AC35 Streaming Data Interface Specification

26 February 2016

Version 1.11

Document Identifier:
AC35 Streaming Data Interface Specification.doc

Technical Committee:
ACTV Tech

Editor(s):
Ken Milnes

Table of Contents

1 Introduction

This document describes the streaming data interface for live Americas Cup race and navigation data. Methods to obtain the streaming data and the contents of the data are described. The intent of the streaming data is to allow Americas Cup teams, commercial vendors and application developers to access the live data containing the status of a race, position of the boats, and wind information during the racing event.

The streaming data is fairly wide bandwidth. Data is streamed from approximately 25 boats at up to 10 hz per boat. Due to this large bandwidth, the data is binary encoded. A binary header precedes each message so that clients can frame individual message packets.

Live data is available during each race and practice event. All of the data is multiplexed into one stream available on a TCP/IP port. The position of the race yachts and race management motorboats is streamed in the “Boat Location” messages. These messages are sent on the stream as soon as they are received from the boats. “Race Status” Messages describes the race course wind direction, speed, and the status of each boat in the race. Race course information such as course limits, course description, and intended mark boat locations is sent in “Race.xml” messages. These messages are sent whenever the Principle 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.

2 TCP/IP Connection

Data is transported via TCP/IP. Each client should connect the server at livedata.americascup.com port 4940 as a TCP/IP client. Once connected, data will start streaming. The server only sends data and does not accept any data sent from the client to the server.

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 source 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 test server is available by connecting to port 4941 instead of port 4940. Port 4941 has the same data format as port 4940, but has a canned race that repeats forever. This port can be used for demonstrations and testing.

The “Chatter” data stream that contains various race events and brief commentary messages is streamed on port 4967 of the same URL

3 Binary 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)

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
CRC	4	CRC32. The CRC is computed from the first byte of the header to the last byte of the Message Body.

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 3.3
Timestamp	6	Number of milliseconds from Jan 1, 1970 when the message was generated for transmission
Source ID	4	Source ID of 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 :
x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

(0x04C11DB7)

CRC-32-IEEE 802.3

Example code for computing the CRC can be found at:

http://www.createwindow.com/programming/crc32/index.htm

http://www.networkdls.com/Software/Release/View/CRC32/1003/

3.4 Message Type Designators

Message Type	Description
1	Heartbeat.
12	Race Status
20	Display Text Message
26	XML Message
27	Race Start Status
29	Yacht Event Code
31	Yacht Action Code
36	Chatter Text
37	Boat Location
38	Mark Rounding
44	Course Wind
47	Avg Wind

4 Binary 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 Race Status

This message is sent from the MDS to clients to inform them of the status of the race.

RaceStatus () {	Number of bytes	Data Type
MessageVersionNumber	1	Message version number. Binary. value is 2 at this time
CurrentTime	6	Current UTC time when message was generated by MDSS
RaceId	4	Race identifier
RaceStatus	1	0 – Not Active 1 – Warning (between 3:00 and 1:00 before start) 2 – Preparatory (less than 1:00 before start) 3 – Started 4 – Finished (obsolete) 5 – Retired (obsolete) 6 – Abandoned 7 – Postponed 8 – Terminated 9 – Race start time not set 10 – Prestart (more than 3:00 until start)
Expected Start Time	6	Number of milliseconds from Jan 1, 1970 for when the data is valid
Race Course Wind Direction	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000
Race Course Wind Speed	2	unsigned short, mm/sec
Number of Boats in Race	1	unsigned char
RaceType	1	Unsigned char 1 -> Match Race 2 -> Fleet Race
{		For each boat
SourceID	4	SourceID of the boat
Boat Status	1	unsigned char 0 -> Undefined 1 -> Prestart 2 -> Racing 3 -> Finished 4 -> DNS (did not start) 5 -> DNF (did not finish) 6 -> DSQ (disqualified) 7 -> OCS (On Course Side – across start line early)
Leg Number	1	unsigned char 0 => Prestart 1 => From start to first mark 2 and above => Sequential numbers as race proceeds NumLegs + 1 => Finished

Number penalties awarded	1	unsigned char
Number penalties served	1	unsigned char
Estimated time at next mark	6	Time in milliseconds at next mark
Estimated time at finish	6	Time in milliseconds at race end
}		
}		

A RaceStatus message is sent by MDSS periodically for each race MDSS knows about. Races under way and in prestart are reported more frequently that races still well in the future or races that have been Terminated or Abandoned.

