



AC35 Streaming Onboard Data Interface Specification

5 May, 2017

Version 1.8

Document Identifier:

AC35 Streaming Onboard Data Interface Specification.doc

Technical Committee:

ACTV Tech

Editor(s):

Ken Milnes
Stan Honey
Mark Sheffield
Richard Slater
Tim Heidmann

Table of Contents

1	Introduction.....	3
2	TCP/IP Connection.....	4
3	Message Format.....	5
3.1	Message Structure.....	5
3.2	Message Header	5
3.3	CRC	5
3.4	Message Type Designators	6
4	Messages	7
4.1	Heartbeat	7
4.2	Display Text Message	8
4.3	XML Message.....	9
4.4	Race Start Status.....	10
4.5	Yacht Event Code.....	11
4.6	YachtActionCode	15
4.7	Boat Position.....	16
4.8	Boat Instruments Data	18
4.9	Boat Instruments Data Version 2.....	21
5	XML Messages.....	24
5.1.1	Regatta.xml	24
5.1.2	Race.xml.....	25
5.1.3	Boats.xml.....	28
	Appendix A. Loom Drawing.....	33
	Appendix B. CRC Calculation	34
	Appendix C. Revision History.....	38

1 Introduction

This document describes the data interface used onboard the AC45s and AC Class Yachts.

There are two objectives for the onboard data interface:

1. To allow the teams to build their own displays for the sailors to use onboard the race yacht to show race limits, penalty status, RC-Comms, and to enter protests.
2. To receive from the teams available instrument data to improve the media presentation of the race and to assist the umpires to make accurate and fair decisions. Each race yacht must send wind, boat speed, accumulator pressure and limited rudder angle information if they are available from the boat's instrument system.

The data feed is available during each race and practice race. All of the data is multiplexed into one stream available on a TCP/IP port. The position of the race yachts and race marks are streamed in the "Boat Position" messages. These messages are sent on the stream as soon as they are received from the boats. Race course information such as course limits is sent in "Race.xml" messages. These messages are sent whenever the Principal Race Officer makes a change to the race course. The "Boat.xml" message describes the physical attributes of all of the race yachts and race management motorboats.

Protocol Article 42.3 describes the method for the teams to send telemetry data to the Event Authority.

Telemetry: As required by ACEA for media purposes, Competitors shall provide live, unaltered and un-skewed telemetry data from their competing yachts in all Events, including boat speed, rudder angle, location and heading, and true and apparent wind speed and direction. This data shall be provided at the same update rate as the data displayed on the yachts own instruments, or at 1Hz, whichever is faster

Section 4.9 describes the **Boat Instruments Data** message that the teams shall send to the server if they have instruments installed on their yacht.

2 TCP/IP Connection

Data is transported via TCP/IP. The ACEA media telemetry box is configured as a TCP/IP server. The team device connecting to the ACEA media telemetry box should be configured as a TCP/IP client. Each client should connect the server at IP address 172.16.1.100 port 4940 as a TCP/IP client. Once connected, data will start streaming.

Once the connection is made, a set of messages that contain xml data will immediately be sent. This xml data establishes the current configuration of the race, including the description and boats that are configured in the race. As the course configuration is changed, xml messages that describe the new course will be sent.

The boat position data is sent at different data rates, depending on what type of boat it is. At the time of this writing, race boats are sent position data at 5 Hz and mark boats are sent data at 2 Hz.

Client applications should attempt to connect to the server until a connection is made. If the connection fails, it should attempt to reconnect on a periodic basis. The retry interval should be on the order of 1 second. A heartbeat message will be sent in the data stream every 5 seconds. If the client does not receive a message in this interval, it should close the connection and reconnect.

Note that if the Client application has a malfunction and loses the most recent race.xml data, the client may disconnect and reconnect in order to receive the latest race.xml file.

The onboard team computer sends BoatInstrumentVer2 messages to the ACEA media telemetry box. The media telemetry box sends Heartbeat, DisplayText, XML, RaceStartStatus, YachtEvent, YachtAction, and BoatPosition messages to the onboard team computer.

3 Message Format

All messages begin with a message header. The message header allows the message client to frame and parse the message. All data uses the little-endian byte ordering (increasing byte significance with increasing byte order) and is binary encoded.

3.1 Message Structure

Name	Number of bytes	Description
Header	15	Message header. See section 3.2
Message Body	Variable	This section contains the actual message. See section 4
CRC	4	CRC32. The CRC is computed from the first byte of the header to the last byte of the Message Body. See section 3.3

3.2 Message Header

Name	Number of bytes	Description
Sync byte 1	1	First sync byte. Value is 0x47
Sync byte 2	1	Second sync byte. Value is 0x83
Message Type	1	Message type. See section 3.4
Timestamp	6	Number of milliseconds from Jan 1, 1970 when the message was generated for transmission
Source ID	4	Source ID of message. Every boat has a unique SourceID. The association between SourceID and boat name can be found in the boats.xml message.
Message Length	2	Length of message. Length is for message body only. Does not include this header or the CRC.

3.3 CRC

The CRC polynomial is :

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

(0x04C11DB7)

CRC-32-IEEE 802.3

See Appendix B for sample code for calculating the CRC bytes.

3.4 Message Type Designators

Message Type	Description
1	Heartbeat.
14	Boat Instruments
20	Display Text Message
26	XML Message
27	Race Start Status
29	Yacht Event Code
31	Yacht Action Code
36	Chatter Text
41	Boat Position
53	Boat Instruments Version 2

4 Messages

4.1 Heartbeat

This message indicates that a communications channel is open. The message contains a sequence number. The message is sent approximately once per 5 seconds. The sequence number will increment by one every time a message is sent.

Message Type : 1

Message Size : 4 bytes

Heartbeat () {	Number of bytes	Data Type
SequenceNum	4	unsigned int
}		

4.2 Display Text Message

Message Type : 20

Message Size : variable

This message is used to display text on the RaceComms Displays on the race yachts. The contents of this message can be used to emulate the existing AC45 RaceComms Display.

DisplayTextMessage() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version 1 at this time. Binary
AckNumber	2	Set to zero
Number of Text Lines	1	unsigned byte Number of lines of text.
{		Loop for each message
Line Number	1	unsigned byte Line number where message should be displayed. Binary 0 => Client software decides where to put the line 1 => 8 Force to line number 15 => Penalty count
MessageTextLength	1	MessageText length.
MessageText	0 to 30	ASCII content of the message. Not null terminated
}		

Each time the RaceComms Display is updated, two Display Text Messages will be sent. The first message will contain a single line with the line number set to zero. The MessageText in the line will be the latest message generated. The first part of the text will have a time stamp followed by the text.

The second Display Text Message will contain up to 9 lines. 8 of the lines will represent a scrolling display of the current plus past 7 messages. Line 1 is the oldest message and line 8 is the most recent. The text in line 8 will be the same as the previous line zero text message without the timestamp. Line Number 15 will also be included and represents the number of penalties the yacht has.

4.3 XML Message

Message Type : 26

Message Size : variable

This message is a container for an xml message. Upon connecting to the TCP/IP server, three of these messages will be sent to the client. The three messages will contain the Regatta.xml, Race.xml, and Boats.xml data. Whenever the Principal Race Officer makes a change to the race course configuration, a new Race.xml message will be sent.

XmlMessage() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version number for this message. Set to 1 at this time
AckNumber	2	Acknowledgement number of message. This value will be set to zero
TimeStamp	6	Number of milliseconds from Jan 1, 1970 for when the data is created
XmlMsgSubType	1	MsgTypeVersionNumber
SequenceNumber	2	Sequence number of XML. Changes whenever the XML content of the particular XmsMsgSubType changes
XmlMsgLength	2	Length of following message
XML Message	Variable	The actual xml paragraph. Null terminated string

Message Sub types

Type	XML Message	Root Name
5	Regatta	RegattaConfig
6	Race	Race
7	Boat	root

See section 5 for descriptions of the xml messages.

