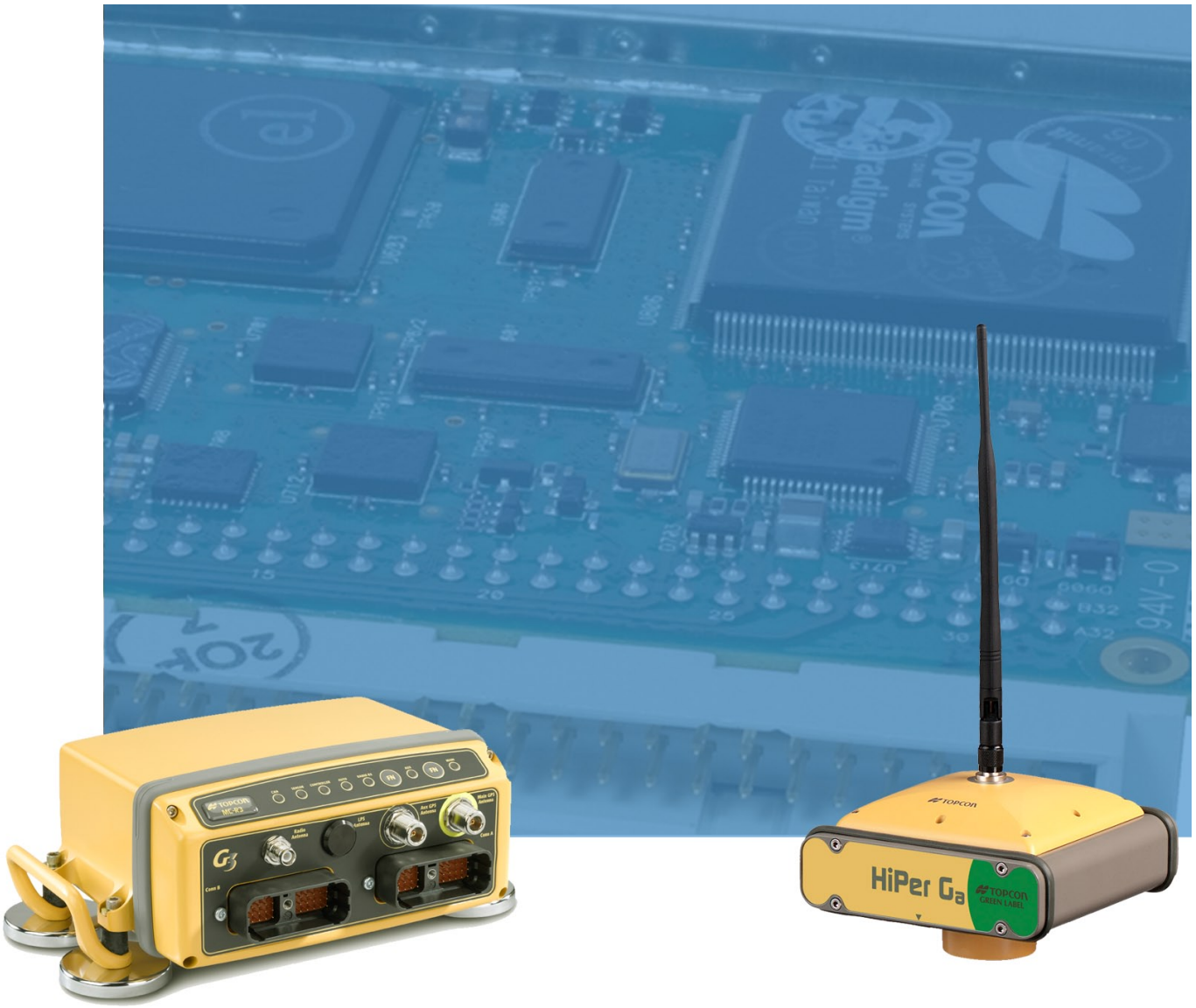
TOPCON

GRIL

GNSS Receiver Interface Language



Reference Manual

TOPCON
POSITIONING SYSTEMS

GRIL
Reference Manual

Part Number 27-040004-01
Rev. A

© Copyright Topcon Positioning Systems, Inc.
June, 2008

All contents in this manual are copyrighted by Topcon. All rights reserved. The information contained herein may not be used, accessed, copied, stored, displayed, sold, modified, published, distributed, or otherwise reproduced without express written consent from Topcon.

Table of Contents

Preface 1

Terms and Conditions 1
About This Manual 3
Intended Audience 3
Manual Organization 3
Additional Documentation 4
Manual Conventions 5
Obtaining Technical Support 7

1 GRIL Concepts and Terms 1-1

What Is GRIL? 1-1
Where Is GRIL Used? 1-2
What Can You Do with GRIL? 1-2
How GRIL Works 1-3
Objects 1-4
Parameters 1-4
Parameter Types 1-5
Command Statements 1-7
Syntax 1-8
Examples 1-9

Messages 1-12
Data Types 1-13
Special Values 1-14
Structure of Standard Messages 1-14
Structure of Non-standard Messages 1-15
Structure of Industry-approved Messages 1-15
Message Groups 1-16
Message Sets 1-16
Examples 1-17
Defining Message Generation 1-18
Flags 1-19
Parsing Message Stream 1-21
Synchronizing 1-21
Skipping to the Next Message 1-21
Performing a Checksum 1-22
Achieving Forward Compatibility 1-22

2 Commands 2-1

Command Specification 2-2
set 2-3
print 2-4
list 2-5
em 2-6
out 2-8
dm 2-9
create 2-11
remove 2-14
init 2-15
event 2-16
get 2-18
put 2-20
fld 2-22

3 Parameters 3-1

Parameter Specification 3-3
Identification Information about Receiver 3-5
Firmware version on GNSS board 3-5

Firmware version on Power board 3-6
Hardware version of GNSS board 3-6
Hardware version of Power board 3-7
Boot loader version 3-7
Board version 3-8
Board version of internal modem 3-8
Receiver serial number 3-9
Receiver electronic identifier 3-9
Receiver configuration word 3-10
Producer of GNSS board 3-10
Receiver model 3-11
Power Management 3-12
Turn receiver off 3-12
Sleep mode 3-13
Low power mode for processor 3-14
Reset receiver 3-15
Voltage of external power supply 3-16
Power source 3-17
Actual power source 3-19
Battery charge mode 3-20
Actual mode used for battery charging 3-21
Battery voltage 3-22
GNSS board voltage 3-23
Backup battery voltage 3-23
Output voltage of DC-DC converter for charge 3-24
Output voltage of DC-DC converter for power distribution 3-24
Output voltage on serial ports 3-25
Voltage across the board's 3-volt power stabilizer (digital) 3-25
Voltage across the board's 3-volt power stabilizer (analog) 3-26
Voltage across the board's 5-volt power stabilizer (digital) 3-26
Voltage across the board's 5-volt power stabilizer (analog) 3-27
External antenna voltage 3-27
External antenna current 3-28
Status of external antenna connection 3-29
Output current of DC-DC converter for charge 3-30
Power output on serial ports 3-31
Default for power output on serial ports 3-32
Power output to internal slots 3-33

Default for power output to internal slots 3-34
Managing internal slots 3-35
Managing specified internal slot 3-36
Default for specified internal slot 3-37
Parameters used only in discontinued receivers 3-38
Battery charging mode 3-38
Actual battery charging mode 3-39
Power output on ports A and B 3-40
Power output on port C 3-41
Memory and File Management 3-42
Memory 3-42
Size of receiver's RAM 3-42
Free memory in NVRAM 3-43
Access time 3-43
Number of blocks available in the receiver's memory card 3-44
Block size in the receiver's memory card 3-44
Working with Log Files 3-45
Information about all existing log files 3-45
Information about specific log file 3-46
Size of specific log file 3-46
Current log file 3-47
Size of current log file 3-48
Recover deleted log files 3-49
File system 3-50
Available memory for raw data logging 3-50
Free memory for raw data logging 3-51
File system block size 3-51
Number of blocks in file system 3-52
Maximum number of files in the file system 3-52
Number of files in the file system 3-53
Number of bad blocks 3-53
Verification of block recording 3-54
File system version 3-54
Time-tag of the last initialization of the file system 3-55
File system initialization status 3-55
Implicit message output period for MINTER and AFRM 3-56
File name prefix 3-57
Automatic File Rotation Mode (AFRM) 3-58

Enable AFRM 3-58
File rotation period 3-59
File rotation phase 3-60
File rotation counter 3-61
File rotation running counter 3-62
Automatically remove old files 3-63
Force file rotation event 3-64
Two Concurrent Log files 3-65
Configuring AFRM and MINTER to control two log files 3-66
Configuring AFRM and MINTER to control a specific concurrent log file 3-67
Frequency Source 3-68
Reference frequency source 3-68
External frequency value 3-69
External frequency status 3-70
Amplitude of external frequency signal 3-71
Eurocard J502 connector I/O pin support 3-72
Configure pins for input/output 3-72
Configure pin logical levels 3-74
Advanced Input Mode (AIM) 3-75
AIM Parameters 3-77
Input specifications 3-77
Input specifications of a particular element 3-78
Operation mode 3-79
Message Identifier 3-80
Value of skip field 3-81
Using the port as a source or destination 3-82
Cinderella Mode 3-83
Configuring Cinderella 3-83
Antenna Parameters 3-84
Antenna input mode 3-84
Current antenna input 3-85
OmniSTAR antenna input mode 3-86
Communication Link Parameters 3-87
Modem connection mode 3-87
Receiver's port GSM modem is connected with 3-88
PIN 3-89
Dial number 3-90
Monitoring modem status 3-91

Last detected modem error 3-92
Wait on data from modem timeout 3-93
Service word repeat period 3-94
GSM module 3-95
GSM debug 3-96
List of PPP parameters 3-97
GSM/GPRS modem type used in PPP connection 3-98
PPP connection status 3-99
PPP connection speed 3-100
IP address of PPP peer 3-101
Debug PPP 3-102
Authentication parameters 3-102
Usage of unencrypted password (PAP) 3-103
Usage of challenge handshake authentication protocol (CHAP) 3-104
PPP compression parameters 3-104
Van Jacobson compression 3-105
Connection-ID compression in Van Jacobson compression 3-106
List of GPRS parameters 3-106
GPRS dial number 3-107
GPRS user name 3-108
GPRS password 3-109
Packet Data Protocol (PDP) context parameters 3-109
PDP context identifier 3-110
Access point name 3-111
Extra Initialization string 3-112
List of dialup parameters 3-112
Dial number for accessing an Internet service provider (ISP) 3-113
User (login) 3-114
Dialup password 3-115
Dialup Initialization string 3-116
PPP errors 3-117
MTU 3-117
Ping 3-118
Idle timeout 3-119
List of CDMA parameters 3-120
CDMA dial number 3-120
CDMA user name 3-121
CDMA password 3-122

Ntrip Parameters 3-123
List of Ntrip parameters 3-123
Ntrip mode 3-124
Ntrip Server Parameters 3-125
List of Ntrip server parameters 3-125
IP address Ntrip server connects to 3-126
Port of Ntrip server 3-127
List of the Ntrip server mountpoint parameters 3-127
Mountpoint name 3-128
Mountpoint description 3-129
Password for an Ntrip server 3-130
Reporting Ntrip data port and input mode 3-130
Ntrip data port 3-131
Input mode for the current Ntrip data port 3-132
Output mode for the current Ntrip data port 3-132
Ntrip errors 3-133
Ntrip Client Parameters 3-134
Ntrip connection state 3-134
List of caster parameters 3-135
IP of Ntrip caster 3-135
Port of Ntrip caster 3-136
Mountpoint of Ntrip caster 3-137
User-ID for accessing data stream(s) of an Ntrip caster 3-138
Password for accessing data stream(s) of an Ntrip caster 3-139
Period of sending NMEA GGA to an Ntrip caster 3-140
Forcing sourcetable download 3-141
Accepting data over TCP/IP (Mobile mode) 3-142
List of TCP/IP parameters 3-142
Activating mobile mode 3-143
Selecting mobile service 3-144
Mobile state 3-145
List of IP parameters 3-146
IP address 3-146
Port of data stream 3-147
User-ID for accessing data stream 3-148
Password for accessing data stream 3-149
Period of sending NMEA GGA to a data source 3-150
List of rover parameters 3-151

IP address of the base to connect 3-151
Port of the base to connect 3-152
Password for the base to connect 3-153
List of base parameters 3-153
IP address of the base after establishing PPP connection 3-154
Port of the base 3-154
Password for the base 3-155
List of mobile data parameters 3-155
Mobile data port 3-156
Input mode for the current mobile data port 3-157
Ntrip errors 3-157
Debug mobile mode 3-158
Port Communication Parameters 3-159
Serial Port Parameters 3-159
Baud rate 3-159
Default baud rate 3-160
Hardware handshaking 3-161
RTS signal 3-162
CTS signal 3-162
Number of stop bits 3-163
Parity 3-164
Number of data bits 3-165
Infrared mode 3-166
Common Communication Parameters 3-167
Input mode 3-167
Output mode 3-168
Stream duplication 3-169
Echo mode 3-170
Echo-off sequence 3-171
Wrapped echo 3-172
Wrapped data size 3-173
TCP/IP Communication Parameters 3-174
IP address 3-174
Network mask 3-175
Gateway 3-176
MAC address 3-177
TCP port 3-178
TCP connection timeout 3-179

FTP port 3-180
FTP connection timeout 3-181
Telnet password 3-182
History Logger Parameters 3-183
Logging period 3-183
Output stream 3-184
Output Time-Frame Parameters 3-185
Time-frame mode 3-185
Time-frame period 3-186
Time-frame length 3-187
Time-frame delay 3-188
Job Programming 3-189
Parameters 3-190
Managing job scheduler 3-190
Restarting job scheduler 3-191
Activity flags for all jobs 3-192
Activity flag for specified job 3-193
Processing definitions for all jobs 3-194
Processing definition for specified job 3-195
Tag of a command set for specified job 3-196
Number of times to execute specified job 3-197
Output port for specified job 3-198
Command sets 3-199
Command set for specified job 3-200
Status for all jobs 3-201
Activity flags for jobs 3-202
Activity flag for specified job 3-203
Number of executions remaining for jobs 3-203
Number of executions remaining for specified job 3-204
Next job to execute 3-204
Time information for current and next jobs 3-205
Execution time for next job 3-205
Current time 3-206
Time left to next job 3-206
Notebook Parameters 3-207
Notes 3-207
Notes contents 3-208
Content of specified note 3-209

Change flags 3-210
Raw Data Management and Signal Processing Parameters 3-211
Raw measurement elevation masks 3-211
Number of epochs output to stream 3-212
Number of messages output to stream 3-212
GPS satellites used for signal acquisition 3-213
Specific GPS satellite used for signal acquisition 3-214
GLONASS satellites used for signal acquisition by frequency channel numbers (FCN) 3-215
Specific GLONASS satellite used for signal acquisition by FCN 3-216
Tracking GPS L2C-L and L2C-M codes 3-217
Tracking GLONASS C/A L2 3-218
Managing GPS P/L1 and P/L2 channel tracking 3-219
Managing GLONASS P/L1 and P/L2 channel tracking 3-220
GPS PRNs codes vs. GPS satellite numbers mapping 3-221
Specific satellite number to GPS PRN code 3-221
Almanac status for all GPS satellites 3-222
Almanac status for specific GPS satellite 3-223
Almanac status for all GLONASS satellites 3-224
Almanac status for specific GLONASS satellite 3-225
Minimum number of GNSS satellites for data output 3-226
Elevation mask for satellites tracking 3-227
Receiver's reaction to unhealthy satellites and satellites with elevation angles smaller than the specified mask 3-228
Raw measurement update rate 3-229
Position update rate 3-230
Current raw measurement update rate 3-231
Current position update rate 3-232
Pseudorange smoothing interval 3-233
Pseudorange smoothing filter type 3-234
Doppler smoothing interval 3-235
Bandwidth of Doppler smoothing filter 3-236
Smart Doppler smoothing 3-237
Nominal ionosphere correction smoothing interval 3-238
Minimum ionosphere correction smoothing interval 3-239
C/A code multipath reduction 3-240
C/A carrier phase multipath reduction 3-241
Anti-jamming 3-242

