Shotlog Software

ICD

- Network Protocol v1.2.3-

1. Network Specification

- Protocol: UDP
- Host: 255.255.255.255 (Default; Broadcast or Multicast capable)
- Port: 37337 (Default; Settable in config files)

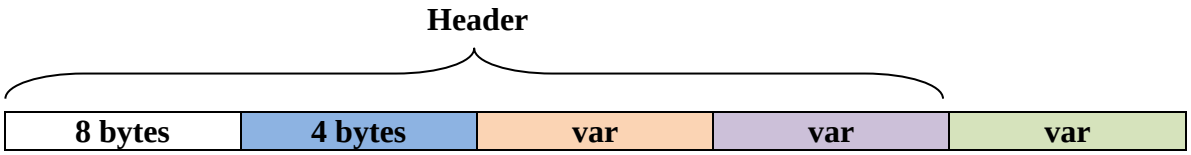
2. Basic message features

- The payload is a UTF-8 encoded string
- UDP Datagram payload is less than 512 bytes
- The UDP Datagram thus is not fragmented
- The data string consists of string fields separated by the ; character
- If in a file, every message is delimited by CRLF
- The number of fields depends on the message type
- All numbers are transmitted as readable strings and have to be converted back to their original data type as stated in the tables
- Sending any event except PlayToTime/JumpToTime should NOT change the displayed timing
- The ACMIPositionPackage is a BINARY package with no limiters nor strings

3. Message Example

```
1379635200000;7;Chaff;Rebel-2;1; \r\n
1379684400000;9;Regenerated;Vodka-1;2; \r\n
1379684915000;3;Remarks;None;3;Bambo in Green Airspace\r\n
1379685005000;13;ShotTime;Puma-1;4;Mirage 2000 -5;272;27.0;Vodka-2;SU-
27 SM;MICA EM;3;A\r\n
1379685055000;14;ShotImpactTime;Puma-1;5;Mirage 2000 -5;272;27.0;Vodka-
2;SU-27 SM;MICA EM;3;A\r\n
1379685528000;8;Flare;Polka-1;6; \r\n
```

4. Message Format



No	Data	Orig. Type	Size	Description
0	Timestamp	long64	8 bytes	Milliseconds since 01.01.1970
1	Event Type	int32 (enum)	4 bytes	See Table
2	Event Name	string	Var <20	See Table
3	Event Source	string	Var < 10	Message Source (Callsign)
4-x	Data	Variable	Variable	Additional Data (Multiple fields)

5. Event Type Table (Informational Messages)

Event Type	Event Name	Field 4	Field 5+ (Data)
0	None	Event ID	
1	Takeoff	Event ID	
2	Landed	Event ID	
3	Remarks	Event ID	1x String (Remark)
4	Safety	Event ID	1x String (Remark)
5	Terminate	Event ID	1x String (Remark)
6	KIO	Event ID	1x String (Remark)
7	Chaff	Event ID	
8	Flare	Event ID	
9	Regenerated	Event ID	
10	Killed	Event ID	
11	MissileBack	Event ID	1x String (Missile Name)
12	Reload	Event ID	
13	ShotTime	Shot ID	10 additional string fields (see table)
14	ShotImpactTime	Shot ID	10 additional string fields (see table)
15	GroundAbort	Event ID	
16	AirAbort	Event ID	
17	MissionFlown	Event ID	
18	EventStart	Event ID	
19	EventEnd	Event ID	
20	SystemEmitting	Event ID	
21	SystemSilent	Event ID	
22	StaffRemarks	Event ID	
23	SystemActive	Event ID	
24	System Inactive	Event ID	

These messages are for informational purposes only and do not require a reply. We use the message timestamp to represent the time of the events.