4.4 Race Start Status

Message Type : 27

Message Size : 20

This message defines when a race is supposed to start

RaceStartStatus() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version number for this message. Set to 1 at this time
TimeStamp	6	Number of milliseconds from Jan 1, 1970 for when the data is created
AckNumber	2	Acknowledgement number of message. Will be set to zero
RaceStartTime	6	Time Race is expected to start
RaceID	4	Race Identifier Number
NotificationType	1	unsigned char See following table
}		

NotificationType	Description
1	Set Race Start Time
2	Race Postponed
3	Race Abandoned
4	Race Terminated

4.5 Yacht Event Code

This message is sent to a yacht by the Umpire or PRO.

The Yacht Event Code message is used to communicate events that are associated with a yacht. For example, if a yacht is assessed a course limits penalty, the Umpire Application will send a Yacht Event Code message with the DestinationSourceID set to the source ID for the penalized yacht and the EventID set to 17, indicating a non-Part 2 penalty has been assessed.

Message Type : 29

Message Size : 22 bytes

YachtEventCode() {	Number of bytes	Data Type
MessageVersionNumber	1	Version Number of this message Value = 1
Time	6	Number of milliseconds from Jan 1, 1970 for when the button was pressed
AckNumber	2	Sequence number of message. This number is will be reflected in an AcknowledgeMessage
RaceID	4	Unique RaceID
DestinationSourceID	4	Source ID for destination device (Subject yacht)
IncidentID	4	Uniquely identifies the specific incident (an OCS, boundary violation, Y-flag penalty, etc) to which this message (protest, penalty, penalty clear,...) refers.
<i>EventID</i>	1	<i>1 => Early entry</i> <i>2 => Entered incorrectly</i> <i>3 => OCS/Prestart Penalty</i> <i>4 => Clear behind start line after being over early</i> <i>5 => Y Flag Penalty</i> <i>6 => No penalty</i> <i>8 => DSQ, Disqualified</i> <i>9 => Penalty complete</i> <i>11 => Finished</i> <i>13 => Penalty Dealt With</i> <i>14 => DNS, Did Not Start</i> <i>15 => DNF, Did Not Finish</i> <i>17 => Boundary (or other) Penalty</i> <i>20 => Clear At Start</i> <i>22 => OCS/Prestart Penalty Complete</i> <i>25 => DNC, Did Not Compete</i> <i>26 => Detected yacht crossing outside boundary</i> <i>27 => Detected yacht clear inside boundary</i> <i>30 => Umplinitiated – create an incident for a penalty initiated by an umpire¹</i>

		<p>31 => <i>PenaltyServed</i>¹, automated preliminary indication to umpires that Part 2 penalty distance has been paid</p> <p>32 => <i>IdPacer</i>, identify the yacht against which a Part 2 penalty should be measured, if not the protesting yacht</p> <p>the following events may accompany a penalty-related event to explicitly indicate the number of penalties currently carried by a yacht¹:</p> <p>100 => <i>Zero penalties</i></p> <p>101 => <i>One penalty</i></p> <p>102 => <i>Two penalties</i></p> <p>103 => <i>Three penalties</i></p> <p>104 => <i>Four penalties</i></p> <p>105 => <i>Five penalties</i></p> <p>106 => <i>Six penalties</i></p> <p>107 => <i>Seven penalties</i></p> <p>108 => <i>Eight penalties</i></p> <p>109 => <i>Nine penalties</i></p> <p>110 => <i>Ten or more penalties</i></p>
		<p>Notes:</p> <p>¹ message sent only to umpires, not to race yachts</p>

Yacht Event Code messages are sent from MDSS, UmpApp and ProApp to communicate events associated with penalties and race outcomes for a particular yacht.

At the starting signal, each yacht is sent either a “*Clear at Start*” message (EventID 20), or an “*Over start line early (OCS)*” message (EventID 3). If Boat AAA is OCS, “*MDS AAA OCS/Prestart Penalty*” appears in the RaceComms display, and when it has completed paying that penalty, MDSS sends the yacht an “*OCS/Prestart Penalty complete*” message (EventID 22), and “*MDS AAA Penalty complete*” appears in the RaceComms display.

As the PRO rules that a boat AAA has successfully crossed the finish line carrying no penalties, they send a “*Finished*” message (EventID 11) to that yacht, and “*PRO AAA Finished*” appears in the RaceComms display. Alternatively, the PRO may send a “*DNC*” (Did Not Compete, EventID 25), “*DNS*” (Did Not Start, EventID 14), “*DNF*” (Did Not Finish, EventID 15), “*DNS*” (Did Not Start, EventID 14), or “*DSQ*” (Disqualified, EventID 8) message to indicate another scoring of that yacht in the race. If this happens, a message such as “*PRO AAA DNF*” appears in the RaceComms display. The umpires may also send “*DSQ*” messages, which appear as “*UMP AAA DSQ*” in the RaceComms display.

The following examples illustrate what messages are sent under various penalty scenarios:

Example 1: Match race: Boat A crosses early (OCS) and Boat B starts within 4 seconds of the start time. There are no pre-start penalties.

- 1) At the start signal, an “*OCS/Prestart Penalty*” message (EventID 3) will be sent for Boat A and their blue penalty light will turn on. RaceComms display penalty field on Boat A will show “OCS”. A “*Clear At Start*” message (EventID 20) will be sent for Boat B. The RaceComms display penalty field on Boat B will show “---”

- 2) Once Boat A completes their OCS penalty, the blue light on Boat A will go out and an “OCS/Prestart Penalty complete” message (EventID 22) will be sent for Boat A. The RaceComms display penalty field on Boat A will show “---”.

Example 2: Match race: Boat A crosses early (OCS). Boat B crosses the start line more than 4 seconds after the start time. There are no pre-start penalties.

- 1) An “OCS/Prestart Penalty” message will be sent for Boat A and their blue penalty light will turn on. A “Clear At Start” message will be sent for Boat B. The RaceComms display penalty field on Boat A will show “OCS”. The RaceComms display penalty field on Boat B will show “---”
- 2) At 4 seconds from the start, since Boat B has not crossed the start line, the blue light on Boat A will stay on until the umpires have determined the penalty is complete under rule 44.3(b). At this time, the boat will receive an “OCS/Prestart Penalty Complete” message (EventID 22) and the blue light will go out.

Example 3: Match race: Boat A is OCS and Boat B starts within 4 seconds of the starting signal. Boat A has a prestart penalty due to entering the pre-start area too early.

- 1) When Boat A enters early, an “Early entry” message (EventID 1) will be sent to Boat A.
- 2) When the umpire assigns a penalty for the entry violation, an “OCS/Prestart Penalty” message (EventID 3) will be sent to Boat A, and the blue penalty light will come on.
- 3) At the start signal, a second “OCS/Prestart Penalty” message will be sent for Boat A and their blue penalty light stays on. RaceComms display penalty field on Boat A will show “OCS”. A “Clear At Start” message will be sent for Boat B.
- 4) Once Boat A drops back 4 boatlengths behind Boat B – 2 boatlengths each for the entry and OCS violations – the umpires will clear both penalties. The blue penalty light will go out and a single “OCS/Prestart Penalty Complete” message (EventID 22) will be sent to Boat A. The RaceComms display penalty field on Boat A will show “---”.

Example 4: Match race: During racing, port-tack Boat A fouls starboard-tack Boat B. Boat B protests.

- 1) Boat B protests. Its onboard red light begins flashing and “B Yankee Flag” appears on the RaceComms display.
- 2) The umpire reviews the incident and issues a penalty. “UMP A Penalty” appears on the RaceComms display. A “Y Flag Penalty” message (EventID 5) is sent to Boat A, and its blue light begins flashing.
- 3) If the umpire rules that Boat A is not slowing to pay its penalty, or if Boat A tacks or gybes while carrying its penalty, the umpire may send an additional “Y Flag Penalty” message to Boat A. Another “UMP A Penalty” message appears on the RaceComms display. The distance by which Boat A must now drop behind Boat B is increased by 2 boatlengths.
- 4) When the umpire rules that Boat A has dropped 2 (or more) boatlengths behind Boat B, or otherwise satisfies its penalty requirements, the booth umpire will send a “Penalty complete” message (EventID 9) to Boat A, and its blue light will go off. “UMP A Penalty complete” will appear on the RaceComms display.

Alternatively

- 2) If the umpire decides no foul was committed, they send a “No Penalty” message (EventID 6) to the protesting boat, Boat B. “UMP B No Penalty” will appear on the RaceComms display.

