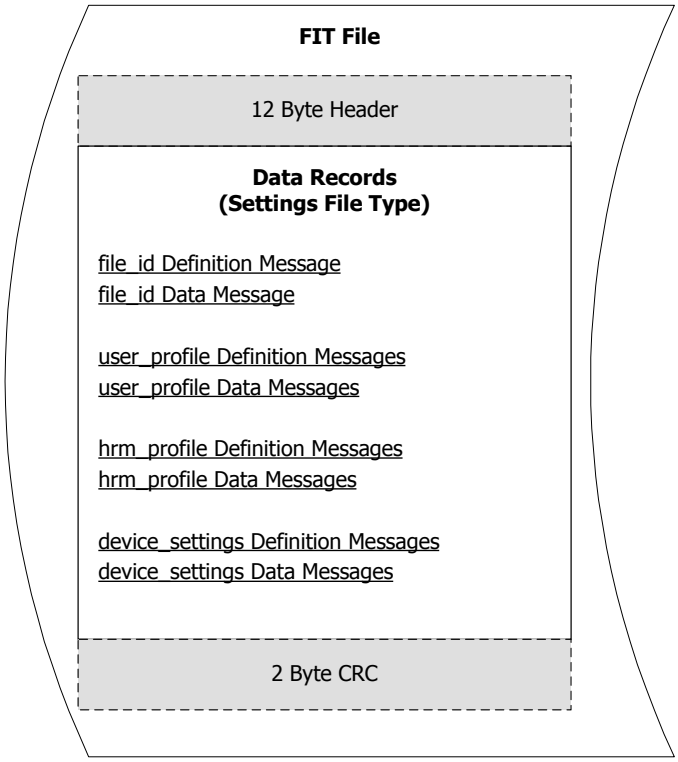

Figure 3-1. Settings File

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

3.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 2** for a settings file. A FIT settings file also includes the following FIT messages:

device_settings

The device settings message currently contains only the UTC offset, allowing for appropriate time coordination between devices.

user_profile

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.

hrm_profile

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 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 data 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.

4.1 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 (Figure 4-1). Other FIT messages, such as user_profile and device_info, may be included if desired.

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 4-1).

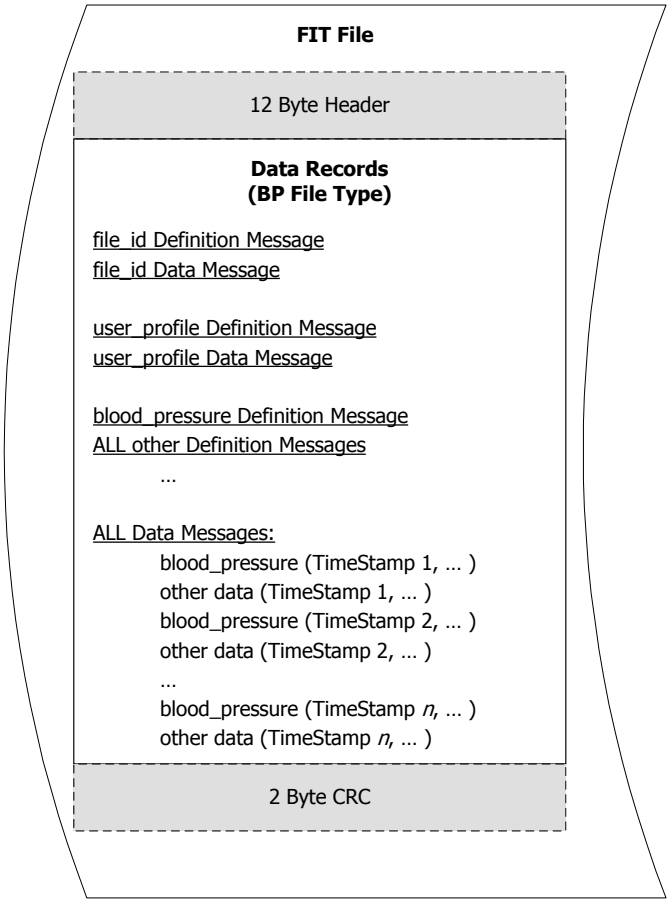

Figure 4-1. Blood Pressure File

The BP file must contain the FIT file_id, and blood_pressure messages as described in Table 4-1. It may also, optionally contain the user_profile and device_info message.

Table 4-1. FIT Messages Contained in BP File

FIT Message	FIT Fields	Required	Type	Value/Units
file_id (files from device)	type	Y	file (enum)	BP file
	manufacturer	Y	Manufacturer (UINT16)	ANT+ managed. Please contact
	product	Y	UINT16	Managed by manufacturer
	serial_number	Y	UINT32z	Managed by manufacturer
file_id (files to device)	type	Y	file (enum)	BP File
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	
	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	mmHg (UINT16)	
	diastolic_pressure	Y	mmHg (UINT16)	
	mean_arterial_pressure	N	mmHg (UINT16)	
	heart_rate	Y	bpm (UINT8)	
	map_3_sample_mean	N	mmHg (UINT16)	
	map_morning_values	N	mmHg (UINT16)	
	map_evening_values	N	mmHg (UINT16)	
	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)	
	device_type	N	device_type (UINT8)	18 (0x12) for ANT+ BP monitor
	manufacturer	N	manufacturer (UINT16)	managed by ANT+ msb (i.e. bit 15) must be set to 1
	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

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

As indicated in the "Required" column, not all of the listed fields shall be included in the BP file. At a minimum, the following is required:

- file_id message must be included to indicate the file type
- blood_pressure message containing systolic pressure, diastolic pressure and pulse (i.e. heart_rate)
- 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. If this message is not recorded, it is implied that user ID's are not supported on any level
- *If the optional device_info message is included, then it must contain the timestamp field in order to link each device_info message to its respective blood_pressure message

4.2 BP File Examples

Figure 4-2 shows an example FIT BP file. Note that the file contains the FIT 12 Byte 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.