6. Event Type Table (Control Messages)

Event Type	Event Name	Field 4+ (Data)	Reply
100	ACMISpeedChange	Speed multiplier (as string)	200
101	ACMIPlay		200
102	ACMIPause		200
103	ACMIPlayToTime	No (Uses timestamp)	200
104	ACMIJumpToTime	No (Uses timestamp)	200
105	ACMIRequestTracks	4 additional fields (see table)	203
106	ACMIRequestPosition	2 additional fields (see table)	203
107	ACMIZoomRectangle	5 additional fields (see table)	201
108	ACMIZoomRectangle2	4 additional fields (see table)	201
109	ACMIZoomLocation	4 additional fields (see table)	201
110	ACMIDrawLockline	6 additional fields (see table)	
111	ACMIDrawLockline2	5 additional fields (see table)	
112	ACMIRequestStatus		200, 201, 202
113	ACMIRequestPlaybackStatus		200
114	ACMIRequestZoomStatus		201
115	ACMIRequestTimeStatus		202
116	ACMIStartStreaming		Start 204
117	ACMIStopStreaming		Stop 204
118	ShotlogRequestFlags	1 additional field	205
119	ShotlogRequestPlatformNames		206
120	ShotlogRequestDMPINames		207
121	ACMIDeleteShot	1 additional field	
122	ACMIDeleteEvent	1 additional field	
123	ShotlogRequestPlatformInfo	1 additional field	208
124	ShotlogRequestAllEvents		0-24

These messages are used to remote control the ACMI and exchange data. They require a response by the ACMI/Shotlog Software as indicated.

Messages 112 to 115 can be used as a PING command to make sure the ACMI server is alive. Thus they can be expected to be requested periodically.

Except the ACMIPlayToTime and the ACMIJumpToTime message the timestamp in the message header indicates the time when the message was sent.

7. Event Type Table (Response Messages)

Event Type	Event Name	Field 4+ (Data)
200	ACMIPlaybackStatus	Returns 1x int32 (Current speed multiplier)
201	ACMIZoomStatus	Returns current center and scale (see table)
202	ACMITimeStatus	Returns earliest/latest time (see table)
203	ACMITrackData	Returns 5 additional fields (see table)
204	ACMIPositionPackage	BINARY PACKAGE! Four magic bytes: !POS
205	ShotlogFlagStatus	Returns track flags at specific time
206	ShotlogPlatformNames	Returns Aircraft Names
207	ShotlogDMPINames	Returns Target Names
208	ShotlogPlatformInfo	Returns platform info

Except the ACMITimeStatus, ACMITrackData and ShotlogFlagStatus messages the timestamp in the message header indicates the time when the message was sent.

8. ShotTime/ ShotImpactTime Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	A/C Type	string	Var<20	Readable aircraft type
5	Shooter Status	string		“Alive” / ”Dead”
6	Tgt BE Radial	int32	4 bytes	BE Radial in degree (0-360)
7	Tgt BE Range	double	8 bytes	BE Range in nautical miles
8	Tgt Name	string	Var<10	Target callsign / name
9	Tgt Type	string	Var<20	Target type / DMPI for ground targets
10	Tgt Status	string		“Alive” / “Dead”
11	Weapon Name	string	Var<20	Readable weapon name
12	Weapon Class	char	1 byte	See weapon class table
13	Weapon Type	string	variable	See weapon type table

- **Weapon Class Table**

  - Single character depicting the class of weapon

Character	Class
“1”	Fox 1
“2”	Fox 2
“3”	Fox 3
“G”	Gun
“5”	AA-ARM
“L”	LGB
“P”	GPS
“D”	Dual Mode

Character	Class
“B”	Dumb Bomb
“R”	HARM
“S”	Standoff
“A”	AAA
“M”	SAM
“C”	Camera
“J”	Jamming
“I”	INS

- **Weapon Type Table**

  - Flag based. If the flag is present inside the string, the weapon can target the respective target type.
  - The flags inside the string are not sorted and not in any particular order.
  - Example: “AGS” -> Weapon can target air/ground and sea targets.