Example 5: Boat A sails out of bounds while racing.

- 1) MDSS sends a “Boundary” message (EventID 26) to the umpires and Boat A.

- 2) If the umpire rules that Boat A did indeed commit a boundary violation, they send a “*Boundary/Other Penalty*” message (EventID 17) to Boat A. The blue light begins flashing on Boat A. “*UMP A Penalty*” appears on the RaceComms display.
- 3) When Boat A completely re-enters the race course boundary, MDSS sends an “*Inside Boundary*” message (EventID 27) to Boat A, and the penalty begins to pay off. The blue light continues flashing and the RaceComms display remains unchanged.
- 4) If the umpire rules Boat A is not slowing to pay off its penalty, or if Boat A tacks while under penalty, the umpire may send another “*Boundary/Other Penalty*” message (EventID 17) to Boat A. Another “*UMP A Penalty*” message appears on the RaceComms display.
- 5) When the umpire rules that Boat A has served its penalty, they send a single “*Penalty Complete*” message (EventID 9) to Boat A. The blue light on Boat A stops flashing. “*UMP A Penalty complete*” appears on the RaceComms display.

Alternatively

- 2) If the umpire rules that Boat A did not commit a violation, they silently dismiss the boundary violation from MDSS. No messages are sent to Boat A, the blue light never flashes, and no messages appear on the RaceComms display.

Example 6: Match race: During racing, Boat A gets a port/starboard penalty, then goes outside the course limits before the penalty is paid off.

- 1) A “*Y Flag Penalty*” message (EventID 5) will be sent for boat A, and their blue light will come on.
- 2) When boat A goes outside the course limits, they will receive a “*Boundary Crossing*” message (EventID 26). When the umpire issues a penalty, a “*Boundary (or other) Penalty*” message (EventID 17) will be sent to Boat A. This tells Boat A that they received another penalty due to the course limits violation.
- 3) When Boat A completely re-enters the race course boundary, MDSS sends an “*Inside Boundary*” message (EventID 27) to Boat A. The blue light continues flashing and the RaceComms display remains unchanged.
- 4) Once boat A completes both their boat on boat penalty and their obligation to slow 2 boatlengths, the umpires will simultaneously clear both penalties, the blue light on boat A will go out, and a single “*Penalty complete*” message (EventID 9) will be sent for Boat A.

4.6 YachtActionCode

This message can be sent from the client to the server to initiate a protest. In this case the OriginatorSourceID field will be initialized by the server to indicate the proper boat before forwarding to MDSS. The client should set the OriginatorSourceID and IncidentID fields to zero.

Message Type : 31

Message Size : 18 bytes

YachtActionCode() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version number for this message. Set to 1 at this time
TimeStamp	6	Time that Action Code was initiated.
AckNumber	2	Sequence number of message. This number is will be set to zero
OriginatorSourceID	4	SourceID of boat who originated this message
IncidentID	4	Unique identifier that ties an action and its subsequent YachtEvents together
EventID	1	1 => Y-flag protest 2 => B-flag protest
}		

4.7 Boat Position

This message describes the location, attitude data from boats. Each message represents one GPS sample from a boat. Each race yacht will receive data for their yacht and mark boat locations. Fields where data is not available will be set to zero.

Message Type : 41

Message Size : 42 bytes

BoatPosition () {	Number of bytes	Data Type
MessageVersionNumber	1	Version Number of this message Value = 1
Time	6	Number of milliseconds from Jan 1, 1970 for when the data is valid
SourceID	4	ID of the boat
SequenceNum	4	unsigned int Sequence number of the message. Number originates at the boat
DeviceType	1	0 -> unknown 1 -> racing yacht (your own boat) 2 -> Signal boat 3 -> Mark 4 -> Pin
Latitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ Positive for north latitude
Longitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ Positive for east longitude
Altitude	4	signed int cm relative to MSL
DistanceToLimit	4	signed int cm Distance from closest point on boat to the course limits. Positive is inside the limits, Negative is outside the limits. This is only valid for your own boat. All others will be set to zero.
Heading	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
Pitch	2	signed int degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Level = 0, Bow down = positive
Roll	2	signed int degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Level = 0, Port down = positive
COG	2	unsigned short

		North = 0x0000 East = 0x4000 South = 0x8000 Course Over Ground relative to true north From GPS IMU or GPS Doppler
SOG	2	unsigned short Speed in mm/sec Speed Over Ground. From GPS IMU or GPS Doppler
}		

Notes:

The latitude, longitude, and altitude values describe the position of the boat's origin. The origin is at the waterline, centerline, and furthest point astern.

Conversion hints:

- To convert the binary latitude or longitude to a floating point number in C++:

```
int BinaryPackedLatitude;
double Latitude = (double)BinaryPackedLatitude*180.0/ 2147483648.0;
```
- To convert the binary Heading to a floating point number in C++:

```
unsigned short BinaryPackedHeading;
double Heading = (double)BinaryPackedHeading*360.0/65536.0;
```
- To convert the binary TrueWindAngle to a floating point number in C++:

```
short BinaryPackedTWA;
double TWA = (double)BinaryPackedTWA*180.0/32768.0;
```

4.8 Boat Instruments Data

This message describes the location, attitude, and sensor data from the boat. It is sent from the client to the server once per second. The teams are required to send all data described below for which they have sensors. For those data fields below that are also displayed for the sailors onboard the race boat, the data will be the same values as displayed to the sailors.

NOTE : This message is obsolete starting with AC35

Message Type : 14

Message Size : 37 bytes

BoatInstruments () {	Number of bytes	Data Type
Time	6	Number of milliseconds from Jan 1, 1970 for when the data is valid
Latitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ North Latitude is positive
Longitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ East Longitude is positive
COG	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
SOG	2	unsigned short Speed in mm/sec
Boat Heading	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
Boat Speed	2	short Speed in mm/sec
True Wind Direction	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
True Wind Speed	2	unsigned short Speed in mm/sec
Apparent Wind Angle	2	signed short degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Relative to centerline of boat
Apparent Wind Speed	2	unsigned short Speed in mm/sec
True Wind Angle	2	signed int degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Relative to centerline of boat
True Wind Speed	2	unsigned short Speed in mm/sec
Rudder Angle	2	signed short degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ 0 = straight, Positive is rudder set to turn yacht to port.
Valid Bitmask	1	unsigned byte Data validity bit mask Bit 0 set if Lat/Lon is valid Bit 1 set COG and SOG is valid Bit 2 is set if Boatspeed and heading are valid Bit 3 is set if True Wind Direction and speed are valid Bit 4 is set if Apparent Wind Speed Angle and Speed are valid Bit 5 is set if True Wind Angle and True Wind speed are valid Bit 6 is set if Rudder Angle is valid
}		

It is acceptable to send zero for rudder angle whenever the rudder angle is within 5 degrees of centerline. The angle may be rounded to 1 degree resolution. See the following table.

Actual rudder angle	Value sent
≤ -5 degrees	Actual angle, rounded to 1 degree
> -5 degrees and < 5 degrees	0
≥ 5 degrees	Actual angle, rounded to 1 degree

4.9 Boat Instruments Data Version 2

This message describes the location, attitude, and sensor data from the boat. It is sent from the client to the server once per second. The teams are required to send all data described below for which they have sensors. For those data fields below that are also displayed for the sailors onboard the race boat, the data will be the same values as displayed to the sailors.

