MT Focus 6000

Open Protocol

Atlas Copco Industrial Technique AB

9839 0732 00
Specification release 2
2017-05
MT Focus 6000 open protocol specification documentation

Note! This manual can be altered without further notice.

For further information log in to Atlas Copco www.atlascopco.com

1 Introduction

This document specifies all the product MT Focus 6000 (MTF6000) specific considerations when using the Open Protocol. For a full description of the Open Protocol see the document **9836 4415 01 Open Protocol Specification**.

1.1 Revision history

Version	Date	Author	Change
1	2017-01-18	Jacob Andersen	First revision
2	2017-05-23	Jacob Andersen	Second revision

2 MTF 6000 specifics

2.1 Communication

Open Protocol on the MTF6000 controller can accept up to 1 RS232 and 4 TCP/IP connections simultaneously. The controller will keep a separate set of session data for each connection.

All connections are of type “Classic” as defined by Open Protocol.

Link level acknowledging and message sequence numbering as defined by Open Protocol is not supported at this time. This means that using Open Protocol over RS232 is less reliable than over TCP/IP.

2.1.1 RS232 Considerations

The implementation of OP over RS232 for MTF6000 uses no handshake.

2.1.2 USB Considerations

The MTF6000 controller does not support Open Protocol over USB at this time.

2.1.3 TCP/IP Considerations

The MTF6000 controller only supports Open Protocol client connections to port 4545 at this time.

2.2 Open Protocol version 2.0 considerations

When sending or receiving messages containing generic PID fields, see notes on each MID for information on what parameter id:s that are used.

Three data node types are defined:

Node Type	**Value**
Pset	301
Job	302
Result	303

All node types are a MicroTorque specific extension of the Open Protocol standard.

The following data types are defined:

Data Type	Value
Integer	02
String	04
Datetime	05
Boolean	06
Scientific float (see notes below)	90

The scientific float datatype is a 12 character string containing the same format as that output from printf_s("%012.*e", 3, value), e.g. "00001.234e+2".

The scientific float datatype is a MicroTorque specific extension of the Open Protocol standard.

The following unit types are defined:

Unit	Value
None	000
Newton meter	001
Percent	009
Degrees	050
Millinewton meter (see notes below)	090
RPM	101
Seconds	200
Degrees Celcius	251

The millinewton meter unit is a MicroTorque specific extension of the Open Protocol standard.

2.3 Special MTF 6000 considerations, general

All value strings returned by MTF 6000 Open Protocol are encoded as UTF-8. Some values where the MTF6000 can store a longer string than Open Protocol supports might be truncated (e.g. Controller Name). The string will always be truncated to the nearest whole utf-8 character.

All value strings sent to the controller using MTF6000 Open Protocol can be encoded as ASCII or UTF-8. Take care to get the message header length field right – it expresses length in **bytes**, not in characters.

No concept of programming control is implemented in MTF6000 Open Protocol. All implemented protocol commands are available to all clients.

If the MTF6000 controller receives a message that has a malformed header, it will discard the entire packet. If the length field in the OP header is incorrect (due to the client sending erroneous data), it is possible to end up in a state where the controller has to discard the following packet entirely in order to

reenter a functional session state. That is, after a malformed command is sent to the controller, depending on the exact nature of the malformed data, the controller can do one of the following

- Discard the current message (Could not even parse it as an OP message - no action, no ACK)
- Return a NAK (e.g. “Unknown mid”)
- ACK the current message but discard the following message (e.g. message was longer than length field indicated – this is discovered when the next message appears garbled).

Due to the last of these cases, it is recommended that mid 9999, keep alive, is sent at twice the minimum rate (every 7 seconds instead of every 15 seconds) as a keep alive message that follows some other, malformed, message, can be discarded, leading to the session closing due to timing out.