Character	Class
“A”	Airborne
“G”	Ground
“S”	Ship

9. ACMISpeedChange Package

This message informs the ACMI software to change its speed to the given speed multiplier. It WILL NOT change the play/pause status. However, if the speed is zero, the ACMI will not increase the time. The ACMI software will reply with an ACMIPlaybackStatus with the current speed multiplier that should be the same as the one sent to the ACMI software using the ACMISpeedChange message.

The following speeds are available at a minimum: -200, -100, -50, -20, -10, -4, -2, -1, 0, 1, 2, 4, 10, 20, 50, 100, 200. However, if other selections are made, the ACMI should do its best to set a speed close to the sent value.

10. ACMIPlay Package

This message informs the ACMI software to play at the speed multiplier currently set in the ACMI. The ACMI software will reply with a ACMIPlaybackStatus with the current speed multiplier.

11. ACMIPause Package

This message will stop the ACMI playback. The ACMI software will reply with a ACMIPlaybackStatus with the speed multiplier of zero. This message will not change the internal playback speed.

12. ACMIPlayToTime Package

This message instructs the ACMI software to play to the timestamp given in the message header, using the speed multiplier currently set in the ACMI (if possible). If the time is in the other direction, the ACMI will multiply the current speed multiplier by (-1). The ACMI software will reply with a ACMIPlaybackStatus with the current speed multiplier.

13. ACMIJumpToTime Package

This message instructs the ACMI software to jump to the timestamp given in the message header. The ACMI software will reply with a ACMIPlaybackStatus with the speed multiplier of zero.

14. ACMIRequestTracks Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Latitude/Radial	double	8 bytes	Latitude or radial based on field 4
6	Longitude/Range	double	8 bytes	Longitude or range based on field 4
7	Circle Range	double	8 bytes	Request all tracks inside range of loc

This message will inform the ACMI software to return the tracks found at the given position with a circle range, and at the current time of the ACMI software. DO NOT use the message header timestamp to get the time.
This package will be replied to by sending the following package with every track inside the circle at the timestamp specified (best interpolated position). If bullseye position is requested, the reply will be supplied in bullseye.

Lat/Lon are stored in WGS-84, decimals of degree, north and east are positive.
Example (Lat/Lon): -35.185;125.625

Bullseye is stored in decimals of degrees and nautical miles.
Example (Degree/Range): 225.23;32.4

15. ACMITrackData Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Track Count	int32	4 bytes	Number of tracks in this package
5	Location type	String	2 bytes	Either “BE” or “LL”
6	Track Callsign	String	Var <10	Callsign of this track
7	Latitude/Radial	double	8 bytes	Latitude or radial based on field 4
8	Longitude/Range	double	8 bytes	Longitude or range based on field 4
9	Heading	double	8 bytes	Interpolated track heading
10	Altitude	double	8 bytes	Interpolated track altitude
The last 5 fields are repeated for every track in the package. If there are more than 10 tracks consider sending a second package to avoid fragmentation.				

Heading is stored in decimals of degrees between 0 and 360. Altitude is stored in feet.

You should not send more than 10 tracks in one package to avoid going over 512 bytes (however the shotlog will accept any number of tracks).

The current time of the ACMI software that corresponds to the returned track data is returned in the timestamp in the message header.

16. ACMIZoomRectangle Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Latitude/Radial 1	double	8 bytes	Latitude or radial based on field 4
6	Longitude/Range 1	double	8 bytes	Longitude or range based on field 4
7	Latitude/Radial 2	double	8 bytes	Latitude or radial based on field 4
8	Longitude/Range 2	double	8 bytes	Longitude or range based on field 4

The two locations in this package are of two aircraft that are shooting at each other.

The ACMI shall zoom in on the rectangle described by these two corner locations, while making sure the aircraft have a generous border on every side of at least 15% screenspace.

This message has to be replied to by sending back the new center of screen coordinate and the set zoom (map scale), so that the zoom setting can then be tweaked by issuing more commands.