Message Type : 53

Message Size :45 bytes

BoatInstrumentsVer2 () {	Number of bytes	Data Type
Version	1	Version Number. Value is 2
Time	6	Number of milliseconds from Jan 1, 1970 for when the data is valid
Latitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ North Latitude is positive
Longitude	4	signed int -180 = 2 ³¹ 180 = 2 ³¹ East Longitude is positive
COG	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
SOG	2	unsigned short Speed in mm/sec
Boat Heading	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
Boat Speed	2	unsigned short Speed in mm/sec
True Wind Direction	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Relative to true north
True Wind Speed	2	unsigned short Speed in mm/sec
Apparent Wind Angle	2	signed short degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Relative to centerline of boat
Apparent Wind Speed	2	unsigned short Speed in mm/sec
True Wind Angle	2	signed int degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ Relative to centerline of boat
True Wind Speed	2	unsigned short Speed in mm/sec
Rudder Angle	2	signed short degrees. -180 = 2 ¹⁵ 180 = 2 ¹⁵ 0 = straight, Positive is rudder set to turn yacht to port.
Valid Bitmask	2	unsigned byte Data validity bit mask. Bit is set if respective data field(s) are valid Bit 0 Lat/Lon Bit 1 COG SOG Bit 2 Boatspeed and heading Bit 3 True Wind Direction and speed Bit 4 iApparent Wind Speed Angle and Speed Bit 5 True Wind Angle and True Wind speed Bit 6 Rudder Angle

		Bit 7 PortAccumPressure Bit 8 StbdAccumPressure Bit 9 AppendageAccumPressure
PortAccumPressure	2	unsigned short. Port accumulator pressure in BAR.
StbdAccumPressure	2	unsigned short. Starboard accumulator pressure in BAR .
AppendageAccumPressure	2	unsigned short. Appendage accumulator pressure in BAR.
}		

It is acceptable to send zero for rudder angle whenever the rudder angle is within 5 degrees of centerline. The angle may be rounded to 1 degree resolution. See the following table.

Actual rudder angle	Value sent
≤ -5 degrees	Actual angle, rounded to 1 degree
> -5 degrees and < 5 degrees	0
≥ 5 degrees	Actual angle, rounded to 1 degree

5 XML Messages

The XML Message contains one of three xml messages. The following sections describe the contents of these XML messages and their purpose.

5.1.1 Regatta.xml

The Regatta.xml file describes global data for the regatta such as regatta name, approximate location of the race, timezone offset, and magnetic variation.

```
<?xml version="1.0" encoding="utf-8"?>
<RegattaConfig>
  <RegattaID>31</RegattaID>
  <RegattaName>Fukuoka World Series 2016</RegattaName>
  <CourseName>Fukuoka</CourseName>
  <CentralLatitude>33.596465</CentralLatitude>
  <CentralLongitude>130.359757</CentralLongitude>
  <CentralAltitude>0.00</CentralAltitude>
  <UtcOffset>9</UtcOffset>
  <MagneticVariation>-7.4</MagneticVariation>
  <ShorelineName>fukuoka_shoreline</ShorelineName>
</RegattaConfig>
```

RegattaID – The RegattaID is a unique number used in the database as a key to the regatta

RegattaName – The name of the regatta

RaceID – The RaceID is a unique number used as a key in the database for the race

CourseName – Name of the race course

CentralLatitude – Latitude in vicinity of the race course.

CentralLongitude – Longitude in the vicinity of the race course

CentralAltitude – Altitude in the vicinity of the race course

UtcOffset – Offset in hours from UTC for the local time of the regatta.

MagneticVariation – Local Magnetic Variation

5.1.2 Race.xml

The race.xml file contains the course boundaries start time, and race type.. Typically more than one race.xml files will be sent before and during a race that describe changes to the race limits, and to the start time if sent prior to the start. This file is generated by an application operated by the Principal Race Officer. Each time the PRO makes a change to the race, a new race.xml file will be committed and sent.

This file should be considered valid until a new version of the file is sent out.

The course limits position accuracy will be exact.

```
<?xml version="1.0" encoding="utf-8"?>
<Race>
  <CreationTimeDate>2016-11-20T13:01:44+09:00</CreationTimeDate>
  <RaceStartTime Start="2016-11-20T13:07:00+09:00" Postpone="False" />
  <RaceID>16112004</RaceID>
  <RaceType>Fleet</RaceType>
  <Participants>
    <Yacht SourceID="101" />
    <Yacht SourceID="102" Entry="Stbd" />
    <Yacht SourceID="103" />
    <Yacht SourceID="104" Entry="Port" />
    <Yacht SourceID="105" />
    <Yacht SourceID="106" />
  </Participants>
  <Course>
    <CompoundMark CompoundMarkID="1" Name="SL1">
      <Mark SeqID="1" Name="Start Line 1" TargetLat="33.6047290" TargetLng="130.3585460" SourceID="122" />
      <Mark SeqID="2" Name="Start Line 2" TargetLat="33.6034350" TargetLng="130.3588330" SourceID="123" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="2" Name="M1">
      <Mark SeqID="1" Name="Mark 1" TargetLat="33.6031430" TargetLng="130.3545840" SourceID="126" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="3" Name="LG1">
      <Mark SeqID="1" Name="Lee Gate 1" TargetLat="33.5985420" TargetLng="130.3580790" SourceID="124" />
      <Mark SeqID="2" Name="Lee Gate 2" TargetLat="33.5981750" TargetLng="130.3568210" SourceID="125" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="4" Name="WG1">
      <Mark SeqID="1" Name="Wind Gate 1" TargetLat="33.6074750" TargetLng="130.3548450" SourceID="131" />
      <Mark SeqID="2" Name="Wind Gate 2" TargetLat="33.6071760" TargetLng="130.3535870" SourceID="127" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="5" Name="LG1">
      <Mark SeqID="1" Name="Lee Gate 1" TargetLat="33.5985420" TargetLng="130.3580790" SourceID="124" />
      <Mark SeqID="2" Name="Lee Gate 2" TargetLat="33.5981750" TargetLng="130.3568210" SourceID="125" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="6" Name="FL1">
      <Mark SeqID="1" Name="Finish Line 1" TargetLat="33.5988940" TargetLng="130.3592550" SourceID="128" />
      <Mark SeqID="2" Name="Finish Line 2" TargetLat="33.5994170" TargetLng="130.3592010" SourceID="129" />
    </CompoundMark>
  </Course>
  <CompoundMarkSequence>
    <Corner SeqID="1" CompoundMarkID="1" Rounding="SP" ZoneSize="3" />
    <Corner SeqID="2" CompoundMarkID="2" Rounding="Port" ZoneSize="3" />
    <Corner SeqID="3" CompoundMarkID="3" Rounding="PS" ZoneSize="3" />
    <Corner SeqID="4" CompoundMarkID="4" Rounding="SP" ZoneSize="3" />
    <Corner SeqID="5" CompoundMarkID="5" Rounding="PS" ZoneSize="3" />
    <Corner SeqID="6" CompoundMarkID="6" Rounding="SP" ZoneSize="3" />
  </CompoundMarkSequence>
  <CourseLimit name="Boundary" draw="1" avoid="1" fill="1" lock="0" colour="000000FF" notes="5,45">
    <Limit SeqID="1" Lat="33.6077220" Lon="130.3511620" />
    <Limit SeqID="2" Lat="33.6087540" Lon="130.3536860" />
    <Limit SeqID="3" Lat="33.6089710" Lon="130.3568840" />
    <Limit SeqID="4" Lat="33.6056350" Lon="130.3580430" />
    <Limit SeqID="5" Lat="33.6061660" Lon="130.3607110" />
    <Limit SeqID="6" Lat="33.6029790" Lon="130.3617080" />
    <Limit SeqID="7" Lat="33.6025070" Lon="130.3595700" />
    <Limit SeqID="8" Lat="33.5990130" Lon="130.3606660" />
    <Limit SeqID="9" Lat="33.5984300" Lon="130.3593270" />
    <Limit SeqID="10" Lat="33.5975320" Lon="130.3568030" />
    <Limit SeqID="11" Lat="33.5975170" Lon="130.3544500" />
  </CourseLimit>
</Race>
```

```

<CourseLimit name="Seaweed Net" draw="1" avoid="1" fill="1" lock="0" colour="000080FF" notes="">
  <Limit SeqID="1" Lat="33.6077970" Lon="130.3282380" />
  <Limit SeqID="2" Lat="33.6075500" Lon="130.3467420" />
  <Limit SeqID="3" Lat="33.6067050" Lon="130.3473710" />
  <Limit SeqID="4" Lat="33.6035850" Lon="130.3473620" />
  <Limit SeqID="5" Lat="33.6016690" Lon="130.3469040" />
  <Limit SeqID="6" Lat="33.6016400" Lon="130.3436430" />
  <Limit SeqID="7" Lat="33.6036220" Lon="130.3424750" />
  <Limit SeqID="8" Lat="33.6035470" Lon="130.3276990" />
</CourseLimit>
<CourseLimit name="Stake Boundary" draw="1" avoid="1" fill="1" lock="0" colour="00800000" notes="">
  <Limit SeqID="1" Lat="33.6098690" Lon="130.3465450" />
  <Limit SeqID="2" Lat="33.6107000" Lon="130.3522940" />
  <Limit SeqID="3" Lat="33.6112310" Lon="130.3556170" />
  <Limit SeqID="4" Lat="33.6109610" Lon="130.3599830" />
  <Limit SeqID="5" Lat="33.6107890" Lon="130.3615010" />
  <Limit SeqID="6" Lat="33.6075870" Lon="130.3613570" />
  <Limit SeqID="7" Lat="33.6017590" Lon="130.3632890" />
  <Limit SeqID="8" Lat="33.5989090" Lon="130.3604320" />
  <Limit SeqID="9" Lat="33.5974800" Lon="130.3567760" />
  <Limit SeqID="10" Lat="33.5976740" Lon="130.3496980" />
  <Limit SeqID="11" Lat="33.6012360" Lon="130.3482600" />
  <Limit SeqID="12" Lat="33.6073780" Lon="130.3485750" />
</CourseLimit>
</Race>