2.4 Special MTF6000 considerations, per mid

The following mids and revisions are supported by MTF6000:

MID	Revisions	Name	Special considerations
0001	6	Application Communication Start	Only supports revision 6.
0002	6	Application Communication Start ACK	Only supports revision 6. The following fields are supported: • Controller name • Controller software version • Tool software version • Controller serial number • Station ID • Station Name
0003	1	Application Communication Stop	N/A
0004	1	Application Communication NAK	Mid accepted field will always contain the requested mid number – no special handling for mid 0008 and mid 0009.
0005	1	Application Communication ACK	Mid accepted field will always contain the requested mid number – no special handling for mid 0008 and mid 0009.
0006	1	Application Data Message Request	Supports requests of the following mids: • 0215 (IO Device status) • 0217 (Relay Function) • 0221 (Digital Input function) • 2501 (Tightening Program download)

0008	1	Application Data Message Subscribe	Supports subscription of the following mids: • 0015 (Program selected) • 0035 (Job information) • 0071 (Alarm, controller will also send 0074 if this subscription is active) • 0217 (Relay Function) • 0221 (Digital input function) • 0900 (Trace data, controller will also send 0901 if this subscription is active) • 1201 (Tightening result overall data, controller will also send 1202 if this subscription is active) See notes on mid 0008 below
0009	1	Application Data Message Unsubscribe	Supports the same mids as for mid 0008
0010	1	Parameter Set ID Upload Request	N/A
0011	1	Parameter Set ID Upload Reply	N/A
0015	1	Parameter Set Selected	Selecting Pset 0 (no Pset selected) is permissible, and will result in a notification if subscribed to. The content of all fields except Pset ID are undefined in this case.
0018	1	Select Parameter Set	Only Pset numbers that are within the licensed range can be selected. Selecting Pset 0 (no Pset selected) is permissible.
0030	2	Job ID Upload Request	Only supports revision 2. The term “Job” is known as “Batch Sequence” in MTF6000.
0031	2	Job ID Upload Reply	Only supports revision 2. The term “Job” is known as “Batch Sequence” in MTF6000.
0035	2	Job Info	Does not support the Job batch mode field. Selecting Job 0 (no Job selected) is permissible, and will result in a notification if subscribed to. The content of all fields except Job ID are undefined in this case. The term “Job” is known as “Batch Sequence” in MTF6000.
0038	2	Select Job	Only supports revision 2. Only Job numbers that are within the licensed range can be selected. Selecting Job 0 (no Job selected) is permissible.The term “Job” is known as “Batch Sequence” in MTF6000.
0040	5	Tool Data Upload Request	Only supports revision 5.

0041	5	Tool Data Upload Reply	Only supports revision 5. The following fields are supported: • Serial number • Number of tightenings • Calibration date • Software version • Max Torque (expressed in mNm * 100) • Gear ratio • Max speed • Tool model
0071	2	Alarm	The following fields are supported: • Error code • ControllerReadyStatus • ToolReadyStatus • Time The term “Alarm” is known as “Event” in MTF6000. See notes on mid 0071 below.
0072	1	Alarm Acknowledge	The term “Alarm” is known as “Event” in MTF6000.
0074	1	Alarm Acknowledged On Controller	The term “Alarm” is known as “Event” in MTF6000. See notes on mid 0074 below.
0076	2	Alarm Status	The term “Alarm” is known as “Event” in MTF6000.
0078	1	Acknowledge Alarm Remotely On Controller	The term “Alarm” is known as “Event” in MTF6000.
0080	1	Read Time Upload Request	N/A
0081	1	Read Time Upload Reply	N/A
0082	1	Set Time	N/A
0150	1	Identifier Download Request	See notes on mid 0150 below
0157	1	Reset All Identifiers	N/A
0215	2	IO Device Status Reply	See notes on mid 0215 below
0217	1	Relay Function	The term “Relay Function” is known as “Digital Output Function” in MTF6000. See notes on mid 0217 below.