Note that in the per-boat section of the message, LegNumber does not change from 0 to 1 until the yacht has crossed the starting line and the PRO (Principal Race Officer) has ruled it as having Started. The leg number does not change from NumLegs to NumLegs+1 until the PRO has ruled it as having Finished.

4.3 Display Text Message

Message Type : 20
Message Size : variable

This message is used to display text on the Stowe Displays on the race yachts.
This message is sent by the MDSS to clients to display text messages destined for the yacht displays.

DisplayTextMessage() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version 1 at this time. Binary
AckNumber	2	
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 => 10 Force to line number 15 => Penalty 16 => Race Start Time 17 => Course limit distance 20 – 25 reserved for internal use 100 => Committee boat large format display. Line 100 will be displayed on all three displays. Line 101 is the first display, Line 102 is the second display, Line 103 is the third display and Line 104 is the fourth display. If Line 100 is blank, Lines 101, 102, 103, and 104 are valid. If 100 is not blank, 101, 102, 103, and 104 are overridden with contents of 100. For lines 100 -> 104, if the text is set to “cdt”, the race Count Down Time is displayed. If it is set to “rst”, the Race Start Time is displayed. 200 – 205 => Rolling display for all committee boat large format displays. If any of the lines between 200 and 205 are non-null, all of the committee boat large displays will roll between the non-blank lines. Each line will be displayed for approximately 3 seconds, then roll to the next non-null line. If all of the lines are blank, the displays will revert to the line 100 -> line 104 methods described above.
MessageTextLength	1	unsigned byte The number of bytes in the MessageText field.
MessageText	0 to 30	UTF-8 text (like ASCII for 7-bit characters, but Unicode

		characters may be encoded in multiple bytes.) Not null-terminated. The text message.
}		

Each time the Stowe 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 test 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.

4.4 XML Message

Message Type : 26
Message Size : variable

This message is a container for an xml message.

XmlMessage() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version number for this message. Set to 1 at this time
AckNumber	2	Sequence number of message. This number is will be reflected in an AcknowledgeMessage
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

Sub types

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

4.5 Race Start Status

Message Type : 20

Message Size : variable

This message defines when a race is supposed to start, if it is postponed, terminated, or abandoned

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	Sequence number of message. This number is will be reflected in an AcknowledgeMessage
RaceStartTime	6	Time Race is expected to start
RaceID	4	Race Identifier Number (from race configuration database
NotificationType	1	unsigned char See following table
}		

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

4.6 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 Y-Flag 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 5.

Message Type : 29

Message Size : 22 bytes

YachtEventCode() {	Number of bytes	Data Type
MessageVersionNumber	1	Version Number of this message Value = 2
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	For each incident, a unique identifier tying together all the yacht events associated with that incident, e.g. for a protest-penalty incident: Y-Flag protest, UmpOwn, Y-Flag Penalty, Penalty Complete.
EventID	*1*	• *1 => Entered pre-start area early. Incident is identified by its incident ID.* • *2 => Entered pre-start area late. Incident is identified by its incident ID.* • *3 => Over start line early. Incident is identified by its incident ID.* • *4 => Clear behind start line after being over early. Incident is identified by its incident ID.* • *5 => Y Flag Penalty violation. Incident is identified by its incident ID.* • *6 => No penalty. Incident is identified by its incident ID.* • *7 => DSQ, 44.1(c). <<Unused in AC35>>* • *8 => DSQ, other* • *9 => Penalty complete. Incident is identified by its incident ID.* • *10 => Penalty offset. <<Unused in AC35>>* *11 => Finished* *12 => Y-flag protest acknowledgement* *13 => Penalty Dealt With – bookkeeping notification that the indicated incident has been addressed and is closed. Incident is identified by its incident ID.*