Configuring anti-jamming for GPS in manual mode 3-243
Configuring anti-jamming for GLONASS in manual mode 3-244
Oscillator Offset Reduction 3-245
Loop Parameters 3-246
C/A PLL bandwidth 3-246
C/A delay loop bandwidth 3-247
P/L1 and P/L2 guided loops bandwidth 3-248
Guiding and common lock loops order 3-249
Order of GPS L2C and GLONASS C/A L2 PLL 3-250
Bandwidth of GPS L2C and GLONASS C/A L2 PLL 3-251
Order of GPS L2C and GLONASS C/A L2 DLL 3-252
Bandwidth of GPS L2C and GLONASS C/A L2 DLL 3-253
Common loops mode 3-254
Individual loops bandwidth 3-255
Common loops bandwidth 3-256
Static Co-Op tracking 3-257
Adaptive guided loops 3-258
Additional (L2-specific) common loop 3-259
Common PLL order 3-260
Time to extrapolate missing observations 3-261
P-code lock loop bandwidth scale factor 3-262
Positioning Parameters 3-263
Satellite elevation mask for position computation 3-263
Minimum allowed SNR for satellites used in position computation 3-264
Position computation mode 3-265
Output of single point position 3-266
Output of code differential position 3-267
Measurement types used for position computation 3-268
GPS constellation for position computation 3-269
Specific GPS satellites used for position computation 3-270
GLONASS constellation (orbital slot) for position computation 3-271
GLONASS satellites (orbital slot) for position computation 3-272
GLONASS constellation (frequency channel #) for position computation 3-273
GLONASS satellites (frequency channel #) for position computation 3-274
Select satellite constellation for position computation 3-275
Correct pseudoranges for ionosheric errors when computing point position 3-276
Correct pseudoranges for tropospheric errors when computing point position 3-277

PDOP mask for position computation 3-278
Altitude used in position computation 3-279
Use of a fixed altitude in the position computation 3-280
Use of a user-defined geoid separation in the position computation 3-281
User-defined geoid separation in the position computation 3-282
Filtering Position Estimates 3-283
Position filter mode 3-283
Position filter type 3-284
Position filter length 3-285
Maximum allowed time gap between adjacent position estimates 3-286
Reset the position filter 3-287
Weighting mode for the position filter 3-288
Timing Parameters 3-289
Fixed time offset between GPS and GLONASS time scales 3-289
Use of a priori (fixed) offset between GPS and GLONASS time scales 3-290
Local time zone (offset) 3-291
Improved timing mode for a static receiver 3-292
Getting date and time from the receiver 3-293
Current receiver time 3-293
Current UTC date 3-294
Current UTC time of day 3-295
Current GPS week number 3-296
Current GPS time of week 3-296
Current GLONASS day number 3-297
Current GLONASS time of day 3-297
Date and time from real-time clock 3-298
Time since last reboot 3-298
Datum Parameters 3-299
General Format 3-300
Almost Fixed Format 3-301
Parameters 3-302
Receiver reference position in Cartesian coordinate system 3-302
Receiver reference position in geodetic coordinate system 3-303
The receiver's reference position in WGS-84 (for GPS) or PE-90 (for GLONASS) - Cartesian and geodetic coordinates 3-304
Current datum for position computation 3-305
User defined datum 3-306
PE-90 datum parameters 3-307

Switching between dynamic constants of PZ-90 and PZ-90.02 3-308
Pre-defined datum's parameters 3-309
Full transformation matrix for datum conversation 3-310
Reference ellipsoid parameters 3-311
7 parameter transformation coefficients 3-312
Reference ellipsoid and 7 parameter transformation data 3-313
Current grid system 3-314
Parameters of user-defined grid system 3-315
Parameters for transformation to local grid coordinate system 3-316
Specific map projection for computing grid coordinates 3-318
Antenna Database 3-319
Antenna type definitions 3-319
Antenna database versions 3-320
Antenna database entries 3-320
Antenna information for a specific antenna 3-321
CMR identifier for a specific antenna 3-322
Vector offset from ARP to L1 phase center for a specific antenna 3-322
Vector offset from L1 to L2 phase centers for a specific antenna 3-323
ID,l1_arp and l2_l1 for antennas with known CMR ID's 3-323
Antenna ID for a specific CMR ID 3-324
Vector offset from ARP to L1 phase center for a specific antenna with CMR ID 3-324
Vector offset from L1 to L2 phase centers for a specific antenna with CMR ID 3-325
Mapping between antenna ID's and CMR ID's 3-325
ID and CMR ID for each antenna in the database 3-326
ID and CMR ID for a specific antenna 3-326
Antenna CMR ID's 3-327
ID and CMR ID for a specific antenna CMR ID 3-327
MINTER Parameters 3-328
Appending data to a specific file 3-328
FN key mode 3-329
Initial dynamic mode 3-330
Automatic File Rotation Mode (AFRM) via MINTER 3-331
Receiver behavior and data recording on start-up 3-332
Governing Optimized Real Time Application (ORTA) Messages 3-333
C/A measurements in rM message 3-333
P/L1 measurements in rM message 3-334

P/L2 measurements in rM message 3-335
Version of rM message 3-336
Time scale for ORTA messages 3-337
Type of coordinates in rV message 3-338
Reference Station Parameters 3-339
Reference station output modes 3-339
Offsets from the marker to the L1 APC 3-340
Offsets from the marker to the ARP 3-341
Offsets from L1 APC to L2 APC 3-342
Cartesian reference receiver position for GPS 3-343
Cartesian reference receiver position for GLONASS 3-344
Cartesian reference receiver position for all satellite systems 3-345
Geodetic reference receiver position for GPS 3-346
Geodetic reference receiver position for GLONASS 3-347
Geodetic reference receiver position for all satellite systems 3-348
WGS84 Cartesian reference receiver position (for GPS) 3-349
WGS84 geodetic reference receiver position (for GPS) 3-349
PZ90 Cartesian reference receiver position (for GLONASS) 3-350
PZ90 geodetic reference receiver position (for GLONASS) 3-350
Coordinates of the ARP 3-351
View coordinates of the ARP 3-352
Maximum tolerance level for the discrepancy between computed and user-defined position of reference station 3-353
Reference position averaging mode 3-354
Stand-alone position averaging interval 3-355
Checking maximum distance between previous reference position and the current averaged position 3-356
Maximum distance between previous reference position and the current averaged position 3-357
Antenna type descriptor for RTCM 2.3 and 3.0 3-358
Antenna setup ID 3-359
Antenna serial number 3-360
RTCM 2.x-related Reference Station Parameters 3-361
RTCM 2.x output mode 3-361
Use of local datum for referencing differential corrections from RTCM messages 1,9,20,21,31,34 3-362
Satellite constellation for RTCM message types 18, 19, 20, 21 3-363
Maximum number of satellites for RTCM message types 18, 19, 20 and 21 3-364
Reference station health 3-365

Reference station identifier 3-366
User-defined text for RTCM message types 16 and 36 3-367
Include C/A measurements in RTCM message types 18, 19, 20 and 21 3-368
Include P/L1 measurements in RTCM message types 18, 19, 20 and 21 3-369
Include P/L2 measurements in RTCM message types 18, 19, 20 and 21 3-370
Use smoothed psuedoranges in RTCM message types 18, 19, 20 and 21 3-371
Enable delimiting characters at the end of RTCM messages 3-372
Specify delimiting character(s) at the end of RTCM message 3-373
RTCM version used at the reference station 3-374
Range rate correction to zero in RTCM messages 3-375
CMR-related Reference Station Parameters 3-376
CMR output mode 3-376
Reference receiver's motion state (this flag is used in CMR message type1) 3-377
Short station ID, COGO core, long station ID (used in CMR message type 2) 3-378
Reference station identifier 3-379
Include P/L2 measurements in CMR observation messages 3-380
Include P/L1 and C/A/L1 measurements in CMR observation messages 3-381
Message type used to transmit GLONASS measurements from the reference station 3-382
Antenna type used 3-383
Receiver type used 3-384
RTCM 3-related Reference Station Parameters 3-385
Reference station ID for RTCM 3 messages 3-385
String for proprietary RTCM 3.0 text message 3-386
TPS-related Reference Station Parameters 3-387
TPS output mode 3-387
Rover Station Parameters 3-388
Input mode 3-388
Receive status of the reference station position on the rover side 3-389
Received position of the reference station 3-390
Marker to ARP offsets 3-390
L1 antenna phase center to L2 antenna phase center vector offset 3-391
Alternative position for the reference station 3-392
Alternative marker to ARP offsets for the reference station’s antenna 3-393
Alternative L1 antenna phase center to L2 antenna phase center vector offset 3-394
Source of reference station position 3-395

Source of marker to ARP offsets 3-396
Reference station position used in RTK 3-397
Use GPS or GLONASS reference position 3-398
Antenna ID transmitted in RTCM message 23, RTCM 3.0, or CMR messages 3-399
Antenna serial number transmitted in RTCM message 23 and RTCM 3.0 3-399
Reference antenna setup ID transmitted in RTCM message 23 and RTCM 3.0 3-400
Reference antenna offset information transmitted in RTCM message 23 and RTCM 3.0 3-400
Reference antenna offset type transmitted in RTCM message 23 and RTCM 3.0 3-401
Reference antenna vector offset transmitted in RTCM message 23 and RTCM 3.0 3-402
RTCM 2.x-related Rover Station Parameters 3-403
RTCM input mode 3-403
Use the local datum for referencing differential corrections transmitted in RTCM messages types 1, 9, 20, 21, 31, 34 3-404
Use data from not monitored reference station 3-405
Verify the field sequence number from the RTCM message header 3-406
Multiple message indicator for RTCM messages 3-407
Complete epoch received logic for RTCM messages 3-408
RTCM 2.x version used at the rover 3-409
Reference point for antenna coordinates in RTCM 2.x messages 3-410
Reset RTCM 2.x decoders 3-411
CMR-related Rover Station Parameters 3-412
CMR input mode 3-412
Message type for GLONASS measurements 3-413
CMR version used at the rover 3-414
RTCM 3-related Rover Station Parameters 3-415
RTCM 3 input mode 3-415
TPS-related Rover Station Parameters 3-416
TPS input mode 3-416
RAIM Parameters 3-417
Turn RAIM on and off 3-417
RAIM alarm limit mode 3-418
Manual RAIM alarm limit 3-419
Parameters Related to NMEA Sentences 3-420
NMEA standard version 3-420

True Heading offset from HDT sentence 3-421
TalkerID in NMEA sentences 3-422
Limit the total number of satellites in GGA sentence to 12 3-423
Resolution Parameters for Fractional Data Fields in NMEA Sentences 3-424
Mantissa length for fractional seconds of UTC time 3-424
Mantissa length for fractional minutes of latitude/longitude 3-425
Mantissa length for fractional minutes for GGA sentence 3-426
Mantissa length for fractional minutes for GLL sentence 3-427
Mantissa length for fractional minutes for GNS sentence 3-428
Mantissa length for fractional meters of geoidal separation and orthometric height 3-429
Mantissa length for fractional part of speed in VTG sentence 3-430
Mantissa length of fractional residuals for GRS sentence 3-431
Pulse Per Second (PPS) Parameters 3-432
PPS signal generation 3-432
PPS signal period 3-433
PPS signal offset (milliseconds) 3-434
PPS signal offset (nanoseconds) 3-435
PPS signal synchronization 3-436
PPS signal reference time 3-437
Period of marked PPS signals 3-438
PPS signal synchronization to the selected reference time 3-439
Event Signal Parameters 3-440
Event Marker signal 3-440
Falling or rising edge of Event Marker signal 3-441
Event Marker's reference time 3-442
Use of the computed receiver clock offset when measuring XA, XB events in the selected reference time 3-443
Synchronize the receiver clock offset with an event signal 3-444
Status of the receiver clock synchronization 3-445
Code Differential Parameters 3-446
Source of differential corrections 3-446
Maximum age of code differential corrections used for position computation 3-447
Maximum age of ionosphere corrections used for position computation 3-448
Use P-code measurements in DGPS 3-449
Ionosphere-free differential correction mode 3-450
Range residual limit 3-451

Use of external source of DGNSS data for code differential solution 3-452
Use DGNSS data from SBAS for code differential solution 3-453
Use DGNSS data from VBS for code differential solution 3-454
Use DGNSS data from any source for code differential solution 3-454
Multi-base DGPS parameters 3-455
Multi-base mode 3-455
Use differential corrections from the nearest reference station 3-456
Run rover in mix mode 3-457
Averaging of individual position estimates or differential corrections from the different reference stations 3-458
Use the best differential corrections 3-459
Corrections (offsets) to the transmitted coordinates of the reference stations used in multi-base mode 3-460
Numerical offset used to distinguish between single and multiple reference stations in solution 3-462
Carrier Phase Differential Parameters 3-463
Position output mode 3-463
Position computation technique used in delay mode 3-464
Ambiguity fixing level 3-465
RTK initialization 3-466
Maximum number of satellites used in RTK 3-467
C/A measurements used by the rover in the position computation 3-468
P/L1 measurements used by the rover in the position computation 3-469
P/L2 measurements used by the rover in the position computation 3-470
Single difference carrier phase weighting scheme 3-471
Differential correction update interval (period) for reference station 3-472
Turn on/off the single-differenced ionospheric model 3-473
The threshold value for the RTK ionospheric modeling 3-474
The memory factor f for the float ambiguity filter 3-475
Extrapolate missing carrier phase measurements in delay mode 3-476
Force the RTK engine to verify fixed ambiguities in regular intervals 3-477
Period of extrapolation for reference station carrier phase measurements 3-478
Doppler in RTK processing 3-479
Period of ambiguities estimation 3-480
Enable measurements quality indicators 3-481
Correlation time for estimating the residual ionosphere 3-482
Minimum C/A signal-to-noise ratio 3-483
RMS for pseudorange measurements noise 3-484

RMS for carrier phase measurements noise 3-485
Maximum distance between Reference station and Rover 3-486
Timeout to wait for data from multiple reference stations 3-487
Minimum number of satellites for fixing integer ambiguities 3-488
User defined rover dynamics 3-489
RTK computational scheme version 3-490
Reset RTK engine 3-491
Factor (f) for standard deviation of the residual ionosphere 3-492
Environmental condition factor 3-493
Invalidate reference station coordinates 3-494
Pseudorange smoothing in RTK 3-495
Maximum extrapolation time 3-496
Maximum number of iterations for float ambiguity 3-497
Maximum number of iterations for fixed ambiguity 3-498
Track C/A L1 only 3-499
Penalty parameter 3-500
Use of FKP data from RTCM message 59 3-501
VRS mode 3-502
Halt RTK engine 3-503
Use of RTK data from external source 3-504
Use of RTK data from HP service 3-505
Use of any RTK data 3-506
Heading mode 3-507
Use of fixed length for the baseline measured 3-508
Value of fixed length for the baseline measured 3-509
Memory factor for smoothing heading angles 3-510
Multiple Reference Stations 3-511
Reference station ID 3-511
Selecting the nearest reference station automatically 3-512
Switching between reference stations 3-513
Maximum counter of attempts to stay the nearest reference station 3-514
SBAS Parameters 3-515
SBAS Parameters for G3- and TG3-based Receivers 3-515
Tracking SBAS satellites 3-515
Tracking specified SBAS satellite 3-516
Smoothing range rate corrections 3-517
SBAS Parameters for Other Receivers 3-518
Tracking selected SBAS satellite 3-518

Use of additional information from SBAS satellite 3-519
Use of true SBAS PRNs in SI message 3-520
Maximum age of SBAS corrections 3-521
Common SBAS Parameters 3-522
Use of message type 0 3-522
Applying ionospheric corrections from SBAS satellite 3-524
Method used for selecting maximum age of SBAS fast corrections 3-525
Maximum age of SBAS fast corrections in manual mode 3-526
OmniSTAR Parameters 3-527
VBS Version 3-527
VBS status flags 3-528
Fugro ID 3-529
Subscription start time 3-529
Subscription expiration time 3-530
Subscription extension time 3-530
Subscription time in hours 3-531
Duplicate OmniSTAR output stream 3-532
Clearing the NVRAM of the OmniSTAR module 3-533
Clearing OmniSTAR almanac in the OmniSTAR NVRAM 3-533
Clearing OmniSTAR NVRAM in the receiver's NVRAM 3-534
Clearing OmniSTAR sites in the OmniSTAR NVRAM 3-534
Clearing OmniSTAR subscription in the OmniSTAR NVRAM 3-535
BINEX Parameters 3-536
Contents of the 0x04 field for the 0x00 record 3-536
Contents of the 0x0f field for the 0x00 record 3-537
Enable output of the 0x04 field to the 0x00 record 3-538
Enable output of the 0x0f field to the 0x00 record 3-539
Enable output of the 0x17 field to the 0x00 record 3-540
Enable output of the 0x19 field to the 0x00 record 3-541
Enable output of the 0x1a field to the 0x00 record 3-542
Enable output of the 0x1b field to the 0x00 record 3-543
Enable output of the 0x1d field to the 0x00 record 3-544
Enable output of the 0x1f field to the 0x00 record 3-545
Enable output of all fields to the 0x00 record with a single parameter 3-546
Status of fields available for the 0x00 record 3-547
List of parameters used to control the 0x00 record 3-548
Receiver Options 3-549
Current value for specified option 3-549

Purchased value for specified option 3-550
Leased value for specified option 3-551
Expiration date for specified option 3-552
Complete information about specified option 3-553
Complete information about all options 3-553
AutoSeed Functionality 3-554
Parameters for working with the current point 3-554
Name of the current point 3-554
Identifier of the current point 3-555
L1PC coordinates of the current point 3-556
ARP coordinates of the current point 3-557
Automatic fill the fields of the current point 3-558
Parameters for working with the given point from the list 3-559
Name of the point 3-559
L1PC coordinates of the point 3-560
ARP coordinates of the point 3-560
Date and time of the point 3-561
Index of the point 3-561
ID of the point 3-562
Point creation status 3-563
Point protection status 3-563
Distance between the current point and the given point 3-564
Other parameters 3-565
Parameters of the points in the list 3-565
Total numbers of the points with the given flag 3-565
Adding current point to the list 3-566
Deleting points from the list 3-566
Sorting points in the list 3-567
Making the point protected from being overwritten 3-568
Miscellaneous Parameters 3-569
Enter exception mode on errors 3-569
Receiver board temperature 3-570
Configuring Messages 3-571