```

RaceID – RaceID is a unique number for each race. The number is typically created on the Expedition program. The ID is created using the following format. YYMMDDNN where YY is the last two digits of the year, MM is the Month, DD is the Day, and NN is the race number.

RaceType – Type of race (Match or Fleet)

RaceStartTime – There are two attributes for the RaceStartTime. ‘Time’ is the local time that the race will start, plus an indication of the local time zone offset from UTC. ‘Postpone’ is true or false. If it is false, the Time describes the time when the race will start (or started). If ‘Postpone’ is true, the race is postponed, and the ‘Time’ field is invalid.

Participants – this element contains a list of the source ids of each yacht in this race. If this is a match race, the Entry attribute indicates on which tack each race yacht will enter the starting area. If this is a fleet race, the Entry attribute may or may not be present, but it has no meaning if it is present.

Course – this element contains a list of the individual mark boats, their target lat long positions, and the source id that identifies the boat. Use the source id to determine which BoatPosition messages to use to get the actual current position of that mark.

CompoundMarkSequence – This element defines the order in which compound marks are rounded in sailing the race. The Rounding attribute is a hint of the direction the yacht should round the compound mark (e.g. Port means leave the single mark to port; PS means leave the 1st mark of the gate to port and the 2nd to starboard, etc.) but **these indications should not be relied on to be accurate**. Single marks are always rounded in the sensible direction to the next mark, and gates are always crossed in the direction from the previous mark.

CreationTimeDate – Local Time and time zone indicating when this race.xml file was created.

CourseLimit – This section describes the course limits. The sequence of ‘Limit’ tags must be clockwise. There may be multiple CourseLimit elements in the document – only the first one defines the racing boundary; the rest represent other features of the course such as marshaling limits and are for information only.

5.1.3 Boats.xml

This file is used to describe all of the boats and other entities involved in the race. One key value in the file is the SourceID. This is a unique number used to associate messages to a particular boat. Another key value is the HullNum. This number uniquely identifies the boat hull. The Boats.xml file is used to map the hull number to boat name and SourceID. To find the SourceID for a particular boat, look in the <Boat> section and find the HullNum or BoatName. Use the SourceID for that boat. The SourceID is associated with the physical navigation system installed in the boat. The SourceID for a particular boat will not change often, but could change if the hardware fails and a spare system is installed in the boat.

Boats.xml will not change very often. Examples of things that would cause the boats.xml file to change are changing the navigation peli case, or adding a new boat to the system.

The BoatShapes section describes the outline of the boats. Each section has a list of vertices that describe the outline. Units are meters.

```
<?xml version="1.0" encoding="utf-8"?>
<BoatConfig>
  <Modified>2016-11-19T06:59:50+0100</Modified>
  <Version>12</Version>
  <Snapshot>275</Snapshot>
  <Settings>
    <RaceBoatType Type="AC45" />
    <BoatDimension BoatLength="14.019" HullLength="13.449" />
    <ZoneSize MarkZoneSize="40.347" CourseZoneSize="53.796" />
    <ZonLimits Limit1="200" Limit2="100" Limit3="53.796" Limit4="0" Limit5="-100" />
  </Settings>
  <BoatShapes>
    <BoatShape ShapeID="0">
      <Vertices>
        <Vtx Seq="3" Y="25" X="0" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="14">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-1" />
        <Vtx Seq="2" Y="0.75" X="-1" />
        <Vtx Seq="3" Y="0.75" X="-0.25" />
        <Vtx Seq="4" Y="3.5" X="-0.25" />
        <Vtx Seq="5" Y="4.5" X="-1" />
        <Vtx Seq="6" Y="6.5" X="-1" />
        <Vtx Seq="7" Y="7" X="-0.5" />
        <Vtx Seq="8" Y="7" X="0.5" />
        <Vtx Seq="9" Y="6.5" X="1" />
        <Vtx Seq="10" Y="4.5" X="1" />
        <Vtx Seq="11" Y="3.5" X="0.25" />
        <Vtx Seq="12" Y="0.75" X="0.25" />
        <Vtx Seq="13" Y="0.75" X="1" />
        <Vtx Seq="14" Y="0" X="1" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="18">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-1.04" />
        <Vtx Seq="2" Y="0.11" X="-1.18" />
        <Vtx Seq="3" Y="0.42" X="-1.28" />
        <Vtx Seq="4" Y="3.74" X="-1.29" />
        <Vtx Seq="5" Y="5.36" X="-1.21" />
        <Vtx Seq="6" Y="6.29" X="-1.08" />
        <Vtx Seq="7" Y="7.15" X="-0.84" />
        <Vtx Seq="8" Y="7.63" X="-0.62" />
        <Vtx Seq="9" Y="7.94" X="-0.34" />
        <Vtx Seq="10" Y="8.06" X="0" />
        <Vtx Seq="11" Y="7.94" X="0.34" />
        <Vtx Seq="12" Y="7.63" X="0.62" />
        <Vtx Seq="13" Y="7.15" X="0.84" />
        <Vtx Seq="14" Y="6.29" X="1.08" />
      </Vertices>
    </BoatShape>
  </BoatShapes>
</BoatConfig>
```

```

    <Vtx Seq="15" Y="5.36" X="1.21" />
    <Vtx Seq="16" Y="3.74" X="1.29" />
    <Vtx Seq="17" Y="0.42" X="1.28" />
    <Vtx Seq="18" Y="0.11" X="1.18" />
    <Vtx Seq="19" Y="0" X="1.04" />
  </Vertices>
</BoatShape>
<BoatShape ShapeID="20">
  <Vertices>
    <Vtx Seq="1" Y="-0.05" X="-0.05" />
    <Vtx Seq="2" Y="0.05" X="-0.05" />
    <Vtx Seq="3" Y="0.05" X="0.05" />
    <Vtx Seq="4" Y="-0.05" X="0.05" />
  </Vertices>
</BoatShape>
<BoatShape ShapeID="24">
  <Vertices>
    <Vtx Seq="1" Y="0" X="-2.5" />
    <Vtx Seq="2" Y="7" X="-2.5" />
    <Vtx Seq="3" Y="12.6" X="-2.2" />
    <Vtx Seq="4" Y="12.6" X="2.2" />
    <Vtx Seq="5" Y="7" X="2.5" />
    <Vtx Seq="6" Y="0" X="2.5" />
  </Vertices>
</BoatShape>
<BoatShape ShapeID="34">
  <Vertices>
    <Vtx Seq="1" Y="0" X="-1.16" />
    <Vtx Seq="2" Y="5.51" X="-1.16" />
    <Vtx Seq="3" Y="5.846" X="-0.84" />
    <Vtx Seq="4" Y="5.846" X="0.84" />
    <Vtx Seq="5" Y="5.51" X="1.16" />
    <Vtx Seq="6" Y="0" X="1.16" />
  </Vertices>
</BoatShape>
<BoatShape ShapeID="35">
  <Vertices>
    <Vtx Seq="1" Y="0" X="-1.461" />
    <Vtx Seq="2" Y="6" X="-1.461" />
    <Vtx Seq="3" Y="7" X="-1.44" />
    <Vtx Seq="4" Y="8" X="-1.38" />
    <Vtx Seq="5" Y="9" X="-1.17" />
    <Vtx Seq="6" Y="10" X="-0.76" />
    <Vtx Seq="7" Y="10.6" X="-0.34" />
    <Vtx Seq="8" Y="10.61" X="0" />
    <Vtx Seq="9" Y="10.6" X="0.34" />
    <Vtx Seq="10" Y="10" X="0.76" />
    <Vtx Seq="11" Y="9" X="1.17" />
    <Vtx Seq="12" Y="8" X="1.38" />
    <Vtx Seq="13" Y="7" X="1.44" />
    <Vtx Seq="14" Y="6" X="1.461" />
    <Vtx Seq="15" Y="0" X="1.461" />
  </Vertices>
</BoatShape>
<BoatShape ShapeID="36">
  <Vertices>
    <Vtx Seq="1" Y="0" X="-3.34" />
    <Vtx Seq="2" Y="5.351" X="-3.46" />
    <Vtx Seq="3" Y="10.544" X="-3.387" />
    <Vtx Seq="4" Y="13.449" X="-3.075" />
    <Vtx Seq="5" Y="13.55" X="0" />
    <Vtx Seq="6" Y="13.449" X="3.075" />
    <Vtx Seq="7" Y="10.544" X="3.387" />
    <Vtx Seq="8" Y="5.351" X="3.46" />
    <Vtx Seq="9" Y="0" X="3.34" />
  </Vertices>
<Catamaran>
  <Vtx Seq="1" Y="1.769" X="-2.752" />
  <Vtx Seq="2" Y="0" X="-2.813" />
  <Vtx Seq="3" Y="0" X="-3.34" />

