


SCANTER Track Management Protocol

Proprietary and intellectual rights of Terma A/S, Denmark, are involved in the subject-matter of this material and all manufacturing, reproduction, use, disclosure, and sales rights pertaining to such subject-matter are expressly reserved. This material is submitted for a specific purpose as agreed in writing, and the recipient by accepting this material agrees that this material will not be used, copied, or reproduced in whole or in part nor its contents (or any part thereof) revealed in any manner or to any third party, except own staff, to meet the purpose for which it was submitted and subject to the terms of the written agreement.

PREP NHW	CHKD CHR	Terma A/S Headquarters Hovmarken 4 DK-8520 Lystrup Denmark 		
APVD MUK	CM NHE			
DATE OF INITIAL RELEASE 2006-03-01	DATE OF THIS RELEASE 2011-06-17			
TITLE SCANTER Track Management Protocol		DOCUMENT NO. 303949 SI	REV F	PAGE 1 of 71

Record of Changes

Description	Rev	Date
Released	A	010306
Added AAZ zone handling		
Changed protocol revision		
Fixed chapter links and minor spelling changes	B	180907
Added set protocol revision for both interfaces	C	241110
Updated Track Control Interface to version 1.2		
Updated Track Data Interface to have a selectable version 2.0 message format.		
The new version 2.0 message format includes adding of associated plot info.		
Added Scanter 5xxx/6xxx track control interface and track data interface ports		
Corrected Track Data Interface protocol revision and port numbers	D	140111
Corrected protocol revision handling and increased version.	E	080411
Added special SCANTER 4xxxx Series section	F	See page 1

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	2 of 71

Contents

1	INTRODUCTION.....	6
1.1	Scope.....	6
2	DEFINITIONS	6
3	SCANTER 2XXX/5XXX/6XXX SERIES.....	7
3.1	Interface protocol revisions.....	7
3.2	Client and Server Messages Summary	7
3.3	Message syntax	9
	3.3.1 Client Syntax	9
	3.3.2 Server Syntax.....	9
3.4	Message transfer description.....	9
	3.4.1 Connection	9
	3.4.1.1 Maintaining the Link	9
	3.4.2 Disconnection.....	10
	3.4.3 Error Detection and Handling	10
3.5	Common Messages	10
	3.5.1 Common Messages from the Client.....	10
	3.5.1.1 The GET Command.....	10
	3.5.1.2 The SET Command.....	11
	3.5.1.3 The PING Command.....	11
	3.5.1.4 The BYE Command.....	11
	3.5.2 Common Messages from the Server	11
	3.5.2.1 The CURRVAL Message	11
	3.5.2.2 The MSGERR Message.....	11
	3.5.2.3 The PONG Message.....	13
	3.5.2.4 The BYE Message.....	14
3.6	Track Control Interface.....	14
	3.6.1 Messages from the Client	14
	3.6.1.1 The GET Command.....	14
	3.6.1.2 The SET Command.....	15
	3.6.1.3 The TRACKCREATE Command.....	15
	3.6.2 The TRACKDELETE Command	16
	3.6.3 The TRACKSWAP Command.....	17
	3.6.3.2 The TRACKSELECT Command.....	18
	3.6.3.3 The TRACKMOVE Command	19
	3.6.3.4 The AAZCREATE Command	20
	3.6.3.5 The AAZDELETE Command	22
	3.6.3.6 The NAAZCREATE Command.....	22
	3.6.3.7 The NAAZDELETE Command	24
	3.6.3.8 The NTZCREATE Command	24
	3.6.3.9 The NTZDELETE Command	26
	3.6.3.10 The ATONCREATE Command.....	26
	3.6.3.11 The ATONDELETE Command.....	28
	3.6.4 Messages from the Server.....	29
	3.6.4.1 The AAZCREATED Message.....	29
	3.6.4.2 The AAZDELETED Message.....	30
	3.6.4.3 The NAAZCREATED Message.....	30

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	3 of 71

3.6.4.4	The NAAZDELETED Message	31
3.6.4.5	The NTZCREATED Message	32
3.6.4.6	The NTZDELETED Message	33
3.6.4.7	The ATONCREATED Message	33
3.6.4.8	The ATONDELETED Message	34
3.7	Track Data Interface	35
3.7.1	Messages from the Client	35
3.7.1.1	The GET Command	35
3.7.1.2	The SET Command	35
3.7.2	Messages from the Server	35
3.7.2.1	Track Message Version 1.1	35
3.7.2.2	Track Message Version 2.0	37
3.7.2.3	Lost Tracks	39
3.7.2.4	Air Tracks	39
3.7.2.5	AtoN Tracks	39
4	SCANTER 4XXX SERIES	40
4.1	Interface protocol revisions	40
4.2	Client and Server Messages Summary	40
4.3	Message syntax	42
4.3.1	Client Syntax	42
4.3.2	Server Syntax	42
4.4	Message transfer description	42
4.4.1	Connection	42
4.4.1.1	Maintaining the Link	42
4.4.2	Disconnection	43
4.4.3	Error Detection and Handling	43
4.5	Common Messages	43
4.5.1	Common Messages from the Client	43
4.5.1.1	The GET Command	43
4.5.1.2	The PING Command	43
4.5.1.3	The BYE Command	44
4.5.2	Common Messages from the Server	44
4.5.2.1	The CURRVAL Message	44
4.5.2.2	The MSGERR Message	44
4.5.2.3	The PONG Message	46
4.5.2.4	The BYE Message	46
4.6	Track Control Interface	47
4.6.1	Messages from the Client	47
4.6.1.1	The GET Command	47
4.6.1.2	The TRACKCREATE Command	48
4.6.1.3	The TRACKDELETE Command	49
4.6.1.4	The TRACKSWAP Command	49
4.6.1.5	The TRACKSELECT Command	50
4.6.1.6	The TRACKMOVE Command	51
4.6.1.7	The AAZCREATE Command	52
4.6.1.8	The AAZDELETE Command	54
4.6.1.9	The NAAZCREATE Command	55
4.6.1.10	The NAAZDELETE Command	56
4.6.1.11	The NTZCREATE Command	57
4.6.1.12	Example	58
4.6.1.13	The NTZDELETE Command	58

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	4 OF 71

4.6.1.14	The ATONCREATE Command.....	59
4.6.1.15	The ATONDELETE Command.....	61
4.6.2	Messages from the Server.....	61
4.6.2.1	The AAZCREATED Message.....	61
4.6.2.2	The AAZDELETED Message.....	63
4.6.2.3	The NAAZCREATED Message.....	63
4.6.2.4	The NAAZDELETED Message.....	64
4.6.2.5	The NTZCREATED Message.....	65
4.6.2.6	The NTZDELETED Message.....	66
4.6.2.7	The ATONCREATED Message.....	66
4.6.2.8	The ATONDELETED Message.....	67
4.7	Track Data Interface.....	68
4.7.1	Messages from the Server.....	68
4.7.1.1	Track Message.....	68

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	5 of 71

1 INTRODUCTION

This document describes the SCANTER Track Management Protocol which is an ASCII protocol built on TCP/IP.

As Terma A/S aims to improve our products continuously, we consequently reserve the right to revise product characteristics without notice.

The document is separated in two main chapters representing the interface for SCANTER 2XXX/5XXX/6XXX series transceivers and the interface for SCANTER 4XXX series transceivers.

1.1 Scope

The scope of the protocol is the transmission of tracking control messages from the client to the server and primary radar track data transmitted from the server to the client. Tracking control messages include handling of AAZ, NAAZ and NTZ as well as dynamic update of AtoN definitions.

The following interfaces are defined:

Interface name	Transceiver	TCP port
Track Control Interface	SC4xxx, channel A/B SC5xxx/SC6xxx SC2xxx	17394
Track Data Interface	SC4xxx, channel A/B SC5xxx/SC6xxx SC2xxx	17396

2 DEFINITIONS

ASCII	American Standard Code for Information Interchange
IP	Internet Protocol
NAAZ	Non-Automatic Acquisition Zone
AAZ	Automatic Acquisition Zone
NTZ	Non-Tracking Zone
TCP	Transmission Control Protocol
AtoN	Aid to Navigation
SC4xxx	The SCANTER 4000 series transceivers
SC5xxx/SC6xxx	The SCANTER 5000/6000 series transceivers
SC2xxx	The SCANTER 2000 series transceivers

In this document, the term “Server” denotes the tracking application whereas the term “Client” denotes a remote system that receives tracks and issues commands to the server.

For all absolute positions given as input or output parameters, the latitude and longitude are referring to the WGS-84 datum.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	6 of 71

3 SCANTER 2XXX/5XXX/6XXX SERIES

3.1 Interface protocol revisions

This document describes the following revisions of the protocol interfaces:

Interface name	Protocol revision
Track Control Interface	1.2
Track Data Interface	2.0 / 1.1

The protocol revision is divided into two versions; major.minor. The major revision is increased when backward compatibility is broken. The minor revision is increased when functionality is added to the protocol, but backward compatibility is kept.

3.2 Client and Server Messages Summary

This chapter provides a summary of all messages that can be sent from either the client or the server and on which interface the message may appear.