4 Messages 4-1

Message Specification 4-3
Using Special Characters in Message Identifiers 4-4

Time Scales Definition and Representation in Standard Messages 4-4
Attention Users of Topcon Tools™ and Other TPS Post-processing Software 4-5
Standard Binary Messages 4-6
Solution Types Definition used in Standard Binary Messages 4-6
Header Messages 4-7
JP – Topcon Positioning Systems file identifier 4-7
MF – Messages format and identifier 4-8
Time Messages 4-9
RD – Receiver date 4-9
RT – Receiver time 4-10
ET – Epoch time 4-11
UO – Universal Coordinated Time (UTC) parameters 4-12
GT – GPS time 4-13
NT – GLONASS time 4-14
WO – WAAS time 4-15
TO – Receiver reference time to receiver time offset 4-16
DO – Derivative of receiver time offset 4-17
GO – GPS time to receiver time offset 4-18
NO – GLONASS time to receiver time offset 4-19
BP – Rough indicator of receiver synchronization to reference time 4-20
Position and Velocity Messages 4-21
ST – Solution time-tag 4-21
PO – Cartesian position 4-22
VE – Cartesian velocity 4-23
PV – Cartesian position and velocity 4-24
PG – Geodetic position 4-25
VG – NEU velocity 4-26
SG – Position and velocity RMS errors 4-27
SP – Position covariance matrix 4-28
SV – Velocity covariance matrix 4-29
DP – Dilution of Precision (DOP) 4-30
BL – Base line 4-31
BI – Base station information 4-32
ha – Heading and pitch 4-33
PS – Position statistics 4-34
PT – Time of continuous position computation 4-35
Satellite Data Messages 4-36
SI – Satellite indices 4-37

NN – GLONASS satellite system numbers 4-38
EL – Satellite elevations 4-39
AZ – Satellite azimuths 4-40
RC – Full L1 C/A pseudoranges 4-41
rc – Delta L1 C/A pseudoranges 4-42
CC – L1 C/A pseudorange smoothing corrections 4-43
cc – Short L1 C/A pseudorange smoothing corrections 4-44
R1 – Full L1 P pseudoranges 4-45
r1 – Delta L1 P pseudoranges 4-46
1R – Relative L1 P pseudoranges 4-47
1r – Relative delta L1 P pseudoranges 4-48
C1 – L1 P pseudorange smoothing corrections 4-49
c1 – Short L1 P pseudorange smoothing corrections 4-50
R2 – Full L2 P pseudoranges 4-51
r2 – Delta L2 P pseudoranges 4-52
2R – Relative L2 P pseudoranges 4-53
2r – Relative delta L2 P pseudoranges 4-54
C2 – L2 P pseudorange smoothing corrections 4-55
c2 – Short L2 P pseudorange smoothing corrections 4-56
R3 – Full GPS L2C and GLONASS L2 C/A pseudoranges 4-57
3R – Relative GPS L2C / GLONASS L2 C/A pseudoranges 4-58
PC – Full L1 C/A carrier phases 4-59
pc – Short L1 C/A carrier phases 4-60
CP – L1 C/A carrier phases computed relative to pseudoranges from the RC message 4-61
cp – L1 C/A carrier phases computed relative to pseudoranges from the rc message 4-62
P1 – Full L1 P carrier phases 4-63
p1 – Short L1 P carrier phases 4-64
1P – L1 P carrier phases computed relative to pseudoranges from the RC message 4-65
1p – L1 P carrier phases computed relative to pseudoranges from the rc message 4-66
P2 – Full L2 P carrier phases 4-67
p2 – Short L2 P carrier phases 4-68
2P – L2 P carrier phases computed relative to pseudoranges from the RC message 4-69
2p – L2 P carrier phases computed relative to pseudoranges from the rc message 4-70

P3 – Full GPS L2C and GLONASS L2 C/A carrier phases 4-71
3P – GPS L2C and GLONASS L2 C/A carrier phases computed relative to pseudo-ranges from the RC message 4-72
DC – L1 C/A Doppler 4-73
D1 – L1 P Doppler 4-74
D2 – L2 P Doppler 4-75
D3 – GPS L2C and GLONASS L2 C/A Doppler 4-76
EC – L1 C/A signal-to-noise ratios (SNR) 4-77
CE – L1 C/A signal-to-noise ratios x 4 4-78
E1 – L1 P signal-to-noise ratios 4-79
1E – L1 P signal-to-noise ratios x 4 4-80
E2 – L2 P signal-to-noise ratios 4-81
2E – L2 P signal-to-noise ratios x 4 4-82
E3 – GPS L2C and GLONASS L2 C/A signal-to-noise ratios 4-83
TC – Time since last loss-of-lock on particular L1 C/A channel 4-84
TT – Time since last loss-of-lock on all L1 C/A channels 4-85
FC – L1 C/A signal lock loop flags 4-86
F1 – L1 P signal lock loop flags 4-88
F2 – L2 P signal lock loop flags 4-89
F3 – GPS L2C and GLONASS L2 C/A signal lock loop flags 4-90
SS – Satellite navigation status 4-91
ID – Ionospheric delays 4-92
GA – GPS almanac 4-93
GE – GPS ephemeris 4-95
NA – GLONASS almanac 4-97
NE – GLONASS ephemeris 4-99
WA – WAAS almanac 4-102
WE – WAAS ephemeris 4-104
GD – GPS raw navigation data 4-106
LD – GLONASS raw navigation data 4-107
PPS and Event Marker Messages 4-108
ZA, ZB – PPS offset 4-108
YA, YB – Time offset at PPS generation time 4-109
XA, XB – External event 4-110
Optimized Real-Time Application (ORTA) Messages 4-111
rE – Reference epoch 4-112
rM – Raw Measurements 4-114
rV – Position and velocity 4-118

rT – Receiver's clock offsets 4-121
Miscellaneous Messages 4-123
IO – Ionospheric parameters 4-123
LT – Message output latency 4-125
>> – Wrapping data 4-126
LH – Logging history 4-127
EV – Event data 4-129
OO – Oscillator offset 4-131
SE – Security 4-132
SM – Security for rM message 4-133
EE – Epoch end 4-134
ND – mmGPS data 4-135
Standard Text Messages 4-136
General Messages 4-138
DL – Data link status 4-138
GS – GPS SV status 4-140
NS – GLONASS SV status 4-142
RS – Reference station status 4-144
JI – Jamming Suppressor information 4-146
JE – Extended Jamming Suppressor information 4-148
JE – Extended Jamming Suppressor information 4-148
MS – RTCM 2.x status message 4-150
TX – Text message 4-151
RM – Results of RAIM processing 4-153
NP – Navigation position 4-155
MP – Position in given map projection or local coordinates 4-158
TR – Time residuals 4-160
TM – Clock offsets and their time derivatives 4-162
RP – Reference station parameters 4-164
RK – RTK parameters 4-166
AP – Position covariance matrix 4-168
AB – Baseline length 4-170
WS – SBAS status 4-172
XD – Data from NMEA XDR sentence 4-173
PM – Receiver parameters 4-174
Command Response Messages 4-175
RE – Receiver ACK reply 4-175
ER – Receiver NAK reply 4-176

Industry-approved Messages 4-177
NMEA sentences 4-177
GGA – Global Positioning System fix data 4-179
GLL – Geographic position – latitude/longitude 4-181
GMP – GNSS map projection fix data 4-183
GNS – GNSS fix data 4-185
GRS – GNSS range residuals 4-187
GSA – GNSS DOP and active satellites 4-189
GST – GNSS pseudorange error statistics 4-191
GSV – GNSS satellites in view 4-193
HDT – Heading, True 4-194
RMC – Recommended minimum specific GNSS data 4-195
ROT – Rate of turn 4-197
VTG – Course over ground and ground speed 4-198
ZDA – Time & Date 4-200
ATT – Attitude parameters 4-201
RTCM 2.x Messages 4-203
List of RTCM 2.x Messages Supported by TPS Receiver 4-203
RTCM 3.x Messages 4-205
CMR Messages 4-206
List of CMR Messages Supported by TPS Receivers 4-206
BINEX Records 4-207

5 Computing Checksum A-1

8-Bit Checksum A-1
16-Bit Checksum A-2

6 TPS Data Transfer Protocol B-1

Protocol Description B-1
Computing Checksum B-3

7 Free-form Events C-1

Preface C-1
Currently supported TPS Free-Form Events C-2
_ANT Event (ANTENNA TYPE) C-2
_ANH Event (ANTENNA HEIGHT) C-2

_DYM Event (DYNAMICS) C-2
_SIT Event (SITE) C-3
_CAN Event (CANCEL) C-3
_SAV Event (SAVE) C-3
_DSC Event (DESCRIPTION) C-4
_FEA Event (FEATURE) C-4
Additional Comments on _CAN and _SAV C-5

8 Satellite Navigation Status Codes D-1

9 Receiver Options E-1

Options Available for a TPS Receiver E-1
Obsolete Options E-9

10 Obsolete Parameters and Messages F-1

Obsolete Parameters F-1
Obsolete Messages F-2

11 Maintenance Parameters G-1

12 Reference Ellipsoids and Local Datums H-1

Reference Ellipsoids and Local Datums H-1
List of DATUMS Supported by TPS Receivers H-3

Notes:

Preface

Thank you for purchasing your Topcon receiver, survey product or accessory (the “Product”). The materials available in this manual (the “Manual”) have been prepared by Topcon Positioning Systems, Inc. (“TPS”) for owners of Topcon products. This Manual is designed to assist owners with the use of software (the “Software”) to be used with the Product and its use is subject to these terms and conditions (the “Terms and Conditions”).



Please read these Terms and Conditions carefully.

Terms and Conditions

USE Please study this manual carefully. The benefits this product provides can be greatly influenced by your applications knowledge. Always wear the required safety attire (safety shoes, hard hat, etc.) when operating or around equipment.

COPYRIGHT All information contained in this Manual is the intellectual property of, and copyrighted material of TPS. All rights are reserved. You may not use, access, copy, store, display, create derivative works of, sell, modify, publish, distribute, or allow any third party access to, any graphics, content, information or data in this Manual without TPS’ express written consent and may only use such information for the care and operation of your Product. The information and data in this Manual are a valuable asset of TPS and are developed by the expenditure of considerable work, time and money, and are the result of original selection, coordination and arrangement by TPS.

TRADEMARKS HiPer®, Topcon® and Topcon Positioning Systems™ are trademarks of TPS. Windows® is a registered trademark of Microsoft Corporation. Product and company names mentioned herein may be trademarks of their respective owners.

DISCLAMER OF WARRANTY EXCEPT FOR ANY WARRANTIES IN A WARRANTY CARD ACCOMPANYING THE PRODUCT, THIS MANUAL, THE PRODUCT, AND ANY ACCOMPANYING SOFTWARE ARE PROVIDED "AS-IS." THERE ARE NO OTHER WARRANTIES. TPS DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR USE OR PURPOSE. TPS AND ITS DISTRIBUTORS SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL, THE SOFTWARE, OR THE PRODUCT. SUCH DISCLAIMED DAMAGES INCLUDE BUT ARE NOT LIMITED TO LOSS OF TIME, LOSS OR DESTRUCTION OF DATA, LOSS OF PROFIT, SAVINGS OR REVENUE, OR LOSS OF THE PRODUCT'S USE. IN ADDITION TPS IS NOT RESPONSIBLE OR LIABLE FOR DAMAGES OR COSTS INCURRED IN CONNECTION WITH OBTAINING SUBSTITUTE PRODUCTS OR SOFTWARE, CLAIMS BY OTHERS, INCONVENIENCE, OR ANY OTHER COSTS. IN ANY EVENT, TPS SHALL HAVE NO LIABILITY FOR DAMAGES OR OTHERWISE TO YOU OR ANY OTHER PERSON OR ENTITY IN EXCESS OF THE PURCHASE PRICE FOR THE PRODUCT.

LICENSE AGREEMENT Use of the Software and any other computer programs or software supplied by TPS or downloaded from a TPS website (the "Software") in connection with a Topcon Product constitutes acceptance of these Terms and Conditions in this Manual and an agreement to abide by these Terms and Conditions. The user is granted a personal, non-exclusive, non-transferable license to use such Software under the terms stated herein and in any case only with a single Product or single computer. You may make one (1) backup copy of the Software. Otherwise, the Software may not be copied or reproduced. You may not assign or transfer the Software or this license without the express written consent of TPS. This license is effective until terminated. You may terminate the license at any time by destroying the Software and Manual. TPS may terminate the license if you fail to comply with any of the Terms or Conditions. You agree to destroy the Software and manual upon termination of your use of the Product. All ownership, copyright and other intellectual property rights in and to the Software belong to TPS. If these license terms are not acceptable, return any unused Software and the Manual.

CONFIDENTIALITY This Manual, its contents and the Software (collectively, the "Confidential Information") are the confidential and proprietary information of TPS. You agree to treat Confidential Information of TPS with a degree of care no less stringent that the degree of care you would use in safeguarding your own most valuable trade secrets. Nothing in this paragraph shall restrict you from disclosing Confidential Information to your employees as may be necessary or appropriate to operate or care for the receiver. Such employees must also keep the Confidential Information confidential. In the event you become legally compelled to disclose any of the Confidential Information, you shall give TPS immediate notice so that it may seek a protective order or other appropriate remedy.

WEBSITE; OTHER STATEMENTS No statement contained at the TPS website (or any other website) or in any other advertisements or TPS literature or made by an employee or independent contractor of TPS modifies these Terms and Conditions (including the Software License Agreement, Disclaimer of Warranty and limitation of liability).

SAFETY Improper use of a Topcon Product can lead to injury to persons or property and/or malfunction of the Product. The Product should only be repaired by authorized TPS warranty service centers. Users should review and heed the safety warnings in the manual accompanying the Product.

MISCELLANEOUS The above Terms and Conditions may be amended, modified, superseded, or canceled, at any time by TPS. The above Terms and Conditions will be governed by, and construed in accordance with, the laws of the State of California, without reference to conflict of laws.

About This Manual

Welcome to the GRIL Reference Manual!

This manual is designed to be a comprehensive guide to the GNSS Receiver Interface Language (GRIL) containing everything you need to know about GRIL in detail. The material in this manual gives exhaustive description of the language elements such as parameters, commands, and messages. It also provides numerous examples of their usage to make them easier to understand.

Intended Audience

This manual is intended for all programmers, system integrators, and engineers who are communicating with TPS receivers using GRIL, and are designing or implementing applications that use GRIL.

This manual assumes that you are familiar with GNSS and also have some knowledge of working with TPS receivers.

Manual Organization

The main contents of this manual are divided into four chapters.

Chapter #	Heading	Description
1	GRIL Concepts and Terms	Introduces GRIL describing its structure and terminology.
2	Commands	Contains a complete list of the GRIL commands and provides detailed description of their function and usage.
3	Parameters	Lists all the GRIL parameters and describes the tasks each parameter serves.

Chapter #	Heading	Description
4	Messages	TBD

Additional reference information about GRIL is contained in the following eight appendixes.

Appendix	Heading	Description
A	Computing Checksum	Explains how to calculate checksums for commands and messages.
B	TPS Data Transfer Protocol	Explains the data protocol used to transfer data to/from the receiver.
C	Free-Form Events	Provides complete description of the supported Free-Form Events.
D	Satellite Navigation Status Codes	Lists available navigation codes.
E	Receiver Options	Lists available receiver options.
F	Obsolete Parameters and Messages	Lists outdated parameters and messages.
G	Maintenance Parameters	Presents the list of debug parameters.
H	Reference Ellipsoids and Local Dataums	Lists the reference ellipsoids and local datums supported by the receiver.

Additional Documentation

Below is a list of manuals that can be useful while operating our GNSS products.