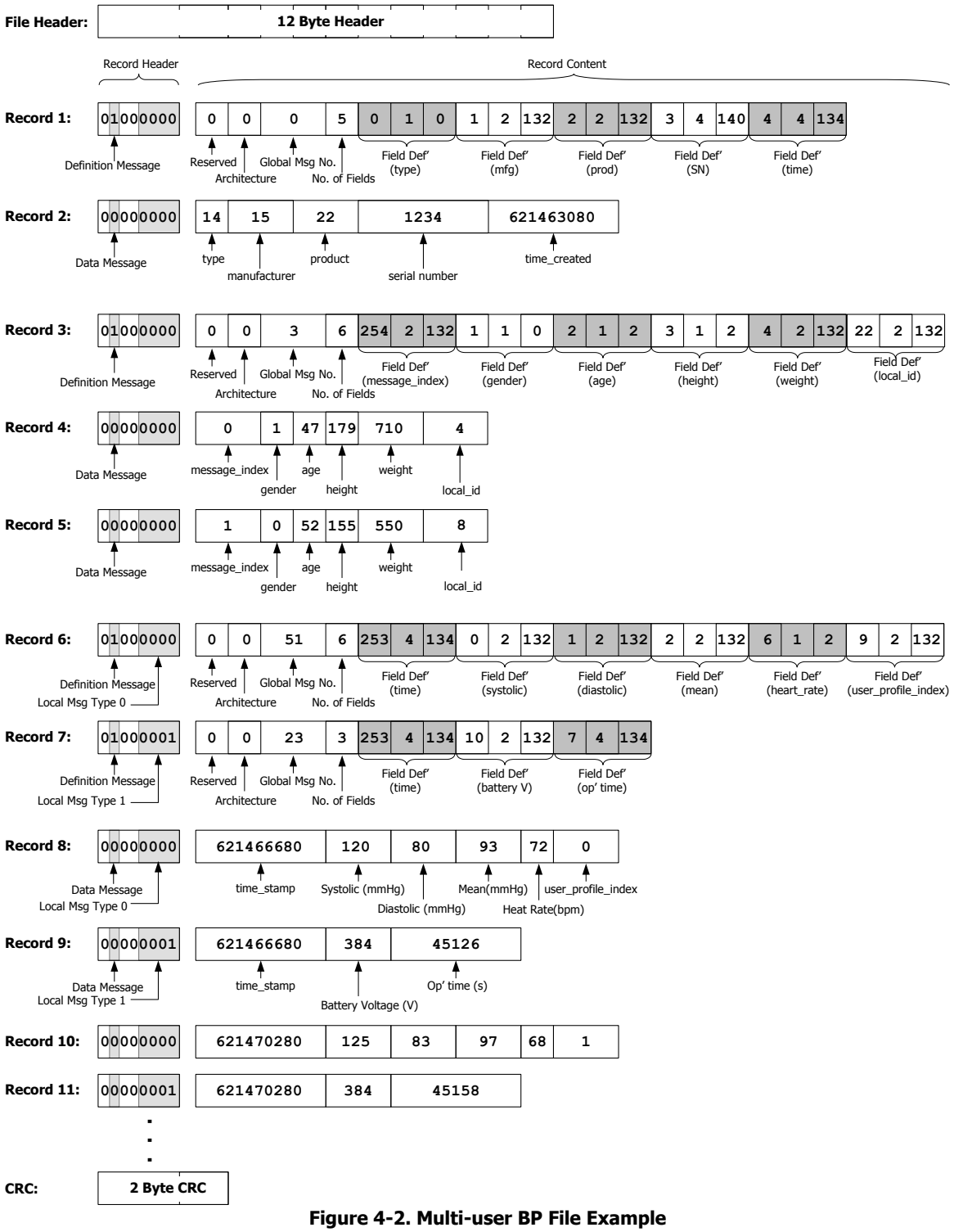

Figure 4-2. Multi-user BP File Example

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, such as device_info, shall use a different local message type.

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.**

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.

Note: 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 4-3, 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 4-4).

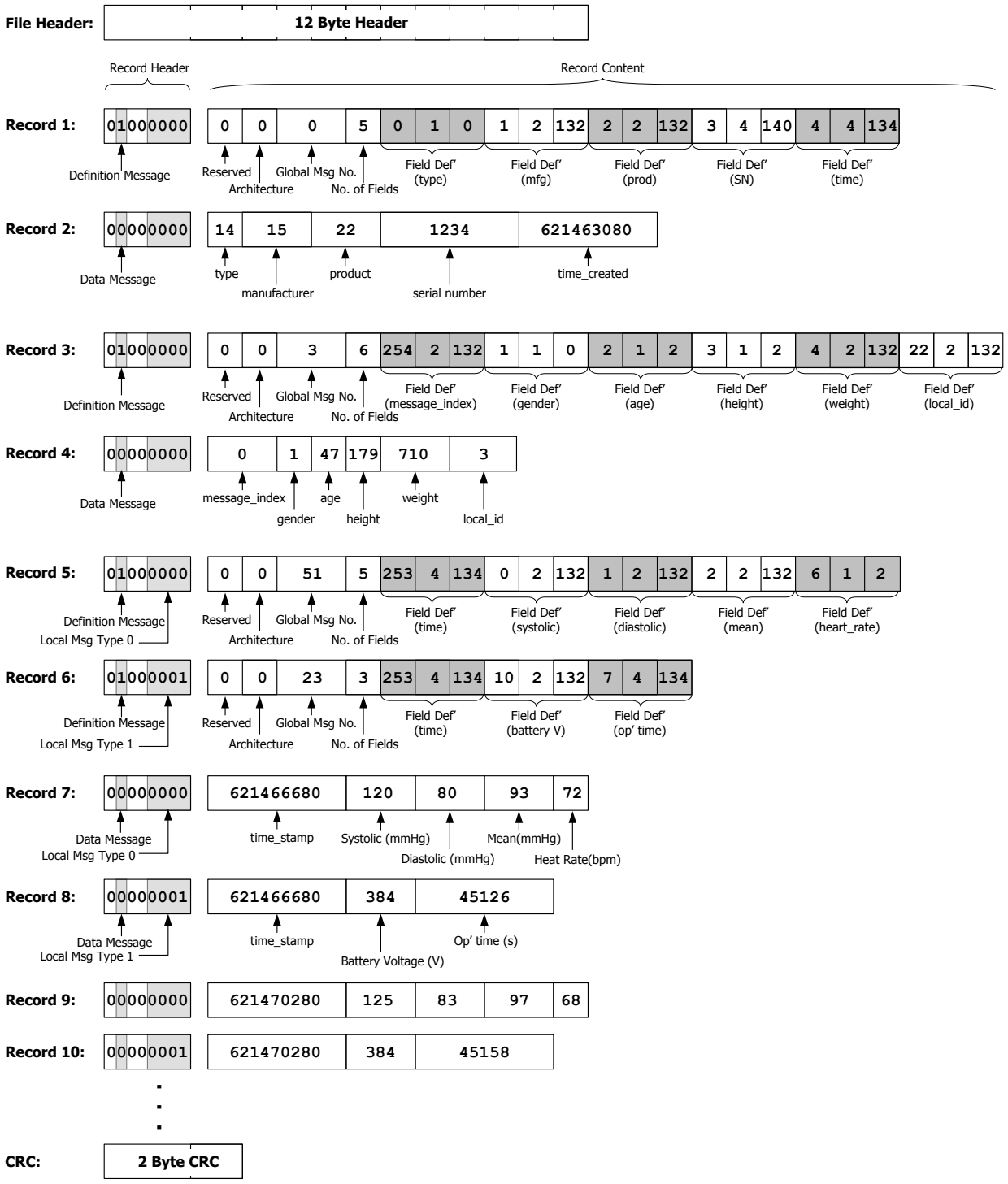

Figure 4-3. Single User BP System

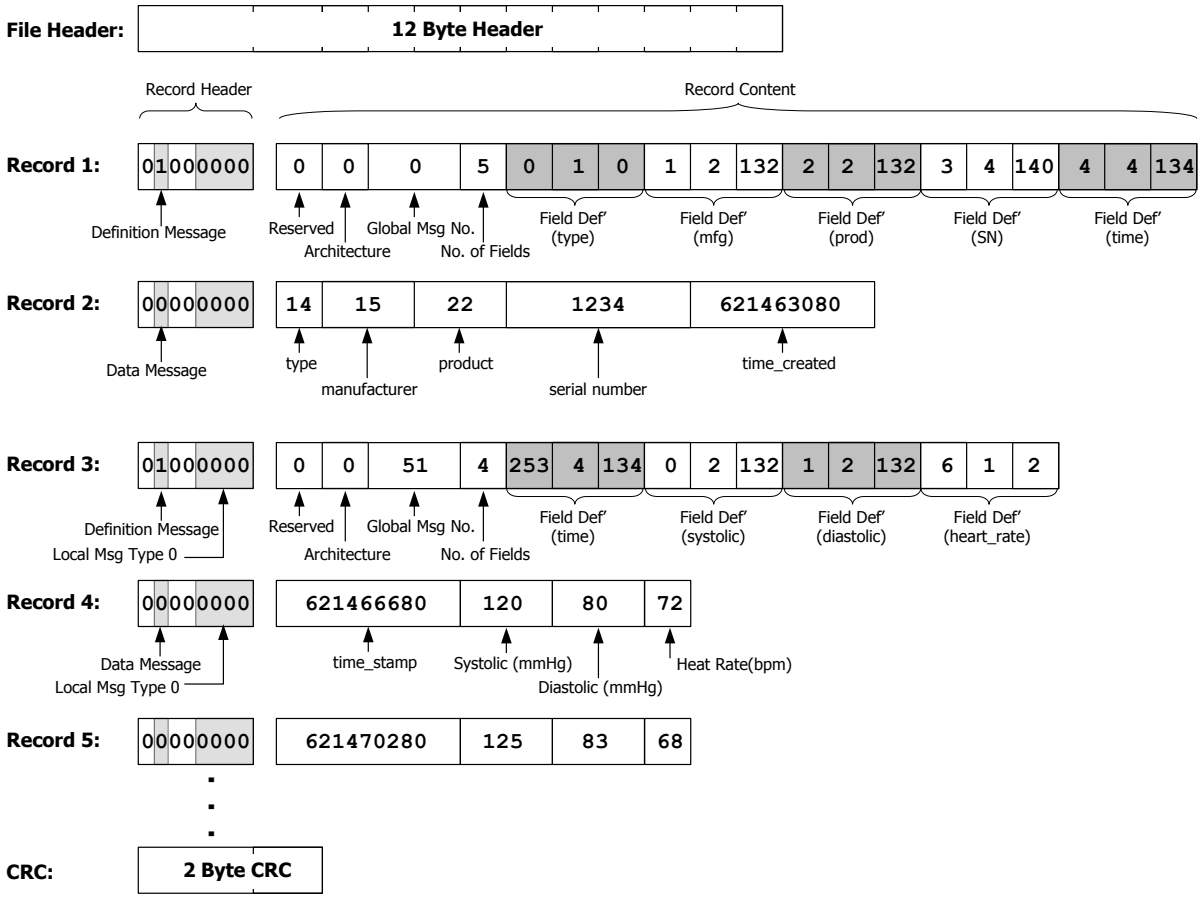

Figure 4-4. BP System without User Profile Support

5 Weight File

A weight file is similar in structure to the BP 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 data 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 5-1).