Messages from Client			
Command message	Purpose	Track Control Interface	Track Data Interface
get	Ask server to transmit the value of a particular property.	+	+
ping	Probe if the connection is alive.	+	+
bye	Client wish to end the communication.	+	+
set	Ask server to set the value of a particular property	+	+
trackcreate	Manually create a track.	+	-
trackdelete	Manually delete a track.	+	-
trackswap	Manually swap two tracks.	+	-
trackselect	Turn a track into a selected one.	+	-
trackmove	Move a track onto another plot.	+	-
aazcreate	Create an AAZ.	+	-
aazdelete	Delete an AAZ.	+	-
naazcreate	Create a NAAZ.	+	-
naazdelete	Delete a NAAZ.	+	-
ntzcreate	Create a NTZ.	+	-
ntzdelete	Delete a NTZ.	+	-
atoncreate	Create an AtoN.	+	-
atondelete	Delete an AtoN.	+	-

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	7 of 71

Messages from Server			
Command message	Purpose	Track Control Interface	Track Data Interface
currval	Reports the current value of a particular property requested by the client.	+	+
msgerr	Reports that there was an error in a previous message received from the client.	+	+
pong	Reply to a ping message received from the client.	+	+
bye	Sent when the server needs to end the communication.	+	+
aazcreated	To report the existence of an AAZ to all connected clients.	+	-
aazdeleted	To report to all connected clients that an AAZ has been deleted.	+	-
naazcreated	To report the existence of a NAAZ to all connected clients.	+	-
naazdeleted	To report to all connected clients that a NAAZ has been deleted.	+	-
ntzcreated	To report the existence of a NTZ to all connected clients.	+	-
ntzdeleted	To report to all connected clients that a NTZ has been deleted.	+	-
atoncreated	To report the existence of an AtoN to all connected clients.	+	-
atondeleted	To report to all connected clients that an AtoN has been deleted.	+	-
track	Track message.	-	+

3.3 Message syntax

The general syntax is described in this chapter.

3.3.1 Client Syntax

The protocol works by reading/writing lines. A line sent from the client shall be formatted as follows:

- `<command>[, <argument#1>][, <argument#2>][, ...][, <argument#n>]<eol>`

The line consists of a command followed by zero or more arguments, and terminated with `<eol>` (end of line). A line is considered to be terminated by any one of a line feed (ASCII value 10_{dec}), a carriage return (ASCII value 13_{dec}), or a carriage return followed immediately by a line feed.

To separate the various elements in the line, a comma (ASCII value 44_{dec}) is used as delimiter. NOTE: Spaces in the arguments are allowed.

3.3.2 Server Syntax

A line sent from the server is formatted as follows:

- `<reply>[, <agument#1>][, <argument#2>][, ...][, <argument#n>]<CR><LF>`

This line has the same elements and constraints as the line sent from the client, except it is always terminated by a carriage return (ASCII value 13_{dec}) followed immediately by a line feed (ASCII value 10_{dec}).

3.4 Message transfer description

This chapter describes how to communicate via this interface.

3.4.1 Connection

When a new connection is created, the server will transmit the protocol revision. This information can be used to determine if the client is able to communicate with the server.

Example #1	
Sent from server:	<code>currval,Protocol revision,1.2</code>

3.4.1.1 Maintaining the Link

The link is maintained on the TCP/IP level. If the client needs to check if the link is alive, a `ping` (see 3.5.1.3) can be sent, which will be replied with a `pong` (see 3.5.2.3) if the link is alive.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	9 OF 71

3.4.2 Disconnection

When connection is no longer desired, the client should send a `bye` command (see 3.5.1.4), which will terminate the link without further notice.

If, however, the client terminates without sending a `bye` command (e.g. if the client fails), it is still possible for the client to immediately initiate a new connection.

3.4.3 Error Detection and Handling

If the client sends a message/request which contains syntax errors or values out of range, the transceiver will respond with an error message (`msgerr`), see 3.5.2.2.

3.5 Common Messages

This section describes those messages that are common for all interfaces described in this document.

3.5.1 Common Messages from the Client

This section describes the messages that the client can send to any of the interfaces.
NOTE: Everything the client can send is case insensitive.

3.5.1.1 The GET Command

The `get` command is used to get information. The command needs one argument, which specifies what information to get. A `msgerr` is returned, if the information does not exist.

Arguments			
No	Name	Possible Values	Description
1	Information to get	revisions	This will request the protocol revision. The protocol revision is also sent automatically by the server upon receiving a connection from a client. See example below.

3.5.1.1.1 Example

Example #1	
Sent from client:	<code>get, revisions</code>
Reply from server:	<code>currval, Protocol revision, 1.2</code>

3.5.1.2 The SET Command

The `set` command is used to set a parameter to a given value. The command needs two arguments – which parameter to set, and the value to set it to. A `msgerr` is returned if the parameter does not exist, or is inaccessible, or if the value is invalid.

Arguments			
No	Name	Possible Values	Description
1	parameter to set	<parameter name>	The name of the parameter to set.
2	value	<legal value>	The value to set the parameter to.

3.5.1.2.1 Example

Example #1	
Sent from client:	<code>set,Protocol revision,1.2</code>
Reply from server:	<code>currval,Protocol revision,1.2</code>

3.5.1.3 The PING Command

This command is used to probe if the connection is alive. It takes no arguments and has no impact on the server settings. The server will immediately respond to the client with `pong` message.

3.5.1.4 The BYE Command

This command is used when the client wishes to end the communication. It takes no arguments, and has no impact on the server settings. The server will immediately close the connection.

3.5.2 Common Messages from the Server

This section describes the messages that the server can send regardless of which of the interfaces described in this document the client is connected to.

3.5.2.1 The CURRVAL Message

This message reports the current value of a particular property requested by the client sending a `get` message. Upon receiving an incoming connection from a client, this message is further more transmitted automatically by the server to report the server's protocol revision. This is illustrated in section 3.4.1.

3.5.2.2 The MSGERR Message

This message is sent as a reply to a client if that client sends a command which the server does not recognise or cannot execute. To be able to identify the error, the message includes two arguments: A textual error description and the original erroneous command.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	11 OF 71

Notice that the `msgerr` message is transmitted only on the TCP connection where the erroneous message originated from.

Arguments			
No	Name	Possible Values	Description
1	Error message	Unknown command	The command was not recognised.
		Incorrect number of arguments	The command required a different number of arguments or the name of an argument was illegal.
		Out of radar scope	The coordinate point is out of radar scope.
		Type mismatch	The parameter could not be set to the type given. E.g. a parameter of type integer cannot be set to "42.5" or "On".
		Illegal value	The parameter cannot be set to the desired value. The value lies outside of the parameter's range of definition.
		Internal error	The operation could not be performed due to an internal error.
		Not a track	The id given by the client does not correspond to an ordinary track. Instead the track might be e.g. an AtoN.
		NAAZ already exists	The name given corresponds to an already existing NAAZ.
		Not a NAAZ	The name given does not correspond to a NAAZ defined in the server.
		AAZ already exists	The name given corresponds to an already existing AAZ.
		Not a AAZ	The name given does not correspond to a AAZ defined in the server.
		NTZ already exists	The name given corresponds to an already existing NTZ.
		Not a NTZ	The name given does not correspond to a NTZ defined in the server.
AAZ overlap	Your defined AAZ overlaps another AAZ.		

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	12 OF 71

Arguments			
No	Name	Possible Values	Description
		AAZ limit reached	The AAZ limit is reached.
		Polygon limit reached	The sum of NAAZ and NTZ polygons has reached the maximum number possible.
		AtoN already exists	The name given corresponds to an already existing AtoN.
		Not an AtoN	The name given does not correspond to an AtoN defined in the server.
		AtoN limit reached	The total number of AtoN's has reached the maximum number possible.
		Unsupported protocol revision	The protocol revision requested is not supported.
2...<eol>	Original command	<The erroneous command the client has sent>	This is a copy of what the client has sent.

3.5.2.2.1 Example

In this section two examples are given.

Example #1	
Sent from client:	<code>get, revisions, now</code>
Reply from server:	<code>msgerr, Incorrect number of arguments, get, revisions, now</code>

Example #2	
Sent from client:	<code>removenaaz, someisland</code>
Reply from server:	<code>msgerr, Unknown command, removenaaz, someisland</code>

3.5.2.3 The PONG Message

This message is a reply to the `ping` command, sent from a client. It includes no arguments, and has no impact on the system.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	13 OF 71

3.5.2.4 The BYE Message

If the server needs to close down the connection for some reason, this message is sent. It includes one argument, explaining in plain text, the reason for the connection shut-down.

Arguments			
No	Name	Possible Values	Description
1	Reason	<A textual description>	Describes the reason for the connection shut-down.

3.5.2.4.1 Example

In this section an example given.

<u>Example</u>	
The server is shut down:	
Message from server:	bye, Shut down

3.6 Track Control Interface

A client application can control the way the server treats the reported tracks using the Track Control Interface. This chapter describes this message exchange on the Track Control Interface.

3.6.1 Messages from the Client

3.6.1.1 The GET Command

Arguments			
No	Name	Possible Values	Description
1	Information to get	naazcreated	This will request a number of <i>naazcreated</i> messages corresponding to the actual number of currently existing NAAZ in the server. The reply is described in section 3.6.4.3.
2	Information to get	ntzcreated	This will request a number of <i>ntzcreated</i> messages corresponding to the actual number of currently existing NTZ in the server. The reply is described in section 3.6.4.5.

3	Information to get	atonscreated	This will request a number of <code>atonscreated</code> messages corresponding to the actual number of currently existing AtoN's in the server. The reply is described in section 3.6.4.7.
4	Information to get	aazcreated	This will request a number of <code>aazcreated</code> messages corresponding to the actual number of currently existing AAZ in the server. The reply is described in section 3.6.4.1.
5	Information to get	Revisions	Returns protocol revision

3.6.1.2 The SET Command

Arguments			
No	Name	Possible Values	Description
1	protocol revision	1.2	This will set the protocol revision to use. Currently only version 1.2 is supported.

3.6.1.3 The TRACKCREATE Command

The `trackcreate` command is used to manually initiate a track in the server.

When using the `trackcreate` command, the resulting track is a “selected” track (see section 3.6.3.2).

Until the track has reached a certain quality-level, it is reported (see section 3.7.2.1) with `STAT="CS"`, i.e. a tentative selected track.