0221	1	Digital Input function	See notes on mid 0221 below
0224	1	Set Digital Input Function	See notes on mid 0224 below
0225	1	Reset Digital Input Function	N/A
0900	1	Trace Curve Data Message	See notes on mid 0900 below
0901	1	Traces Plot Parameters Message	See notes on mid 0901 below
1201	1	Operation Result Overall Data	See notes on mid 1201 below
1202	1	Operation Result Object Data	See notes on mid 1202 below
2500	1	Tightening Program Message Download	See notes on mid 2500 below
2501	1	Tightening Program Message Upload	See notes on mid 2501 below
9999	1	Keep Alive	See notes on mid 9999 below

2.4.1 Notes on mid 0008 Application data message subscribe

When subscribing to mid 0035, the controller will immediately respond with a mid 0005 ACK and a mid 0035 Job Information.

When subscribing to mid 0900, two separate mid 0900 messages will be sent for each tightening result, one for angle data and one for torque data. They will be immediately followed by one mid 0901 message.

When subscribing to mid 1201, only send alternative 0 (new data) is available. Mid 1201 will be followed by exactly one instance of mid 1202 containing the object data from one single controller/tool.

2.4.2 Notes on mid 0071 Alarm

Only those events that are saved in the event log on the controller will result in a mid 0071 alarm message. Note that a subscription on 0071 also results in 0074 Alarm Acknowledged on Controller being sent to the integrator.

The following error codes are supported:

Code	**Explanation**	**Recommended Action**
101	Supply voltage exceeded	Check power supply
102	Supply voltage too low	Check power supply
103	Internal 24V exceeded	Disconnect external equipment
104	Internal 24V too low	Disconnect external equipment
105	Internal 12V exceeded	Disconnect tool if connected

106	Internal 12V too low	Disconnect tool if connected
107	Internal 5V exceeded	Disconnect tool if connected
108	Internal 5V too low	Disconnect tool if connected
109	External 24V error	Disconnect external equipment
110	DC/DC temperature error	Replace the controller
120	Zero offset error, Current	Replace the controller
121	Zero offset error, Sensor	Replace the tool transducer or the tool
122	Motor current too low	Replace the tool
123	Motor current exceeded	Replace the tool
124	Motor driver voltage too low	Replace the tool
125	Motor driver short circuit	Replace the tool
126	Motor driver temperature exceeded	Replace the tool
130	EEPROM read error	Replace the controller
131	EEPROM write error	Replace the controller
132	Internal hardware error	Replace the controller
133	File system error	Replace IAM MT
134	USB host overcurrent	Disconnect USB devices
140	No tool connected	Connect tool
141	Tool not supported	Connect supported tool
142	Tool communication error	Replace the tool
143	Tool 5V error	Replace the tool
144	Tool 12V error	Replace the tool
150	Tool initializing error	Check that the tool is running freely during start up
151	Tool friction too high	Check that the tool is running freely during start up
152	Tool angle encoder error	Connect tool
153	Tool direction error	Connect tool
154	Bit was blocked	Check that the tool is running freely during start up
155	Tool motor temperature exceeded	Allow time for tool to cool down
156	Tool not supported	Replace the tool
201	Runtime parameter error	Restart the controller
202	Invalid controller parameters	Replace IAM MT
203	Invalid tool parameters	Replace the tool
204	Invalid I/O parameters	Replace IAM MT
205	Invalid identifier parameters	Replace IAM MT
206	Invalid Pset parameters	Replace IAM MT
207	Invalid Batch sequence parameters	Replace IAM MT
208	Invalid password parameters	Replace IAM MT