		14 => DNS *15 => DNF* *17 => VMX Penalty violation. <<Unused in AC35>>* *18 => Course Limits. <<Unused in AC35>>* *20 => Clear At Start* *21 => Not A Finish* *22 => OCS Penalty Complete. Incident is identified by its incident ID.* *23 => VMX Penalty payoff active. <<Unused in AC35>>* *24 => Boat on Boat Penalty payoff active. <<Unused in AC35>>* *25 => DNC – Yacht flagged as not competing in this race* *26 => Boundary – Yacht has crossed outside the boundary. Incident is identified by its incident ID.* *27 => BoundaryInside – Yacht has cleared back inside the boundary. Incident is identified by its incident ID.* *28 => IdOffender – Indicates against which yacht a protest has been raised. Incident is identified by its incident ID.* *29 => UmpOwn – Indicates Source ID UmpApp taking responsibility for this incident. Incident is identified by its incident ID.* *30 => UmpInitiated – Indicates Source ID of UmpApp that has initiated a penalty. Incident is identified by its incident ID.* *31 => PenaltyServed – the indicated yacht has dropped back behind the penalty distance to be served, and umps might consider clearing its penalty. Incident is identified by its incident ID.* *32 => IdPacer – identify the yacht against which a boat-on-boat penalty should be measured, e.g. for an ump-initiated penalty, or if the protesting boat receives the penalty. Incident is identified by its incident ID.* *the following events may accompany a penalty-related event to explicitly indicate the number of penalties currently carried by a yacht:* *100 => Zero penalties* *101 => One penalty* *102 => Two penalties* *103 => Three penalties* *104 => Four penalties* *105 => Five penalties* *106 => Six penalties* *107 => Seven penalties* *108 => Eight penalties* *109 => Nine penalties* *110 => Ten or more penalties*

Revision history:

version 2, 7/2015, adds a 4-byte unsigned integer IncidentID field, and several new event codes.

Yacht Event Code messages are sent from MDSS, UmpApp and ProApp to communicate events associated with penalties.

The following examples illustrate what messages are send under various penalty scenarios.

Example 1: Match race: Boat A crosses early (OCS) and Boat B starts within 3 seconds of the start time. There are no pre-start penalties.

1) After the Principal Race Officer makes the OCS call, an "*Over Early*" message will be sent for Boat A and their blue penalty light will turn on. A "*Clear At Start*" message will be send for Boat B. Stowe display penalty field on Boat A will show "OCS". The Stowe display penalty field on Boat B will show "---"
2) As soon as Boat B starts, a "*Boat on Boat Penalty payoff active"* message will be sent for Boat A. The Stowe display on Boat A will show "BoB".
3) Once Boat A completes their OCS penalty, the blue light on Boat A will go out and an "*OCS Penalty complete*" message will be sent for Boat A. The Stowe display penalty field on Boat A will show "---".

Example 2: Match race: Boat A crosses early (OCS). Boat B crosses the start line 3 seconds after the start time. There are no pre-start penalties.

1) After the Principal Race Officer makes the OCS call. An "Over Early" message will be sent for Boat A and their blue penalty light will turn on. A "Clear At Start" message will be send for Boat B. The Stowe display penalty field on Boat A will show "OCS". The Stowe display penalty field on Boat B will show "---"
2) At 3 seconds from the start, a "*VMX Penalty payoff active* " message will be sent to Boat A. The Stowe display penalty field on Boat A will show "VMX".
3) Once Boat A completes their OCS penalty, the blue light on Boat A will go out and a "*OCS Penalty complete*" message will be sent for Boat A. The Stowe display penalty field on Boat A will show "---"

Example 3: Match race: Boat A crosses early (OCS) and Boat B starts within 3 seconds of the start time. Boat A has a prestart penalty due to entering the pre-start area too early.

1) When Boat A gets their prestart penalty, a "*Entered pre-start area early* " message will be sent to Boat A and their blue penalty light will turn on. Boat A will also receive a "*VMX Penalty payoff active*" message. The Stowe display penalty field will show "VMX" on Boat A.
2) After the Principal Race Officer makes the OCS call, an "*Over Early*" message will be sent for Boat A and their blue penalty light stay on. A "*Clear At Start*" message will be send for Boat B. Stowe display penalty field on Boat A will show "OCS".
3) As soon as Boat B starts, a "*Boat on Boat Penalty payoff active* " message will be sent for Boat A. The Stowe display penalty field on Boat A will show "BoB".

4) Once Boat A completes their OCS BoB penalty, a “*OCS Penalty complete*” message will be sent to Boat A. A “*VMX Penalty payoff active*” message will be sent. The blue penalty light will remain on. The Stowe display penalty field on Boat A will show “VMX”
5) Once Boat A pays off their VMX penalty the blue penalty light will go out and a “*Penalty complete* message will be sent to Boat A. The Stowe display penalty field on Boat A will show “---”.