If a track is created at a place where there are no plots at all (or if a plot association cannot be established), it will be reported with `STAT="CS"` a number of times and finally reported with `STAT="LS"` when the tracker has deemed it to be lost.

The command needs three arguments:

Arguments			
No	Name	Possible Values	Description
1	LAT	Floating point decimal with 5 digits after the decimal point.	Latitude in decimal degrees of the point near a plot on which a manual correlated tracks should be created.
2	LONG	Floating point decimal with 5 digits after the decimal point.	Longitude in decimal degrees of the point near a plot on which a manual correlated tracks should be created.
3	RADIUS	Integer > 0	Integer number of meters controlling the plot search around the absolute coordinates.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	15 OF 71

If the command is accepted as error-free, no reply is issued by the server.

However, in case of errors in the command issued by the client, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Out of radar scope
Type mismatch
Illegal value

3.6.1.3.1 Example

Example #1	
Sent from client:	<code>trackcreate,-45.00231,32.94276,25</code>

3.6.2 The TRACKDELETE Command

The `trackdelete` command is used to manually delete a track (including an AtoN track) in the server. Notice that a deleted AtoN track will immediately be recreated by the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Positive integer identifying the track.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if the parameter value is bad or the track does not exist, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a track

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	16 OF 71

3.6.2.1.1 Example

<u>Example #1</u>	
Sent from client:	<code>trackdelete,12</code>

3.6.3 The TRACKSWAP Command

The `trackswap` command is used to manually swap the ids of two tracks in the server. It is only possible to swap two tracks within the same video channel (i.e. within either channel A or channel B).

Notice that the `trackswap` message does not change the status (i.e. “automatic or “selected”) of a track.

The command needs two arguments:

Arguments			
No	Name	Possible Values	Description
1	ID1	Integer 0...9999	Id identifying the first one of the two tracks.
2	ID2	Integer 0...9999	Id identifying the second one of the two tracks.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if any of the track ids does not exist, if any of the tracks are not confirmed or if any of the tracks is an AtoN. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a track

3.6.3.1.1 Example

<u>Example #1</u>	
Sent from client:	<code>trackswap,8,14</code>

3.6.3.2 The TRACKSELECT Command

The `trackselect` command is used to turn an automatically created track into being considered as a “selected” one.

A selected track has the following characteristics:

- During the creation phase, a “selected” track may have a larger number of detection lacks than an automatic track before being deemed as lost by the tracker. This number of extra lacks is a configuration parameter in the server.
- A confirmed “selected” track may have more detection lacks than a confirmed automatic track without being deemed as lost by the tracker. This number of extra lacks is a configuration parameter in the server.
- A “selected” track that is lost is reported as lost a configurable number of times. This number is a configuration parameter in the server.

The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Id identifying the track.

If the command is accepted as error-free, no reply is issued by the server. However, the command fails if the track id does not exist. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a track

3.6.3.2.1 Example

Example #1	
Sent from client:	<code>trackselect,29</code>

As a design idea, a client application may choose to issue the `trackselect` command on an automatically created track in any of the following circumstances:

- Whenever the operator on the client side choose to name, classify or otherwise look at that particular track.
- Whenever the client application invoke the `trackmove` command.
- Whenever the client application invoke the `trackswap` command.

3.6.3.3 The TRACKMOVE Command

The `trackmove` command is used to move a track onto another plot at an absolute position. This is typically used if the tracker has failed tracking a target. The track is then predicted for a while after which it is deleted. However, using this command it is possible to manually moving it back onto the plot.

The command needs four arguments:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Id identifying the track to be moved.
2	LAT	Floating point decimal with 5 digits after the decimal point.	Latitude in decimal degrees of the point near a plot on which a manual correlated tracks should be moved.
3	LONG	Floating point decimal with 5 digits after the decimal point.	Longitude in decimal degrees of the point near a plot on which a manual correlated tracks should be moved.
4	RADIUS	Integer > 0	Integer number of meters controlling the plot search around the absolute coordinates.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if the track id does not exist or the message is otherwise malformed. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Out of radar scope
Type mismatch
Illegal value
Not a track

3.6.3.3.1 Example

<u>Example #1</u>	
Sent from client:	<code>trackmove, 4, -45.00231, 32.94276, 25</code>

3.6.3.4 The AAZCREATE Command

The `aazcreate` command is used to create an automatic track acquisition zone (AAZ) in which the tracker will automatically initiate new tracks.

A created track that leaves an AAZ will keep getting updated as it would be inside the AAZ. A track created manually inside an AAZ will be updated provided that there is a plot to associate with the track.

A limit is imposed on the sum of AAZ's of 16 zones and no AAZ may overlap another within same video coverage. An AAZ defined as NR may overlap an AAZ defined as MTI, but an AAZ defined NR or MTI may not overlap an AAZ defined as ALL.

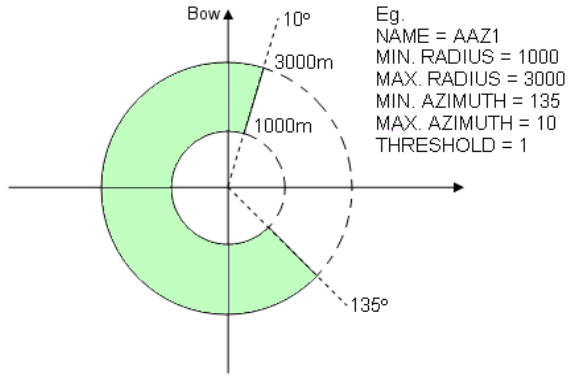
Notice that if an AAZ already exists with the given name, it is considered as an error. An already-existing AAZ must be deleted using the `aazdelete` command before a new AAZ with the same name may be created.

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	MIN. RADIUS	Integer 0...300000	An integer indicating the crown minimum radius in metres.
4	MAX. RADIUS	Integer 0...300000	An integer indicating the crown maximum radius in metres.
5	MIN. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The minimum crown azimuth in decimal degrees.
6	MAX. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The maximum crown azimuth in decimal degrees.
7	THRESHOLD	Integer 0...500	Within an AAZ with area threshold "AT" and area threshold factor "AF" a plot can only be used for automatic track creation if the plot area $PA \geq AF * AT$. If this is a feature you do not want to use, set threshold value to 1.

If the command is accepted as error-free, no reply is issued by the server.

The definition is following:



The AAZ is relative to ship heading (or north on stationary platforms, unless a heading value is fed into the VDT), so if the ship turns, the zone will turn too. To create a donut shape, use a minimum azimuth value of 0 and a azimuth maximum value of 360.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
AAZ already exists
AAZ limit reached
AAZ overlap

3.6.3.4.1 Example

Example #1	
Sent from client:	<code>aazcreate,someaaz,NR,1000,20000,90.0,10.0,1</code>

3.6.3.5 The AAZDELETE Command

The `aazdelete` command is used to manually delete an AAZ crown in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if an AAZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a AAZ

3.6.3.5.1 Example

Example #1	
Sent from client:	<code>aazdelete,someaaz</code>

3.6.3.6 The NAAZCREATE Command

The `naazcreate` command is used to create a non-automatic track acquisition zone (NAAZ) in which the tracker will not automatically initiate new tracks.

An existing track that enters an NAAZ will keep getting updated as it would be outside the NAAZ. A track created manually inside an NAAZ will be updated provided that there is a plot to associate with the track.

A limit is imposed on the sum of NAAZ's and NTZ's: Within any radar coverage area (= a circle around the radar), at least 125 zones may be created. In total, the system will allow the operator to create and store 10000 (ten thousand) zones.

A NAAZ is defined as a polygon of 4...127 vertices where the first and the last vertex must be the same.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	22 OF 71

When giving coordinates for an NAAZ, there must be no lines overlapping each other. It is not considered an error if the sequence of coordinates defines an area of no size.

Notice that if a NAAZ already exists with the given name, it is considered as an error. An already-existing NAAZ must be deleted using the `naazdelete` command before a new NAAZ with the same name may be created.

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NAAZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
NAAZ already exists
Polygon limit reached

3.6.3.6.1 Example

Example #1	
Sent from client:	naazcreate, someisland, NR, 5, 45.50000, 15.00000, 45.50000, 15.50000, 45.0000, 15.50000, 45.00000, 15.00000, 45.50000, 15.00000

3.6.3.7 The NAAZDELETE Command

The `naazdelete` command is used to manually delete a NAAZ polygon in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NAAZ polygon.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if a NAAZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message	
Possible Values	
Incorrect number of arguments	
Type mismatch	
Illegal value	
Not a NAAZ	

3.6.3.7.1 Example

Example #1	
Sent from client:	naazdelete, someisland

3.6.3.8 The NTZCREATE Command

The `ntzcreate` command is used to create a NTZ in which no tracking takes place at all. It is defined as a polygon of 4...127 vertices where the first and the last vertex must be the same.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	24 OF 71

When giving coordinates for an NTZ, there must be no lines overlapping each other. It is not considered an error if the sequence of coordinates defines an area of no size.

Notice that if a NTZ already exists with the given name, it is considered as an error. An already-existing NTZ must be deleted using the `ntzdelete` command before a new NTZ with the same name may be created.

Please read section 3.6.3.6 which defines a common limit for NAAZ's and NTZ's.

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NTZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NTZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NTZ vertex longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
NTZ already exists
Polygon limit reached

3.6.3.8.1 Example

Example #1	
Sent from client:	<code>ntzcreate, someisland, NR, 5, 45.50000, 15.00000, 45.50000, 15.50000, 45.00000, 15.50000, 45.00000, 15.00000, 45.50000, 15.00000</code>

3.6.3.9 The NTZDELETE Command

The `ntzdelete` command is used to manually delete a NTZ polygon in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NTZ polygon.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if a NTZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message	
Possible Values	
Incorrect number of arguments	
Type mismatch	
Illegal value	
Not a NTZ	

3.6.3.9.1 Example

Example #1	
Sent from client:	<code>ntzdelete, someisland</code>

3.6.3.10 The ATONCREATE Command

The `atoncreate` command is used to dynamically create an AtoN.