209	Tool statistics error	Replace the tool
210	Unable to load fonts from IAM MT	Replace IAM MT
220	Result database error	Replace IAM MT
221	Graph database error	Replace IAM MT
225	Reset result database	Result database was reset
226	Reset graph database	Graph database was reset
230	Pset not configured for connected tool	Run another Pset or change tool
231	Select error source	Change select source. Can be changed in ToolsTalk MT Controller settings
232	Select error busy	Wait until tool is not running
301	Tool connected	Press OK to initialize the tool
302	Tool initializing	Please wait
303	Tool software updating	Please wait
304	Tool software update error	Check tool cable or replace the tool
305	Tool calibration required	Calibrate the tool
306	Service required	Service the tool
310	Software has been updated	Press OK to continue
311	Software file was not found on USB	Copy software file to USB flash drive
312	Software update failed	Software file on USB may be broken. Update software file on USB and try again
313	Copying software	Please wait. Software is being copied to/from USB
320	Login expired	User was logged out
321	User has logged in	User was logged in
322	User has logged out	User was logged out
401	Tightening error	Press OK to continue
402	Batch sequence error	Press OK to continue
403	Batch sequence parameter error	Check controller to see which parameter is incorrect
404	Pset not supported	Select another Pset
901	IAM MT not present	Insert IAM MT
902	Failed to initialize IAM MT	Restart the controller or reconnect the IAM MT
903	IAM MT read error	Try again or change IAM MT
904	IAM MT write error	Try again or change IAM MT
905	Invalid software on IAM MT	Update software on IAM MT (manually from PC)
906	Flash programming error	Try again or replace the controller
907	Invalid license on IAM MT	Update software on IAM MT (manually from PC)

2.4.3 Notes on mid 0074 Alarm Acknowledged on Controller

This message will be sent to integrator if a subscription to mid 0071 was done.

2.4.4 Notes on mid 0150 Identifier Download Request

Identifiers downloaded to the controller will be handled in the same manner as scanned identifiers, i.e. through simple or advanced identifier scan settings. An identifier that does not match the identifier settings in the controller will not be accepted as input during the batch sequence – however, the controller will respond to the mid 0150 request with a mid 0005 ACK regardless.

2.4.5 Notes on mid 0215 IO Device Status

Note that this message contains the status of the *physical IO pins*, **not** the status of the *IO functions*. The IO device status can be requested using mid 0006. The mid 0215 reply currently only contains the IO status of the internal device, i.e. the state of the IO pins of the controller itself. The IO device ID field will therefore currently always contain "00".

2.4.6 Notes on mid 0217 Relay Function

All status changes for both tracking and non-tracking events will be sent.

The following output functions can be sent for mid 0217:

ID	Output function	Description
1	Ready	Ready to start
2	Busy	Busy performing an operation
3	Error	An error is preventing start
4	OK	Tightening OK
5	NOK	Tightening not OK
6	Batch OK	Batch OK
7	Event active	An event is active
8	Event blocking	Active event is blocking operation
9	Event clearable	Active event is acknowledgeable
10	Tool initializing	Tool is being initialized
11	Tool disabled	Tool is disabled
12	Vacuum enabled	Vacuum pump enabled / disabled
13	Start signal	Mirror of tool start signal
14	Loosening signal	Mirror of tool loosening trigger signal
15	Push to start signal	Mirror of tool push to start signal

16	Bseq output pin 1	Available for use in batch sequence
17	Bseq output pin 2	Available for use in batch sequence
18	Bseq output pin 3	Available for use in batch sequence
19	Bseq output pin 4	Available for use in batch sequence
20	Bseq output pin 5	Available for use in batch sequence
21	Bseq output pin 6	Available for use in batch sequence
22	Bseq output pin 7	Available for use in batch sequence
23	Bseq output pin 8	Available for use in batch sequence
24	Bseq OK	Batch sequence OK
25	Bseq error	Batch sequence error

2.4.7 Notes on mid 0221 Digital Input Function

All status changes for both tracking and non-tracking events will be sent.

The enumeration of digital input functions is specific to MTF6000. See 2.4.82.4.8.

2.4.8 Notes on mid 0224 Set Digital Input Function

The following input functions can be triggered using mid 0224:

ID	Input function	Description
1	Start tightening	Start tightening operation
2	Start tightening (hold)	Start tightening with stop on negative flank.
3	Start loosening	Start loosening operation
4	Start loosening (hold)	Start loosening with stop on negative flank.
5	Stop operation	Stop ongoing operation
6	Reset	Stop operation if busy, reset tightening result if idle
7	Disable communication	Disables all communication while set to high
8	Disable tool	Disable tool while set to high
9	Initialize tool	Initialize tool without confirmation
10	Disconnect tool	Disconnect tool
11	Clear event	Clear active event
12	Clear all events	Clear all active events
21	Start Batch sequence	Start Batch sequence
22	Abort Batch sequence	Abort Batch sequence