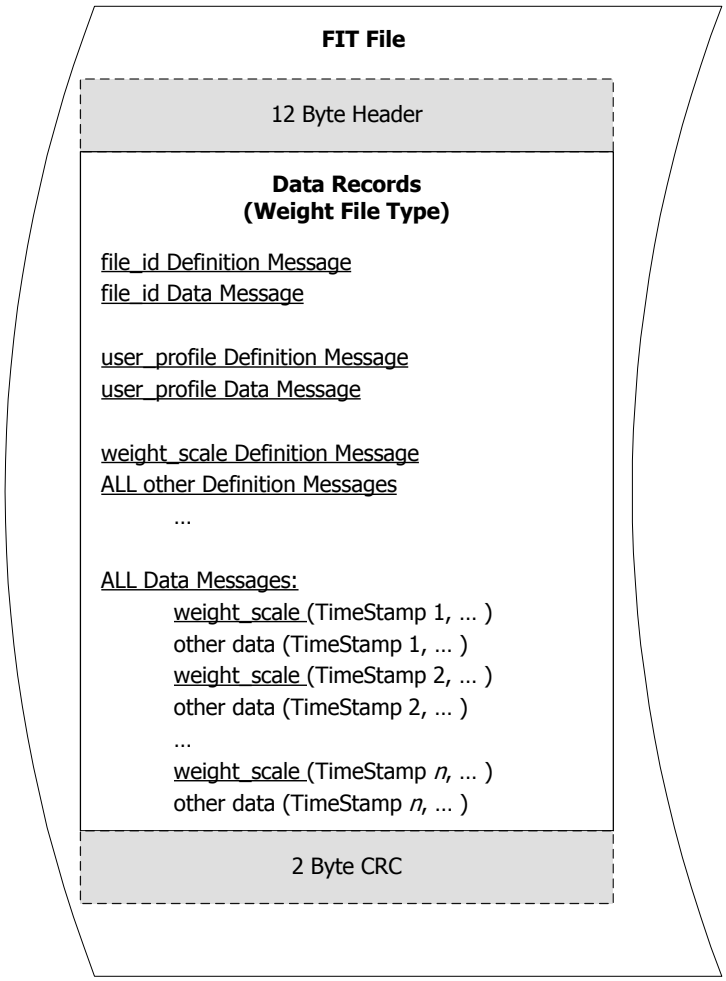

Figure 5-1. Weight File

5.1 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 FIT file_id, user_profile (if user profiles supported) and weight_scale messages as described in Table 5-1. It may also, optionally, contain the device_info message.

Table 5-1. FIT Messages Contained in Weight File

FIT Message	FIT Fields	Required	Type	Value/Units
file_id (files from device)	type	Y	file (enum)	Weight File
	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
file_id (files to device)	type	Y	file (enum)	Weight File
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	UINT16	Weight scale's local user ID
	friendly_name	N	string	
	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
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)	
	device_type	N	device_type (UINT8)	18 (0x12) for ANT+ Weight scale
	manufacturer	N	manufacturer (UINT16)	managed by ANT+ msb (i.e. bit 15) must be set to 1
	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

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

As indicated in the "Required" column, not all of the listed fields shall be included in the weight file. At a minimum, the following is required:

- file_id message must be included to indicate the file type
- weight_scale message containing weight
- 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
- * If optional device_info message is included, then it must contain the timestamp field in order to link each device_info message to its respective blood_pressure message

5.2 Weight File Examples

Figure 5-2 shows an example FIT weight file. Note that the file contains the FIT 12 Byte 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.

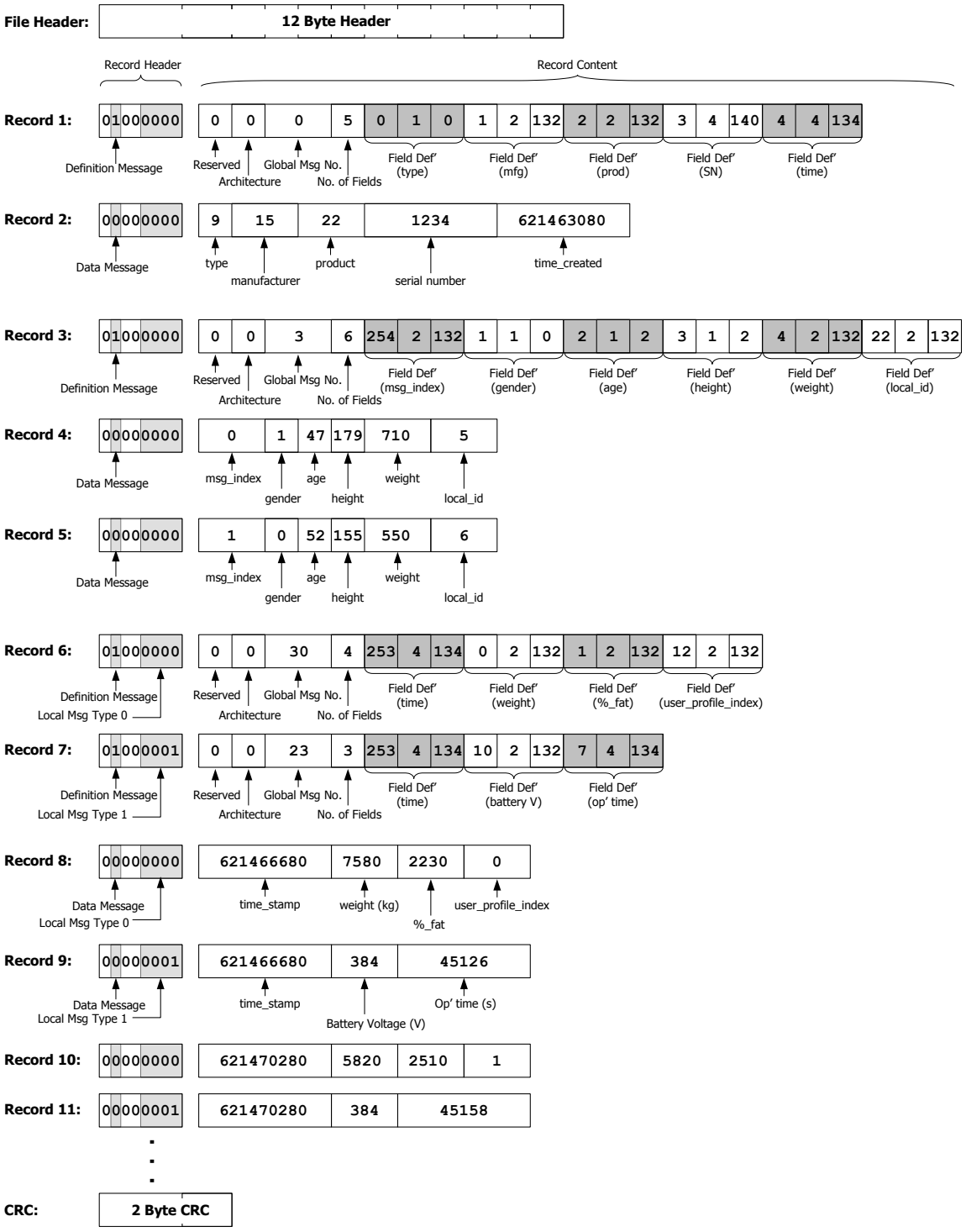

Figure 5-2. Multi-user Weight File Example

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:

- **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 5-2, 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 number of local user ID's will be dependent on the weight scale devices capabilities (i.e. user profile ID). For simple weight scales that do not support user profiles, the user_profile message does not need to be included, indicating that the system that does not support user profiles.