In total, the system will allow 10000 (ten thousand) AtoN's to be stored.

An AtoN that is created using this command, but not detected by the radar, will be reported with the `STAT="CS"` and `TYPE="ATON"` in the track message of section 3.7.2.1.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	26 OF 71

Notice that if an AtoN already exists with the given name, it is considered as an error. An already-existing AtoN must be deleted using the `atondelete` command before a new AtoN with the same name may be created.

Notice further more, that a track of TYPE="TARGET" cannot be converted into being a track of TYPE="ATON" by using this (or any other) command. If an AtoN is created at the same or at a nearby lat/long position as a non-AtoN track, it is up to the tracker to determine the future behaviour of the track that was already located at that position (i.e. the track may continue or cease to exist).

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AtoN.
2	RADIUS	Integer ≥ 0	The accepted radius that the AtoN may drift away from its absolute coordinates.
3	LAT	Floating point decimal with 5 digits after the decimal point.	AtoN latitude in decimal degrees.
4	LONG	Floating point decimal with 5 digits after the decimal point.	AtoN longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
AtoN already exists
AtoN limit reached

3.6.3.10.1 Example

<u>Example #1</u>	
Sent from client:	<code>atoncreate, kb01, 25, 56.22182, 10.44495</code>

Notice that it is possible for a client to maintain a data model of defined AtoN's. Issuing a `get, atonscreated` command to the server will list all the AtoN's currently defined. Whenever AtoN's are created or deleted by any client, corresponding `atoncreated` and `atondeleted` messages are transmitted to all connected clients.

3.6.3.11 The ATONDELETE Command

The `atondelete` command is used to delete a dynamically created AtoN in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AtoN.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if an AtoN of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not an AtoN

3.6.3.11.1 Example

<u>Example #1</u>	
Sent from client:	<code>atondelete, kb01</code>

3.6.4 Messages from the Server

Below is listed those messages from the server which are specific for the Track Control Interface.

3.6.4.1 The AAZCREATED Message

This message is transmitted in two cases:

1. Whenever a client has created a AAZ using the `aazcreate` message, the server transmits a `aazcreated` to all connected clients.
2. When a client has asked for the complete list of existing AAZ using the `get, aazcreated` command, the server replies with multiple `aazcreated` messages, one for each existing AAZ. Notice that the protocol does not transmit the actual number of AAZ currently existing.
- 3.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	MIN. RADIUS	Integer 0...300000	An integer indicating the crown minimum radius in metres.
4	MAX. RADIUS	Integer 0...300000	An integer indicating the crown maximum radius in metres.
5	MIN. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The minimum crown azimuth in decimal degrees.
6	MAX. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The maximum crown azimuth in decimal degrees.
7	THRESHOLD	Integer 0...500	Within an AAZ with area threshold "AT" and area threshold factor "AF" a plot can only be used for automatic track creation if the plot area $PA \geq AF * AT$. If this is a feature you do not want to use, set threshold value to 1.

3.6.4.1.1 Example

In the following example, a single AAZ is reported:

<u>Example #1</u>	
Sent from server:	aazcreated, someaaz, ALL, 1000, 20000, 90.0, 10.0, 1

3.6.4.2 The AAZDELETED Message

This message is sent to all connected clients when an AAZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AAZ crown which has been deleted.

3.6.4.2.1 Example

<u>Example #1</u>	
Sent from server:	aazdeleted, someaaz

3.6.4.3 The NAAZCREATED Message

This message is transmitted in two cases:

- Whenever a client has created a NAAZ using the `naazcreate` message, the server transmits a `naazcreated` to all connected clients.
- When a client has asked for the complete list of existing NAAZ using the `get, naazcreated` command, the server replies with multiple `naazcreated` messages, one for each existing NAAZ. Notice that the protocol does not transmit the actual number of NAAZ currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NAAZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.

Arguments			
No	Name	Possible Values	Description
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex longitude in decimal degrees.

3.6.4.3.1 Example

In the following example, a single NAAZ is reported:

Example #1	
Sent from server:	naazcreated,someisland,ALL,5,45.50000,15.00000,45.50000,15.50000,45.0000,15.50000,45.00000,15.00000,45.50000,15.00000

3.6.4.4 The NAAZDELETED Message

This message is sent to all connected clients when a NAAZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NAAZ polygon which has been deleted.

3.6.4.4.1 Example

Example #1	
Sent from server:	naazdeleted,someisland

3.6.4.5 The NTZCREATED Message

This message is transmitted in two cases:

1. Whenever a client has created a NTZ using the `ntzcreate` message, the server transmits a `ntzcreated` to all connected clients.
2. When a client has asked for the complete list of existing NTZ using the `get,ntzcreated` command, the server replies with multiple `ntzcreated` messages, one for each existing NTZ. Notice that the protocol does not transmit the actual number of NTZ currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NTZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NTZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NTZ vertex longitude in decimal degrees.

3.6.4.5.1 Example

In the following example, a single NTZ is reported:

Example #1	
Sent from server:	<code>ntzcreated,someisland,ALL,5,45.50000,15.00000,45.50000,15.50000,45.00000,15.50000,45.00000,15.00000,45.50000,15.00000</code>

3.6.4.6 The NTZDELETED Message

This message is sent to all connected clients, when a NTZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NTZ polygon which has been deleted.

3.6.4.6.1 Example

<u>Example #1</u>	
Sent from server:	ntzdeleted,someisland

3.6.4.7 The ATOCREATED Message

This message is transmitted in two cases:

1. Whenever a client has created an AtoN using the `atoncreate` message, the server transmits a `atoncreated` to all connected clients.
2. When a client has asked for the complete list of existing AtoN's using the `get,atonscreated` command, the server replies with multiple `atoncreated` messages, one for each existing AtoN. Notice that the protocol does not transmit the actual number of AtoN's currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AtoN.
2	RADIUS	Integer ≥ 0	The accepted radius that the AtoN may drift away from its absolute coordinates.
3	LAT	Floating point decimal with 5 digits after the decimal point.	AtoN latitude in decimal degrees.
4	LONG	Floating point decimal with 5 digits after the decimal point.	AtoN longitude in decimal degrees.

3.6.4.7.1 Example

In the following example, a single AtoN is reported:

<u>Example #1</u>	
Sent from client:	atoncreated, kb01, 25, 56.22182, 10.44495

3.6.4.8 The ATONDELETED Message

This message is sent to all connected clients when an AtoN is dynamically deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AtoN which has been deleted.

3.6.4.8.1 Example

<u>Example #1</u>	
Sent from server:	atondeleted, kb01

3.7 Track Data Interface

This chapter describes the interface for transmitting tracks from the server to a client. Multiple clients may connect to this interface at the same time and all connected clients will receive all of the `track` messages.

Notice that it is not possible for a client to request a list of currently existing tracks. However, for each antenna rotation, all existing tracks are reported.

3.7.1 Messages from the Client

3.7.1.1 The GET Command

Arguments			
No	Name	Possible Values	Description
1	Information to get	revisions	Returns protocol revision

3.7.1.2 The SET Command

Arguments			
No	Name	Possible Values	Description
1	Protocol revision	1.1 2.0	This will set the protocol revision to use.

3.7.2 Messages from the Server

3.7.2.1 Track Message Version 1.1

This is the default revision when connection to the interface. It can be upgraded using the set command.

This version of the track message contains the following arguments:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Track number identifying the track.
2	Y	Integer 0...9999	Year of track time stamp.
3	MM	Integer 1...12	Month of track time stamp.
4	DD	Integer 1..31	Day of track time stamp.
5	HH	Integer 0..23	Hour of track time stamp.
6	MIN	Integer 0..59	Minute of track time stamp.
7	SEC	Integer 0...59	Second of track time stamp.
8	MSEC	Integer 0...999	Millisecond of track time stamp.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	35 OF 71

Arguments			
No	Name	Possible Values	Description
9	STAT	CA CS FA FS LA LS	CA = Tentative automatic track. CS = Tentative selected track. FA = Update of automatic track. FS = Update of a selected track. LA = Automatic track is lost. LS = Selected track is lost.
10	TYPE	TARGET ATON	TARGET = Air or sea target. ATON = AtoN.
11	NAME	String of letters [a-z] and digits [0-9]	When TYPE="ATON", this is the textual name representing the AtoN. When TYPE="TARGET", this field is empty.
12	LINEMASK	Integer 0...63	Proprietary service information.
13	SIZE	Integer >= 0	Size (= plot area) of the track.
14	RANGE	Integer >= 0	Track slant range in meters relative to radar.
15	AZIMUTH	Floating point deci- mal with 5 digits af- ter the decimal point.	Track azimuth radians.
16	LAT	Floating point deci- mal with 5 digits af- ter the decimal point.	Track position latitude in decimal de- grees.
17	LONG	Floating point deci- mal with 5 digits af- ter the decimal point.	Track position longitude in decimal de- grees.
18	SPEED	Floating point deci- mal with 5 digits af- ter the decimal point.	Absolute speed vector length in m/s.
19	COURSE	Floating point deci- mal with 5 digits af- ter the decimal point.	Absolute speed vector azimuth in radians.
20	QUALITY	Integer 0...30	Track quality.
21	L16QUALITY	Integer 0...15	STANAG5516 track quality.
22	LACKS	Integer >= 0	Track detection lacks.
23	WINRGW	Integer >= 0	Track search window range width in me- ters
24	WINAZW	Floating point deci- mal with 5 digits af- ter the decimal point.	Track search window azimuth width in radians.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	36 OF 71

Arguments			
No	Name	Possible Values	Description
25	STDERR	Floating point decimal with 2 digits after the decimal point in the range: [0.00 ... 100000.00]	The tracker's calculated standard error on the filtered track position. The unit is meters.