If you are looking for	Read this document	Part Number
A description on how to use the HiPer GD and GGD	HiPer GD/GGD Operator's Manual	7010-0626
A description on how to use the HiPer+	HiPer+ Operator's Manual	7010-0526
A description on how to use the HiPer Lite&HiPer Lite+	HiPer Lite & HiPer Lite+ Operator's Manual	7010-0557
A description on how to use the HiPer Pro	HiPer Pro Operator's Manual	7010-0681
A description on how to use the HiPer Ga/Gb	HiPer Ga/Gb Operator's Manual	7010-0816

If you are looking for	Read this document	Part Number
A description on how to use the Odyssey-E	Odyssey-E Operator's Manual	7010-0507
A description on how to use the Odyssey-RS	Odyssey-RS Operator's Manual	7010-0511
A description on how to use the GB-500	GB-500 Operator's Manual	31-050506-01
A description on how to use the GB-1000	GB-1000 Operator's Manual	31-050402-01
A description on how to use the GR-3	GR-3 Operator's Manual	7010-0736
A description on how to use the NET-G3	NET-G3 Operator's Manual	7010-0761
A description on how to use the Map-RT	GMS-110 Operator's Manual	7010-0727
A description on how to use the Map-HP	GMS-200 Operator's Manual	7010-0683
A description on how to use the GMS-2	GMS-2 Operator's Manual	7010-0752
A description on how to install/update the firmware in a TPS receiver	FLoader User's Manual	n/a

Manual Conventions

This manual uses the following conventions:

Example	Description
file rotation event	Indicates definitions or when it is necessary to emphasize that the marked term is technical jargon.
`RE035%% clp`	Indicates GRIL elements such as parameters, command statements, responses, and messages. Also indicates code examples.
<CR> or `<CR>`	Indicates a non-printing character(s).
`<port>`	In command statements, indicates an element that you should replace with a value.
[0.0...1800.0]	In command statements, indicates a closed interval for a set of values. As shown, [0.0...1800.0] will be all numbers between 0.0 and 1800.0, including 0.0 and 1800.0.

[0.0...5.0)	In command statements, indicates a half-closed interval for a set of values. As shown, [0.0...5.0) will be all numbers between 0.0 and 5.0, including 0.0 but except 5.0. Note that the numbers may come as close to 5.0 as it is allowed by the value format, including 4.9, 4.99 and so forth, but 5.0 is not included.
`a | b`	In command statements, indicates alternative choices between the given values. The values might be grouped within angle brackets.



Supplementary information that can help you configure, maintain, or set up a system.



Supplementary information that can have an affect on system operation, system performance, measurements, or personal safety.



Notification that an action has the potential to adversely affect system operation, system performance, data integrity, personal health.



Notification that an action *will* result in system damage, loss of data, loss of warranty, or personal injury.



UNDER NO CIRCUMSTANCES SHOULD THIS ACTION BE PERFORMED.

Obtaining Technical Support

If you need assistance and this reference manual has not provided an answer, contact TPS Customer Support.

Phone

To contact TPS Customer Support by phone, call:

1-866-4TOPCON (1-866-486-7266)
Monday through Friday, 8am to 8pm, Eastern Standard time

E-mail

To contact TPS Customer Support by e-mail, use the electronic mail addresses shown below:

For Questions Related To...	Use...
GPS+ and 3DMC	psg@topcon.com
If in doubt about which e-mail address to use for your particular question, e-mail the support group	support@topcon.com

Website

The Topcon Positioning Systems website provides current information about Topcon's line of products. The support area of the website provides access to various support information, including contact details, dealer locations, service, etc.

To access the Precision GPS area of the TPS website, use:

www.topcongps.com

Notes:

GRIL Concepts and Terms

This chapter gets you acquainted with GNSS Receiver Interface Language (GRIL). It introduces the fundamental terms and elements of the language and describes their syntax, semantics and other essential aspects of the language.

After reading this chapter, you will have a general understanding of the language structure and usage. Further chapters will provide information required to manage your receiver easily and effectively.

In this chapter:

- "What Is GRIL?" on page 1-1
- "How GRIL Works" on page 1-3
- "Objects" on page 1-4
- "Parameters" on page 1-4
- "Command Statements" on page 1-7
- "Messages" on page 1-12

What Is GRIL?

GRIL is a domain-specific interfacing language enabling you to effectively communicate with TPS receivers by accessing all of their capabilities and functions.

GRIL represents a generic receiver language structure for the entire range of GNSS receivers produced by Topcon. This language structure is receiver-independent and open to future modification or expansion.

GRIL is based on a unified approach allowing you to control a TPS receiver with an appropriate set of predefined language elements—commands and messages. There are no specific constraints on the number or type of the receiver commands and messages.

Designed with expandability in mind, GRIL allows you to develop applications that will be both backward compatible with older firmware versions and forward compatible with newer releases of firmware. This advantage means that your application will be able to work with the receiver regardless of the firmware loaded.

Where Is GRIL Used?

GRIL is used in any system that is communicating with a TPS receiver through one of its hardware interfaces (serial, parallel, USB, Ethernet, Bluetooth, etc.). Such a system normally uses a set of GRIL commands and messages to accomplish the required task.

A pair of typical applications where GRIL plays a very important part are:

- using hand-held controllers to communicate with the receivers during field operation in survey and RTK projects
- when downloading data from the receivers into desktop systems for further post processing.

In addition, post-processing software should recognize a certain set of GRIL messages for a correct parsing and further processing of the native TPS raw data files.

What Can You Do with GRIL?

GRIL is an easy-to-use, yet very powerful and versatile language allowing you an extensive, yet intuitive way to adjust the various receiver parameters for optimum performance.

Thanks to unchallenged GRIL flexibility, this language can be effectively used for the automatic or manual control of TPS receivers. For automatic control, you can create sets of command statements and program them to run at the desired time, thus making the receiver to perform multi-step operations automatically. For manual control, enter necessary GRIL commands into the receiver through a terminal, for example, using Manual Mode of the PC-CDU software.

Numerous important functions that you can achieve with GRIL include:

- Receiver file management
- Satellite tracking control
- Raw measurement data management and logging
- Receiver power management
- Survey control
- Receiver monitoring and diagnostics
- Connection with external devices.

These examples are just few from a long list of features that GRIL supports.

How GRIL Works

GRIL works by sending instructions, called *command statements*, to the receiver. When you run command statements, the receiver parses and directs them to the appropriate *objects*. They respond to the statements by performing the requested actions and generating the *messages* to report results.

Figure 1-1 depicts an example on how the appropriate object in the receiver responds to the user's request to display information about the current voltage for the batteries. Note that this example uses PC-CDU's Manual Mode as a user application for sending the command statements and viewing the receiver responds. Actually, you might use any terminal with a command-line interface or a dedicated GUI for sending the command statements and displaying the results.

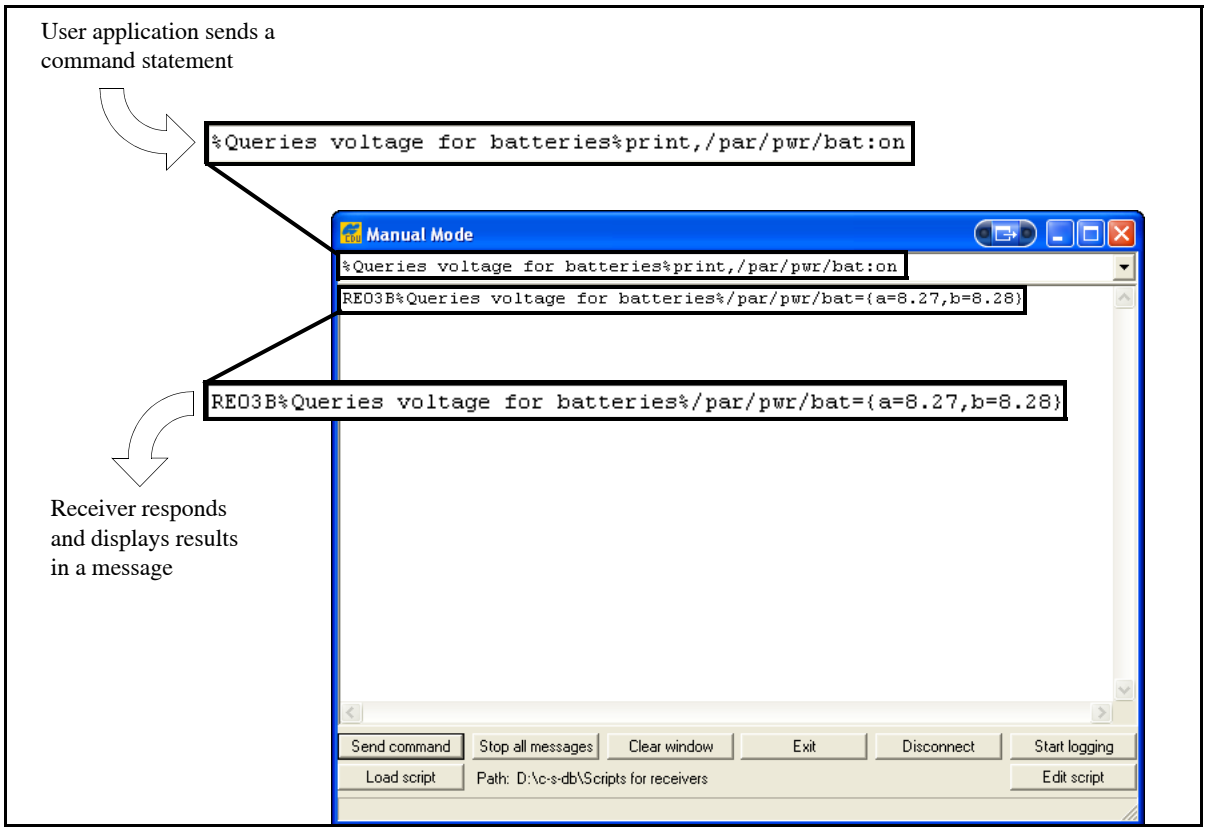


Figure 1-1. GRIL Typical Workflow

The following sections will explain each term defined above in greater detail.

Objects

In the context of the model that GRIL is designed on, a TPS receiver is identified with a set of named objects.

Object is defined as a hardware or firmware entity of the receiver that you can manipulate using the corresponding set of parameters and messages. Typical objects are receiver ports, memory card and raw data files on it, processes such as the receiver RTK engine, and so on.

Each object has one or more parameters which you specify in a command statement in order to access the object for management. The object responds to the command statement by performing the requested action and reporting the result of action in a message(s).

Parameters

As mentioned above, a receiver is considered as a set of objects in the context of the GRIL model. Each object is controlled with a unique set of parameters.

All parameters are logically orginized into a tree-like hierarchy starting at the single root node. For most computer users, this representation resembles the organization of files into directories (folders).

With such a hierarchy, a unique parameter begins with a root node followed by none, one, or more internal nodes, which in turn are followed by an end node or leaf. A parameter might consist of a single node. In this case, the node is considered as a root and leaf. The forward slash (/) is used as the delimiter between nodes. Figure 1-2 illustrates a typical structure of a parameter.

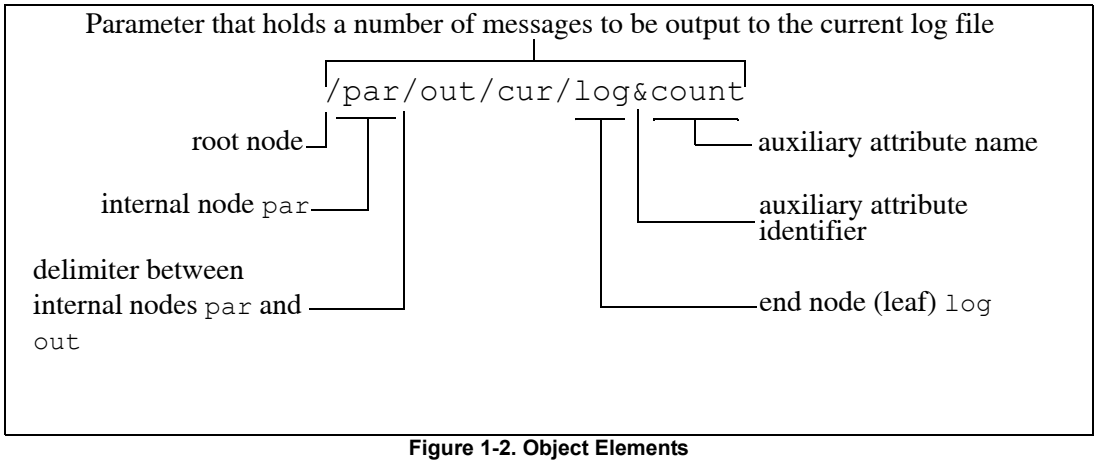


Figure 1-2. Object Elements

The receiver has five commonly used internal nodes that follow immediately after the / root node:

- par – for parameters that control firmware objects other than devices and files. An example of the parameter that includes this node is `/par/rcvid` – receiver ID.
- dev – for parameters that control hardware objects. An example of the parameter that includes this node is `/dev/blk/a&blocks` – memory card information.
- log – for parameters that control log files. An example of the parameter that includes this node is `/log/NAME` – contents of the file NAME.
- msg – for parameters that control various messages. An example of the parameter that includes this node is `/msg/def` – default set of messages.
- cur – for parameters that control current objects (e.g., current communication port and current log file). An example of the parameter that includes this node is `/cur/log` – name of the current log file.

Parameters should contain only alphanumeric characters or an underscore and should not begin with a digit.

As also shown in the figure above, a parameter may have one or more auxiliary attributes that follow immediately after the auxiliary attribute identifier (`&`). These attributes, if available, define properties of the addressed parameter.

Parameter Types

Each parameter that is defined by GRIL has one or more values. You assign a value to a parameter using a command statement. The assigned value has a *type*. The type determines the set of valid values that apply to the parameter.

In the context of GRIL, *primitive* and *composite* types are employed. Primitive types are fundamental types used in the language. They also serve as the building blocks for composite types, which allow assembling of values of different or the same primitive types.

Table 1-1 lists all primitive and composite value types supported by GRIL.

Table 1-1. Types of Parameters

Type	Definition
Primitive Types	
integer	Positive and negative numbers without a fractional part such as -1, -2, -3,...,1, 2, 3,..., and the number zero (0). They can be represented in three numeral systems, specifically decimal, octal, and hexadecimal. • decimal – start with an optional plus (+) or minus (-) sign, followed by one or more digits ranging from 0 to 9. Decimal integers cannot contain leading zeros if more than one digit is used. Examples of decimal integers: -1, 0, 31, 1035 • octal – start with an optional plus (+) or minus (-) sign, followed by the prefix zero (0) and one or more digits ranging from 0 to 7. Examples of oct integers: -01, 00, 037, 02013 • hexadecimal – start with an optional plus (+) or minus (-) sign, followed by the prefix 0x (or 0X), and one or more symbols ranging from 0 to 9, A through F, or a through f. Examples of hex integers: -0x1, 0x0, 0x1F, 0X40B

Table 1-1. Types of Parameters

Type	Definition
float	Positive and negative numbers that may contain both a whole and a fractional part. They are represented in the following format: start with an optional plus (+) or minus (-) sign, followed by one or more digits ranging from 0 to 9 with an optional period (.), and finally an optional exponent part consisting of the letter E or e, an optional sign, and one or more digits. Examples of floating values: 0.8, -543.25, 1.23E-04
boolean	Values with two possible states: *true* and *false*. The true states are represented by on, yes, y, while the false states are off, no, n
enumerated	A predefined set of values represented by alphanumeric identifiers. For example, ca, p1, p2, ionofree.
string	A sequence of alphanumeric symbols. An empty string and white spaces are allowed. When specifying a string with variable length, its minimum and maximum allowed lengths are given inside square brackets as *[min...max]*. Example of string: 3.1 Jan,24,2007 p1
Composite Types	
list	A collection of ordered values enclosed inside curly braces and delimited with commas. Values in the list may have different types and may also be empty. Examples of lists: {on,off,off}; {{off,off,off,0,{3600,0,0}},{y,n},{{1.00}},{{1.00}}}; {,off}
array	A collection of indexed values of the same type. Empty values are allowed. An individual value is accessed by specifying the value's position (or its index) in the array, which is a decimal integer. GRIL utilizes zero-, one-, and two-based arrays, for which the starting value has index zero, one, or two, respectively. When referring to an array, the following notation is used: array *[N...M]* of *type*, where *N* – the index of the first value in the array, *M* – the index of the last value in the array, *type* – type of values in the array. Visually, arrays look almost like lists. Examples of arrays: {n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n}; {on,on}

Table 1-1. Types of Parameters

Type	Definition
time_spec	An alphanumeric combination of characters used to denote date&time. This combination represents in the following format: DdHHhMMmSSs, where D, HH, MM, and SS are either integer numbers in corresponding range or a special symbol "__" (double underscore) used as a wildcard character. • D – number of day inside a week [1..7]. 1-Sunday ... 7-Saturday. • HH – number of hour inside a day [0..23]. • MM – number of minute inside an hour [0..59]. • SS – number of second inside a minute [0..59]. Examples of valid time_spec are: _4d17h40m18s __d__h00m__s The `print` command always outputs time_spec in its canonical form. The `set` command, however, might also accept time_spec in simplified forms. The rules for the `set` command are as follows: • If a field is omitted, it is assumed to be "__". • Single underscore is the same as double underscore. • Any number of integer numbers is accepted. For example, empty time_spec is equal to `__d__h__m__s`, and `8h2s` is equal to `__d08h__m02s`.

Command Statements

A *command statement*, which is the minimal entity the receiver can handle, contains an instruction for a receiver to perform. When you want to request an action or result from the receiver, you send a command statement(s) to the receiver where it is parsed and executed by the command interpreter. In response to the executed command statements, the receiver usually generates message(s).

Each command statement is aimed at a *target*, which is the receiver parameter that replies to the statement. Note however that not all command statements can be utilized with all parameters. When specifying a command statement, you must be sure to supply the correct command with a parameter you want to control.