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 5-3, 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

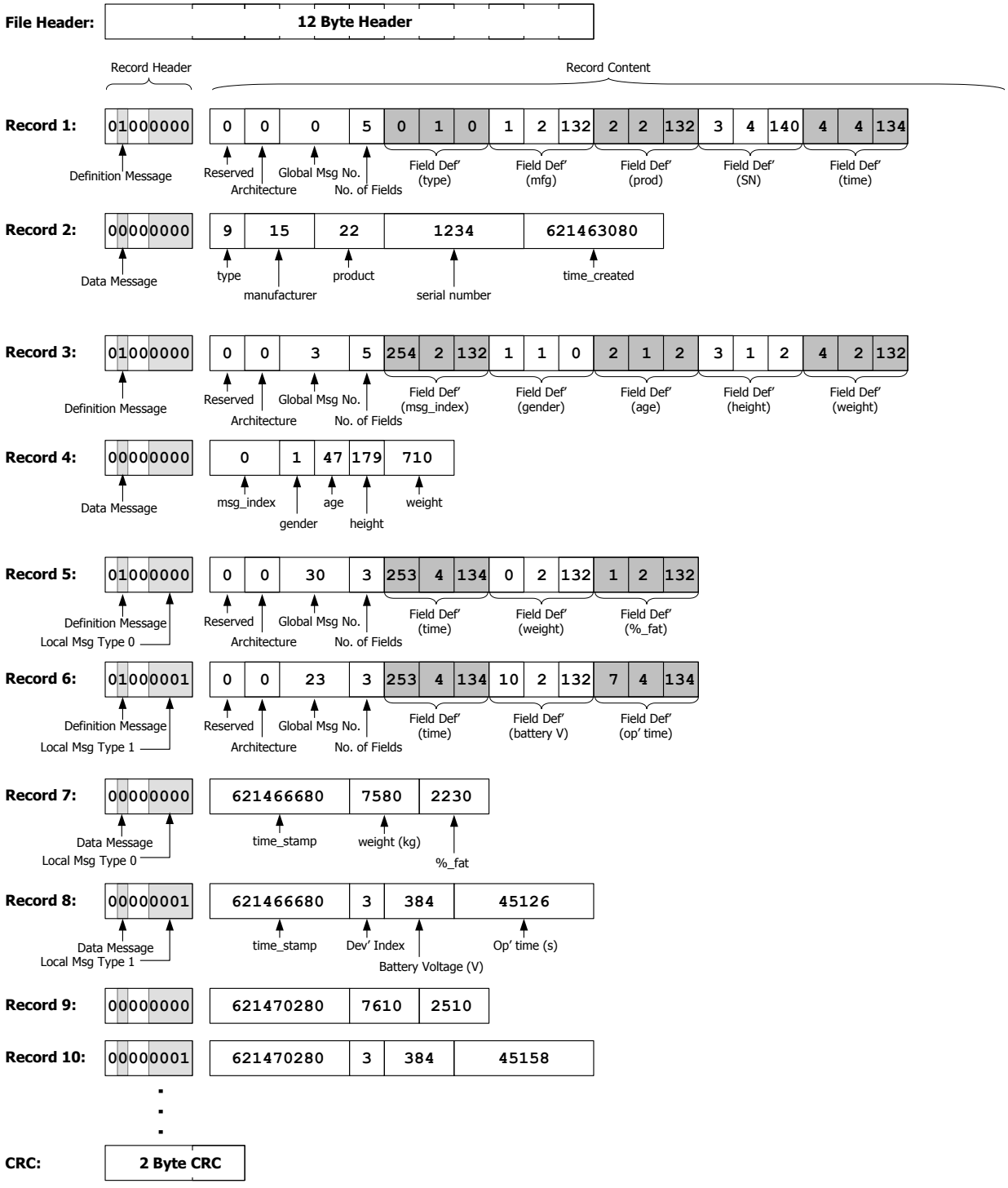

Figure 5-3. Single or Unidentified User Systems

6 Activity File

Activity files are used to record sensor data and events from an active session. All data messages in a session file are related by a timestamp (Figure 6-1).

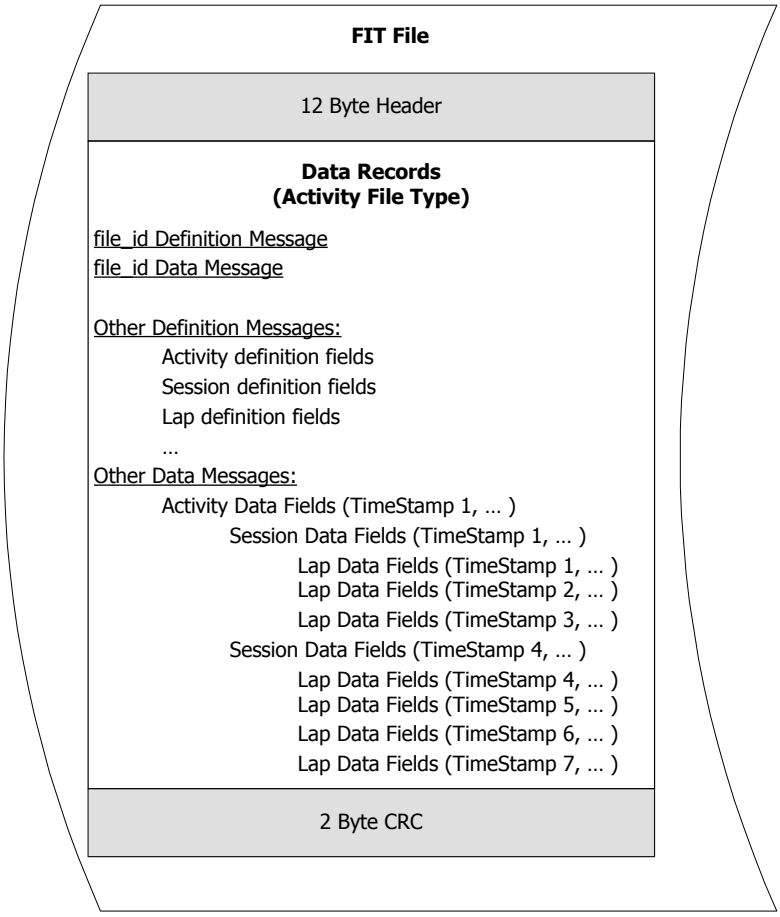

Figure 6-1. Activity, Session and Lap Message Structure

6.1 FIT Messages

All FIT files must start with a file_id message. The FIT **file_id.type = 4** for an activity file. The following FIT messages can also be included in an activity file:

activity, session and lap

Activity, session and lap messages have a similar structure and provide summary data over a discrete time period with increasing detail. As shown Figure 6-1 each activity file will have one activity message, any number of session messages within each activity, and any number of laps within each session. These three types of summary message may be grouped together at the start of the file or interleaved with event and record messages. In either case the messages must be in chronological order.

An Activity message provides a high level description of the overall activity file. This includes overall time, number of sessions and the type of each session.

The Session message adds more detail including totals and averages over the entire session while Lap messages provide this detail over the duration of a single lap.

Depending on the device, there may be a limit to the number of sessions that are allowed per activity file, or number of laps that are allowed per session.

record

Record messages are a time-stamped data message carrying information about the user activity in the current session. This message carries instantaneous data such as speed, position, heart rate and bicycle power. Record messages must be in chronological order.

event

These messages are used to record events within a session including starting and stopping the timer, but also alerts. Event messages must be in chronological order.

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.

7 Workout File

A workout file describes a structured activity that can be designed on a computer and transferred to a display device to guide a user through the activity. All FIT files must start with a file_id message. The FIT **file_id.type = 5** for a workout file.

The workout file should, at a minimum, contain the file_id, workout and at least one workout_step FIT messages (Figure 7-1). Messages should be defined and recorded sequentially, using only local message type 0. 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. Redefining local message type 0 for all messages will ensure simple processors can handle all workout data.