23	Increment batch	Skip a tightening in a Batch
24	Decrement batch	Redo a tightening in a Batch
25	Reset batch	Reset batch
26	Bseq input pin 1	Available for use in batch sequence
27	Bseq input pin 2	Available for use in batch sequence
28	Bseq input pin 3	Available for use in batch sequence
29	Bseq input pin 4	Available for use in batch sequence
30	Bseq input pin 5	Available for use in batch sequence
31	Bseq input pin 6	Available for use in batch sequence
32	Bseq input pin 7	Available for use in batch sequence
33	Bseq input pin 8	Available for use in batch sequence
34	Bseq input pin 9	Available for use in batch sequence
35	Bseq input pin 10	Available for use in batch sequence
36	Bseq input pin 11	Available for use in batch sequence
37	Bseq input pin 12	Available for use in batch sequence
38	Standby	Put controller into standby, resume on I/O activity.
39	Reboot	Reboot controller.
40	Increment Batch Sequence	Skip a step in a Batch Sequence
41	Decrement Batch Sequence	Redo a step in a Batch Sequence
42	Enable Vacuum	Enable Vacuum

2.4.9 Notes on mid 0900 Trace curve data message

Each mid 0900 message will contain the data for one trace type, currently supported types are angle and torque.

The message will contain the PID 02213 OR the PID 02214 coefficient for converting sample points. The data value of both these PIDs will be encoded as char[12], format is that output from printf_s("%.*e", 3, value), e.g. ”00001.234e+2”.

The resolution fields in the message will encode the time between two consecutive samples as char[12], format is that output from printf_s("%.*e", 3, value), e.g. ”00001.234e+2”, expressed in seconds.

The samples contained in the binary part are encoded as little endian 16 bit two complement signed integers, expressed as degrees for angle trace, and as mNm for torque trace.

Note that the controller settings influence which results are sent out through Open Protocol – only those tightening results and graphs that are stored locally on the controller are sent out.

2.4.10 Notes on mid 0901 Traces Plot Parameters Message

The following PIDs are defined for MTF6000:

ID	Content	Data Representation
30400	Graph marker seating index	Value is char[5], integer; zero based index of graph sample point
30401	Graph marker seating detected index	Value is char[5], integer; zero based index of graph sample point
30402	Graph marker snug index	Value is char[5], integer; zero based index of graph sample point
30403	Graph marker snug detected index	Value is char[5], integer; zero based index of graph sample point
30404	Graph marker final peak torque index	Value is char[5], integer; zero based index of graph sample point
30405	Graph marker final clamp torque index	Value is char[5], integer; zero based index of graph sample point
30420	Graph marker seating value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30421	Graph marker seating detected value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30422	Graph marker snug value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30423	Graph marker snug detected value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30424	Graph marker final peak torque value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30425	Graph marker final clamp torque value	Value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”

2.4.11 Notes on mid 1201 Operation result Overall data

When subscribing to mid 1201, only send alternative 0 is supported. Time stamps and indexes are ignored on subscription. Send object data field is ignored on subscription – object data (mid 1202) will always be sent.

Note that the controller settings influence which results are sent out through Open Protocol – only those tightening results and graphs that are stored locally on the controller are sent out.

2.4.12 Notes on mid 1202 Operation result object data

The following PIDs are defined for MTF6000:

ID	Content	Data Representation
30200	Result identifier	value is char[11], integer
30201	Result type	value is char[1]: - 0 = undefined, - 1 = tightening, - 2 = loosening
30202	Result code	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = cancel, - 3 = error
30203	Tightening start time	value is char[19], OP datetime, e.g. "2016-09-23:11:50:20"
30204	Error step number	value is char[3], integer
30205	Error code	value is char[2]: - 0 = undefined, - 1 = ok, - 2 = valuehigh, - 3 = valuelow, - 4 = triggerlost, - 5 = bitslipdetected,