```

```

<Vtx Seq="4" Y="5.351" X="-3.46" />
<Vtx Seq="5" Y="10.544" X="-3.387" />
<Vtx Seq="6" Y="13.449" X="-3.075" />
<Vtx Seq="7" Y="10.851" X="-2.793" />
<Vtx Seq="8" Y="6.669" X="-2.699" />
<Vtx Seq="9" Y="6.669" X="2.699" />
<Vtx Seq="10" Y="10.851" X="2.793" />
<Vtx Seq="11" Y="13.449" X="3.075" />
<Vtx Seq="12" Y="10.544" X="3.387" />
<Vtx Seq="13" Y="5.351" X="3.46" />
<Vtx Seq="14" Y="0" X="3.34" />
<Vtx Seq="15" Y="0" X="2.813" />
<Vtx Seq="16" Y="1.769" X="2.752" />
</Catamaran>
<Bowsprit>
<Vtx Seq="1" Y="6.669" X="-0.2" />
<Vtx Seq="2" Y="11.377" X="-0.2" />
<Vtx Seq="3" Y="14.019" X="0" />
<Vtx Seq="4" Y="11.377" X="0.2" />
<Vtx Seq="5" Y="6.669" X="0.2" />
</Bowsprit>
<Trampoline>
<Vtx Seq="1" Y="2" X="-2.699" />
<Vtx Seq="2" Y="6.438" X="-2.699" />
<Vtx Seq="3" Y="6.438" X="2.699" />
<Vtx Seq="4" Y="2" X="2.699" />
</Trampoline>
</BoatShape>
<BoatShape ShapeID="39">
<Vertices>
<Vtx Seq="1" Y="0" X="-6.58" />
<Vtx Seq="2" Y="33.25" X="-6.54" />
<Vtx Seq="3" Y="35.5" X="-6.4" />
<Vtx Seq="4" Y="37.58" X="-5.8" />
<Vtx Seq="5" Y="38.91" X="-5.11" />
<Vtx Seq="6" Y="39.19" X="-4.79" />
<Vtx Seq="7" Y="39.37" X="0" />
<Vtx Seq="8" Y="39.19" X="4.79" />
<Vtx Seq="9" Y="38.91" X="5.11" />
<Vtx Seq="10" Y="37.58" X="5.8" />
<Vtx Seq="11" Y="35.5" X="6.4" />
<Vtx Seq="12" Y="33.25" X="6.54" />
<Vtx Seq="13" Y="0" X="6.58" />
</Vertices>
</BoatShape>
</BoatShapes>
<Boats>
<Boat Type="RC" SourceID="121" ShapeID="35" StoweName="PRO" ShortName="PRO" ShorterName="PRO"
BoatName="REGARDLESS" HullNum="RG02" Skipper="Iain Murray" Helmsman="Iain Murray" PeliID="121">
<GPSposition Z="3.97" Y="4.15" X="0" />
<FlagPosition Z="0" Y="3.77" X="0" />
</Boat>
<Boat Type="Mark" SourceID="122" ShapeID="34" StoweName="SL1" ShortName="SL1" ShorterName="SL1"
BoatName="Start Line 1" HullNum="Mark-02" Skipper="" PeliID="122">
<GPSposition Z="5.573" Y="1.12" X="0" />
<FlagPosition Z="0" Y="0.74" X="0" />
</Boat>
<Boat Type="Mark" SourceID="123" ShapeID="34" StoweName="SL2" ShortName="SL2" ShorterName="SL2"
BoatName="Start Line 2" HullNum="Mark-03" Skipper="" PeliID="123">
<GPSposition Z="5.573" Y="1.12" X="0" />
<FlagPosition Z="0" Y="0.74" X="0" />
</Boat>
<Boat Type="Mark" SourceID="124" ShapeID="34" StoweName="LG1" ShortName="LG1" ShorterName="LG1"
BoatName="Lee Gate 1" HullNum="Mark-04" Skipper="" PeliID="124">
<GPSposition Z="5.573" Y="1.12" X="0" />
<FlagPosition Z="0" Y="0.74" X="0" />
</Boat>
<Boat Type="Mark" SourceID="125" ShapeID="34" StoweName="LG2" ShortName="LG2" ShorterName="LG2"
BoatName="Lee Gate 2" HullNum="Mark-05" Skipper="" PeliID="125">
<GPSposition Z="5.573" Y="1.12" X="0" />

```

```

    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="126" ShapeID="34" StoweName="M1" ShortName="M1" ShorterName="M1"
  BoatName="Mark 1" HullNum="Mark 1" Skipper="" PeliID="126">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="127" ShapeID="34" StoweName="WG2" ShortName="WG2" ShorterName="WG2"
  BoatName="Wind Gate 2" HullNum="Mark-07" Skipper="" PeliID="127">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="128" ShapeID="34" StoweName="FL1" ShortName="FL1" ShorterName="FL1"
  BoatName="Finish Line 1" HullNum="Mark-08" Skipper="" PeliID="128">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="129" ShapeID="34" StoweName="FL2" ShortName="FL2" ShorterName="FL2"
  BoatName="Finish Line 2" HullNum="Mark-09" Skipper="" PeliID="129">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="130" ShapeID="34" StoweName="SP1" ShortName="SP1" ShorterName="Sp1"
  BoatName="Entry Mark" HullNum="Mark-10" Skipper="" PeliID="130">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Mark" SourceID="131" ShapeID="34" StoweName="WG1" ShortName="WG1" ShorterName="WG1"
  BoatName="Wind Gate 1" HullNum="Wind Gate 1" Skipper="" PeliID="131">
    <GPSposition Z="5.573" Y="1.12" X="0" />
    <FlagPosition Z="0" Y="0.74" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="101" ShapeID="36" StoweName="USA" ShortName="ORACLE" ShorterName="USA"
  BoatName="ORACLE TEAM USA" HullNum="AC4515" Skipper="SPITHILL" Helmsman="SPITHILL" Country="USA"
  PeliID="101">
    <GPSposition Z="1.53" Y="-0.331" X="-0.006" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="102" ShapeID="36" StoweName="SWE" ShortName="ARTEMIS" ShorterName="SWE"
  BoatName="ARTEMIS RACING" HullNum="AC4517" Skipper="OUTTERIDGE" Helmsman="OUTTERIDGE" Country="SWE"
  PeliID="102">
    <GPSposition Z="1.477" Y="-0.359" X="-0.0121" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="103" ShapeID="36" StoweName="NZL" ShortName="ETNZ" ShorterName="NZL"
  BoatName="EMIRATES TEAM NZ" HullNum="AC4503" Skipper="ASHBY" Helmsman="BURLING" Country="NZL"
  PeliID="103">
    <GPSposition Z="1.631" Y="-0.291" X="-0.003" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="104" ShapeID="36" StoweName="JPN" ShortName="JAPAN" ShorterName="JPN"
  BoatName="SOFTBANK TEAM JAPAN" HullNum="AC4504" Skipper="BARKER" Helmsman="BARKER" Country="JPN"
  PeliID="109">
    <GPSposition Z="1.555" Y="-0.322" X="-0.003" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="105" ShapeID="36" StoweName="FRA" ShortName="FRANCE" ShorterName="FRA"
  BoatName="GROUPAMA TEAM FRANCE" HullNum="AC4505" Skipper="CAMMAS" Helmsman="CAMMAS" Country="FRA"
  PeliID="107">
    <GPSposition Z="1.613" Y="-0.3" X="-0.003" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
  <Boat Type="Yacht" SourceID="106" ShapeID="36" StoweName="GBR" ShortName="GBR" ShorterName="GBR"
  BoatName="LAND ROVER BAR" HullNum="AC4516" Skipper="AINSLIE" Helmsman="AINSLIE" Country="GBR"
  PeliID="106">