Since this section focuses on command statements, the following discussion goes on to describe their basic syntax elements and explain how those elements are put together to form up command statements.

For the sake of brevity, a notation *command* will be used when referring to a command statement hereinafter in the manual.

Chapter 2 contains an exhaustive reference of the GRIL commands.

Syntax

A complete command consists of optional *prefix*, optional *body*, optional *checksum*, and a *termination character(s)*. The term *optional* means that a command may be issued with or without any of the elements.

`[<prefix>][<body>][<checksum>]<EOC>`

- prefix – a command element that is made up of two percent signs (`%%`) and an arbitrary set of alphanumeric characters between them. This combination is called a *command identifier*. It can be added to any command or used alone.

  Any command with an identifier will always generate a response from the receiver. The identifier, if present, is copied unchanged by the receiver into the response message for the command. A command that contains only an identifier is also allowed. In this case, the receiver will generate a response message to the command containing only the identifier. The alphanumeric part between the percent signs may also be empty (i.e., `%%` is a valid identifier).



It is strongly recommended that command identifiers be used whenever possible, since they allow you to effectively manage the commands by providing easy association of the issued commands and the corresponding replies.

- body – command's key element that is made up of a command name followed by none, one or more arguments. The command name denotes an action to be made. Arguments comprise receiver parameters, their values, and options (if any). Command names and parameters may have one or more options (option list) associated with them. In each command, the command name, parameter, and its value are separated by commas (`,`) whereas options are delimited from them with colons (`:`). When applying the options, the following rules should be observed:
  - The option list is usually surrounded by curly brackets {}.
  - If the option list consists of only one element, curly brackets surrounding the option list may be omitted. For example, {elem1:{10},elem2:{20}} is equivalent to {elem1:10,elem2:20}.
  - If the option list is attached to a list of elements, options will be set for each element in the list. For example, {elem1,elem2}:10 is equivalent to {elem1:10,elem2:10}.
  - If an option is attached to a given element, then this option takes precedence over any option attached to a surrounding list.
  - If no options are specified, default values are assumed.

  Possible options, their relative positions, meaning and default values are described in the corresponding sections of Chapter 2.
- checksum – a command element that is used to ensure the integrity of a command before executing it. The checksum is an 8-bit value formatted as a 2-byte hexadecimal number. It is appended to the end of a command with the @ character. Before running a command with the checksum, the receiver will compare the input checksum against that computed by the firmware. Checksum is computed starting with the command's first non-blank character until and including the @ character is reached. For information about computing 8-bit checksum, see "Computing Checksum" on page A-1.
- terminator – the indicator of the end of a command (EOC). The command interpreter recognizes the following basic termination characters: the carriage return (CR), the line feed (LF), and the CR/LF combination and treats them uniformly. In addition, three extended end-of-command

conventions can be used, specifically the semicolon (;), two ampersands (&&), and two vertical bars (||). These extended EOCs allow you to specify several commands on the same line[1] and to have different execution patterns.

– If a command ends with the semicolon (;), then the next command on the same line will run regardless of the result returned by the previous command.

– If a command ending with two ampersands (&&) is successful (i.e., returns no error), then the next command on the same line will run.

– If a command ending with two vertical lines (||) is unsuccessful (i.e., returns an error), then the next command on the same line will run.

The commands are case-sensitive. This means that receiver distinguishes uppercase from lowercase characters and treats them differently.

Any number of spaces, or blanks, are allowed before and after each element in a command. The command interpreter normally ignores these spaces (blanks) unless a checksum mechanism is invoked.

If you start a command with a hash mark (#), the receiver will regard all further input on the line as comment and will not interpret the input as commands until the end of the line.

Response to a command is usually put to a *command response message*. When a command has been accepted successfully, the receiver responds with RE message provided the command identifier (%%) has been used with the command. When a command is not accepted, the receiver always responds with ER message. For examples of these messages, see Table 1-2. For information about the format of these messages, see "Command Response Messages" on page 4-175. After generating a response message, the receiver might start outputting other messages if requested by the command.

Examples

This topic provides some examples of commands to make the concept information above more concrete. Do not include non-printing characters (i.e., <CR> and <LF>) to the commands as they are shown in the examples below. They are just reminders that each command must be ended with a valid EOC. Throughout "Commands" on page 2-1 and "Parameters" on page 3-1, you will find more examples on commands usage.

Table 1-2. Examples of Commands and Replies

Task	Command (C) and Reply (R)
Check that the receiver replies to the commands by sending only command identifier.	C: %%<CR> R: RE002%%<CR><LF>
• Enable output of NMEA sentence GGA to the receiver's port D.	C: em,/dev/ser/d,/msg/nmea/GGA<CR> R: The receiver will not generate a reply because the command is accepted.

1. Currently, GRIL supports lines of up to 256 characters in length.

Table 1-2. Examples of Commands and Replies (Continued)

Task	Command (C) and Reply (R)
• Force the receiver to generate a reply to the above task.	C: %%em,/dev/ser/d,/msg/nmea/GGA<CR> R: RE002%%<CR><LF>
• Query receiver board temperature.	C: %%print,/par/dev/thermo/out<LF> R: RE007%% 42.8<CR><LF>
• Make a reply to the above task more meaningful by adding comment between %%.	C: %Celsius%print,/par/dev/thermo/out<LF> R: RE013%Celsius% 42.4<CR><LF>
• Query the size of the receiver's RAM and force the receiver to generate a reply.	C: %%print,/par/rcv/mem<CR><LF> R: RE007%% 4096<CR><LF>
• Make a reply to the above task more meaningful by adding comment between %%.	C: %RAM_in_KB%print,/par/rcv/mem<CR><LF> R: RE010%RAM_in_KB% 4096<CR><LF>
• Query the list of parameters belonging to /par/pwr/charge with their current values.	C: %%print,/par/pwr/charge<CR> R: RE00C%%{auto,off}<CR><LF>
• Query the list of the same parameters as above together with their names and current values.	C: %%print,/par/pwr/charge:on<CR> R: RE027%%/par/pwr/charge={bat=auto,curbat=off}<CR><LF>
• Specify PDOP for position computation. Note the correct usage of blank spaces between command elements.	C: %% set , /par/pos/pdop , 25<CR> R: RE002%%<CR><LF>
• Specify PDOP for position computation. Note the incorrect usage of blank space within a command element.	C: %%set,/par/ pos/pdop,25<CR> R: ER01A%%{2,,wrong 1st parameter}<CR><LF>
• Set a wrong value for position computation mode.	C: set,/par/pos/mode/cur,abs<CR> R: ER018{2,,wrong 2nd parameter}<CR><LF> The receiver generates a reply even without %%, because the commad is not accepted.
• Query non-existent parameter.	C: print,/par/pos/mode/glo<CR> R: ER018{2,,wrong 1st parameter}<CR><LF>

Table 1-2. Examples of Commands and Replies (Continued)

Task	Command (C) and Reply (R)
• Query non-existent parameter and add %%.	C: %%print,/par/pos/mode/glo<LF> R: ER01A%%{2,,wrong 1st parameter}<CR><LF>
• Query GNSS board version with the correct checksum.	C: %%print,/par/rcv/ver/board@64<CR><LF> R: RE008%% EG3_2
• Query GNSS board version with the incorrect checksum.	C: %%print,/par/rcv/ver/board@67<CR><LF> R: ER00E%%{4,checksum}
Instruct the receiver to do the following: • Add NMEA GGA to the default message set. • If successful then create a log file with an automatically generated name. • If successful then set the elevation mask to 15 degrees for data outputted to the log file. • If successful then enable the output of default set of messages to the log file.	C: %1%create,/msg/def/nmea/GGA&&%2%create,:a&3&set,/par/out/elm/cur/log,15&4&em,/cur/log,/msg/def<CR> R: RE003%1%<CR><LF> R: RE003%2%<CR><LF> R: RE003%3%<CR><LF> R: RE003%4%<CR><LF>

Messages

Another chief element of the language is its messages. A *message* is the receiver output product that contains a set of data in a specific GRIL format. These data include diagnostic information such as battery voltage, raw data measurements such as pseudoranges and carrier phases, and any other information that receiver can produce.

The receiver generates messages in response to the commands it receives. The messages are then conveyed to the specified destination for further manipulation. *Destination* could be a port or log file.

GRIL supports a broad number of messages in binary and text (ASCII) forms. Some of these messages are designed to be TPS proprietary while the others comply with the standards accepted by the industry or third-party manufacturers.

The language architecture divides these messages into standard TPS, non-standard TPS, and industry-approved messages.

Standard TPS messages include TPS proprietary binary and ASCII messages. These messages are fundamental entities supported by the message handler in the receiver. Their syntax and semantics are precisely defined and optimized providing you with the most efficient and reliable way of interacting with TPS receivers. For details about the structure of these messages, see “Structure of Standard Messages” on page 1-14.

Non-standard TPS messages include only ASCII messages. Currently, GRIL does not define any non-standard TPS messages except for a special kind of utility messages which will be described in “Structure of Non-standard Messages” on page 1-15.

Industry-approved messages include:

- NMEA – Interface standard NMEA 0183 that was developed by the National Marine Electronics Association (NMEA). It defines electrical signal requirements, data transmission protocol and timing, and specific ASCII messages (sentences) for communications between various electronic devices, including GNSS receivers. In GRIL, NMEA sentences might also be considered as non-standard TPS messages, because they formally comply with the format defined for non-standard messages. GRIL supports versions 2.1, 2.2, 2.3, 3.0 of this standard. See more details about the structure of non-standard TPS messages on page 1-15.
- RTCM 2.x, 3.x – Standard of binary messages that was developed by the Radio Technical Commission for Maritime Services Special Committee 104. This standard guarantees commonality and efficiency in transmitting differential data between the reference and rover receivers even from various manufacturers. GRIL supports versions 2.1, 2.2, 2.3, 3.0, 3.1 of this standard.
- BINEX – Standard of binary messages that was developed by UNAVCO, a non-profit, membership-governed consortium of research institutions. BINEX from BINary EXchange, was designed to allow encapsulation of all (or most) of the information currently allowed for in:
  - RINEX observation, navigation, and meteorological files,
  - IONEX,
  - SP3,

– SINEX, and so on.

Plus other GNSS-related data and metadata as encountered, including next-generation GNSS.

- CMR/CMR+ – The public version of standard for binary messages that was developed by Trimble Navigation Limited for RTK surveying.

For information about the structure and meaning of the messages, see "Structure of Industry-approved Messages" on page 1-15.

When you request messages to be sent to a desired destination, the receiver makes up a message stream consisting of the requested messages. If this stream comprises only standard TPS and non-standard TPS messages, it is called an *ordinary message stream*. Its structure and parsing are completely defined. If you add any other messages to the ordinary message stream, for example, from the industry-approved group or arbitrary messages, you will break its standard structure. As a result, you cannot treat this stream as an ordinary one and should apply your own parsing rules to the stream.



NMEA sentences do not break the structure of the ordinary message stream.



The flexible nature of the language allows seamless integration of any messages to an ordinary message stream without destroying its structure and rules. If you use any messages that might break the ordinary message stream, encapsulate them to a dedicated standard message >>. This will ensure consistency and easy analysis of the stream. See page 4-126 for a structure of >> message.

Further topics explain data types, special values, structures, sets and groups, rules for generation and parsing of the standard and non-standard TPS messages.

Data Types

Standard TPS messages use the following data types for storing the values.

Table 1-3. Data Types in Standard TPS Messages

Data Type	Description
a1	1-byte ASCII character
i1	1-byte signed integer
i2	2-byte signed integer
i4	4-byte signed integer
u1	1-byte unsigned integer
u2	2-byte unsigned integer
u4	4-byte unsigned integer
f4	single-precision floating-point value; 4 bytes in length; complies with IEEE-754

Table 1-3. Data Types in Standard TPS Messages (Continued)

Data Type	Description
f8	double-precision floating-point value; 8 bytes in length; complies with IEEE-754
str	zero-terminated sequence of ASCII characters

Special Values

For TPS standard binary messages, some of their integer and floating-point entities may contain special values. These values are intended to warn you when the meaningful data for the entities are unavailable. Entities that might contain these special values are shown in **`bold`** in the message body.



Always check for special values, when you parse a message that might contain them.

Table 1-4 lists special values that might appear in various types of data.

Table 1-4. Special Values in TPS Binary Messages

Data Type	Special Value	Hex Representation
i1	127	7F
u1	255	FF
i2	32767	7FFF
u2	65535	FFFF
i4	2147483647	7FFF_FFFF
u4	4294967295	FFFF_FFFF
f4	NaN	7FC0_0000
f8	NaN	7FF8_0000_0000_0000

Structure of Standard Messages

Each standard binary and text TPS message begins with its *identifier*. A message identifier is followed by the *length* of the message body, which in turn is followed by the message *body* itself.

- Identifier – Specifies a unique message tag. It comprises two ASCII characters in the range from 0 to ~ (i.e., decimal ASCII codes in the range from 48 to 126).
- Length – Specifies the length of the message body in bytes. It comprises three uppercase hexadecimal digits. The maximum length of message body is 4095 (0xFFF) bytes.
- Body – Specifies the contents of data in the message. There are no restrictions on the contents of the message body. Recall that a message's body can have a maximum of 4095 characters.

Chapter 4 provides detailed description of these elements for each standard TPS message.

Structure of Non-standard Messages

Non-standard TPS messages do not meet the standard message format's requirements. The main purpose that non-standard messages serve is that you can insert them between standard TPS messages thus generating a combined stream of standard and non-standard messages.

These messages have the following elements.

- Identifier – Specifies a unique message tag. It comprises a single ASCII character in the range from ! to / (i.e., decimal ASCII codes in the range from 33 to 47).



The character $ is already reserved for NMEA sentences. Do not use this character as an identifier in any other non-standard TPS messages.

- Body – Specifies the contents of data in the message. It comprises an arbitrary number of ASCII characters except <CR> and <LF> characters, and an identifier used in the message.
- EOM – Signals the end of a message. An end-of-message indication might comprise an arbitrary combination of <CR> and <LF> characters.

There are non-standard messages of special purpose which comprise only <CR> and/or <LF> characters. The message handler inserts them between standard TPS messages to make ordinary TPS message streams look more human-readable when outputting data to a general-purpose terminal.

Structure of Industry-approved Messages

These messages are from NMEA, RTCM, CMR/CMR+, BINEX standards.

This manual provides some general information about these standards. Refer to the following pages to get familiar with the standards.

- NMEA – page 4-177.
- RTCM – page 4-203 and page 4-205.
- CMR/CMR+ – page 4-206.
- BINEX – page 4-207.

The complete description of the standards is outside the scope of the manual. For more information on any of the standards, refer to the corresponding websites listed below.

- NMEA – www.nmea.org
- RTCM – www.rtcm.org
- CMR/CMR+ – www.trimble.com
- BINEX – www.unavco.org

Message Groups

Each message from standard, non-standard, and industry-approved message families belongs to a specific *group* that comprises related messages. Currently, TPS receivers support seven groups of messages:

- TPS – Standard binary and ASCII TPS messages.
- RTCM 2.x – Binary RTCM messages from versions 2.1, 2.2, and 2.3 of RTCM-104 standard.
- RTCM 3.x – Binary RTCM messages from versions 3.0 and 3.1 of RTCM-104 standard.
- CMR – CMR/CMR+ binary messages.
- NMEA – ASCII NMEA messages from versions 2.1, 2.2, 2.3, and 3.0 of NMEA 0183 standard.
- BINEX – Binary messages developed by UNAVCO.
- MISC – Two special messages of arbitrary structure and contents used to communicate with various external meteo and tilt sensors.

These groups always have a fixed (unchangeable) number of messages with fixed scheduling parameters. To output one or more messages with a single `em` command, you have to explicitly define the message names in the command. If you want to change the scheduling parameters, you must also specify them explicitly when outputting the message with the `em` command. See “Configuring Messages” on page 3-571 for parameters that you might use to manage message groups.

Message Sets

The *Message set* is a customized collection of messages that can be enabled with a single command without specifying each message explicitly. A message set might comprise up to 49 messages from various groups.

Currently, there are three message sets defined in the language:

- Default message set. In its original form, it comprises 40 binary TPS messages designed to meet the requirements of various surveying software programs. It is the primary message set used by the receiver when you start data logging.
- Minimum RTK message set. In its original form, it comprises eight binary TPS messages designed to meet slow data links in RTK.
- Maximum RTK message set. In its original form, it comprises twelve binary TPS messages designed to meet fast data links in RTK.

You might completely change these message sets to make them more suitable for your applications. For example, you might add or remove messages or modify their scheduling parameters. The receiver will keep the customized message set in the NVRAM for further usage.

See “Configuring Messages” on page 3-571 for parameters that you might use to manage message sets.

Examples

This section shows you some examples of how various messages appear on the screen without interpreting them using dedicated software. This will give you a general understanding of what you might expect to see.