		- 6 = rehitdetected, - 7 = noseatingdetected, - 8 = damagedthreaddetected
30206	Error value	value is char[2]: - 0 = undefined, - 1 = none, - 2 = totalTime, - 3 = stepTime, - 4 = torque, - 5 = stepAngle, - 6 = totalAngle, - 7 = clampTorque, - 8 = clampAngle, - 9 = tighteningAngle
30207	Controller serial number	value is char[12]
30208	Controller name	value is char[90], encoding is utf8
30209	Controller ID	value is char[11], integer
30210	Station name	value is char[90], encoding is utf8
30211	Station ID	value is char[11], integer
30212	Line name	value is char[90], encoding is utf8
30213	Line ID	value is char[11], integer
30214	Tool serial number	value is char[12]
30215	Tool name	value is char[30]
30216	Pset number	value is char[11], integer
30217	Pset name	value is char[90], encoding is utf8
30218	Pset revision	value is char[9], integer
30219	Pset created date	value is char[19], OP datetime
30220	Pset modified date	value is char[19], OP datetime

30221	Batch sequence number	value is char[5], integer
30222	Batch sequence name	value is char[90], encoding is utf8
30223	Batch sequence revision	value is char[9], integer
30224	Batch sequence created date	value is char[19], OP datetime
30225	Batch sequence modified date	value is char[19], OP datetime
30226	Batch sequence step count	value is char[5], integer
30227	Batch sequence step number	value is char[5], integer
30228	Batch size	value is char[3], integer
30229	Batch count	value is char[3], integer
30230	Peak torque	value is char[12], expressed in mNm, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30231	Total angle	value is char[12], expressed in degrees, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30232	Total duration	value is char[12], expressed in seconds, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30233	Tool temperature	value is char[12], expressed in degrees celsius, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30234	Total angle status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30235	Total duration status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low,

		- 3 = high
30236	Final torque type	value is char[1]: - 0 = undefined, - 1 = peak torque, - 2 = clamp torque
30237	Final torque	value is char[12], expressed in mNm, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30238	Final angle	value is char[12], expressed in degrees, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30239	Final angle type	value is char[1]: - 0 = undefined, - 1 = step angle, - 2 = clamp angle, - 3 = tightening angle
30240	Final report step	value is char[3], integer
30241	Final torque status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30242	Final angle status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30243	Torque tuning	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30244	Custom identifier key 1	value is char[48]

30245	Custom identifier value 1	value is char[100]
30246	Custom identifier key 2	value is char[48]
30247	Custom identifier value 2	value is char[100]
30248	Custom identifier key 3	value is char[48]
30249	Custom identifier value 3	value is char[100]
30250	Custom identifier key4	value is char[48]
30251	Custom identifier value 4	value is char[100]
30300	Step start	value is char[3], integer, denotes start of data relating to step NNN – all following PID fields contain data for step NNN
30301	Step type	value is char[1]: - 1 = thread engagement, - 2 = angle, - 3 = torque, - 4 = torque seating monitoring step, - 5 = seating control step
30302	Step peak torque	value is char[12], expressed in mNm, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30303	Step angle	value is char[12], expressed in degrees, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30304	Step duration	value is char[12], expressed in seconds, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30305	Step transition torque	value is char[12], expressed in mNm, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30306	Step tightening method	value is char[1], 0 = undefined, 1 = torque, 2 = angle

30307	Step tightening angle	value is char[12], expressed in degrees, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30308	Step clamp torque	value is char[12], expressed in mNm, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30309	Step clamp angle	value is char[12], expressed in degrees, format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30310	Step torque status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30311	Step angle status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30312	Step duration status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30313	Step tightening angle status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high
30314	Step clamp torque status	value is char[1]: - 0 = undefined, - 1 = ok,

		- 2 = low, - 3 = high
30315	Step clamp angle status	value is char[1]: - 0 = undefined, - 1 = ok, - 2 = low, - 3 = high

For each step in the tightening result, a PID field "30300" marks the start of the fields pertaining to that particular step – all PIDs following this PID are a part of that step, up until the next “30300” PID or the end of the message.