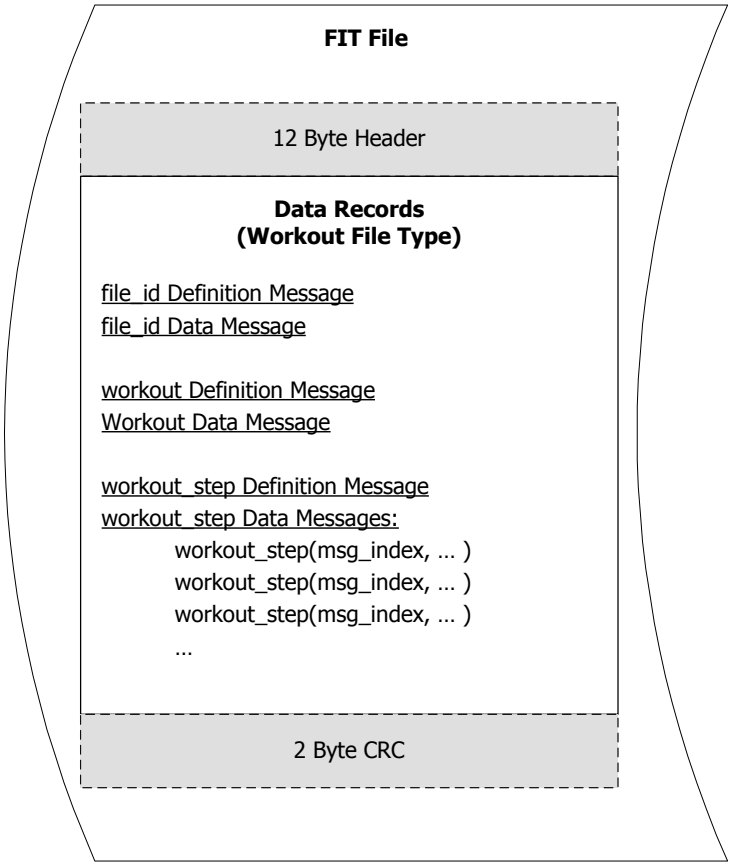

Figure 7-1. Workout File

Workouts are described as a series of steps. Each step is used to define a target effort for a set duration (Figure 7-2, step A and B), or to define a repetition pattern (Figure 7-2, step C).

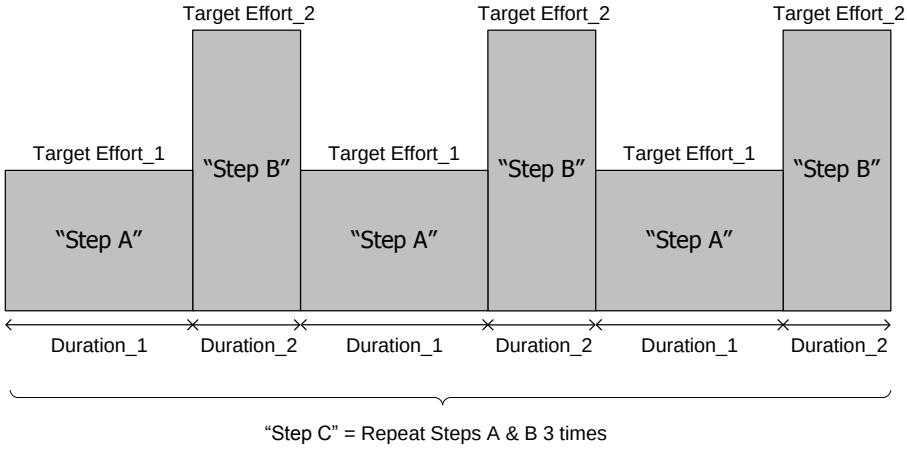

Figure 7-2. Defining Workout Steps

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

7.1 FIT Messages

The general message structure for both the workout and workout_step messages are show below in Figure 7-3.

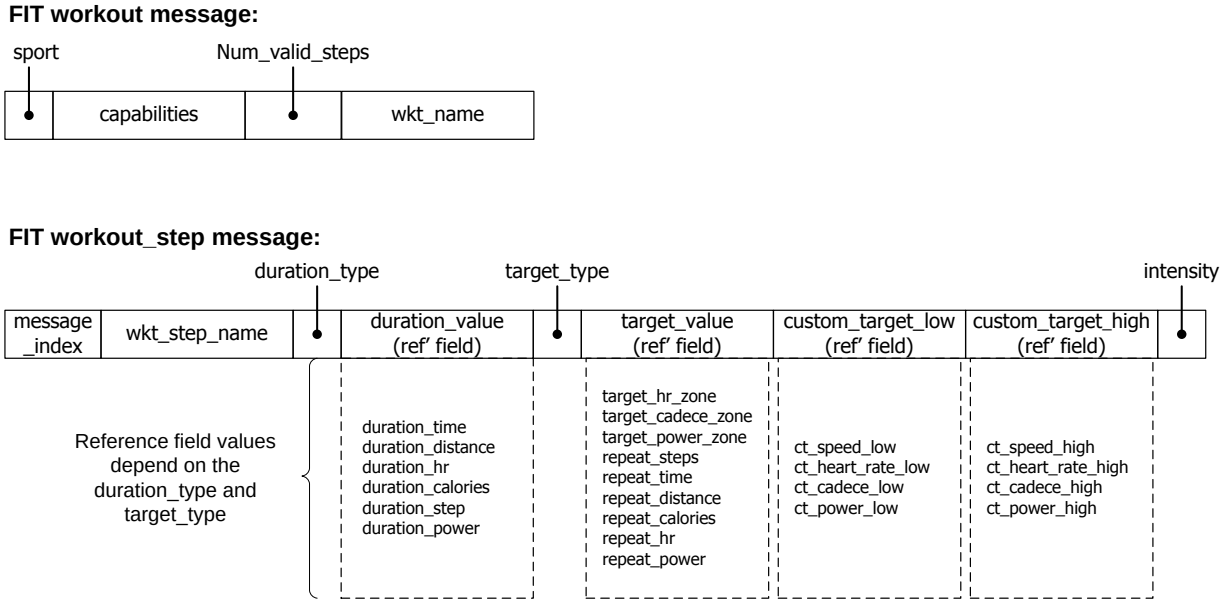

Figure 7-3. FIT workout and workout_step Message Structure

The full list of FIT messages and fields contained in a workout file are outlined in Table 7-1. Note that not all fields are required.

Table 7-1. FIT Messages Contained in Workout File

FIT Message	FIT Fields	Required	Type	Value/Units
file_id (files from device)	type	Y	file (enum)	Workout file
	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
file_id (files to device)	type	Y	file (enum)	Workout file
	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
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 contained in the file
	wkt_name	N	String	User friendly string identifying name of workout
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 steps' 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 7-2)
	target_type	Y	wkt_step_target (enum)	Indicates the type of parameter that will define the workout steps' target range/zone.
	target_value	N	UINT32, workout_hr, or workout_power	Dynamic field representing the value of the target. The value in this field depends on either duration_type or target type as outline in Table 7-3.
	custom_target_value_low	N	UINT32, workout_hr,	If the workout target uses a custom range, rather than a

			or workout_power	defined zone, this field is used to specify the lower boundary. Dynamic field dependent on target_type (Table 7-3)
	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 7-3)
	intensity	N	intensity (enum)	Represents the workout steps intensity level (Table 7-4)

7.2 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 example of Figure 7-2, the number of defined steps is 3 (i.e. steps A, B and C).

7.3 Workout_steps Message

The workout_steps 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 7-2)
- target value: number of repeats, time limit of repeats, etc

The workout_steps message contains dynamic fields which are described in sections 7.3.1 and 7.3.2.

7.3.1 Duration_type Referenced 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 7-2.

Table 7-2. 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	

7.3.2 Target_type Referenced 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 7-3.

Table 7-3. List of target_types and Relevant Dynamic Field Values

target_type	target_value (dynamic field value)	custom_target_low (dynamic field value)	custom_target_high (dynamic field value)
speed		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
grade			
resistance			

7.3.3 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. The workout_step can then be used to set a target heart rate, power or other

zone value. 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. Refer to the FIT SDK for specific field/zone values.