Example 4: 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 violation*” and a “*Boat on Boat Penalty payoff active* “ message will be sent for boat A. The Stowe display penalty field on boat A will show “BoB”.
2) When boat A goes outside the course limits and the umpire issues a penalty, a “*VMX Penalty violation*” message will be sent to Boat A. This tells Boat A that they received another penalty due to the course limits violation. Immediately a “*Boat on Boat Penalty payoff active* “ message will be sent to inform Boat A that they must pay off the penalty as a boat on boat penalty. The Stowe display penalty field on Boat A will show “BoB”
3) Once boat A completes their boat on boat penalty, the blue light on boat A will go out and a “*Penalty complete*” message will be sent for Boat A. The Stowe display penalty field on Boat A will show “---”

4.7 YachtActionCode

Message Type : 31

Message Size : 18 bytes

This message is sent by yacht when one of the 4 buttons are pressed on the yacht

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 reflected in an AcknowledgeMessage
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
}		

4.8 Chatter Text

Message Type : 36

Message Size : Variable

This message is used communicate the Chatter Text messages. The chatter text is an assembly of messages created by boats, race PRO, umpires, commentary, and machine generated messages. The purpose of the chatter text is to inform the audience of actions and events in the race.

ChatterText() {	Number of bytes	Data Type
MessageVersionNumber	1	unsigned byte Version number for this message. Set to 1 at this time
MessageType	1	unsigned byte 1 -> Yacht Generated Message 2 -> Umpire Generated Message 3 -> Race Officer Generated Message 4 -> Commentary 5 -> Machine Generated Message
Length	1	unsigned byte The number of bytes in the Text field.
Text	variable	UTF-8 text (like ASCII for 7-bit characters, but Unicode characters may be encoded in multiple bytes.) Not null-terminated. The text message.
}		

4.9 Boat Location

Message Type : 37

Message Size : 56 bytes

This message describes the location, attitude, and sensor data from the boat. Each message represents one GPS sample from the boat. Not all of the boats have wind and boat speed instrumentation on the boats. For example, the AC45 yachts do not have any boat speed, rudder, or wind instrumentation. The mark boats do not have any boat speed, or rudder instrumentation. Fields where data is not available will be set to zero.

BoatLocation () {	**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 2 -> Committee boat 3 -> Mark 4 -> Pin (unused) 5 -> Chase boat 6 -> Medical boat 7 -> Marshall boat 8 -> Umpire boat 9 -> Umpiring software application 10 -> Principal Race Officer (PRO) s/w app 11 -> Weather station 12 -> Helicopter 13 -> Data processing s/w app
Latitude	4	signed int -180 = 2^31 180 = 2^31
Longitude	4	signed int -180 = 2^31 180 = 2^31
Altitude	4	signed int cm relative to MSL
Heading	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000
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
BoatSpeed	2	unsigned short Speed in mm/sec
COG	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000 Course Over Ground. From GPS IMU or GPS Doppler
SOG	2	unsigned short Speed in mm/sec Speed Over Ground. From GPS IMU or GPS Doppler
ApparentWindSpeed	2	unsigned short Speed in mm/sec
ApparentWindAngle	2	signed short degrees -180 = -2^15 180 = 2^15 Wind over Starboard = positive
TrueWindSpeed	2	unsigned short Speed in mm/sec
TrueWindDirection	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000
TrueWindAngle	2	signed short degrees -180 = 2^15 180 = 2^15 0 = straight ahead. Positive to Starboard
CurrentDrift	2	unsigned short Speed in mm/sec
CurrentSet	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000
RudderAngle	2	signed short degrees. -180 = 2^15 180 = 2^15 0 = straight, Positive is rudder set to turn yacht to port
}		

Notes:

The latitude, longitude, and altitude of the boat is at 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;
  ```

4.10 Mark Rounding

Message Type : 38

Message Size : 21 bytes

This message is sent from MDSS to clients when a mark, start line, or finish line is rounded. The purpose of this message is to record the time when yachts cross marks.

MarkRounding() {	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 mark is rounded
AckNumber	2	Sequence number of message. This number is will be reflected in an AcknowledgeMessage. If the AckNumber is set to zero, client applications should not send an AcknowledgeMessage.
RaceID	4	Unique RaceID. RaceID will be zero if the yacht is not in a race.
SourceID	4	Source ID for the subject yacht
BoatStatus	1	Status of the boat in the race 0 => Unknown 1 => Racing 2 => DSQ 3 => Withdrawn
RoundingSide	1	0 => Unknown 1 => Port 2 => Starboard
MarkType	1	0 => Unknown 1 => Rounding Mark 2 => Gate (windward, leeward, start, finish, etc.)
MarkID	1	1 to 50 Mark Number 100 => Entry Limit Line 101 => Entry Line 102 => Race Start Startline 103 => Race Finishline 104 => Speed test start 105 => Speed test finish 106 => ClearStart

4.11 CourseWind

This structure is used to send periodic messages containing one or more aggregate course wind speed and direction reports. MDSS receives these messages on port 4925, and sends them on port 4926 to all clients configured in boats.xml with Flag=”CourseWind”

Message Type : 44

Message Size : Variable; 3 + (20 * # wind sources) bytes

CourseWind () {	**Number of bytes**	**Data Type**
MessageVersionNumber	1	unsigned byte Version Number of this message; Value = 1
SelectedWindID	1	unsigned char WindID of the CourseWind currently selected in MDSS
LoopCount	1	unsigned char Number records in loop
{		Loop
WindID	1	unsigned byte The logical identifier of this wind data source
Time	6	Number of milliseconds from Jan 1, 1970 UTC indicating when this source’s wind data is valid
RaceID	4	unsigned int ID of the race for which this wind is valid. 0 if the value is not race-specific
WindDirection	2	unsigned short North = 0x0000 East = 0x4000 South = 0x8000
WindSpeed	2	unsigned short Wind speed in mm/sec
BestUpwindAngle	2	unsigned short Optimum upwind sailing angle, expressed as positive true wind angle, typically around 45 degrees. 0 degrees = 0x0000 to 180 degrees = 0x8000
BestDownwindAngle	2	unsigned short Optimum downwind sailing angle, expressed as positive true wind angle, typically around 135 degrees. 0 degrees = 0x0000 to 180 degrees = 0x8000

Flags	1	unsigned byte, bit field (x = don't care) xxxxxx00 direction value not valid xxxxxx01 direction value valid (hex 01) xxxx00xx speed value not valid xxxx01xx speed value valid (hex 04) xx00xxxx best upwind angle value not valid xx01xxxx best upwind angle value valid (hex 10) 00xxxxxx best downwind angle value not valid 01xxxxxx best downwind value valid (hex 40)
}		
}		

MDSS collects and reports an aggregate course wind direction and/or speed from several sources. For example, the Umpiring application may use the wind reported by all the mark boats and combined into a single wind value by MDSS in the RaceStatus message, but it might also allow the umpires to select which wind reports to watch and to manually adjust the wind direction and sailing angles to better reflect the actual conditions on the water. The Umpire application uses this message to report that wind data back to MDSS, and then MDSS uses this message to forward that info on to clients like the LiveLine on-air graphics application. MDSS also stores these values in logs and a central database for use later.

There may be several people originating a course wind value – VirtualEye does its own adjustment of wind direction based on the trails of the yachts; umpires watching different races may manage separate winds for each race. MDSS collects, stores, and forwards all these values, each identified by a unique ID. Clients can agree to use a common source so that all realtime statistics - like time to mark, time behind, etc. – agree, or they can each decide to use whatever source is best suited for their immediate needs.

In addition to Direction and Speed, originators of CourseWind may also choose to specify the optimum upwind and/or downwind sailing angles, typically to match the observed trails of the yachts. Normally these values are determined by each client as needed from the current wind speed and the yachts' polars, but these fields provide a direct way to override those numbers.

MDSS caches all the CourseWind messages it receives, then retransmits them in a single combined message at some interval, typically 5 Hz. MDSS sends this message over port 4926 to all clients configured with the Flag="…CourseWind…" value in boats.xml. If a source reports its CourseWind more frequently than MDSS sends its combined CourseWind, there may be multiple wind reports from that source in one combined CourseWind message. If a source reports less frequently than MDSS retransmits, that source may not appear in some messages, or MDSS may insert duplicated values in each message. These duplicates could be recognized because they would share the same timestamp. MDSS may also filter or interpolate/extrapolate incoming values so there is exactly one unique value for a given source in each message it sends.

The MDSS Operator selects one of the primary wind sources to be the "official" wind, and this wind speed and direction is reported in the RaceStatus messages, and is also sent out from MDSS with WindID 10 in its combined CourseWind message. The value is filtered over time, so the operator may switch between wind sources without causing an abrupt jump in any animated graphics. If the MDSS operator selects a wind source that does not supply all the fields – speed, direction, tacking, gybing – those values are obtained from other sources, typically the MDSS averaged/manual wind.

Originators of aggregate wind data must choose their WindID so that it is unique among all the sources active at any one time. To help avoid conflicts, use these ranges:

WindID 10-19, MDSS-generated values.

> 10 = MDSS “official” filtered complete selected wind,
>
> 11 = MDSS averaged mark boat or manual wind only, with sailing angles from polars.

WindID 20-29, other services such as VirtualEye.

> 20 = Primary VirtualEye wind (typically direction-only)

WindID 30-39, Umpires.

> 31 = Primary Umpire wind (typically direction and tacking/gybing angle only)

WindID 40, Current Set and Drift

> 41 = Course set and drift.   Set angle is direction current is flowing (opposite convention of wind)

Version History

> v.1 – Original message version

4.12 Average Wind

This message is sent by MDSS to clients with the current true ground wind speed with various averaging periods. The first wind field contains the raw sample data. The raw data is collected from the signal boat using a Gill WindSonic wind sensor 10 meters above the water running at a 1 Hz sample rate.

The True Wind Speed is computed using the measured apparent wind angle and apparent wind speed corrected using GPS COG and SOG and boat heading to generate ground wind direction and speed.

The AveragePeriod fields describe the time period over which the averaging is done. The WindSpeed field is the result of the average.

The averaging filter is a rolling average (i.e. box car) filter. For example, a 30 second average is the simple arithmetic mean of the most recent 30 one second samples.

Message Type : 47

Message Size : 23 bytes

(YachtFeatureEnable) {	**Number of bytes**	**Data Type**
MessageVersionNumber	1	unsigned byte Version Number of this message; Value = 1
Time	6	Number of milliseconds from Jan 1, 1970 for when the data is valid
RawPeriod	2	Raw sample rate period Units are tenths of seconds 0 => Invalid field
RawSampleSpeed	2	unsigned short Speed in mm/sec Raw Wind Speed
Period2	2	Wind Speed Average Period Units are tenths of seconds 0 => Invalid field
Speed2	2	unsigned short Speed in mm/sec Wind Speed 2
Period3	2	Wind Speed Average Period Units are tenths of seconds 0 => Invalid field
Speed3	2	unsigned short Speed in mm/sec Wind Speed 3
Period4	2	Wind Speed Average Period Units are tenths of seconds 0 => Invalid field
Speed4	2	unsigned short Speed in mm/sec Wind Speed 4
}		

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 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
<?xml version="1.0" encoding="utf-8"?>
<RegattaConfig>
  <RegattaID>3</RegattaID>
  <RegattaName>New Zealand Test</RegattaName>
  <CourseName>North Head</CourseName>
  <CentralLatitude>-36.82791529</CentralLatitude>
  <CentralLongitude>174.81218919</CentralLongitude>
  <CentralAltitude>0.00</CentralAltitude>
  <UtcOffset>12</UtcOffset>
  <MagneticVariation>14.1</MagneticVariation>
</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 center of the race course.

CentralLongitude – Longitude in the vicinity of the center of the race course

CentralAltitude – Altitude in the vicinity of the center of the race course

UtcOffset – Offset from UTC for the race.

MagneticVariation – Local Magnetic Variation

5.2 Race.xml

The race.xml file describes specific details for a race.   It contains the course boundaries, race participants, race type, and the course description.   A race may have more than one race.xml files associated with the race.   This file is generated by an application operated by the Principal Race Officer. Each time the PRO makes a change to the race, such as course boundaries, a new race.xml file will be committed and sent in the log file.   It is typical that the course limits will change multiple times before and during a race.   It is not typical that the participants or race course description will change during the race. In the time leading up to the start, the race start time may change.

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