2.4.13 Notes on mid 2500 Tightening Program Download

The format for the mid 2500 message content is exactly the same as for mid 2501. This means that uploading a Tightening Program from the controller and saving it e.g. to file, and then sending the exact same data back to the controller using mid 2501 is permissible.

Modifying the contents of a tightening program when using these two mids is not recommended, with one exception – the Pset number field can be modified to download the program into a different Pset on the controller. For all other fields, the same limits on Pset values apply as usual, and changes to a Pset “offline” will quite probably result in a Pset that cannot be used on the controller. The Open Protocol implementation will not try to adjust any modified values to within acceptable parameter limits.

Note that only Pset numbers that are within the licensed range can be down- or uploaded.

If more than one connected clients simultaneously tries to download the same Pset, the Pset saved in the controller will be the last one downloaded. One client cannot partially overwrite the Pset data of another – saving Psets to the controller is performed using exclusive access.

2.4.14 Notes on mid 2501 Tightening Program Upload

When requesting a tightening program upload using mid 0006, the requested node type field is ignored.

The contents of the mid 2501 message is structured in the following manner: The PID fields at the root node correspond to the global Pset parameters. The root node will have one child node for each step in the Pset. Each child node will contain the PID fields relevant to that step.

Note that only Pset numbers that are within the licensed range can be down- or uploaded.

The following PIDs are defined for MTF6000:

ID	**Content**	**Data representation**

1000	Pset number	value is char[3], integer
30001	Step count	value is char[3] integer, e.g. "003" for 3 steps
30002	Pset name	value is char[91], encoding is utf8
30003	Pset revision	value is char[9] integer
30004	Pset created	value is char[19] OP datetime, e.g. "2016-09-23:11:50:20"
30005	Pset modified	value is char[19] OP datetime, e.g. "2016-09-23:11:50:20"
30006	Configured tool type	value is char[5], integer
30007	Configured tool name	value is char[90], encoding is utf8
30008	Minimum time	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30009	Maximum time	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30010	Min total angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30011	Max total angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30012	Loosening torque	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30013	Loosening angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30014	Loosening speed	value is char[4] integer, expressed in rpm
30015	Loosening vacuum enabled	value is char[1]: - 0 = no, - 1 = yes

30016	Loosening graph enabled	value is char[1]: - 0 = no, - 1 = yes
30017	Loosening start delay	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30018	Graph start step	value is char[2] integer
30019	Torque tuning	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in percent
30020	Final report step	value is char[2] integer
30021	Final report torque type	value is char[1]: - 0 = peak, - 1 = clamp torque
30022	Final report angle type	value is char[1]: - 0 = step, - 1 = tightening angle
30023	Bit slip detection	value is char[1], - 0 = disabled, - 1 = enabled
30024	Loosening max time	value is char[1], - 0 = disabled, - 1 = enabled
30025	Trigger lost torque	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30026	Custom field count	Value is char[1] integer
30027	Custom field 1 type	value is char[1], - 0 = general, - 1 = step

30028	Custom field 1 data	Value is char[2] integer
30029	Custom field 1 step number	Value is char[2] integer
30030	Custom field 1 step data	Value is char[2] integer
30031	Custom field 2 type	value is char[1], - 0 = general, - 1 = step
30032	Custom field 2 data	Value is char[2] integer
30033	Custom field 2 step number	Value is char[2] integer
30034	Custom field 2 step data	Value is char[2] integer
30035	Custom field 3 type	value is char[1], - 0 = general, - 1 = step
30036	Custom field 3 data	Value is char[2] integer
30037	Custom field 3 step number	Value is char[2] integer
30038	Custom field 3 step data	Value is char[2] integer
30039	Custom field 4 type	value is char[1], - 0 = general, - 1 = step
30040	Custom field 4 data	Value is char[2] integer
30041	Custom field 4 step number	Value is char[2] integer
30042	Custom field 4 step data	Value is char[2] integer
30043	Pset GUID	Value is char[36]
30044	Screw pickup enabled	value is char[1]: - 0 = disabled - 1 = enabled
30045	Screw pickup vacuum	value is char[1]: - 0 = disabled