7.3.4 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 7-4.Workout Intensity Values

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

7.3.5 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 as 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 integer 0 to 100 (heart rate) and 0 to 1000 (power) range is reserved for relative values, specific heart rate and power values must be incremented by 100 bpm or 1000 watts respectively. Examples are provided in

Table 7-5. 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

7.4 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.

7.4.1 Defining Individual Workout Steps

Figure 7-4 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").

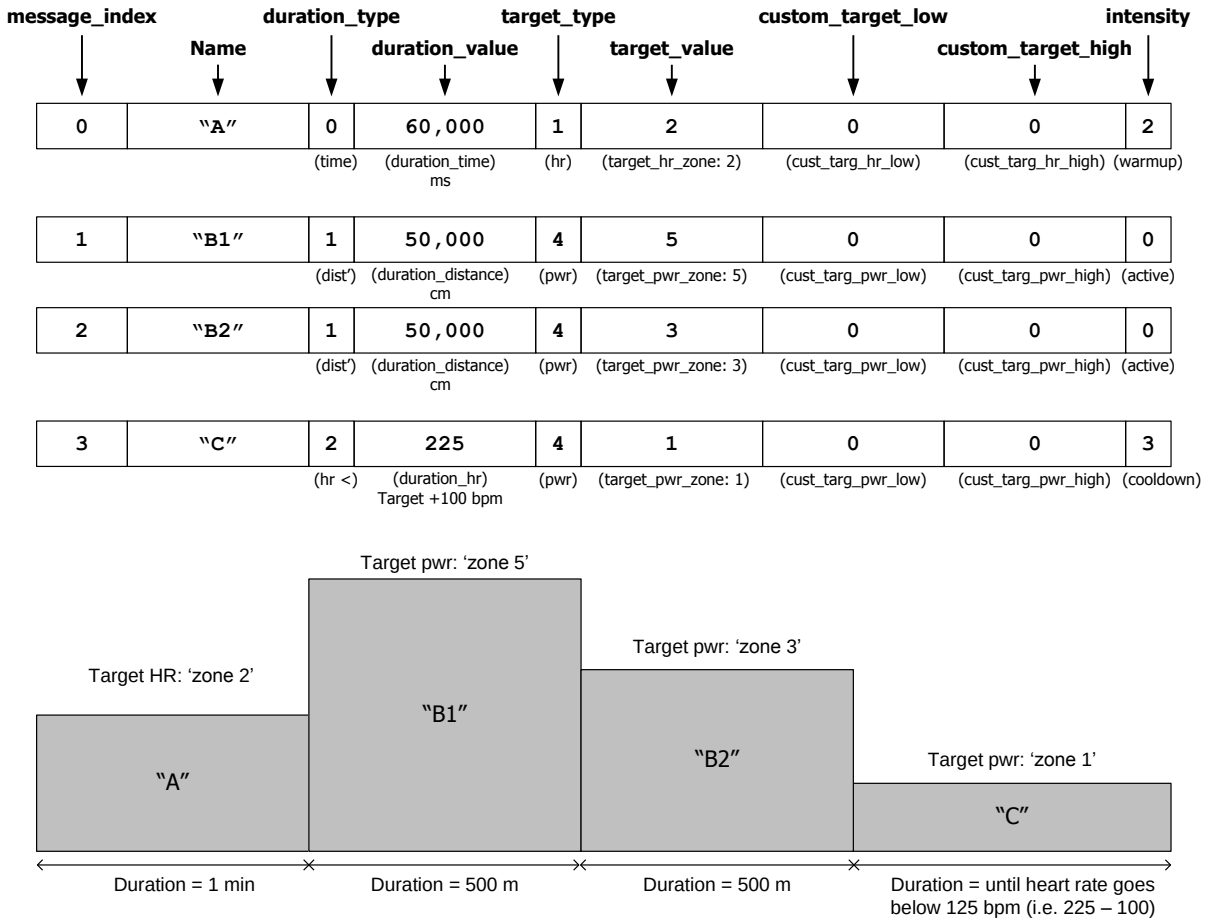

Figure 7-4. Example Workout_Steps

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 7.3.5 for details on setting heart rate or power values.

7.4.2 Defining Repeat Steps Example

Figure 7-5 uses the same steps from the example in Figure 7-4, 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**.

0	"A"	0	60,000	1	2	0	0	2
1	"B1"	1	50,000	4	5	0	0	0
2	"B2"	1	50,000	4	3	0	0	0
3	"Rep"	6	1	2	3	0	0	0
		(rep' until steps cmplt)	(duration_step) Repeat from msg_index 1	(open)	(repeat_steps) 3 times	(custom_target_low)	(custom_target_high)	(active)
4	"C"	2	225	4	1	0	0	3

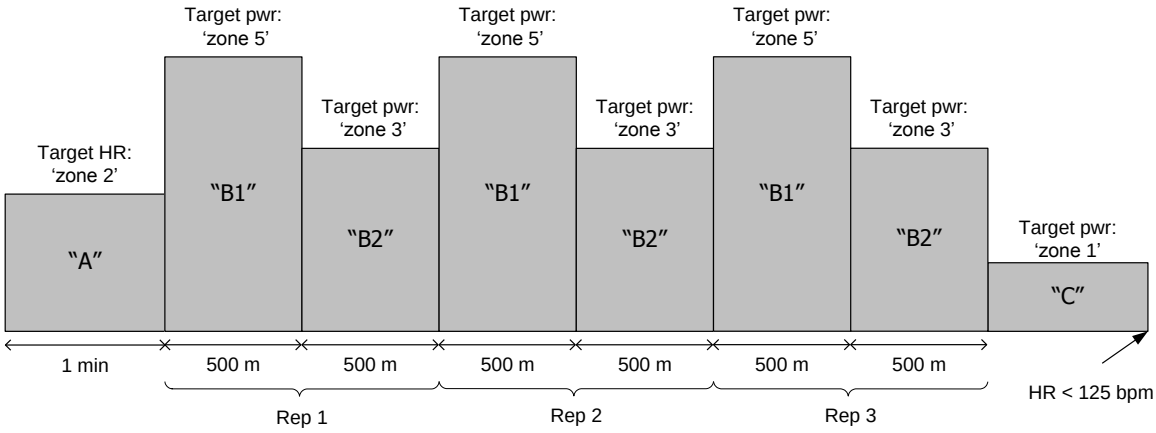

Figure 7-5. Workout_Steps for Repeating Steps

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").

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 met, drop out of the current step and immediately drop into the next step. This scenario is illustrated in Figure 7-6.

0	"A"	0	60,000	1	2	0	0	2
1	"B1"	1	50,000	4	5	0	0	0
2	"B2"	1	50,000	4	3	0	0	0
3	"Rep"	11	1	1	80	0	0	0
		(rep' until hr >)	(duration_step) Repeat from msg_index 1	(hr)	(repeat_hr) Hr > 80% max hr	(custom_target_low)	(custom_target_high)	(active)
4	"C"	2	225	4	1	0	0	3

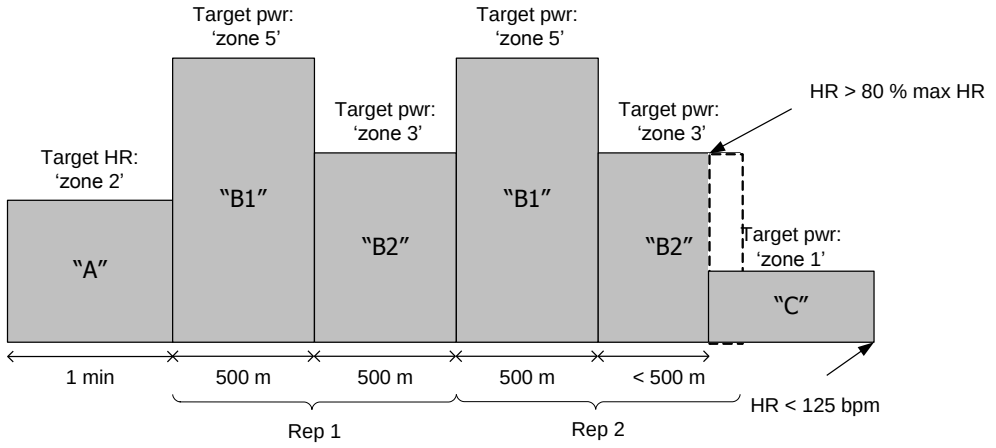

Figure 7-6. Repeat Steps Using "greater than" or "less than" Duration Types.