3.7.2.1.1 Example

This example shows a track message for a TYPE="TARGET" track where the NAME argument is empty.

Example #1	
Sent from client:	<pre>track, 1234,1999,5,21,12, 49,24,5,CA,TARGET,,6,5,5928, 0.67083,12.34567,45.12345,587.92845,4.18472, 27,12,2,18,0.23985,0.00</pre>

3.7.2.2 Track Message Version 2.0

This version adds associated plot information to the track messages. It contains track message version 1.0 and adds following fields:

Arguments			
No	Name	Possible Values	Description
26	PLOTLINE	Integer 0...6	Tracking line number. 0 (zero) if no associated plot is available
27	PLOTTYPE	MTI/ NR/NONE	Indicates what type of tracking line the plot originated from. NONE if no associated plot is available.
28	PLOTRANGE	Integer >= 0	Plot range in meters relative to radar. Set to 0 (zero) if no associated plot.
29	PLOTAZIMUTH	Floating point decimal with 5 digits after the decimal point.	Plot azimuth radians. Set to 0 (zero) if no associated plot.
30	PLOTLAT	Floating point decimal with 5 digits after the decimal point.	Plot position latitude in decimal degrees. Set to 0 (zero) if no associated plot.
31	PLOTLONG	Floating point decimal with 5 digits after the decimal point.	Plot position longitude in decimal degrees. Set to 0 (zero) if no associated plot.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	37 OF 71

Arguments			
No	Name	Possible Values	Description
32	RGW	Integer ≥ 0	Plot range width in meters. Set to 0 (zero) if no associated plot.
33	AZW	Floating point decimal with 5 digits after the decimal point.	Plot azimuth width in radians. Set to 0 (zero) if no associated plot.
34	PA	Integer 0..255	Peak amplitude. Set to 0 (zero) if no associated plot.
35	TA	TA Integer ≥ 0	Total amplitude (amplitude sum of samples in plot). Set to 0 (zero) if no associated plot.
36	SC	Integer ≥ 0	Sample count. The number of samples in the plot. Set to 0 (zero) if no associated plot.
37	PC	Integer 0..1000	Plot credibility measure. The plot credibility is a function of peak amplitude and sample count. Set to 0 (zero) if no associated plot.

3.7.2.2.1 Example

This example shows a track message for a TYPE="TARGET" track where the NAME argument is empty.

Example #1	
Sent from client:	track,1999,5,27,72,49,24,507,CA,TARGET,,6,558,5928,0.67083,12.34561,45.12345,87.92845,4.184,28,I0,0,300,0.23985,143.61,2,NR,5928,0.67083,12.34561,45.12345,51,0.07232,250,12050,80,751

3.7.2.2.2 Associated plot note

A target is often tracked in more than one tracking line; that means, that the track correlator is supplied with one track + associated plot message from each of the relevant tracking lines. The correlator uses a quality criterion to select one of these 'elementary tracks' to be the correlated track transmitted to the outside world. The associated plot in this extended track message is the plot corresponding to the selected 'elementary track'.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	38 OF 71

3.7.2.3 Lost Tracks

Lost tracks are reported as follows:

- a. When no delete message is sent from a connected client:
 1. Automatic tracks are sent as lost (i.e. STAT="LA") to all connected clients only once with their last position and the timestamp of this last position.
 2. Selected tracks are sent as lost (i.e. STAT="LS") to all connected clients a configurable number of times with their last position and the timestamp of this last position. A track ID that has been reported lost will never return to a non-lost state.
- b. When a delete message is received from a connected client, the concerned track or AtoN track is deleted and the following lost-track behaviour can be observed:
 1. For an automatic track, the track is sent as lost (i.e. STAT="LA") to all connected clients only once with its last position and the timestamp of this last position.
 2. For a selected track, the track is sent as lost (i.e. STAT="LS") to all connected clients only once with its last position and the timestamp of this last position.

A track message for a lost track (STAT="LA" or STAT="LS") will always include LINEMASK=0.

Notice that a stale period for a track ID cannot be guaranteed.

3.7.2.4 Air Tracks

Air tracks, as well as sea tracks, are reported using the track message defined in 3.7.2.1. For all tracks that are physically not located in the ground plane (e.g. air tracks), the projection of slant range/azimuth coordinates onto lat/long coordinates will imply a projection error.

3.7.2.5 AtoN Tracks

This section contains advice on how to administer AtoN tracks.

AtoN's are defined in absolute coordinates using the `atoncreate` command.

Within its configured range, the server will always report AtoN's regardless of whether they are being tracked or not. Outside its configured range, the server will not report any AtoN tracks.

For an AtoN that is not being tracked, the track message will have STAT="CS". The corresponding coordinates of the AtoN will in this case be those that were used to define the AtoN using the `atoncreate` command, e.g. its static coordinates.

For an AtoN that is being tracked, the track message will have STAT="FS". The coordinates reported are in this case the tracked position that may differ from the static coordinates.

If the client wants the static coordinates of a tracked AtoN, the client must issue a `get, atonscreated` command and match the NAME field of the TYPE="ATON" track message with the corresponding name in the resulting list of AtoN's. The client may maintain a data model of defined AtoN's as described in section 3.6.3.10.1.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	39 OF 71

4 SCANTER 4XXX SERIES

4.1 Interface protocol revisions

This document describes the following revisions of the protocol interfaces:

Interface name	Protocol revision
Track Control Interface	1.1
Track Data Interface	1.0

The protocol revision is divided into two versions; major.minor. The major revision is increased when backward compatibility is broken. The minor revision is increased when functionality is added to the protocol, but backward compatibility is kept.

4.2 Client and Server Messages Summary

This chapter provides a summary of all messages that can be sent from either the client or the server and on which interface the message may appear.

Messages from Client			
Command message	Purpose	Track Control Interface	Track Data Interface
get	Ask server to transmit the value of a particular property.	+	+
ping	Probe if the connection is alive.	+	+
bye	Client wish to end the communication.	+	+
trackcreate	Manually create a track.	+	-
trackdelete	Manually delete a track.	+	-
trackswap	Manually swap two tracks.	+	-
trackselect	Turn a track into a selected one.	+	-
trackmove	Move a track onto another plot.	+	-
aazcreate	Create an AAZ.	+	-
aazdelete	Delete an AAZ.	+	-
naazcreate	Create a NAAZ.	+	-
naazdelete	Delete a NAAZ.	+	-
ntzcreate	Create a NTZ.	+	-
ntzdelete	Delete a NTZ.	+	-
atoncreate	Create an AtoN.	+	-
atondelete	Delete an AtoN.	+	-

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	40 OF 71

Messages from Server			
Command message	Purpose	Track Control Interface	Track Data Interface
currval	Reports the current value of a particular property requested by the client.	+	+
msgerr	Reports that there was an error in a previous message received from the client.	+	+
pong	Reply to a ping message received from the client.	+	+
bye	Sent when the server needs to end the communication.	+	+
aazcreated	To report the existence of an AAZ to all connected clients.	+	-
aazdeleted	To report to all connected clients that an AAZ has been deleted.	+	-
naazcreated	To report the existence of a NAAZ to all connected clients.	+	-
naazdeleted	To report to all connected clients that a NAAZ has been deleted.	+	-
ntzcreated	To report the existence of a NTZ to all connected clients.	+	-
ntzdeleted	To report to all connected clients that a NTZ has been deleted.	+	-
atoncreated	To report the existence of an AtoN to all connected clients.	+	-
atondeleted	To report to all connected clients that an AtoN has been deleted.	+	-
track	Track message.	-	+

4.3 Message syntax

The general syntax is described in this chapter.

4.3.1 Client Syntax

The protocol works by reading/writing lines. A line sent from the client shall be formatted as follows:

- `<command>[, <argument#1>][, <argument#2>][, ...][, <argument#n>]<eol>`

The line consists of a command followed by zero or more arguments, and terminated with `<eol>` (end of line). A line is considered to be terminated by any one of a line feed (ASCII value 10_{dec}), a carriage return (ASCII value 13_{dec}), or a carriage return followed immediately by a line feed.

To separate the various elements in the line, a comma (ASCII value 44_{dec}) is used as delimiter. NOTE: Spaces in the arguments are allowed.

4.3.2 Server Syntax

A line sent from the server is formatted as follows:

- `<reply>[, <agument#1>][, <argument#2>][, ...][, <argument#n>]<CR><LF>`

This line has the same elements and constraints as the line sent from the client, except it is always terminated by a carriage return (ASCII value 13_{dec}) followed immediately by a line feed (ASCII value 10_{dec}).

4.4 Message transfer description

This chapter describes how to communicate via this interface.

4.4.1 Connection

When a new connection is created, the server will transmit the protocol revision. This information can be used to determine if the client is able to communicate with the server.

Example #1	
Sent from server:	<code>currval,Protocol revision,1.1</code>

4.4.1.1 Maintaining the Link

The link is maintained on the TCP/IP level. If the client needs to check if the link is alive, a `ping` (see 4.5.1.2) can be sent, which will be replied with a `pong` (see 4.5.2.3) if the link is alive.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	42 OF 71

4.4.2 Disconnection

When connection is no longer desired, the client should send a `bye` command (see 4.5.1.3), which will terminate the link without further notice.

If, however, the client terminates without sending a `bye` command (e.g. if the client fails), it is still possible for the client to immediately initiate a new connection.

4.4.3 Error Detection and Handling

If the client sends a message/request which contains syntax errors or values out of range, the transceiver will respond with an error message (`msgerr`), see 4.5.2.2.

4.5 Common Messages

This section describes those messages that are common for all interfaces described in this document.