```

```

    <GPSposition Z="1.484" Y="-0.352" X="0" />
    <MastTop Z="21.496" Y="5.2" X="0" />
    <FlagPosition Z="0" Y="6.2" X="0" />
  </Boat>
</Boats>
</BoatConfig>

```

BoatShapes – The BoatShapes section contain a set of BoatShape objects. Each BoatShape object describes the shape of a boat.

BoatShape – The BoatShape tag contains a ShapeID value. This ID is referenced in the Boats section *to associate a boat shape with a boat. The list of vertices have an assume start vertex at the boat origin, 0,0.* The shape must be described clockwise. One of the boat shapes will have tags for **Catamaran**, **Bowsprit**, and **Trampoline**. These shapes describe the approximate shape of the AC45 or AC Class Yacht's race yacht.

Boats – The Boats tag lists that boats that are tracked during the race. It includes the committee boat, mark boats, and racing yachts

Boat – The boat tag describes attributes of the boats.

SourceID – SourceID is the number in the mdMsg header used to identify the source of a message

ShapeID – ShapeID is the reference ID to the BoatShape in the BoatShapes tag.

BoatName – Official name of the boat. Normally the name printed on the boat

ShortName – Abbreviated name for the boat.

StoweName – Abbreviation to use when sending messages to the RaceComms Display. If this tag is missing, use the **ShortName** tag.

HullNum – Unique hull number for the boat. This number will follow the physical boat hull, regardless of team or owner.

Skipper – Name of the boat skipper.

Country – Three letter abbreviation of the team country.

Type – Type of device. Valid names are

- Yacht
- RC
- Mark
- Umpire
- Marshall
- Pin

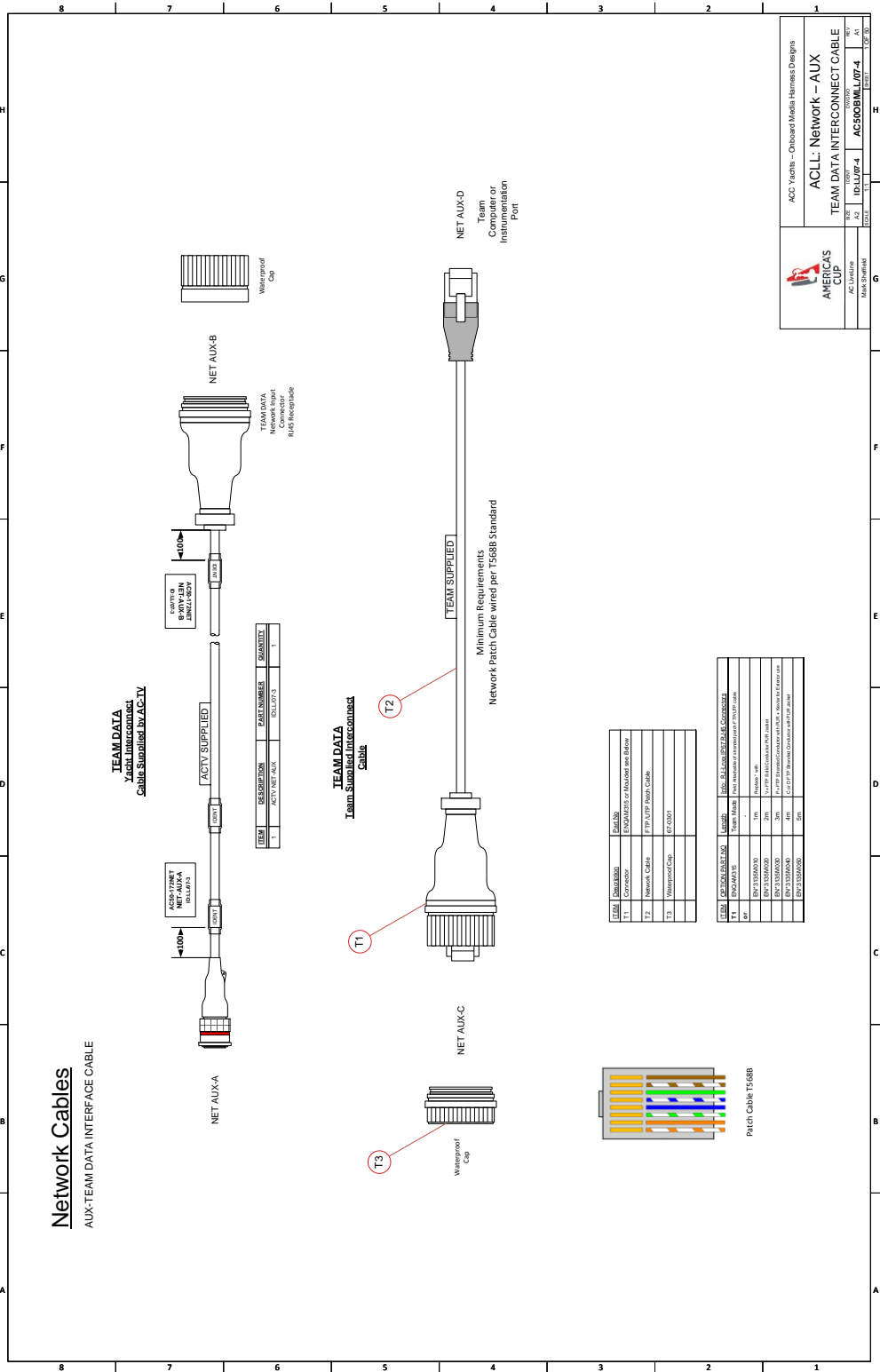
GPSposition – Location of the GPS Antenna relative to the boat coordinate system

FlagPosition – Location of the coordinate on the boat where the position should be reported relative to the boat coordinate system. This is used for the mark boats to indicate where the flag or center of the boat is.