```xml
<?xml version="1.0" encoding="utf-8"?>
<Race>
  <RaceID>11080703</RaceID>
  <RaceType>Match</RaceType>
  <CreationTimeDate>2011-08-06T13:25:00-0000</CreationTimeDate >
  <RaceStartTime Time="2011-08-06T13:30:00-0700" Postpone="false" />
  <Participants>
    <Yacht SourceID="107" Entry="Port" />
    <Yacht SourceID="108" Entry="Stbd" />
  </Participants>
  <Course>
    <CompoundMark CompoundMarkID="1" Name="StartLine">
      <Mark SeqID="1" Name="PRO" TargetLat="-36.83" TargetLng="174.83" SourceID="101" />
      <Mark SeqID="2" Name="PIN" TargetLat="-36.84" TargetLng="174.81" SourceID="102" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="2" Name="M1">
      <Mark Name="M1" TargetLat="-36.63566590" TargetLng="174.88543944" SourceID="103" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="3" Name="M2">
      <Mark Name="M2" TargetLat="-36.83" TargetLng="174.80" SourceID="102" />
    </CompoundMark>
    <CompoundMark CompoundMarkID="4" Name="Gate">
      <Mark SeqID="1" Name="G1" TargetLat="-36.63566590" TargetLng="174.97205159" SourceID="104" />
      <Mark SeqID="2" Name="G2" TargetLat="-36.64566590" TargetLng="174.98205159" SourceID="105" />
    </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="Stbd" ZoneSize="6" />
    <Corner SeqID="4" CompoundMarkID="4" Rounding="PS" ZoneSize="6" />
    <Corner SeqID="5" CompoundMarkID="1" Rounding="SP" ZoneSize="3"/>
  </CompoundMarkSequence>
  <CourseLimit>
    <Limit SeqID="1" Lat="-36.8325" Lon="174.8325"/>
    <Limit SeqID="2" Lat="-36.82883" Lon="174.81983"/>
    <Limit SeqID="3" Lat="-36.82067" Lon="174.81983"/>
    <Limit SeqID="4" Lat="-36.811" Lon="174.8265"/>
    <Limit SeqID="5" Lat="-36.81033" Lon="174.83833"/>
    <Limit SeqID="6" Lat="-36.81533" Lon="174.8525"/>
    <Limit SeqID="7" Lat="-36.81533" Lon="174.86733"/>
    <Limit SeqID="8" Lat="-36.81633" Lon="174.88217"/>
    <Limit SeqID="9" Lat="-36.83383" Lon="174.87117"/>
    <Limit SeqID="10" Lat="-36.83417" Lon="174.84767"/>
  </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 starft. '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.