4.5.1 Common Messages from the Client

This section describes the messages that the client can send to any of the interfaces.
NOTE: Everything the client can send is case insensitive.

4.5.1.1 The GET Command

The `get` command is used to get information. The command needs one argument, which specifies what information to get. A `msgerr` is returned, if the information does not exist.

Arguments			
No	Name	Possible Values	Description
1	Information to get	revisions	This will request the protocol revision. The protocol revision is also sent automatically by the server upon receiving a connection from a client. See example below.

4.5.1.1.1 Example

Example #1	
Sent from client:	<code>get, revisions</code>
Reply from server:	<code>currval, Protocol revision, 1.0</code>

4.5.1.2 The PING Command

This command is used to probe if the connection is alive. It takes no arguments and has no impact on the server settings. The server will immediately respond to the client with `pong` message.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	43 OF 71

4.5.1.3 The BYE Command

This command is used when the client wishes to end the communication. It takes no arguments, and has no impact on the server settings. The server will immediately close the connection.

4.5.2 Common Messages from the Server

This section describes the messages that the server can send regardless of which of the interfaces described in this document the client is connected to.

4.5.2.1 The CURRVAL Message

This message reports the current value of a particular property requested by the client sending a `get` message. Upon receiving an incoming connection from a client, this message is further more transmitted automatically by the server to report the server's protocol revision. This is illustrated in section 4.4.1.

4.5.2.2 The MSGERR Message

This message is sent as a reply to a client if that client sends a command which the server does not recognise or cannot execute. To be able to identify the error, the message includes two arguments: A textual error description and the original erroneous command.

Notice that the `msgerr` message is transmitted only on the TCP connection where the erroneous message originated from.

Arguments			
No	Name	Possible Values	Description
1	Error message	Unknown command	The command was not recognised.
		Incorrect number of arguments	The command required a different number of arguments or the name of an argument was illegal.
		Out of radar scope	The coordinate point is out of radar scope.
		Type mismatch	The parameter could not be set to the type given. E.g. a parameter of type integer cannot be set to "42.5" or "On".
		Illegal value	The parameter cannot be set to the desired value. The value lies outside of the parameter's range of definition.
		Internal error	The operation could not be performed due to an internal error.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	44 OF 71

Arguments			
No	Name	Possible Values	Description
		Not a track	The id given by the client does not correspond to an ordinary track. Instead the track might be e.g. an AtoN.
		NAAZ already exists	The name given corresponds to an already existing NAAZ.
		Not a NAAZ	The name given does not correspond to a NAAZ defined in the server.
		AAZ already exists	The name given corresponds to an already existing AAZ.
		Not a AAZ	The name given does not correspond to a AAZ defined in the server.
		NTZ already exists	The name given corresponds to an already existing NTZ.
		Not a NTZ	The name given does not correspond to a NTZ defined in the server.
		AAZ overlap	Your defined AAZ overlaps another AAZ.
		AAZ limit reached	The AAZ limit is reached.
		Polygon limit reached	The sum of NAAZ and NTZ polygons has reached the maximum number possible.
		AtoN already exists	The name given corresponds to an already existing AtoN.
		Not an AtoN	The name given does not correspond to an AtoN defined in the server.
		AtoN limit reached	The total number of AtoN's has reached the maximum number possible.
2...<eol>	Original command	<The erroneous command the client has sent>	This is a copy of what the client has sent.

4.5.2.2.1 Example

In this section two examples are given.

<u>Example #1</u>	
Sent from client:	<code>get, revisions, now</code>
Reply from server:	<code>msgerr, Incorrect number of arguments, get, revisions, now</code>

<u>Example #2</u>	
Sent from client:	<code>removenaaz, someisland</code>
Reply from server:	<code>msgerr, Unknown command, removenaaz, someisland</code>

4.5.2.3 The PONG Message

This message is a reply to the `ping` command, sent from a client. It includes no arguments, and has no impact on the system.

4.5.2.4 The BYE Message

If the server needs to close down the connection for some reason, this message is sent. It includes one argument, explaining in plain text, the reason for the connection shut-down.

Arguments			
No	Name	Possible Values	Description
1	Reason	<A textual description>	Describes the reason for the connection shut-down.

4.5.2.4.1 Example

In this section an example given.

<u>Example</u>	
The server is shut down:	
Message from server:	<code>bye, Shut down</code>

4.6 Track Control Interface

A client application can control the way the server treats the reported tracks using the Track Control Interface. This chapter describes this message exchange on the Track Control Interface.

4.6.1 Messages from the Client

4.6.1.1 The GET Command

Arguments			
No	Name	Possible Values	Description
1	Information to get	naazcreated	This will request a number of <code>naazcreated</code> messages corresponding to the actual number of currently existing NAAZ in the server. The reply is described in section 4.6.2.3.
2	Information to get	ntzcreated	This will request a number of <code>ntzcreated</code> messages corresponding to the actual number of currently existing NTZ in the server. The reply is described in section 4.6.2.5.
3	Information to get	atonscreated	This will request a number of <code>atonscreated</code> messages corresponding to the actual number of currently existing AtoN's in the server. The reply is described in section 4.6.2.7.
4	Information to get	aazcreated	This will request a number of <code>aazcreated</code> messages corresponding to the actual number of currently existing AAZ in the server. The reply is described in section 4.6.2.1.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	47 OF 71

4.6.1.2 The TRACKCREATE Command

The `trackcreate` command is used to manually initiate a track in the server.

When using the `trackcreate` command, the resulting track is a “selected” track (see section 4.6.1.5).

Until the track has reached a certain quality-level, it is reported (see section 4.7.1.1) with `STAT="CS"`, i.e. a tentative selected track.

If a track is created at a place where there are no plots at all (or if a plot association cannot be established), it will be reported with `STAT="CS"` a number of times and finally reported with `STAT="LS"` when the tracker has deemed it to be lost.

The command needs three arguments:

Arguments			
No	Name	Possible Values	Description
1	LAT	Floating point decimal with 5 digits after the decimal point.	Latitude in decimal degrees of the point near a plot on which a manual correlated tracks should be created.
2	LONG	Floating point decimal with 5 digits after the decimal point.	Longitude in decimal degrees of the point near a plot on which a manual correlated tracks should be created.
3	RADIUS	Integer > 0	Integer number of meters controlling the plot search around the absolute coordinates.

If the command is accepted as error-free, no reply is issued by the server.

However, in case of errors in the command issued by the client, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Out of radar scope
Type mismatch
Illegal value

4.6.1.2.1 Example

Example #1	
Sent from client:	<code>trackcreate, -45.00231, 32.94276, 25</code>

4.6.1.3 The TRACKDELETE Command

The `trackdelete` command is used to manually delete a track (including an AtoN track) in the server. Notice that a deleted AtoN track will immediately be recreated by the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Positive integer identifying the track.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if the parameter value is bad or the track does not exist, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message	
Possible Values	
Incorrect number of arguments	
Type mismatch	
Illegal value	
Not a track	

4.6.1.3.1 Example

Example #1	
Sent from client:	<code>trackdelete,12</code>

4.6.1.4 The TRACKSWAP Command

The `trackswap` command is used to manually swap the ids of two tracks in the server. It is only possible to swap two tracks within the same video channel (i.e. within either channel A or channel B).

Notice that the `trackswap` message does not change the status (i.e. “automatic or “selected”) of a track.

The command needs two arguments:

Arguments			
No	Name	Possible Values	Description
1	ID1	Integer 0...9999	Id identifying the first one of the two tracks.
2	ID2	Integer 0...9999	Id identifying the second one of the two tracks.

If the command is accepted as error-free, no reply is issued by the server.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	49 OF 71

However, the command fails if any of the track ids does not exist, if any of the tracks are not confirmed or if any of the tracks is an AtoN. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a track

4.6.1.4.1 Example

Example #1	
Sent from client:	<code>trackswap, 8, 14</code>

4.6.1.5 The TRACKSELECT Command

The `trackselect` command is used to turn an automatically created track into being considered as a “selected” one.

A selected track has the following characteristics:

- During the creation phase, a “selected” track may have a larger number of detection lacks than an automatic track before being deemed as lost by the tracker. This number of extra lacks is a configuration parameter in the server.
- A confirmed “selected” track may have more detection lacks than a confirmed automatic track without being deemed as lost by the tracker. This number of extra lacks is a configuration parameter in the server.
- A “selected” track that is lost is reported as lost a configurable number of times. This number is a configuration parameter in the server.

The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Id identifying the track.

If the command is accepted as error-free, no reply is issued by the server.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	50 OF 71

However, the command fails if the track id does not exist. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a track

4.6.1.5.1 Example

Example #1	
Sent from client:	<code>trackselect,29</code>

As a design idea, a client application may choose to issue the `trackselect` command on an automatically created track in any of the following circumstances:

- Whenever the operator on the client side choose to name, classify or otherwise look at that particular track.
- Whenever the client application invoke the `trackmove` command.
- Whenever the client application invoke the `trackswap` command.

4.6.1.6 The TRACKMOVE Command

The `trackmove` command is used to move a track onto another plot at an absolute position. This is typically used if the tracker has failed tracking a target. The track is then predicted for a while after which it is deleted. However, using this command it is possible to manually moving it back onto the plot.

The command needs four arguments:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Id identifying the track to be moved.
2	LAT	Floating point decimal with 5 digits after the decimal point.	Latitude in decimal degrees of the point near a plot on which a manual correlated tracks should be moved.
3	LONG	Floating point decimal with 5 digits after the decimal point.	Longitude in decimal degrees of the point near a plot on which a manual correlated tracks should be moved.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	51 OF 71

Arguments			
No	Name	Possible Values	Description
4	RADIUS	Integer > 0	Integer number of meters controlling the plot search around the absolute coordinates.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if the track id does not exist or the message is otherwise malformed. In that case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Out of radar scope
Type mismatch
Illegal value
Not a track

4.6.1.6.1 Example

Example #1	
Sent from client:	<code>trackmove, 4, -45.00231, 32.94276, 25</code>

4.6.1.7 The AAZCREATE Command

The `aazcreate` command is used to create an automatic track acquisition zone (AAZ) in which the tracker will automatically initiate new tracks.