17. ACMIZoomRectangle2 Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Shooter Callsign	String	Var <10	Callsign to get lat/lon position from
6	Latitude/Radial 2	double	8 bytes	Latitude or radial based on field 4
7	Longitude/Range 2	double	8 bytes	Longitude or range based on field 4

The two locations in this package are of two aircraft that are shooting at each other. The first aircraft is selected by callsign in this message.

The current time of the ACMI software will be used, thus the ACMI gets the aircraft position at the current time. DO NOT use the message header timestamp to get the time.

The ACMI shall zoom in on the rectangle described by these two corner locations, while making sure the aircraft have a generous border on every side of at least 15% screenspace.

This message has to be replied to by sending back the new center of screen coordinate and the set zoom (map scale), so that the zoom setting can then be tweaked by issuing more commands.

18. ACMIZoomLocation Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Latitude/Radial	double	8 bytes	Latitude or radial based on field 4
6	Longitude/Range	double	8 bytes	Longitude or range based on field 4
7	New map scale	double	8 bytes	Zoom of the map

The ACMI shall set the map center to the described location and zoom to the new map scale. **Both of these should happen smoothly** if the location is already on screen and otherwise instantly.

The graining of "map scale" has to be in a way that smooth scaling is possible (so not 20x, 40x, 100x zoom but also all the value between).

This message has to be replied to by sending back the new center of screen coordinate and the set zoom (map scale), so that the zoom setting can then be tweaked by issuing more commands.

19. ACMIZoomStatus Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Latitude	double	8 bytes	Latitude of map center
5	Longitude	double	8 bytes	Longitude of map center
6	Map scale	double	8 bytes	Current scale of the map

This package is sent by the ACMI to respond to change requests. The graining of "map scale" has to be in a way that smooth scaling is possible.

20. ACMIDrawLockline Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either "BE" or "LL"
5	Latitude/Radial 1	double	8 bytes	Latitude or radial based on field 4
6	Longitude/Range 1	double	8 bytes	Longitude or range based on field 4
7	Latitude/Radial 2	double	8 bytes	Latitude or radial based on field 4
8	Longitude/Range 2	double	8 bytes	Longitude or range based on field 4
9	Color Index	Int32	4 bytes	Color Index as below

Location 1 contains the shooter location. Location 2 contains the target location. There does not have to be an actual track at that location, the drawing can also be used to mark a location or to show where a tgt pod is looking.

The ACMI shall draw a clearly visible marker at the target location (below the aircraft) using the color as specified in the message.

The ACMI shall further draw a clearly visible line of the same color from location 1 to location 2. Location 1 and location 2 can be the same location, which would result in the drawing of a dot.

This message will NOT change the map center, nor the map scale. It would usually be used after issuing a **ACMIZoomRectangle** command, but can also be used to mark locations around the map for other purposes.

The drawing shall be temporary. That means any later **ACMIZoomLocation, ACMIZoomRectangle** or **ACMIDrawLockline** command will remove this lockline

from the ACMI. Additionally the lockline is only valid at the time for which it was issued. If the time changes by more than a few seconds (exact graining depends on the graining of the underlying ACMI time data), the lockline should be removed.

Color Index Table

Color Index	Resulting Color
0	Affiliation 1: Own
1	Affiliation 2: Enemy
2	Affiliation 3: Neutral
3	Affiliation 4: Unknown
4	Blue
5	Red
6	Yellow
7	Orange
8	Green
9	Cyan
10	Magenta

21. ACMIDrawLockline2 Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Shooter Callsign	string	Var <10	Callsign to get lat/lon position from
6	Latitude/Radial 2	double	8 bytes	Latitude or radial based on field 4
7	Longitude/Range 2	double	8 bytes	Longitude or range based on field 4
8	Color Index	Int32	4 bytes	Color Index as in table above

Location 1 is taken from the respective shooter aircraft. Location 2 contains the target location. There does not have to be an actual track at that location, the drawing can also be used to mark a location or to show where a tgt pod is looking.

The current time of the ACMI software will be used, thus the ACMI gets the aircraft position at the current time. DO NOT use the message header timestamp to get the time.