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 7.3.5 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").

7.4.3 Using Custom Target Values

If predefine target zones are unavailable or undesired, custom target values may be used instead. Figure 7-7 below uses the same workout steps from the example in Figure 7-4, 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 7-3.

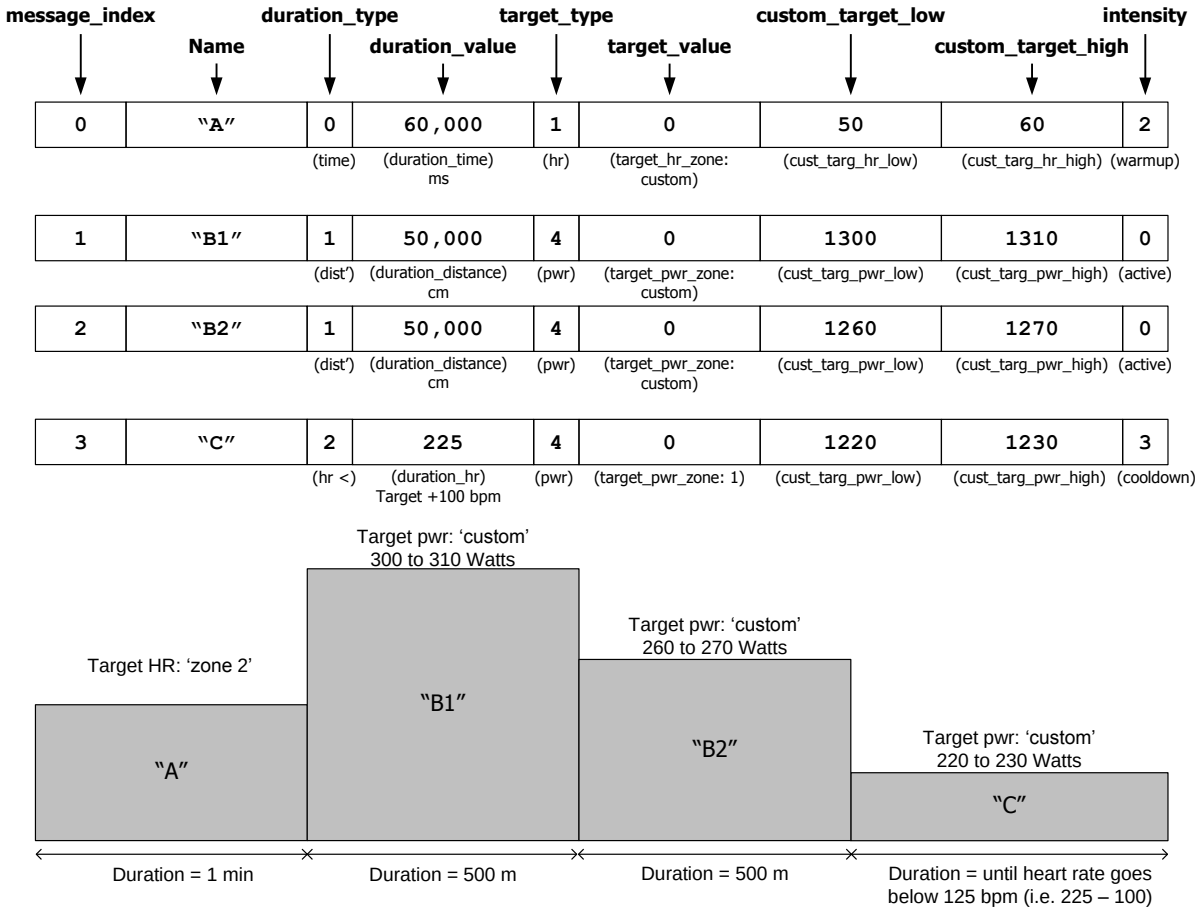

Figure 7-7. Example Workout_Steps Using Custom Target Values

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 7.3.5 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 Course Files

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 8-1).

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 8-1).

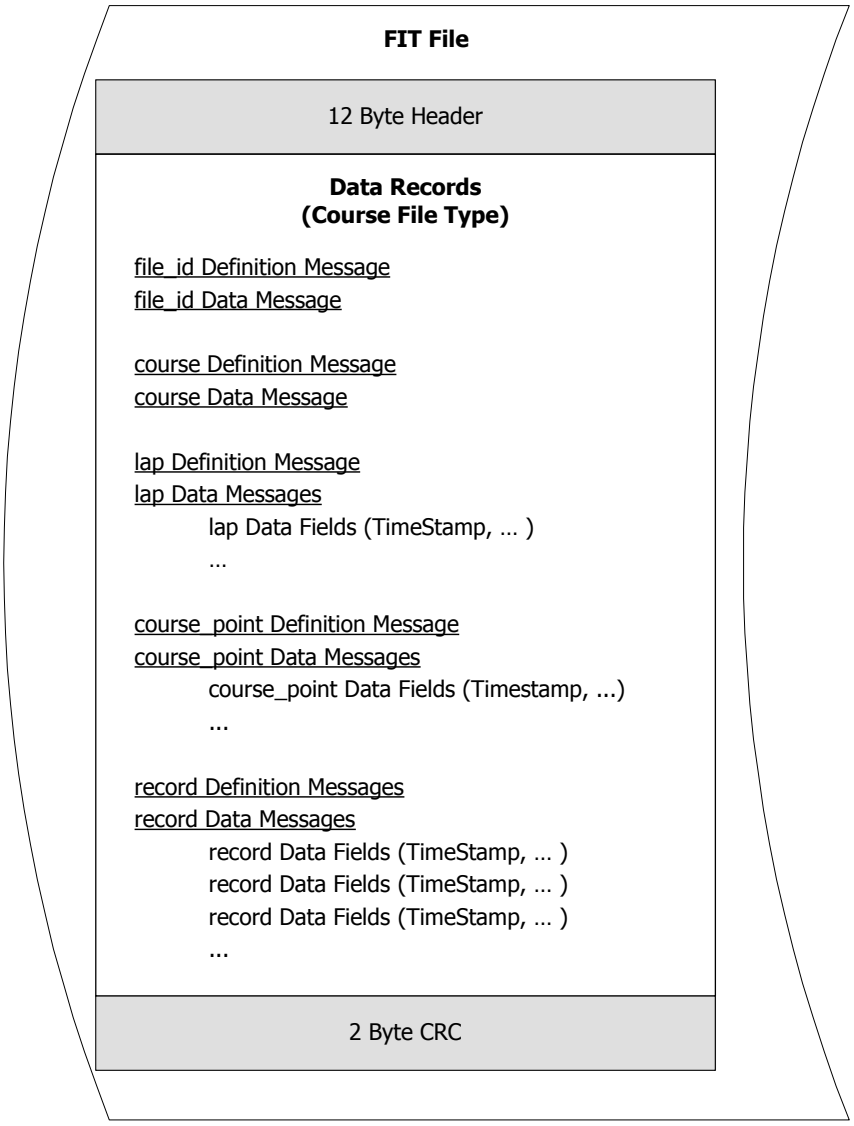

Figure 8-1. Course File

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 8-2). 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.