A created track that leaves an AAZ will keep getting updated as it would be inside the AAZ. A track created manually inside an AAZ will be updated provided that there is a plot to associate with the track.

A limit is imposed on the sum of AAZ's of 16 zones and no AAZ may overlap another within same video coverage. An AAZ defined as NR may overlap an AAZ defined as MTI, but an AAZ defined NR or MTI may not overlap an AAZ defined as ALL.

Notice that if an AAZ already exists with the given name, it is considered as an error. An already-existing AAZ must be deleted using the `aazdelete` command before a new AAZ with the same name may be created.

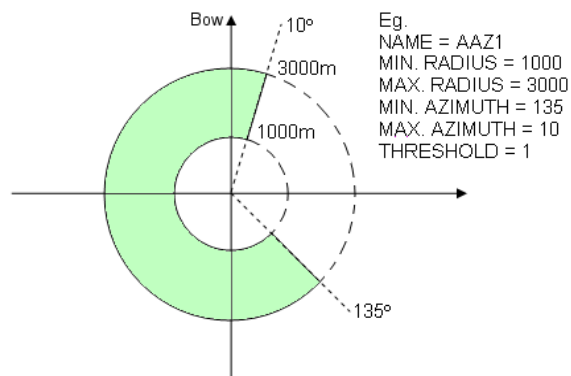
The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	52 OF 71

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	MIN. RADIUS	Integer 0...300000	An integer indicating the crown minimum radius in metres.
4	MAX. RADIUS	Integer 0...300000	An integer indicating the crown maximum radius in metres.
5	MIN. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The minimum crown azimuth in decimal degrees.
6	MAX. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The maximum crown azimuth in decimal degrees.
7	THRESHOLD	Integer 0...500	Within an AAZ with area threshold "AT" and area threshold factor "AF" a plot can only be used for automatic track creation if the plot area $PA \geq AF * AT$. If this is a feature you do not want to use, set threshold value to 1.

If the command is accepted as error-free, no reply is issued by the server.

The definition is following:



The AAZ is relative to ship heading (or north on stationary platforms, unless a heading value is fed into the VDT), so if the ship turns, the zone will turn too. To create a donut shape, use a minimum azimuth value of 0 and a azimuth maximum value of 360.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	53 OF 71

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
AAZ already exists
AAZ limit reached
AAZ overlap

4.6.1.7.1 Example

Example #1	
Sent from client:	<code>aazcreate, someaaz, NR, 1000, 20000, 90.0, 10.0, 1</code>

4.6.1.8 The AAZDELETE Command

The `aazdelete` command is used to manually delete an AAZ crown in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if an AAZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a AAZ

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	54 OF 71

4.6.1.8.1 Example

<u>Example #1</u>	
Sent from client:	aazdelete, someaaz

4.6.1.9 The NAAZCREATE Command

The `naazcreate` command is used to create a non-automatic track acquisition zone (NAAZ) in which the tracker will not automatically initiate new tracks.

An existing track that enters an NAAZ will keep getting updated as it would be outside the NAAZ. A track created manually inside an NAAZ will be updated provided that there is a plot to associate with the track.

A limit is imposed on the sum of NAAZ's and NTZ's: Within any radar coverage area (= a circle around the radar), at least 125 zones may be created. In total, the system will allow the operator to create and store 10000 (ten thousand) zones.

A NAAZ is defined as a polygon of 4...127 vertices where the first and the last vertex must be the same.

When giving coordinates for an NAAZ, there must be no lines overlapping each other. It is not considered an error if the sequence of coordinates defines an area of no size.

Notice that if a NAAZ already exists with the given name, it is considered as an error. An already-existing NAAZ must be deleted using the `naazdelete` command before a new NAAZ with the same name may be created.

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NAAZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex latitude in decimal degrees.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	55 OF 71

Arguments			
No	Name	Possible Values	Description
	LONG	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message	
Possible Values	
Incorrect number of arguments	
Type mismatch	
Illegal value	
NAAZ already exists	
Polygon limit reached	

4.6.1.9.1 Example

Example #1	
Sent from client:	<code>naazcreate, someisland, NR, 5, 45.50000, 15.00000, 45.50000, 15.50000, 45.0000, 15.50000, 45.00000, 15.00000, 45.50000, 15.00000</code>

4.6.1.10 The NAAZDELETE Command

The `naazdelete` command is used to manually delete a NAAZ polygon in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NAAZ polygon.

If the command is accepted as error-free, no reply is issued by the server.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	56 OF 71

However, the command fails if a NAAZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a NAAZ

4.6.1.10.1 Example

Example #1	
Sent from client:	<code>naazdelete,someisland</code>

4.6.1.11 The NTZCREATE Command

The `ntzcreate` command is used to create a NTZ in which no tracking takes place at all. It is defined as a polygon of 4...127 vertices where the first and the last vertex must be the same.

When giving coordinates for an NTZ, there must be no lines overlapping each other. It is not considered an error if the sequence of coordinates defines an area of no size.

Notice that if a NTZ already exists with the given name, it is considered as an error. An already-existing NTZ must be deleted using the `ntzdelete` command before a new NTZ with the same name may be created.

Please read section 4.6.1.9 which defines a common limit for NAAZ's and NTZ's.

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NTZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.

Arguments			
No	Name	Possible Values	Description
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NTZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NTZ vertex longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
NTZ already exists
Polygon limit reached

4.6.1.12 Example

Example #1	
Sent from client:	<code>ntzcreate, someisland, NR, 5, 45.50000, 15.00000, 45.50000, 15.50000, 45.0000, 15.50000, 45.00000, 15.00000, 45.50000, 15.00000</code>

4.6.1.13 The NTZDELETE Command

The `ntzdelete` command is used to manually delete a NTZ polygon in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the NTZ polygon.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	58 OF 71

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if a NTZ of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not a NTZ

4.6.1.13.1 Example

Example #1	
Sent from client:	<code>ntzdelete,someisland</code>

4.6.1.14 The ATONCREATE Command

The `atoncreate` command is used to dynamically create an AtoN.

In total, the system will allow 10000 (ten thousand) AtoN's to be stored.

An AtoN that is created using this command, but not detected by the radar, will be reported with the `STAT="CS"` and `TYPE="ATON"` in the track message of section 3.7.2.1.

Notice that if an AtoN already exists with the given name, it is considered as an error. An already-existing AtoN must be deleted using the `atondelete` command before a new AtoN with the same name may be created.

Notice further more, that a track of `TYPE="TARGET"` cannot be converted into being a track of `TYPE="ATON"` by using this (or any other) command. If an AtoN is created at the same or at a nearby lat/long position as a non-AtoN track, it is up to the tracker to determine the future behaviour of the track that was already located at that position (i.e. the track may continue or cease to exist).

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	59 OF 71

The command needs the following arguments:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AtoN.
2	RADIUS	Integer >= 0	The accepted radius that the AtoN may drift away from its absolute coordinates.
3	LAT	Floating point decimal with 5 digits after the decimal point.	AtoN latitude in decimal degrees.
4	LONG	Floating point decimal with 5 digits after the decimal point.	AtoN longitude in decimal degrees.

If the command is accepted as error-free, no reply is issued by the server.

In case of an error in the command, a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
AtoN already exists
AtoN limit reached

4.6.1.14.1 Example

Example #1	
Sent from client:	<code>atoncreate, kb01, 25, 56.22182, 10.44495</code>

Notice that it is possible for a client to maintain a data model of defined AtoN's. Issuing a `get, atonscreated` command to the server will list all the AtoN's currently defined. Whenever AtoN's are created or deleted by any client, corresponding `atoncreated` and `atondeleted` messages are transmitted to all connected clients.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	60 OF 71

4.6.1.15 The **ATONDELETE** Command

The `atondelete` command is used to delete a dynamically created AtoN in the server. The command needs one argument:

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AtoN.

If the command is accepted as error-free, no reply is issued by the server.

However, the command fails if an AtoN of that name does not exist or there is a syntax error in the command, in which case a `msgerr` is returned by the server describing one and only one of the following error conditions:

Error message
Possible Values
Incorrect number of arguments
Type mismatch
Illegal value
Not an AtoN

4.6.1.15.1 Example

Example #1	
Sent from client:	<code>atondelete, kb01</code>

4.6.2 Messages from the Server

Below is listed those messages from the server which are specific for the Track Control Interface.

4.6.2.1 The **AAZCREATED** Message

This message is transmitted in two cases:

- Whenever a client has created a AAZ using the `aazcreate` message, the server transmits a `aazcreated` to all connected clients.
- When a client has asked for the complete list of existing AAZ using the `get, aazcreated` command, the server replies with multiple `aazcreated` messages, one for each existing AAZ. Notice that the protocol does not transmit the actual number of AAZ currently existing.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	61 OF 71

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9] Max length: 50	A textual name representing the AAZ crown.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	MIN. RADIUS	Integer 0...300000	An integer indicating the crown minimum radius in metres.
4	MAX. RADIUS	Integer 0...300000	An integer indicating the crown maximum radius in metres.
5	MIN. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The minimum crown azimuth in decimal degrees.
6	MAX. AZIMUTH	Floating point decimal with 1 digit after the decimal point. 0.0 ... 360.0	The maximum crown azimuth in decimal degrees.
7	THRESHOLD	Integer 0...500	Within an AAZ with area threshold "AT" and area threshold factor "AF" a plot can only be used for automatic track creation if the plot area PA >= AF*AT. If this is a feature you do not want to use, set threshold value to 1.