1. The example shows an excerpt from a typical ordinary stream of messages. More precisely, it is the beginning of a stream that contains the default set of standard messages. The message identifiers and lengths are shown in bold. Notice the body of the messages. Some of them appear as ASCII text while the others appear as *garbage* because they are in binary format.

```
JP055RLOGF JPS HIPER Receiver Log File                                  <LF>
MF009JP010008F<CR><LF>
PM025rcv/ver/main="3.1 Jan,24,2007 p1",@8A<CR><LF>
PM018rcv/id="8RWFM15BD34",@D4<CR><LF>
PM01Brcv/ver/board="HGGDT_5",@4B<CR><LF>
PM015rcv/sn="385-0343",@60<CR><LF>
PM01E/cur/stream={0,/cur/file/a}@6F<CR><LF>
PM048/par={rcv={8RWFM15BD34,HIPER,4096, 6d22h09m47s,{"3.1 Jan,24,2007
p1",@16<CR><LF>
PM039   0,71,HGGDT_5,none,{44,4}},1,385-0343,0x201,Topcon}}@4F<CR><LF>
~~005.b\..<LF>
RD006......<LF>
SI0071......<LF>
EL007T2E.#..<LF>
FC00DS.S.S.S.S.S.U<LF>
RC031.F.E.K.?isu..&.?.}.$5..?.jdi.A.?.{.....?S..K...?.<LF>
DC019.%.......t...X..7.;.....`<LF>
EC0074-/&,(k<LF>
TC00Dg&.+Z$..p....<LF>
CP019...2H.I3...2wP.2.4.2!..2.<LF>
1R019.........5X.= ..Ii[...[.O<LF>
```

2. The following examples show RTCM 2.x and CMR messages as they appear on the screen. These messages are unreadable and appear as *garbage* on the screen.

CMR

```
...$`b..>.s.~D..=........3V.U.K........2......`...>.[..Y..vk..f....K.
.>..X...@..G.P}...k.......#..&..N....R.@...S........(......?.......%.
jJ..zQ.......e.
5..@...:.5....T.JC.............U.....
...............$`b..>.s.{..........l.3V.../
....B...2......`...>.[......k..?.
..K..V..8...0..g.P.`..
```

RTCM 2.x

```
fQB@mgxzWPkGA@`Hds@JzEhbs_HUB@fVfEA@cgrw~Lp}oDfOwg@mqlsijjIfQB@mgxzzF
X}WN|OrQGaE@ac^G`uD}xtYz~ygRs~Pp}oDC@pHXe@mqILVUUSfQB@HXgCEzwBLDtOpeY
PB@xINpeR|_ysYz~ygbu~Up}oDC@pHXe@m~[LVUUvY~|^XgGA{wCDK{LtIG_OM~\bUvqN
fQB@mghpzfGChzT@`IRu@CyR|cu_Py@@wRfEA@cgb~~\p}oDC@pHXe@m~[LVUUvYn}RXw
HEPH}s{nOpiReC@]\C}oka]jYz~\X}EAZp}oDC@pHXe@m~[LVUUvRE007
```

3. This is an example of NMEA sentences. They are in ASCII format and can be easily read on the screen.

```
$GPGGA,090650.00,5543.3511138,N,03739.0686378,E,1,11,1.04,151.5021,M,
14.4309,M,,*5A<CR><LF>
$GNRMC,090650.00,A,5543.3511138,N,03739.0686378,E,0.1212,345.320,0210
07,8.099,W,A*0D<CR><LF>
$GPGGA,090651.00,5543.3511127,N,03739.0686382,E,1,11,1.04,151.5033,M,
14.4309,M,,*53<CR><LF>
$GNZDA,090651.00,02,10,2007,00,00*75<CR><LF>
$GPGGA,090652.00,5543.3511123,N,03739.0686380,E,1,11,1.04,151.5050,M,
14.4309,M,,*53<CR><LF>
```

Defining Message Generation

When you request messages, the receiver will output them according to the *schedule specification* defined for each message. Schedule specification consists of four parameters—period, phase, count, and flags—that completely describe an output scheme of a message.

They are represented in the following format: {period,phase,count,flags}.

- *period* specifies the message output interval. It is a floating-point value ranging from 0.0 (zero) seconds to 86400.0 seconds. If set to zero, the corresponding message will be output at the highest possible rate[2].

2. The receiver controls this rate with the *measurement update rate* parameter and option _RAW.

- *phase* specifies a time shift in the output of a message with respect to the moment when the receiver time is a multiple of the specified period. It is a floating-point value ranging from 0.0 (zero) seconds to 86400.0 seconds.

  How do the related parameters *period* and *phase* govern the process of message outputting? The receiver will output the message only at the receiver times $\{t_{out}\}$ satisfying the following two equations:

  $t_{out}=N \cdot step$,

  t_{out} {modulo period } = phase,

  where *step* is the value of the parameter `/par/raw/curmsint` expressed in seconds, *N* =0, 1, 2, 3….

  Note that the receiver time is counted from the start of the day in the specified reference scale. Consider a couple of examples illustrating this mechanism:

  – Suppose period = 10 s, phase = 2.2 s whereas step = 0.2 s. The receiver will output the message at the moments 2.2 s, 12.2 s, 22.2 s, etc.

  – Suppose period = 10 s, phase = 2.2 s whereas step = 0.5 s. The receiver will not output the message since the above pair of simultaneous equations is never satisfied.

  – Suppose phase > period. The receiver will not output the message at all .

  Also note that if 86400 mod period is not equal to 0, the actual interval between the last message sent before the day rollover and the first message after will be different from period.

- *count* serves two purposes. Firstly, if set to a positive number, specifies how many times the message outputs. Zero means an unlimited number of times the message outputs. Secondly, if set to a negative number, instructs the receiver to encapsulate a message into >> message before output. This option is an integer value ranging from -256 to 32767. See page 4-126 for details on how this option affects >> message.

- *flags* provides further adjustment of the way the message is output. This is a 16-bit word, where each of the bits is a separate flag having its own, defined meaning. The following section explains the meaning of message scheduling flags is detail.

Flags

The flag specification allows you further flexibility in the process of outputting messages. This specification is represented as a binary word (16 bits) in which each bit corresponds to the specific flag.

The following table describes the message scheduling flags. Hexadecimal representations are followed by flag names. Note that these names are introduced here only for the reader's convenience and are used nowhere except this particular appendix (remember that these names can't be used when you explicitly specify the "flags" option for your em or out command).

Table 1-5. Scheduling Flags

Flag Name	Bit #	Hex representation	Description
F_OUT	0	0x0001	If this flag is set, the first of the messages invoked by the corresponding command will be output at the epoch closest to the command execution time (whatever the specified "period" option).

Table 1-5. Scheduling Flags

Flag Name	Bit #	Hex representation	Description
F_CHANGE	1	0x0002	If this flag is set, the corresponding message will be output only if the message data have changed. Note that the receiver checks whether the message data have changed specifically at the moments matching the given "period". The field "phase", which loses its original function in this case, now plays the role of a "forced output period" ("forced output" means that the corresponding message will be output whether its contents will have changed or not). If the field "phase" is set to zero, then the receiver performs no "forced output" so that the corresponding message will be output only on condition that its data have changed.
F_OUT_ON_ADD	2	0x0004	If this flag is set, then the (first) message will be output immediately after executing the corresponding em or out command. Messages that do not allow asynchronous output will just ignore this flag. Currently only two TPS messages, JP and MF, can be output asynchronously. When setting messages from the default set of messages to their factory defaults, these kinds of flags are always set to unity.
F_NOTENA	3	0x0008	Suppose you are going to set F_NOTENA for a message which is already "checked" in the message set associated with the chosen output stream and which already has F_DISABLE=1 in this message set (see the description of F_DISABLE below). In this case, the receiver will retain the F_DISABLE value as it is (i.e., F_DISABLE=1). The receiver uses this flag in order not to output the file header every time you change the "period" value when recording data into the log file (for, example, if you send em,,def:1 followed by em,,def:5, you instruct the receiver to change the output interval from 1 second to 5 seconds).
F_FIX_PER	4	0x0010	The flags F_FIX_PER, F_FIX_PH, F_FIX_CNT and F_FIX_FL are used to disable the user to change the option values ("period", "phase", "count", "flags") through the corresponding commands. For example, the command em,,def:2 will not allow you to set the jps/XA message output period to 2 seconds since, for jps/XA, the flag F_FIX_PER is set to unity (as established for the default set of messages "/msg/def").
F_FIX_PH	5	0x0020	
F_FIX_CNT	6	0x0040	
F_FIX_FL	7	0x0080	
reserved	8	0x0100	N/A
reserved	9	0x0200	N/A
reserved	10	0x0400	N/A

Table 1-5. Scheduling Flags

Flag Name	Bit #	Hex representation	Description
F_DISABLE	11	0x0800	This flag is not explicitly programmable by the user. When one enables a message with a nonzero "count", then, after this message has been output "count" times, the message scheduler sets this flag to "1" for the corresponding message set.
reserved	12-15	0x1000-0x8000	N/A

Parsing Message Stream

In this section, you will find some guidelines on how to write code intended to parse a TPS receiver's message streams. These guidelines and the prudent format of TPS messages will help you effectively process/parse nearly any TPS message stream you may encounter in practice.

When referring to any message defined in GRIL, the following byte notations apply:

- a byte consists of 8 bits.
- bit 0 is the least significant bit of the byte.
- multi-byte integer and floating-point values are stored as little endianness in the message bodies.

Refer to Chapter 4 for a detailed reference of all binary and text messages supported in GRIL.

Synchronizing

When parsing a message stream, you need to find out the messages boundaries. This is called a *synchronization task*. Message synchronization is carried out when parsing is started or when synchronization is lost due to an error in the message stream.

First, your code should search for the nearest two adjacent characters which may be taken for a message identifier and which are followed by another three uppercase characters representing a non-negative hexadecimal number (say L).

Once you have detected such a quintet of characters, you treat the following L characters as the message body.

You will make your synchronization technique more robust if you make the decision after examining more than one successive message. Alternatively, you may prefer to search the stream for a message whose structure is a priori known (including the message body).

Skipping to the Next Message

To go from the current message to the next one, take the following steps:

1. Assume the current message starts at position N. Determine the current message length (decode characters ## N+2, N+3, N+4). Assume the message length is equal to L. Skip the first L+5 characters starting from position N.
2. Skip all of CR and LF characters (if any).

Strictly speaking, it is not recommended to use in your parsing code any a priori information about the sizes and the contents of the message bodies. If you respect this recommendation, you will not have trouble with the parsing program should some of the messages be changed.

Performing a Checksum

Most of TPS messages contain a checksum. Use it to ensure the integrity of a message before extracting message contents. This can be done with comparison of the checksum computed by you against that computed by the firmware and displayed in the message.

The checksum (if available) is always placed at the very end of the message body. The checksum is computed using all elements of a message, specifically identifier, length, and body. For detailed information about the checksum procedure TPS receivers utilize for messages, see "Computing Checksum" on page A-1.

Achieving Forward Compatibility

In prospective firmware versions, new fields might appear in the existing TPS binary and ASCII messages. Adding new fields to such messages will conform to the following rules:

- For binary messages, new data fields are always added at the end of a message body immediately before the checksum field (if available).
- For ASCII messages, new data fields are added at the end of a message body immediately before the checksum field or are inserted immediately before any right-hand curly bracket (`}`). For example, a message having the following data structure `...1,2,{3,4},27,@CS` can be supplemented with the new fields (shown in bold) `...1,2,{3,4,`**`5`**`},27,`**`M,`**`@CS`.

With the addition of new fields, your software might encounter compatibility issues and as a result will behave incorrectly if it receives a message with new data. Consider the following parsing strategies for making your applications forward compatible.

- Regard the size of any message body as a temporary value, which might actually differ from the size given in this documentation.
- Address the checksum field relative to the end of the message body.
- Address all other data fields relative to the beginning of the message body.

Develop your software with the above guidelines in mind. This will allow the software to parse a message with new data properly without rejecting the whole message or reporting an error.

Commands

This chapter provides definitions of the GRIL commands including their functions, syntax, and examples of usage. GRIL's flexibility allows for future expansion of the current command list or use of only part of the existing command list.

In This Chapter:

- "Command Specification" on page 2-2
- "set" on page 2-3
- "print" on page 2-4
- "list" on page 2-5
- "em" on page 2-6
- "out" on page 2-8
- "dm" on page 2-9
- "create" on page 2-11
- "remove" on page 2-14
- "init" on page 2-15
- "event" on page 2-16
- "get" on page 2-18
- "put" on page 2-20
- "fld" on page 2-22

Command Specification

Each command is defined with its specifications as per the following categories:

- *Command Name* – provides the title of a command and appears as a second-level heading followed by the rest of the command specification. Actually, the command name is an inherent part of a complete command but for clarity, the Command Name category excludes any other elements of a command. The complete command syntax is displayed in the Syntax category.
- *Description* – contains a concise explanation of what the command does.
- *Syntax* – provides a template for the command showing how to enter the command elements, including command name and its arguments. Command identifier and checksum are omitted. Syntax conventions used in a command are the same as elsewhere in the manual.
  - `<target>`: a word in angle brackets indicates a placeholder you must replace with the actual value.
  - `[value]`: a word in square brackets indicates an optional element, i.e., the element that can be omitted.
  - `{opt1,opt2,...}`: one or more words in curly brackets indicate options used to customize the command or a group of objects to which the command is applied.
- *Arguments* – while the command name tells what to do with the elements that follow, the arguments are objects of the action. A command might have one, two, or more arguments. These arguments can be receiver parameters, values for parameters, as well as options for commands.
- *Usage Guidelines* – provides information that ensures the proper use of a command. This might include detailed explanation of the command operation and result, prohibitions, and any special considerations.
- *Examples* – provides one or more examples demonstrating how to use the command and might also include an example of the expected result.
- *History* – lists the firmware versions in which the command was introduced and modified, if relevant.
- *See Also* – directs you to other commands, parameters, or messages that might contain related information.

set

Description: The `set` command applies a value to a target.

Syntax: `set,<target>,<value>`

Arguments:

- `target` – a parameter to address by the command. If a parameter does not begin with `/`, then `/par/` will be automatically inserted before the parameter prior to executing the command.
- `value` – an alphanumeric quantity applied to one or more parameters. The type and range of values depend on a particular parameter.

Usage Guidelines: This command generates no response unless it starts with a command identifier or an error occurs.

Examples: Enable tracking the GPS L2C signal.

```
set,/par/lock/gps/l2c,{on,on}
```

Or

```
set,lock/gps/l2c,{on,on}
```

Set the input frequency source to external and force a reply.

```
%%set,/par/frq/input,ext
```

Or

```
%%set,frq/input,ext
```

Set a user-defined geoidal separation to 1.5 meter and force a reply.

```
%%set,/par/pos/geoidh,1.5
```

Or

```
%%set,pos/geoidh,1.5
```

History: This command was introduced in the first release of firmware.

See Also: “create” on page 2-11 for how you can use the `set` command with `create`.

print

Description: The `print` command prints a value of a target.

Syntax: `print,<target>[:<option>]`

Arguments:

- `target` – one or more parameters to print. If a parameter does not begin with `/`, then `/par/` will be automatically inserted before the parameter prior to executing the command.
- `option` – auxiliary entity that can be set to `off` and `on`. If the option is omitted, its default state is automatically applied.
- `off` prints only parameter values (default).
- `on` prints parameter values along with the corresponding identifiers.

Usage Guidelines: This command will always generate one or more response messages. If the parameters specified in the command do not exist, a single error message will be reported. Otherwise, one or multiple positive response messages might be reported depending on the amount of queried parameters.

Examples: Report firmware version.

```
print,/par/rcv/ver/main
```

Or

```
print,rcv/ver/main
```

Report the receiver model and force a reply.

```
%%print,/par/rcv/model
```

Or

```
%%print,rcv/model
```

Print values of parameters that govern serial port A.

```
print,dev/ser/a
```

Reply (only a part of the full reply is shown).

```
RE01C{115200,off,on,on,8,N,1,off,
RE02D cmd,std,/dev/null,"#OFF#",off,128,/dev/null,
```

Or print the same values with the corresponding parameters.

```
print,dev/ser/a:on
```

Reply (only a part of the full reply is shown).

```
RE045/par/dev/ser/
a={rate=115200,rtscts=off,rts=on,cts=on,bits=8,parity=N,
RE045 imode=cmd,omode=std,echo=/dev/
null,eoff="#OFF#",ewrap=off,wsize=128,
RE00F dup=/dev/null,
```

History: This command was introduced in the first release of firmware.

See Also: N/A

list

Description: The `list` command lists a name of a target.

Syntax: `list[,<target>]`

Arguments: `target` – one or more parameters to list. If a parameter does not begin with `/`, then `/log/` will be automatically inserted before the parameter prior to executing the command. If a target is omitted, `/log` is assumed.

Usage Guidelines: This command will always generate one or more response messages. If the parameters specified in the command do not exist, a single error response message will be reported. Otherwise, one or multiple positive response messages might be reported depending on the amount of queried parameters.

Examples: Obtain a list of existing log files.

```
list,/log
```

Or

```
list
```

Obtain a list of the default set of messages.

```
%%list,/msg/def
```

Obtain a list of all messages supported by the receiver.

```
%%list,/msg/
```

History: This command was introduced in the first release of firmware.