CreationTimeDate – UTC Time when this race.xml was created.

Participants – This section describes what race yachts are in the race. For match races, the Entry tag describes what side of the course the boats start on (Port or Stbd). The SourceID describes what boat is participating

Course – This section describes the race course. Within this tag, there is a sequence of 'CompoundMark's. Each CompoundMark has a Name used to describe the compound mark. The is a symbolic name and does not necessarily describe a particular boat. A compound mark is either a single mark or a pair of marks that make up the starting line or the gate. Each boat in a CompoundMark has a 'Name' tag. The name is the name that written on the stern of the boat. The 'CompountMarkID' is used as a reference in the CompoundMarkSequence tag. Marks within the CompoundMark tag can optionally have a TargetLat and TargetLng tag to describe the target location for a mark boat.

When a CompoundMark has two boats (start line, gate, finish line), the TargetLat and TargetLon will be the same.

CompoundMarkSequence – This section describes the sequence of mark roundings. The SeqID tag describes the order. The CompoundMarkID tag is associated with the CompounMarkID tag in the Course section. The Rounding tag describes how the mark is passed. Valid values are "Port", Stbd", "SP", or "PS". Port means that the mark should be rounded to port, Stbd means that the mark should be rounded to starboard. SP means that the boat within the compound mark with the SeqID of 1 should be rounded to starboard and the boat within the compound mark with the SeqID of 2 should be rounded to port. PS means the opposite of SP.

The CompoundMarkSequence is slightly different in a Match race when an entry boat is utilized. The first CompoundMarkSequence will reference a single boat with and the rounding will be Port or Starboard. This is the entry mark. The second CompoundMarkSequence will be the startline.

ZoneSize – Size of the zone around a mark in boatlengths.

CourseLimit – This section describes the course limits. The sequence of 'Limit' tags must be clockwise.

5.3 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.