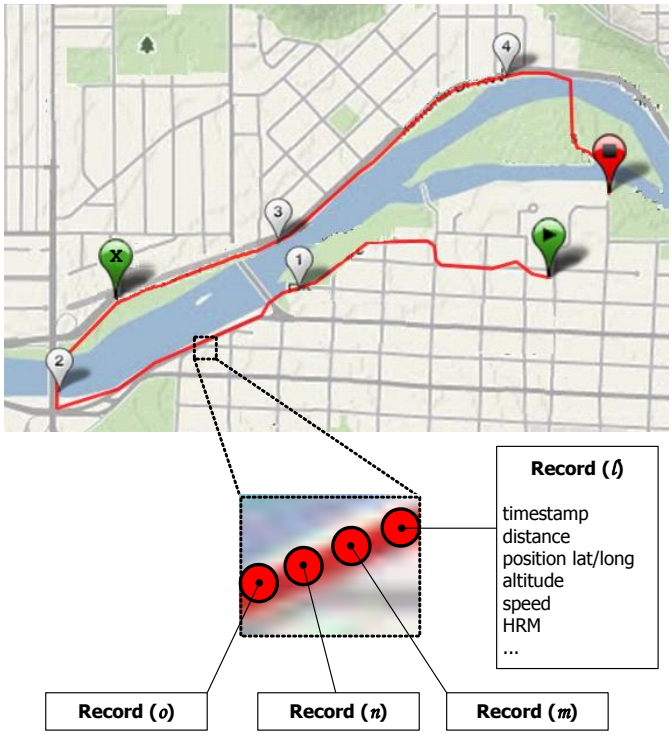

Figure 8-2. Activity Record Messages Used to Create a "River Run" Course

Course files also contain lap and course_point messages to provide summary activity data, and key course milestones and/or landmarks. Figure 8-3 shows the example "River Run" course file with lap and course_point messages.

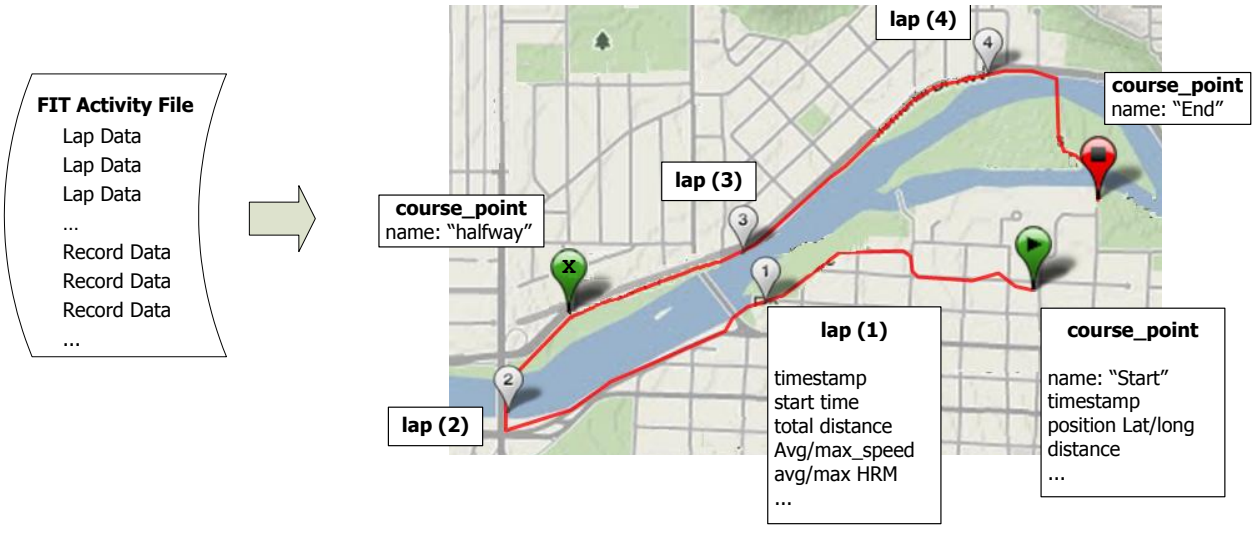

Figure 8-3. "River Run" Course File with Laps and Course_points

8.1 FIT Messages

The list of FIT messages and fields contained in a course file are outlined in Table 8-1. Note that not all fields are required.

Table 8-1. FIT Messages Contained in Course File

FIT Message	FIT Fields	Required	Type	Value/Units
file_id (files from device)	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
file_id (files to device)	type	Y	file (enum)	Course file
	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
course	sport	N	sport (enum)	Type of activity 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		
	position_lat	N		
	position_long	N		
	…			

* Only required if optional message included
** Fields for FIT message only partially listed. Refer to FIT SDK for full listing.

8.2 Course File Example

Figure 8-4 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.

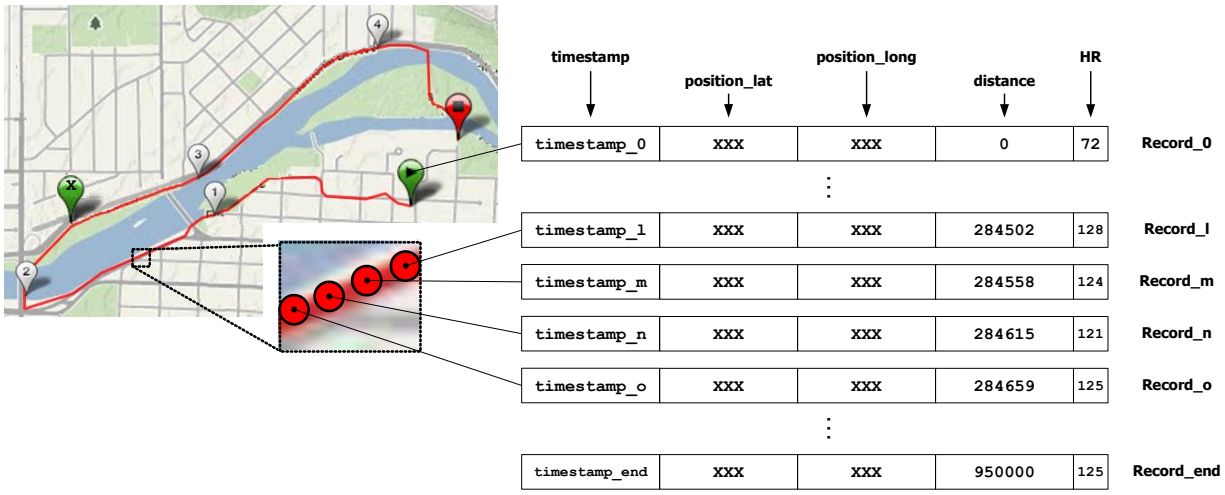

Figure 8-4. Example record Data Messages of a Course File

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 8-5). 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.

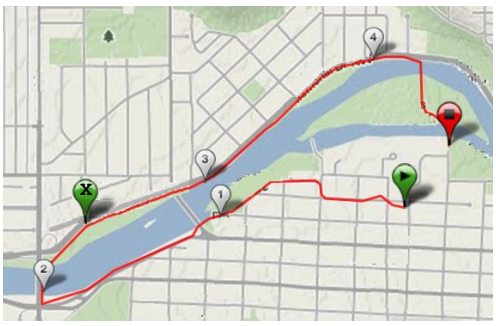

Figure 8-5. Example lap and course_point Data Messages of a Course File

file_id Definition Message (local message type=0)				
F	6	X	1234	timestamp_created

course Definition Message (local message type=0)	
C	"River Run"

lap Definition Message (local message type=0)				
L	timestamp_L1	timestamp_0	121	138
L	timestamp_L2	timestamp_L1	134	143
L	timestamp_L3	timestamp_L2	145	151
L	timestamp_L4	timestamp_L3	146	153
L	timestamp_end	timestamp_L4	133	147

course_point Definition Message (local message type=0)		
CP	timestamp_0	"start"
CP	timestamp_X	"halfway"
CP	timestamp_end	"end"

record Definition Message (local message type=0)					
R	timestamp_0	XXX	XXX	0	72
⋮					
R	timestamp_L1	XXX	XXX	200000	128
⋮					
R	timestamp_l	XXX	XXX	284502	128
R	timestamp_m	XXX	XXX	284558	124
R	timestamp_n	XXX	XXX	284615	121
R	timestamp_o	XXX	XXX	284659	125
⋮					
R	timestamp_L2	XXX	XXX	400000	128
⋮					
R	timestamp_X	XXX	XXX	450000	128
⋮					
R	timestamp_L3	XXX	XXX	600000	128
⋮					
R	timestamp_L4	XXX	XXX	800000	128
⋮					
R	timestamp_end	XXX	XXX	950000	125

HEADER BYTE (F: file_id, C: course, L: lap, CP: course_point, R: record)

Figure 8-6. Example Course File

The resultant course file is formatted as shown in Figure 8-6.

- 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