See Also: N/A

em

Description: The em command initiates the periodic output of the specified messages into a target.

Syntax: em,[<target>],<messages>[:{[<o1>],[<o2>],[<o3>],[<o4>]}]

Arguments:

- target – a communication port or the current log file to which messages are to be enabled. If no target is specified, the receiver assumes the current terminal (the port through which you are sending commands).
- messages – one or more message names or a message set to enable to a target. If some of the message names do not begin with /, then /msg/ is automatically inserted before these names prior to executing the command. Several message names are usually surrounded by curly brackets and delimited with commas {msg1,msg2,...}.
- o1,...,o4 – auxiliary entities, called the *schedule specification*, are intended to customize the messages output pattern. It consists of *period* (o1), *phase* (o2), *count* (o3) and *flags* (o4). See "Defining Message Generation" on page 1-18 for details about the schedule specification.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- If you execute the em command containing unsupported message name(s), the message handler will report an error message. Output of other supported messages will not be stopped.
- If some messages are being output to a target and you then enable additional messages to the same target, the message handler will add them to the end of the list of the messages to generate.
- If you execute the em command containing a message that is already outputting to a target, the message handler will only change the schedule specification (if it differs). Its position with respect to the other messages in the output list will be left intact.
- When you enable the receiver to output a specific message (as a separate message or as part of a message set) to a target, the operating scheduling parameters governing the output are selected taking into account various conditions, specifically:
  1. Values explicitly specified in options of the em command.
  2. Default values of options for the em command.
  3. Scheduling parameters specified for the given message as part of the corresponding message set. These are taken into account only when enabling a message set, not an individual message.
  4. Current scheduling parameters for the message that is to be output to a target.
  5. Default scheduling parameters specified for the given message as part of the corresponding message group.
- The above categories are listed in the order of their precedence (highest to lowest). In other words, values from the first item will override values from the second one and so on.

Examples: Enable the output of the default set of messages to the current log file A using the default options.

```
em,/cur/file/a,/msg/def
```

Or

```
em,/cur/file/a,def
```

Enable output of the default set of messages to the current log file A every 10 seconds. Other options—phase, count, and flags—use their default values.

```
em,/cur/file/a,def:10
```

Or

```
%%em,/cur/file/a,def:10
```

Enable output of the default set of messages to the current terminal using the default options.

```
em,/cur/term,/msg/def
```

Or

```
em,,/msg/def
```

Or

```
em,,def
```

Enable output of TPS messages ~~ and RD to the current terminal.

```
em,,/msg/jps/RT,/msg/jps/RD
```

Or

```
em,,jps/RT,jps/RD
```

Or

```
em,,jps/{RT,RD}
```

Enable output of NMEA sentences GGA and ZDA to the current terminal every 0.5 seconds.

```
em,,nmea/{GGA,ZDA}:0.5
```

Or

```
em,,nmea/{GGA,ZDA}:.5
```

Enable output of NMEA sentence GGA to serial port D every 5 seconds.

```
em,/dev/ser/d,nmea/GGA:5
```

Enable the output of messages SI, EL and AZ to serial port A. The interval between any two subsequent SI messages will be equal to 10 seconds, if they coincide, and 1 second, otherwise. Only the first fifty SI messages will be output to the port. In addition, the receiver will output both EL and AZ messages every other second.

```
em,/dev/ser/a,jps/{SI:{1,10,50,0x2},EL,AZ}:2
```

Enable output of RTCM 2.x message types 1 and 31 to serial port B every 3 seconds, and RTCM 2.x message types 18, 19, 3, 22 to serial port C every 1 second for types 18 and 19, and 10 seconds for types 3 and 22. The use of semicolon allows you to run two commands on the same line and run the second command even if the first one returns an error.

```
em,/dev/ser/b,/msg/rtcm{/1,/31}:3.0; em,/dev/ser/c,/msg/
rtcm{/18:1.0,/19:1.0,/22,/3}:10.0
```

History: This command was introduced in the first release of firmware.

See Also:
- “out” on page 2-8 for how to request a one-time output of messages.
- “dm” on page 2-9 for how to disable the output of messages.

out

Description: The `out` command initiates a one-time, non-periodic output of the specified messages into a target.

Syntax: `out,[<target>],<messages>[:<options>]`

Arguments: This command uses the same arguments as the `em` command. For details, see the `em` command on page 2-6. One important difference between `out` and `em` is the default value for the `count` specification. For `out` it is unity (1), while for `em` it is zero (0). Recall that `count` defines how many times the message will be output. Unity means a one-time output; zero – an unlimited number of times.

Usage Guidelines: The `out` and `em` commands apply the same guidelines. For details, see the `em` command on page 2-9.

Examples: Enable a one-time output of TPS message RD to serial port D.

```
out,/dev/ser/d,/msg/jps/RD
```

Or

```
out,,jps/RD
```

Enable a one-time output of NMEA sentence GGA to the current terminal.

```
out,,nmea/GGA
```

History: This command was introduced in the first release of firmware.

See Also: “em” on page 2-6 for how to request a periodic output of messages.

dm

Description: The dm command stops the periodic output of the specified messages into a target.

Syntax: dm,[<target>][,<messages>]

Arguments:

- target – a communication port or the current log file to which messages are to be stopped. If no target is specified, the receiver assumes the current terminal (the port through which you are sending commands).
- messages – one or more message names or a message set to stop, directed towards the command target. If some of the message names do not begin with /, then /msg/ is automatically inserted before these names prior to executing the command. Several message names are usually surrounded by curly brackets {} and delimited with comma. If no messages are specified, the receiver will stop all periodic outputs to a target.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- If you execute dm with the current log file as a target and without any messages specified, the receiver will stop the output to the log file, and then close the log file.
- If you stop a message that is not currently in the list of outputted messages, the receiver will not generate an error message.

Examples:

Disable all of the messages being output into the current log file A and then close the log file.
```
dm,/cur/file/a
```

Disable all of the messages being outputted into the current terminal.
```
dm,/cur/term
```
Or
```
dm
```

Disable output of TPS message PV into serial port D.
```
dm,/dev/ser/d,/msg/jps/PV
```

Disable output of the TPS message DP into the current log file B.
```
dm,/cur/file/b,/msg/jps/DP
```

Examples: (continued) Disable output of the NMEA messages GGA and ZDA into the current terminal.

```
dm,/cur/term,/msg/nmea/GGA,/msg/nmea/ZDA
```

Or

```
dm,,/msg/nmea/GGA,/msg/nmea/ZDA
```

Or

```
dm,,nmea/GGA,nmea/ZDA
```

Or

```
dm,,nmea/{GGA,ZDA}
```

History: This command was introduced in the first release of firmware.

See Also:
- "em" on page 2-6 for how to request a periodic output of messages.
- "out" on page 2-8 for how to request a one-time output of messages.

create

Description: The `create` command makes or supplements a new target.

Syntax: `create[,<target>][:<option>]`

Arguments:

- `target` – a log file to create or a message to add. If a target does not begin with `/` or `/msg`, then `/log/` will be automatically inserted before the target prior to executing the command. If no target is specified, the receiver creates a log file with an automatically generated name.
- `option` – this option has two values, `a` and `b`, allowing you to associate the created file with the log file A or B, respectively. If no option is specified, the created file is associated with the log file A.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- The files with automatically generated names use the following naming scheme for all receivers with firmware 3.1p3 or older:
  P...PMMDDL...L.tps; where:
  P...P – the beginning part of the name taken from the `/par/cmd/create/pre/<a|b>/` parameter.
  MM – the month value (1 through 12).
  DD – the day of the month (1 through 31).
  L...L – the combination of lowercase letters (a through z) allowing you to avoid confusion between files created on the same day. The receiver will change the combination sequentially for each new file using the following pattern a, b,...,z, za, zb,..., zz, zza, zzb,... . This combination resets to the letter "a" every day.
- The files with automatically generated names use the following naming scheme for all receivers with firmware 3.2 or newer, except the NET-G3:
  P...PMMDDHL...L.tps; where:
  P...P – the beginning part of the name taken from the `/par/cmd/create/pre/<a|b>/` parameter.
  MM – the month value (1 through 12).
  DD – the day of the month (1 through 31).
  H – the lowercase letter (a through x) that will change sequentially every hour. The letter "a" corresponds to the interval 00:00:00-00:59:59 in GPS time, "b" to 01:00:00-01:59:59 in GPS time, etc.
  L...L – the combination of lowercase letters (a through z) that will be added to the files starting with the second file within an hour. The receiver will change the combination sequentially for each new file using the following pattern a, b,...,z, za, zb,..., zz, zza, zzb,... . This combination resets to the letter "a" every day.

Usage Guidelines: (continued)

- The files with automatically generated names use the following naming scheme for the NET-G3 receiver only starting with firmware 3.2:
  P...PMMDDLXXX.tps; where
  P...P – the beginning part of the name taken from the `/par/cmd/create/pre/<a|b>/` parameter.
  MM – the month value (1 through 12).
  DD – the day of the month (1 through 31).
  L – the lowercase letter (a through x) that will change sequentially every hour.
  XXX – the numeric portion (000 through 999) that will be added to the files starting with the second file within an hour and will increase by one with each new file. This number resets to zero every hour.
- If you specify the file name manually, make sure it conforms to the following restrictions:
  The name of the log file should not be longer than 32 characters.
  The following alphanumeric characters can be used in the names—A through Z, a through z, and 0 through 9.
  Other printable ASCII characters can also be used in the names—#, :, ., +, -, ;, ^, ~, _, =, [ ].
  Do not use the following characters in the names—space, comma, { }, ( ), @, &, ", / \, $, *, !, |.
- The receiver produces an error message if you try to create a file whose name coincides with the name of already created file.
- If the receiver is recording messages into the current log file when you run `create`, this log file will be closed and the receiver will continue outputting messages to the newly created log file.
- You can add messages to a message set only. By default, these messages will have scheduling parameters defined by the message group they belong to. Use the `set` command if you need to change these parameters.
- The `create` command is not applicable to the GMS-2 receiver with the TG3_x board.

Examples:

Create a new log file with an automatically generated name and assign it to the current log file A.

```
create
```

Or

```
create,:a
```

Create a new log file with the name point#7 and assign it to the current log file A.

```
create,point#7
```

Or

```
create,/log/point#7
```

Create log files Spirit and Opportunity, and assign them to the current log file A and B, respectively.

```
create,Spirit:a
create,Opportunity:b
```

Or

```
create,Spirit:a; create,Opportunity:b
```

Examples: (continued)

Add TPS message DL to the default set of messages with scheduling parameters taken from a group this message belongs to.

```
create,/msg/def/jps/DL
```

Add TPS message RP to the default set of messages with customized scheduling parameters. They will instruct the receiver to output this message with the following pattern: output it every second only if the message data have changed since the previous output; force the message output every 10 seconds no matter whether the message contents will have changed or not; prevent changing of phase with the `em` or `out` commands.

```
create,/msg/def/jps/RP
set,/msg/def/jps/RP,{1,10,,0x22}
```

History: This command was introduced in the first release of firmware.

See Also:

- For how to close the current log file, refer to the `dm` command on page 2-9.
- For how to enable outputting of messages to a log file, refer to the `em` or `out` commands on page page 2-6 and page 2-8, respectively.
- For how to change the scheduling parameters in the added messages, refer to the `set` command on page 2-3.

remove

Description: The `remove` command deletes one or more targets.

Syntax: `remove,<target>`

Arguments: `target` – one or more log files or messages to delete. If a target does not begin with `/`or `/msg`, then `/log/` will be automatically inserted before the target prior to executing the command. If no target is specified, or if the target cannot be removed, an error message is generated.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- If you try to delete a log file that is recording, an error message will be generated and the file will not be deleted.
- You can remove messages from a message set only.

Examples: Remove the log file with the name Phobos.

```
remove,/log/Phobos
```

Or

```
remove,Phobos
```

Remove all log files.

```
remove,/log/
```

Remove TPS message GA from the default set of messages.

```
remove,/msg/def/jps/GA
```

Remove all messages from the default set of messages.

```
remove,/msg/def/
```

History: This command was introduced in the first release of firmware.

See Also: For how to create the current log file or add a message to a message set, refer to the `create` command on page 2-11.

init

Description: The `init` command initializes one or more targets. In other words, this command restores a target to its original state.

Syntax: `init,<target>`

Arguments: `target` – one or more parameters to initialize. If no target is specified, or if the target cannot be initialized, an error message is generated.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- This command causes the receiver's automatic reboot when initializing some types of targets.
- All targets that you can initialize currently are listed in the examples.

Examples: Clear the NVRAM and reboot the receiver. All receiver parameters will be set to their default values. Almanacs, ephemerides, etc., will be erased. This operation will not delete any raw data files stored in the receiver's memory.

```
init,/dev/nvm/a
```

Set all the receiver parameters to their default values and reboot the receiver.

```
init,/setup/
```

Format the file system. All raw data files stored in the receiver's memory will be erased.

```
init,/dev/blk/a
```

Restore the default values of schedule parameters for all message sets.

```
init,/msg/
```

History:

- The command was introduced in the first release of firmware.
- The command `init,/par/` was replaced with `init,/setup/` starting with firmware 3.0

See Also: N/A

event

Description: The `event` command generates a text of arbitrary contents called a *free-form event*.

Syntax: `event,<string>`

Arguments: `string` – an arbitrary text comprising up to 63 characters.

Usage Guidelines:

- The free-form event mechanism is intended to forward arbitrary text information to any output stream of the receiver without interpreting this information in the receiver. A typical application for the `event` command is the sending of predefined free-form events to a log file. As a result, post-processing software will treat the contents of the file taking into account the recorded free-form events.
- The receiver's firmware never generates free-form events on its own. Nor does it interpret the information generated and output with the `event` command.
- This command generates no response unless it starts with a command identifier or an error occurs.
- If you use any of the characters having a special meaning in the language, enclose the string in double quotes. This ensures that the string is transferred to the event message == in its original form. The following characters and their combinations might affect the string contents: comma, { }, #, ;, :, ||, &&.
- After you send this command, the string will be transferred to the receiver's special buffer. If the receiver detects that the event message == is enabled to any of the output streams, it sends the data from the buffer to all the streams where the event message is enabled. The data will be sent to an event message that is closest to the command execution time.
- The current firmware provides a buffer large enough to store up to sixteen 64-byte free-form events. This might be useful if you send the free-form events more frequently than the output period of the event message.
- If the event message is disabled for output to all the streams, sending the `event` command will not have any result.
- All of the strings starting with the underscore character (ASCII 0x5F) are reserved for some TPS application software (Topcon Tools™, etc.). Care should be taken that such strings are not used with the `event` commands unless you cannot accomplish your task otherwise.

Examples: Generate a free-form event with the string Mars Pathfinder.

```
event,Mars Pathfinder
```

Generate a free-form event Mars Pathfinder with characters that affect the resulting look of the event.

```
event,Mars {Pathfinder}
```

Reply-> `ER018{2,,wrong 1st parameter}`

Put the event from the above example into double quotes and get the correct result (event is shown in the event message ==).

```
event,"Mars {Pathfinder}"
```

Reply-> `==017.....Mars {Pathfinder}.`

Generate a free-form event reserved for TPS application software. This event informs a post-processing software that the occupation is static.

```
event,"_DYN=STATIC"
```

History: The command was introduced in the first release of firmware.

See Also:
- For a structure of the event message, refer to “EV – Event data” on page 4-129.
- For free-form events reserved for TPS software, refer to “Free-form Events” on page C-1.

get

Description: The `get` command downloads a target to the host computer using the TPS Data Transfer Protocol (DTP).

Syntax: `get,<target>[:{[<o1>],[<o2>],[<o3>],[<o4>]}[,<offset>]]`

Arguments:

- `target` – a file to download. If a file does not begin with `/`, then `/log/` will be automatically inserted before the file prior to executing the command. If the target does not exist, or if the target cannot be downloaded, an error message is generated.
- `o1`,...,`o4` – options used to customize file downloading.
  - `o1` or `timeout` specifies how long the receiver will wait for an acknowledgment from the host computer before returning to command mode. It is an integer value ranging from 0 (zero) to 86400 seconds. The default value is 10 seconds.
    If this option is set to zero, the *DTP transmitter* (i.e., GNSS receiver) will stream data blocks without waiting for ACKs from the *DTP receiver* (i.e., the host computer). The transmitter will immediately abort data transfer should NAK be received. This approach allows sustained high-speed data transfer over connections having high latencies (such as TCP) or significant overhead in bidirectional communication (such as USB).
  - `o2` or `block_size` specifies the size of a data block transferred from the receiver to the computer at one time. It is an integer value ranging between 1 byte and 16384 bytes. Default block size is 512 bytes. For receivers that do not support TCP or USB interfaces, the maximum block size is 2048 bytes.
  - `o3` and `o4` specify the sampling `period` and `phase`, respectively. Use them if you need to thin the original raw data in the file to a new sampling interval. The receiver will download data only for the epochs where the receiver time modulo one day (T_r) satisfies the following equation: $T_r\{\text{mod } \texttt{period}\} = \texttt{phase}$. The period and phase are floating-point values ranging from 0 (zero) to 86400 seconds.
- `offset` – specifies the byte from which to start the file downloading. Note that this argument starts counting bytes from zero. If this argument is set to N, the receiver will skip the first N bytes before downloading the file. If an offset is not explicitly specified, the receiver will start downloading from the very first byte (i.e. N=0). This option allows you to resume a file download that has been interrupted for any reason from the point of interruption.