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. The next section, Boats, describe the physical and functional attributes of a boat or application.

```
<?xml version="1.0" encoding="utf-8"?>
<BoatConfig>
  <Modified>2012-05-17T07:49:40+0200</Modified>
  <Version>12</Version>
  <Settings>
    <RaceBoatType Type="AC45" />
    <BoatDimension BoatLength="14.019" HullLength="13.449" />
    <ZoneSize MarkZoneSize="40.347" CourseZoneSize="40.347" />
    <ZoneLimits Limit1="200" Limit2="100" Limit3="40.347" Limit4="0" Limit5="-100" />
  </Settings>
  <BoatShapes>
    <BoatShape ShapeID="0">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-2.659" />
        <Vtx Seq="2" Y="18.359" X="-2.659" />
        <Vtx Seq="3" Y="18.359" X="2.659" />
        <Vtx Seq="4" Y="0" X="2.659" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="1">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-1.278" />
        <Vtx Seq="2" Y="8.876" X="-1.278" />
        <Vtx Seq="3" Y="8.876" X="1.278" />
        <Vtx Seq="4" Y="0" X="1.278" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="2">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-1.1" />
        <Vtx Seq="2" Y="8.3" X="-1.1" />
        <Vtx Seq="3" Y="8.3" X="1.1" />
        <Vtx Seq="4" Y="0" X="1.1" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="3">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-0.75" />
        <Vtx Seq="2" Y="3" X="-0.75" />
        <Vtx Seq="3" Y="3" X="0.75" />
        <Vtx Seq="4" Y="0" X="0.75" />
      </Vertices>
    </BoatShape>
    <BoatShape ShapeID="4">
      <Vertices>
        <Vtx Seq="1" Y="0" X="-3.46" />
        <Vtx Seq="2" Y="13.449" X="-3.46" />
        <Vtx Seq="3" Y="14.019" X="0" />
        <Vtx Seq="4" Y="13.449" X="3.46" />
        <Vtx Seq="5" Y="0" X="3.46" />
      </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="5" />
  </BoatShapes>
  <Boats>
    <Boat Type="RC" SourceID="121" ShapeID="0" HullNum="RG01" StoweName="PRO" ShortName="PRO"
BoatName="Regardless">
      <GPSposition Z="6.840" Y="7.800" X="0.000" />
      <FlagPosition Z="0.000" Y="7.800" X="0.000" />
    </Boat>
    <Boat Type="Mark" SourceID="122" ShapeID="1" HullNum="LC05" StoweName="CON" ShortName="Constellation"
BoatName="Constellation">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat Type="Mark" SourceID="123" ShapeID="1" HullNum="LC04" StoweName="MIS" ShortName="Mischief"
BoatName="Mischief">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat Type="Mark" SourceID="124" ShapeID="1" HullNum="LC03" ShortName="Atalanta" BoatName="Atalanta">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat SourceID="125" ShapeID="1" StoweName="VOL" HullNum="LC01" ShortName="Volunteer"
BoatName="Volunteer">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat Type="Mark" SourceID="126" ShapeID="1" HullNum="LC13" StoweName="MS2" ShortName="Defender"
BoatName="Defender">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat Type="Mark" SourceID="128" ShapeID="1" HullNum="LC01" ShortName="Shamrock" BoatName="Shamrock">
      <GPSposition Z="5.334" Y="3.804" X="0.000" />
      <FlagPosition Z="0.000" Y="3.426" X="0.000" />
    </Boat>
    <Boat Type="Yacht" SourceID="101" ShapeID="4" HullNum="AC4501" StoweName="KOR" ShortName="TEAM KOREA"
BoatName="TEAM KOREA" Country="KOR">
      <GPSposition Z="1.738" Y="0.625" X="0.001" />
      <MastTop Z="21.496" Y="4.233" X="0.000" />
    </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 AC72 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 Stowe 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. Revision History

Revision	Date	Editor	Changes Made
1.0	27 Sept 2011	Ken Milnes	Initial Draft
1.1	14 Oct 2011	Ken Milnes	Add Mark Rounding and Chatter text messages. Add additional eventId codes to YachtEventCode message.
1.2	22 Oct 2011	Ken Milnes	Add additional documentation (Port numbers)
1.3	26 Oct 2011	Ken Milnes	Added additional text to message descriptions.
1.4	12 Dec 2011	Ken Milnes	Fix documentation errors. Add documentation for YachtActionCode and RaceStartStatus messages
1.5	23 Mar 2012	Ken Milnes	Fix documentation omissions in Mark Rounding and Boat Location messages.
1.6	4 June 2012	Ken Milnes	Add CourseWind documentation. Clarify length meaning and text format in ChatterText, DisplayText, and SendText messages. Add the definitions of Boat Status field values to the RaceStatus message.
1.7	27 March 2013	Ken Milnes	Add additional documentation and codes to Yacht Event Code message. Add Current set/drift to CourseWind documentation Add Settings section in Boats.xml
1.8	27 June 2013	Ken Milnes	Add Average Wind Message
1.9	2 July 2015	Ken Milnes	Update for AC35
1.10	15 Feb 2016	Ken Milnes Tim Heidmann	Add revision to Yacht Event Code. Add additional fields.
1.11	26 Feb 2016	Ken Milnes Tim Heidmann	Revise YachtActionCode message. Add IncidentID field.