4.6.2.1.1 Example

In the following example, a single AAZ is reported:

<u>Example #1</u>	
Sent from server:	aazcreated,someaaz,ALL,1000,20000,90.0,10.0,1

4.6.2.2 The AAZDELETED Message

This message is sent to all connected clients when an AAZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AAZ crown which has been deleted.

4.6.2.2.1 Example

<u>Example #1</u>	
Sent from server:	aazdeleted, someaaz

4.6.2.3 The NAAZCREATED Message

This message is transmitted in two cases:

8. Whenever a client has created a NAAZ using the `naazcreate` message, the server transmits a `naazcreated` to all connected clients.
9. When a client has asked for the complete list of existing NAAZ using the `get, naazcreated` command, the server replies with multiple `naazcreated` messages, one for each existing NAAZ. Notice that the protocol does not transmit the actual number of NAAZ currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NAAZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NAAZ vertex longitude in decimal degrees.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	63 OF 71

4.6.2.3.1 Example

In the following example, a single NAAZ is reported:

<u>Example #1</u>	
Sent from server:	naazcreated,someisland,ALL,5, 45.50000,15.00000,45.50000,15.50000,45.0000, 15.50000,45.00000,15.00000,45.50000,15.00000

4.6.2.4 The NAAZDELETED Message

This message is sent to all connected clients when a NAAZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NAAZ polygon which has been deleted.

4.6.2.4.1 Example

<u>Example #1</u>	
Sent from server:	naazdeleted,someisland

4.6.2.5 The NTZCREATED Message

This message is transmitted in two cases:

3. Whenever a client has created a NTZ using the `ntzcreate` message, the server transmits a `ntzcreated` to all connected clients.
4. When a client has asked for the complete list of existing NTZ using the `get,ntzcreated` command, the server replies with multiple `ntzcreated` messages, one for each existing NTZ. Notice that the protocol does not transmit the actual number of NTZ currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NTZ polygon.
2	VIDEO	NR MTI ALL	NR = Normal radar video MTI = Moving Target Indication video ALL = Cover both NR and MTI
3	VERTEXNB	Integer 4...127	An integer indicating the number of vertices in this polygon.
The following is a list of 4...127 vertices (as indicated by the 'VERTEXNB' argument), each defined by a LAT and a LONG:			
4...257	LAT	Floating point decimal with 5 digits after the decimal point.	NTZ vertex latitude in decimal degrees.
	LONG	Floating point decimal with 5 digits after the decimal point.	NTZ vertex longitude in decimal degrees.

4.6.2.5.1 Example

In the following example, a single NTZ is reported:

Example #1	
Sent from server:	<code>ntzcreated,someisland,ALL,5,45.50000,15.00000,45.50000,15.50000,45.0000,15.50000,45.00000,15.00000,45.50000,15.00000</code>

4.6.2.6 The NTZDELETED Message

This message is sent to all connected clients, when a NTZ has been manually deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the NTZ polygon which has been deleted.

4.6.2.6.1 Example

<u>Example #1</u>	
Sent from server:	ntzdeleted,someisland

4.6.2.7 The ATONCREATED Message

This message is transmitted in two cases:

- Whenever a client has created an AtoN using the `atoncreate` message, the server transmits a `atoncreated` to all connected clients.
- When a client has asked for the complete list of existing AtoN's using the `get,atonscreated` command, the server replies with multiple `atoncreated` messages, one for each existing AtoN. Notice that the protocol does not transmit the actual number of AtoN's currently existing.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AtoN.
2	RADIUS	Integer ≥ 0	The accepted radius that the AtoN may drift away from its absolute coordinates.
3	LAT	Floating point decimal with 5 digits after the decimal point.	AtoN latitude in decimal degrees.
4	LONG	Floating point decimal with 5 digits after the decimal point.	AtoN longitude in decimal degrees.

4.6.2.7.1 Example

In the following example, a single AtoN is reported:

<u>Example #1</u>	
Sent from client:	atoncreated, kb01, 25, 56.22182, 10.44495

4.6.2.8 The ATONDELETED Message

This message is sent to all connected clients when an AtoN is dynamically deleted by one of the clients.

Arguments			
No	Name	Possible Values	Description
1	NAME	String of letters [a-z] and digits [0-9]	A textual name representing the AtoN which has been deleted.

4.6.2.8.1 Example

<u>Example #1</u>	
Sent from server:	atondeleted, kb01

4.7 Track Data Interface

This chapter describes the interface for transmitting tracks from the server to a client. Multiple clients may connect to this interface at the same time and all connected clients will receive all of the `track` messages.

Notice that it is not possible for a client to request a list of currently existing tracks. However, for each antenna rotation, all existing tracks are reported.

4.7.1 Messages from the Server

4.7.1.1 Track Message

A track message contains the following arguments:

Arguments			
No	Name	Possible Values	Description
1	ID	Integer 0...9999	Track number identifying the track.
2	Y	Integer 0...9999	Year of track time stamp.
3	MM	Integer 1...12	Month of track time stamp.
4	DD	Integer 1..31	Day of track time stamp.
5	HH	Integer 0..23	Hour of track time stamp.
6	MIN	Integer 0..59	Minute of track time stamp.
7	SEC	Integer 0...59	Second of track time stamp.
8	MSEC	Integer 0...999	Millisecond of track time stamp.
9	STAT	CA CS FA FS LA LS	CA = Tentative automatic track. CS = Tentative selected track. FA = Update of automatic track. FS = Update of a selected track. LA = Automatic track is lost. LS = Selected track is lost.
10	TYPE	TARGET ATON	TARGET = Air or sea target. ATON = AtoN.
11	NAME	String of letters [a-z] and digits [0-9]	When TYPE="ATON", this is the textual name representing the AtoN. When TYPE="TARGET", this field is empty.
12	LINEMASK	Integer 0...63	Proprietary service information.
13	SIZE	Integer >= 0	Size (= plot area) of the track.
14	RANGE	Integer >= 0	Track slant range in meters relative to radar.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	68 OF 71

Arguments			
No	Name	Possible Values	Description
15	AZIMUTH	Floating point decimal with 5 digits after the decimal point.	Track azimuth radians.
16	LAT	Floating point decimal with 5 digits after the decimal point.	Track position latitude in decimal degrees.
17	LONG	Floating point decimal with 5 digits after the decimal point.	Track position longitude in decimal degrees.
18	SPEED	Floating point decimal with 5 digits after the decimal point.	Absolute speed vector length in m/s.
19	COURSE	Floating point decimal with 5 digits after the decimal point.	Absolute speed vector azimuth in radians.
20	QUALITY	Integer 0...30	Track quality.
21	L16QUALITY	Integer 0...15	STANAG5516 track quality.
22	LACKS	Integer >= 0	Track detection lacks.
23	WINRGW	Integer >= 0	Track search window range width in meters
24	WINAZW	Floating point decimal with 5 digits after the decimal point.	Track search window azimuth width in radians.
25	STDERR	Floating point decimal with 2 digits after the decimal point in the range: [0.00 ... 100000.00]	The tracker's calculated standard error on the filtered track position. The unit is meters.

4.7.1.1.1 Example

This example shows a track message for a TYPE="TARGET" track where the NAME argument is empty.

Example #1	
Sent from client:	<pre>track, 1234,1999,5,21,12, 49,24,5,CA,TARGET,,6,5,5928, 0.67083,12.34567,45.12345,587.92845,4.18472, 27,12,2,18,0.23985,0.00</pre>

4.7.1.1.2 Lost Tracks

Lost tracks are reported as follows:

- c. When no delete message is sent from a connected client:
 1. Automatic tracks are sent as lost (i.e. STAT="LA") to all connected clients only once with their last position and the timestamp of this last position.
 2. Selected tracks are sent as lost (i.e. STAT="LS") to all connected clients a configurable number of times with their last position and the timestamp of this last position. A track ID that has been reported lost will never return to a non-lost state.
- d. When a delete message is received from a connected client, the concerned track or AtoN track is deleted and the following lost-track behaviour can be observed:
 1. For an automatic track, the track is sent as lost (i.e. STAT="LA") to all connected clients only once with its last position and the timestamp of this last position.
 2. For a selected track, the track is sent as lost (i.e. STAT="LS") to all connected clients only once with its last position and the timestamp of this last position.

A track message for a lost track (STAT="LA" or STAT="LS") will always include LINEMASK=0.

Notice that a stale period for a track ID cannot be guaranteed.

4.7.1.1.3 Air Tracks

Air tracks, as well as sea tracks, are reported using the track message defined in 3.7.2.1. For all tracks that are physically not located in the ground plane (e.g. air tracks), the projection of slant range/azimuth coordinates onto lat/long coordinates will imply a projection error.

4.7.1.1.4 AtoN Tracks

This section contains advice on how to administer AtoN tracks.

AtoN's are defined in absolute coordinates using the `atoncreate` command.

Within its configured range, the server will always report AtoN's regardless of whether they are being tracked or not. Outside its configured range, the server will not report any AtoN tracks.

For an AtoN that is not being tracked, the track message will have STAT="CS". The corresponding coordinates of the AtoN will in this case be those that were used to define the AtoN using the `atoncreate` command, e.g. its static coordinates.

For an AtoN that is being tracked, the track message will have STAT="FS". The coordinates reported are in this case the tracked position that may differ from the static coordinates.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	70 OF 71

If the client wants the static coordinates of a tracked AtoN, the client must issue a `get, atonscreated` command and match the NAME field of the TYPE="ATON" track message with the corresponding name in the resulting list of AtoN's. The client may maintain a data model of defined AtoN's as described in section 4.6.1.14.1.

The use and/or disclosure, etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.			
TITLE	DOCUMENT NO.	REV	PAGE
SCANTER Track Management Protocol	303949 SI	F	71 OF 71