		- 1 = enabled
30046	Screw pickup rotation	value is char[1]: - 0 = disabled - 1 = enabled
30047	Screw pickup timeout	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30048	Damaged thread detection	value is char[1], - 0 = disabled, - 1 = enabled
30100	Step type	value is char[1]: - 1 = thread engagement, - 2 = angle, - 3 = torque, - 4 = torque seating monitoring step, - 5 = seating control step
30101	Step speed	value is char[4] integer, expressed in rpm
30102	Step transition torque	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30103	Step transition angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30104	Step direction	value is char[1]: - 0 = CW, - 1 = CCW
30105	Step start delay	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30106	Min step time	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seonds

30107	Max step time	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in seconds
30108	Step vacuum enabled	value is char[1]: - 0 = no, - 1 = yes
30109	Step target angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30110	Step min torque limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30111	Step max torque limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30112	Step target torque	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30113	Step min angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30114	Step max angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30115	Step tightening angle trigger	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30116	Step min tightening angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30117	Step max tightening angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30118	Step seating angle displacement	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees

30119	Step gradient trigger point	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”
30120	Step min clamp torque limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30121	Step max clamp torque limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30122	Step min clamp angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30123	Step max clamp angle limit	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30124	Step final tightening method	value is char[1]: - 0 = torque, - 1 = angle
30125	Step clamp torque	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in mNm
30126	Step clamp angle	value is char[12], format is that output from printf_s(”%012.*e”, 3, value), e.g. ”00001.234e+2”, expressed in degrees
30127	Fast speed change	value is char[1]: - 0 = disabled, - 1 = enabled

2.4.15 Notes on mid 9999 Keep Alive

As per the Open Protocol specification, the timeout value of the Open Protocol for MTF6000 implementation is 15 seconds. I.e. a client must send a message or a keep alive more often than each 15:th second. However, it is recommended that the client sends keep alive at least each 7:th second.

2.5 Guide for previous users of Open Protocol

The Open Protocol for MTF6000 implementation supports a subset of the full Atlas Copco Open Protocol. While not all mids of the full Open Protocol are supported, the functionality itself sometimes is, using different mids. This is mostly due to the differences in the MTF6000 platform versus other platforms.

This is a short guide that tries to map use cases from other uses of Open Protocol to this specific implementation.

I want to use…	**For MTF6000 Open Protocol, instead use**
Mid 0012, 0013, parameter set data upload	Mid 2501, tightening program upload
Mid 0014, parameter set selected subscribe	Mid 0008, general subscribe
Mid 0017, parameter set selected unsubscribe	Mid 0009, general unsubscribe
Mid 0034, job info subscribe	Mid 0008, general subscribe
Mid 0037, job info unsubscribe	Mid 0009, general unsubscribe
Mid 0042, disable tool	Mid 0224 with input function 8
Mid 0043, enable tool	Mid 0225 with input function 8
Mid 0044, disconnect tool	Mid 0224 with input function 10
Mid 0050, VIN download	Mid 150, identifier download
Mid 0060, last tightening result subscribe	Mid 0008, general subscribe
Mid 0063, last tightening result unsubscribe	Mid 0009, general unsubscribe
Mid 0070, alarm subscribe	Mid 0008, general subscribe
Mid 0073, alarm unsubscribe	Mid 0009, general unsubscribe
Mid 0127, abort job	Mid 224 with input function 22
Mid 0128, job batch increment	Mid 224 with input function 23
Mid 0129, job batch decrement	Mid 224 with input function 24
Mid 0216, relay function subscribe	Mid 0008, general subscribe
Mid 0219, relay function unsubscribe	Mid 0009, general unsubscribe
Mid 0220, digital input function subscribe	Mid 0008, general subscribe
Mid 0223, digital input function unsubscribe	Mid 0009, general unsubscribe

Mid 0270, controller reboot request	Mid 0224 with input function 39