The ACMI shall draw a clearly visible marker at the target location (below the aircraft) using the shooter aircrafts color (or the supplied one if the shooter aircraft is not found).

The ACMI shall further draw a clearly visible line of the same color from location 1 to location 2. Location 1 and location 2 can be the same location, which would result in the drawing of a dot.

This message will NOT change the map center, nor the map scale. It would usually be used after issuing a **ACMIZoomRectangle** command, but can also be used to mark locations around the map for other purposes.

The drawing shall be temporary. That means any later **ACMIZoomLocation, ACMIZoomRectangle** or **ACMIDrawLockline** command will remove this lockline from the ACMI. Additionally the lockline is only valid at the time for which it was issued. If the time changes by more than a few seconds (exact graining depends on the graining of the underlying ACMI time data), the lockline should be removed.

22. ACMIRequestPosition Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Location type	string	2 bytes	Either “BE” or “LL”
5	Aircraft Callsign	string	Var<10	Callsign to get position from

This package will be replied to by sending a ACMITrackData package with the position of the requested callsign at the timestamp specified (best interpolated position). If bullseye position is requested, the reply will be supplied in bullseye.

The current time of the ACMI software will be used, thus the ACMI gets the aircraft position at the current time. DO NOT use the message header timestamp to get the time.

Lat/Lon are stored in WGS-84, decimals of degree, north and east are positive.
Example (Lat/Lon): -35.185;125.625

Bullseye is stored in decimals of degrees and nautical miles.
Example (Degree/Range): 225.23;32.4

23. ACMITimeStatus Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Earliest data time	double	8 bytes	Earliest time of ACMI data
5	Latest data time	double	8 bytes	Latest time of ACMI data

Time is in Milliseconds since 01.01.1970. The timestamp in the header contains the current time displayed in the ACMI.

24. ACMIPositionPackage

Byte	Data	Var Type	Size	Description
0-3	Magic Bytes	String	4 bytes	Always: !POS
4-11	Timestamp	long64	8 bytes	Milliseconds since 01.01.1970
12-19	Latitude	Double	8 bytes	Latitude
20-27	Longitude	Double	8 bytes	Longitude
28-31	Heading	Float	4 bytes	Heading
32-35	Altitude	Float	4 bytes	Altitude
36-39	Speed	Float	4 bytes	Speed
40-49	Callsign / ID	string	10 bytes	ID of this track (null terminated)
50-59	A/C type	string	10 bytes	A/C type of track (null terminated)

Once ACMIStartStreaming is issued any incoming realtime track data will be streamed to the network via the above package. This is a BINARY package and does not feature strings nor limiters.

ACMIStopStreaming will stop these messages.

25. ShotlogRequestFlags Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Track Callsign	string	Var < 10	Callsign of track

The timestamp in the header contains the time at which the tracks status is requested.

26. ShotlogFlagStatus Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Track Callsign	string	Var < 10	Callsign of track
5	Flags	Int32	4 bytes	Flags of aircraft status

The timestamp in the header contains the time at which the tracks status is returned. The flag value contains the following bits:

Value	Flag
1	Active
2	Emitting
4	Damaged
8	Destroyed
16	Overkill

27. ShotlogPlatformNames Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	List of Callsigns	string	Var	Callsigns separated by '|'

28. ShotlogDMPINames Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	List of DMPIs	string	Var	DMPI names separated by '|'

29. ShotlogRequestPlatformInfo

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Track Callsign	string	Var < 10	Callsign of track

30. ShotlogPlatformInfo Package

No	Data	Orig. Type	Size	Description
0 – 3	Message Header			
4	Callsign	string	Var<10	Callsign
5	Aircraft Type	string	Var<20	Aircraft Type
6	Platform Class	Int32	4 bytes	Platform Class
7	Color	Int32	4 bytes	Color Index

This protocol is subject to change and will be extended as needed. Please report errors or suggestions to the below email addresses.

Ralf "Minime" Mengwasser

TacDev
http://www.tacdev.eu
support@tacdev.eu