Appendix A. Loom Drawing

Appendix B. CRC Calculation

Here's some example code in C# for calculating the message CRC. The URL of the original source is listed, but the link no longer seems to be working.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Security.Cryptography;

namespace Messaging
{
    /// <summary>
    /// Compute a Crc32
    /// code lifted from http://tomkaminski.com/crc32-hashalgorithm-c-net
    /// CRC("message digest") = 7F 9D 15 20
    /// </summary>
    public class Crc32 : HashAlgorithm
    {
        public const uint DefaultSeed = 0xffffffff;

        readonly static uint[] CrcTable = new uint[]
        {
            0x00000000, 0x77073096, 0xEE0E612C, 0x990951BA, 0x076DC419,
            0x706AF48F, 0xE963A535, 0x9E6495A3, 0x0EDB8832, 0x79DCB8A4,
            0xE0D5E91E, 0x97D2D988, 0x09B64C2B, 0x7EB17CBD, 0xE7B82D07,
            0x90BF1D91, 0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE,
            0x1ADAD47D, 0x6DDDE4EB, 0xF4D4B551, 0x83D385C7, 0x136C9856,
            0x646BA8C0, 0xFD62F97A, 0x8A65C9EC, 0x14015C4F, 0x63066CD9,
            0xFA0F3D63, 0x8D080DF5, 0x3B6E20C8, 0x4C69105E, 0xD56041E4,
            0xA2677172, 0x3C03E4D1, 0x4B04D447, 0xD20D85FD, 0xA50AB56B,
            0x35B5A8FA, 0x42B2986C, 0xDBBBC9D6, 0xACBCF940, 0x32D86CE3,
            0x45DF5C75, 0xDCD60DCF, 0xABD13D59, 0x26D930AC, 0x51DE003A,
            0xC8D75180, 0xBFDD06116, 0x21B4F4B5, 0x56B3C423, 0xCFBA9599,
            0xB8BDA50F, 0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924,
            0x2F6F7C87, 0x58684C11, 0xC1611DAB, 0xB6662D3D, 0x76DC4190,
            0x01DB7106, 0x98D220BC, 0xEF05102A, 0x71B18589, 0x06B6B51F,
            0x99FBFE4A5, 0xE8B8D433, 0x7807C9A2, 0x0F00F934, 0x9609A88E,
            0xE10E9818, 0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01,
            0xB6B6B51F4, 0x1C6C6162, 0x856530D8, 0xF262004E, 0x6C0695ED,
            0x1B01A57B, 0x8208F4C1, 0xF50FC457, 0x65B0D9C6, 0x12B7E950,
            0x8BBEB8EA, 0xFCB9887C, 0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3,
            0xFBD44C65, 0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2,
            0x4ADFA541, 0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB, 0x4369E96A,
            0x346ED9FC, 0xAD678846, 0xDA60B8D0, 0x44042D73, 0x33031DE5,
            0xAA0A4C5F, 0xDD0D7CC9, 0x5005713C, 0x270241AA, 0xBE0B1010,
            0xC90C2086, 0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F,
            0x5EDEF90E, 0x29D9C998, 0xB0D09822, 0xC7D7A8B4, 0x59B33D17,
            0x2EB40D81, 0xB7BD5C3B, 0xC0BA6CAD, 0xEDB88320, 0x9ABFB3B6,
            0x03B6E20C, 0x74B1D29A, 0xEAD54739, 0x9DD277AF, 0x04DB2615,
            0x73DC1683, 0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8,
            0xE40ECF0B, 0x9309FF9D, 0x0A00AE27, 0x7D079EB1, 0xF00F9344,
            0x8708A3D2, 0x1E01F268, 0x6906C2FE, 0xF62575D, 0x806567CB,
            0x196C3671, 0x6E6B06E7, 0xFED41B76, 0x89D32BE0, 0x10DA7A5A,
            0x67DD4ACC, 0xF9B9DF6F, 0x8EBEFFF9, 0x17B7BE43, 0x60B08ED5,
            0xD6D6A3E8, 0xA1D1937E, 0x38D8C2C4, 0x4FFF252, 0xD1BB67F1,
            0xA6BC5767, 0x3FB506DD, 0x48B2364B, 0xD80D2BDA, 0xAF0A1B4C,
            0x36034AF6, 0x41047A60, 0xDF60EFC3, 0xA867DF55, 0x316E8EEF,
            0x4669BE79, 0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236,
            0xCC0C7795, 0xBB0B4703, 0x220216B9, 0x5505262F, 0xC5BA3BBE,
            0xB2BD0B28, 0x2BB45A92, 0x5CB36A04, 0xC2D7FFA7, 0xB5D0CF31,

```

```

        0x2CD99E8B, 0x5BDEAE1D, 0x9B64C2B0, 0xEC63F226, 0x756AA39C,
        0x026D930A, 0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713,
        0x95BF4A82, 0xE2B87A14, 0x7BB12BAE, 0x0CB61B38, 0x92D28E9B,
        0xE5D5BE0D, 0x7CDCEFB7, 0x0BDBDF21, 0x86D3D2D4, 0xF1D4E242,
        0x68DDB3F8, 0x1FDA836E, 0x81BE16CD, 0xF6B9265B, 0x6FB077E1,
        0x18B74777, 0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C,
        0x8F659EFF, 0xF862AE69, 0x616BFFD3, 0x166CCF45, 0xA00AE278,
        0xD70DD2EE, 0x4E048354, 0x3903B3C2, 0xA7672661, 0xD06016F7,
        0x4969474D, 0x3E6E77DB, 0xAED16A4A, 0xD9D65ADC, 0x40DF0B66,
        0x37D83BF0, 0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9,
        0xBDBDF21C, 0xCABAC28A, 0x53B39330, 0x24B4A3A6, 0xBAD03605,
        0xCDD70693, 0x54DE5729, 0x23D967BF, 0xB3667A2E, 0xC4614AB8,
        0x5D681B02, 0x2A6F2B94, 0xB40BBE37, 0xC30C8EA1, 0x5A05DF1B,
        0x2D02EF8D
    };

    uint crcValue = 0;

    public override void Initialize()
    {
        crcValue = 0;
    }

    protected override void HashCore( byte[] buffer, int start, int length )
    {
        crcValue ^= DefaultSeed;

        unchecked {
            while ( --length >= 0 ) {
                crcValue = CrcTable[ ( crcValue ^ buffer[ start++ ] ) & 0xFF ] ^ (
crcValue >> 8 );
            }
        }

        crcValue ^= DefaultSeed;
    }
    protected override byte[] HashFinal()
    {
        this.HashValue = new byte[] { (byte)((crcValue >> 24) & 0xff),
            (byte)((crcValue >> 16) & 0xff),
            (byte)((crcValue >> 8) & 0xff),
            (byte)(crcValue & 0xff) };

        return this.HashValue;
    }
    public uint CrcValue
    {
        get
        {
            return (uint) ( ( HashValue[ 0 ] << 24 ) | ( HashValue[ 1 ] << 16 ) | (
HashValue[ 2 ] << 8 ) | HashValue[ 3 ] );
        }
    }
    public override int HashSize
    {
        get { return 32; }
    }
}
}

```

Here is example C# code for calling the CRC methods.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Messaging;

namespace TestCrc32 {
    class Program {

        static void Main( string[] args ) {
            Crc32 crc32 = new Crc32();
            string message = "message digest";
            byte[] message_bytes = Encoding.ASCII.GetBytes( message );

            crc32.Initialize();
            byte[] crc32_bytes = crc32.ComputeHash( message_bytes );

            Console.WriteLine( "Message: \"{0}\"", crc32: {1:X2} {2:X2} {3:X2} {4:X2} ",
                message, crc32_bytes[3], crc32_bytes[2], crc32_bytes[1], crc32_bytes[0]
            );
        }
    }
}
```

This program produces the following output.

```
Message: "message digest", crc32: 7F 9D 15 20
```

Note the bytes should be placed into the CRC field of the message least significant byte first.

Appendix C. Revision History

Revision	Date	Editor	Changes Made
1.0	16 Dec 2011	Ken Milnes Stan Honey Mark Sheffield	Initial Draft for Teams
1.1	4 June 2012	Ken Milnes	Update xml examples.
1.2	3 Sep 2012	Ken Milnes	Include message type for Boat Instruments.
1.3	23 Jan 2013	Ken Milnes Stan Honey	Document changes to feed starting in April 2013 Naples regatta. 1) Target Lat/Lon included in Race.xml 2) Mark boat position accuracy not degraded.
1.4	27 Mar 2013	Ken Milnes	Add additional codes in YachtActionCode message. Add documentation to this message Add Settings section in Boats.xml
1.5	2 Dec 2016	Ken Milnes Richard Slater Tim Heidmann Mark Sheffield	Update for AC35 Add field to YachtActionCode and YachtEventCode messages; document example YachtEvent message sequences Add sample CRC32 code Additional attributes in race.xml message CourseLimit tag, multiple CourseLimit tags.
1.6	17 Jan 2017	Ken Milnes	Add new BoatInstrumentsVer2 message incorporating hydraulic pressure values.
1.7	10 Apr 2017	Ken Milnes	Correct error in message size for BoatInstruments2 message. Change from 53 to 45 bytes
1.8	5 May 2017	Ken Milnes	Change Boat Speed in BoatInstruments Data version 2 from short to unsigned short