Usage Guidelines:

- The essential difference between the `get` and `print` commands: the former is based on an error-detection data exchange protocol (i.e., DTP) whereas the latter is not.
- To download the file using the `get` command, it is not enough to simply send it with the desired options to the receiver. You must also have a piece of software or a dedicated software with the receiver part of DTP implemented on the host computer.
- When you apply an offset to the downloaded file, it is recommended to add an identifier to the command and wait for a positive reply from the receiver before running DTP receiver. This will ensure that the GNSS receiver accomplishes the search for the offset before DTP is started on the host computer.
- Should you use the offset together with specified period and phase and the download was interrupted for any reason, it is recommended to restart the download from the beginning rather than resuming it from where it was interrupted.
- The `get` command is not applicable to the GMS-2 receiver with the TG3_x board.

Examples:

Start downloading the file Phoenix from the very first byte using DTP.

```
get,/log/Phoenix
```

Or

```
get,Phoenix
```

Start downloading the file Phoenix from byte 3874416 (counting bytes from zero) and wait for the positive reply as shown below before starting DTP on the computer.

```
%Offset skipped. Begin download.%get,/log/Phoenix,3874416
```

Reply-> `RE021%Offset skipped. Begin download.%`

Start downloading a 10-Hz file Phoenix using a 1-Hz sampling interval. The resulted file will have the 1-Hz data logging interval.

```
get,Phoenix:{,,1}
```

Start downloading the file Phoenix from byte 3000 with timeout set to half a minute and block size set to 9216 bytes.

```
get,Phoenix:{30,9216},3000
```

History:

- The command was introduced in the first release of firmware.
- A new meaning of the zero value in the `timeout` option was implemented in firmware 2.4.

See Also: For an explanation about how DTP works and how to implement it on a host computer, refer to “TPS Data Transfer Protocol” on page B-1.

put

Description: The `put` command uploads data from a host computer to a target in the receiver using the TPS Data Transfer Protocol (DTP).

Syntax: `put,<target>[:{[<opt1>],[<opt2>]}[,<offset>]]`

Arguments:

- `target` – a file where to upload data. If a file does not begin with `/`, then `/log/` will be automatically inserted before the file prior to executing the command.
- `opt1`, `opt2` – options used to customize data uploading.
  - `opt1` or `timeout` specifies how long the receiver will wait for an acknowledgment from the host computer before returning to command mode. It is an integer value ranging from 0 (zero) to 86400 seconds. The default value is 10 seconds.
  - `opt2` or `block_size` specifies the size of a data block transferred from the computer to the receiver at one time. It is an integer value ranging between 1 byte and 16384 bytes. Default block size is 512 bytes. For receivers that do not support TCP or USB interfaces, the maximum block size is 2048 bytes.
- `offset` – specifies the byte from which to start data uploading to an existing file. Note that this argument starts counting bytes from zero. If this argument is set to N, the host computer will skip the first N bytes before uploading the data to a file. If an offset is not explicitly specified, the host computer will start uploading the data from the very first byte (i.e. N=0). This option allows you to resume data upload that has been interrupted for any reason from the point of interruption.
  If you issue a command with a zero offset, the receiver will create and open a new file with the name specified by the command. If the receiver already contains a file with the same name as a file specified in the command, an error message will be generated.
  If you issue a command containing a name of the existing file and an offset that matches the size of the file, the receiver will allow the host to append data to the file. If the receiver does not have such a file or the size of the file does not match the offset value, an error message will be generated.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- To upload data to a file using the `put` command, it is not enough to simply send it with the desired options to the receiver. You must also have a piece of software or a dedicated software with the transmitter part of DTP implemented on the host computer.
- Before appending data to an existing file in the receiver, request the size of the file with the command `print,/log/<name>&size`. This size instructs the host computer how many bytes to skip in the source file before starting its upload to the receiver.
- The `put` command is not applicable to the GMS-2 receiver with the TG3_x board.

Examples: Start data uploading to an empty file Viking from the very first byte using DTP.

```
put,/log/Viking
```

Or

```
put,Viking
```

Request the size of the file Viking and then start data appending to it from byte 3874416 (counting bytes from zero). Default timeout and block size of 5120 bytes are used.

```
print,/log/Viking&size
```

Reply-> `RE008 3874416`

```
put,/log/Viking:{,5120},3874416
```

Start data uploading to an empty file Viking using a 30-second timeout and a 2048-byte block size.

```
put,Viking:{30,2048}
```

History: The command was introduced in firmware 2.4.

See Also: For an explanation about how DTP works and how to implement it on a host computer, refer to “TPS Data Transfer Protocol” on page B-1.

fld

Description: The `fld` command loads the firmware from a target to the receiver and then resets the receiver.

Syntax: `fld,<id>,<target>[:{[<opt1>],[<opt2>]}]`

Arguments:

- `id` – a string containing the receiver electronic ID. If specified ID does not correspond to the actual electronic ID of the connected receiver, the command will return an error message.
- `target` – a source from where to load the firmware. The source can be a file in the receiver or the USB port.
- `opt1`, `opt2` – options used to customize the firmware loading. They are applied only when loading the firmware via USB.
  - `opt1` or `timeout` specifies how long the receiver will wait for an acknowledgment from the host computer before returning to command mode. It is an integer value ranging from 0 (zero) to 86400 seconds. The default value is 10 seconds.
  - `opt2` or `block_size` specifies the size of a data block transferred from the computer to the receiver at one time. It is an integer value ranging between 1 byte and 16384 bytes. Default block size is 512 bytes. For receivers that do not support TCP or USB interfaces, the maximum block size is 2048 bytes.

Usage Guidelines:

- This command generates no response unless it starts with a command identifier or an error occurs.
- If you load the firmware from a file, you need to upload the file to the receiver in advance using the `put` command. The receiver will then check whether the file contains a valid firmware for the receiver. This will take a number of seconds to complete. If the check succeeds, the receiver will load the firmware and then perform a reset. The reply to the command (if any) will be sent after the check is performed but before firmware loading begins.
- If you load the firmware via the USB port, the receiver will wait for a reply (if any) on the command. After getting a positive reply, the GNSS receiver will start DTP receiver and will wait for DTP transmitter to start on the host computer. The receiver will perform a reset after the firmware loads completely or an interruption occurs.
- To load the firmware via USB, it is not enough to simply send the command with the desired options to the receiver. You must also have a piece of software or a dedicated software with the transmitter part of DTP implemented on the host computer.
- Do not interrupt the firmware loading. Should an interruption of firmware loading occur for any reason, the receiver will go into a non-working state. To restore the receiver to the working state, you need to complete the firmware loading using the receiver's serial port and a power-cycling operation.
- The `fld` command is not applicable to the GMS-2 receiver with the TG3_x board.

Examples: Start firmware loading from the file, `main.ldp`, to the receiver having an `ID = AFUM 12JVP4V`.

```
fld,AFUM 12JVP4V,/log/main.ldp
```

Start firmware loading from the USB port using a block size 16384 bytes and timeout of 30 seconds.

```
fld,AFUM12JVP4V,/dev/usb/a:{30,16384}
```

History: The command was introduced in firmware 2.4.

See Also: For an explanation about how DTP works and how to implement it on a host computer, refer to “TPS Data Transfer Protocol” on page B-1.

Notes:

Parameters

This chapter contains the complete reference of GRIL parameters used to control the operation and behavior of the receiver. Each listed parameter accompanies with its description, syntax and usage examples.

In This Chapter:

- "Parameter Specification" on page 3-3
- "Identification Information about Receiver" on page 3-5
- "Power Management" on page 3-12
- "Parameters used only in discontinued receivers" on page 3-38
- "Memory and File Management" on page 3-42
- "Frequency Source" on page 3-68
- "Eurocard J502 connector I/O pin support" on page 3-72
- "Advanced Input Mode (AIM)" on page 3-75
- "Cinderella Mode" on page 3-83
- "Antenna Parameters" on page 3-84
- "Communication Link Parameters" on page 3-87
- "Ntrip Parameters" on page 3-123
- "Accepting data over TCP/IP (Mobile mode)" on page 3-142
- "Port Communication Parameters" on page 3-159
- "TCP/IP Communication Parameters" on page 3-174
- "History Logger Parameters" on page 3-183
- "Output Time-Frame Parameters" on page 3-185
- "Job Programming" on page 3-189
- "Notebook Parameters" on page 3-207

- "Raw Data Management and Signal Processing Parameters" on page 3-211
- "Loop Parameters" on page 3-246
- "Positioning Parameters" on page 3-263
- "Timing Parameters" on page 3-289
- "Datum Parameters" on page 3-299
- "Antenna Database" on page 3-319
- "MINTER Parameters" on page 3-328
- "Governing Optimized Real Time Application (ORTA) Messages" on page 3-333
- "Reference Station Parameters" on page 3-339
- "Rover Station Parameters" on page 3-388
- "RAIM Parameters" on page 3-417
- "Parameters Related to NMEA Sentences" on page 3-420
- "Pulse Per Second (PPS) Parameters" on page 3-432
- "Event Signal Parameters" on page 3-440
- "Code Differential Parameters" on page 3-446
- "Carrier Phase Differential Parameters" on page 3-463
- "SBAS Parameters" on page 3-515
- "OmniSTAR Parameters" on page 3-527
- "BINEX Parameters" on page 3-536
- "Receiver Options" on page 3-549
- "AutoSeed Functionality" on page 3-554
- "Miscellaneous Parameters" on page 3-569
- "Configuring Messages" on page 3-571

Parameter Specification

Each parameter is defined with its specifications as per the following categories:

- *Name of the parameter*.
- *Parameter syntax* – a unique representation of a parameter as it appears in a command. Command identifier and name, value for the parameter, options and checksum are omitted. Syntax conventions used in a parameter are the same as elsewhere in the manual.
  - `<port>` an element in angle brackets indicates a placeholder you must replace with the actual value.
  - `<a|b>` characters in angle brackets separated with a vertical bar indicate mutually exclusive elements, one of which you should select.
  - `[/par/]` an element in square brackets indicates an optional element, i.e., the element that can be omitted.
  - `&def` auxiliary attribute used to indicate that the parameter's value is stored in the permanent area of NVRAM.
  - `rs_port` indicates serial ports of the RS232C standard. Four ports are defined:

    `dev/ser/a` for serial port A.

    `dev/ser/b` for serial port B.

    `dev/ser/c` for serial port C.

    `dev/ser/d` for serial port D.
  - `port` indicates any ports that can used for input and output, specifically:

    `rs_port` – serial ports.

    `dev/usb/a` – USB port.

    `dev/tcp/a`, `dev/tcp/b`, `dev/tcp/c`, `dev/tcp/d`, `dev/tcp/e` – TCP ports.

    `dev/prl/a` – parallel port (obsolete).

    `dev/modem/a` – modem port (obsolete).
  - `stream` designates either a port from the `port` category or the current log file.
  - `cur/term` designates any `port` used for issuing the current command.
- *Description* – contains a concise explanation of what the parameter does.
- *Access* – indicates which commands can be used with the parameter. All of the parameters described in this chapter are mainly intended for use with `set` and/or `print` commands. Most of these parameters are both readable with the `print` command and writable with the `set` command. Some of them, however, are either read-only or write-only.
  - read & write: a parameter can be used with the `print` and `set` commands.
  - write-only: a parameter can be used with the `set` command only.

  – read-only: a parameter can be used with the `print` command only.

  If the parameter can also be used with any other commands, it will be shown explicitly.

- *Type* – indicates the type of values. See “Parameter Types” on page 1-5 for available types.
- *Values* – indicates values that can be applied to the parameter.
  - `cur_value` indicates the current value for the parameter.
  - Double quotation marks (`""`) indicate an empty value.
- *Default* – indicates the default value of parameter.
- *Options* – indicates the options that you can apply to the parameter with the `set` command. Currently, only one parameter uses the option and this category will appear for this parameter only.
- *Usage Guidelines* – provides information that ensures the proper use of the parameter. This might include detailed explanation of the parameter operation and result, typical use, prohibitions, and any special considerations.
- *Command Examples* – provides one or more examples demonstrating how to use the parameter with a command, including the result, if needed.
- *Receiver Availability* – indicates the receivers where the parameter is available.
- *History* – lists the firmware versions in which the parameter was introduced and modified, if relevant.
- *See Also* – directs you to other parameters, commands, or messages that might contain related information.

Some of the categories might have the N/A abbreviation standing for Not Applicable. It should be considered as a normal feature of the parameter and not a lack of functionality.

Identification Information about Receiver

This section describes parameters that identify the receiver by displaying its hardware and firmware versions, model, serial number, and so on.

Firmware version on GNSS board

`[/par/]rcv/ver/main`

Description:	The version of firmware running on the GNSS board.
Access:	read-only
Type:	string
Values:	up to 32 alphanumeric symbols
Default:	N/A
Usage Guidelines:	If your program configures its functionality according to the firmware version loaded on the connected receiver, use this parameter to let the program know the version. With this parameter, you might also check if you are using up-to-date firmware.
Command Examples:	Displays firmware version. `print,/par/rcv/ver/main` Or `print,rcv/ver/main`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Firmware version on Power board

`[/par/]rcv/ver/pow/fw`

Description: The version of firmware running on the Power board.

Access: read-only

Type: string [0...4]

Values: up to four alphanumeric symbols
`none` means the receiver does not have firmware in power board.

Default: N/A

Usage Guidelines: With this parameter, you might check if you are using up-to-date firmware.

Command Examples: Displays firmware version.
```
print,/par/rcv/ver/pow/fw
```
Or
```
print,rcv/ver/pow/fw
```

Receiver Availability: This parameter is not available in the GMS-2.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Hardware version of GNSS board

`[/par/]rcv/ver/hw`

Description: The numerical identification of the hardware version of the GNSS board.

Access: read-only

Type: integer

Values: `[0...1023]`

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays hardware version.
```
print,/par/rcv/ver/hw
```
Or
```
print,rcv/ver/hw
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Hardware version of Power board

`[/par/]rcv/ver/pow/hw`

Description: The numerical identification of the hardware version of the Power board.

Access: read-only

Type: string [0...32]

Values:
- up to 32 alphanumeric symbols
- `none` means the receiver does not have a power board.

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays hardware version.
```
print,/par/rcv/ver/pow/hw
```
Or
```
print,rcv/ver/pow/hw
```

Receiver Availability: This parameter is available in receivers with Power board such as HiPer family, GR-3.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Boot loader version

`[/par/]rcv/ver/boot`

Description: The boot loader version.

Access: read-only

Type: integer

Values: `[0...128]`

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays the version of boot loader.
```
print,/par/rcv/ver/boot
```
Or
```
print,rcv/ver/boot
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Board version

`[/par/]rcv/ver/board`

Description: The GNSS board version and its revision number.

Access: read-only

Type: string [0...128]

Values: up to 128 alphanumeric symbols

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays the board version.
`print,/par/rcv/ver/board`
Or
`print,rcv/ver/board`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Board version of internal modem

`[/par/]rcv/ver/modem`

Description: The board version of internal modem.

Access: read-only

Type: string [0...128]

Values: up to 128 alphanumeric symbols

Default: N/A

Usage Guidelines: It is not recommended to use this parameter for checking the modem's board version because it currently shows `none` for all TPS receivers. Use Modem-TPS or corresponding modem commands to retrieve this information.

Command Examples: Displays the board version.
`print,/par/rcv/ver/modem`
Or
`print,rcv/ver/modem`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver serial number

`[/par/]rcv/sn`

Description: The serial number of receiver.

Access: read-only

Type: string

Values: up to 32 alphanumeric symbols

Default: N/A

Usage Guidelines: The value of this parameter can be set only by a special proprietary command.

Command Examples: Displays the serial number.
`print,/par/rcv/sn`
Or
`print,rcv/sn`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver electronic identifier

`[/par/]rcv/id`

Description: This parameter contains a piece of text uniquely identifying the receiver.

Access: read-only

Type: string [0...16]

Values: up to 16 alphanumeric symbols

Default: N/A

Usage Guidelines: The value of this parameter can be set only by a special proprietary command.

Command Examples: Displays the electronic identifier.
`print,/par/rcv/id`
Or
`print,rcv/id`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver configuration word

`[/par/]rcv/cfgw`

Description: This parameter displays the receiver configuration word in a hexadecimal format or an empty string (if not available).

Access: read-only

Type: string [0...128]

Values: up to 128 alphanumeric symbols

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays the configuration word.
`print,/par/rcv/cfgw`
Or
`print,rcv/cfgw`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Producer of GNSS board

`[/par/]rcv/vendor`

Description: This parameter contains a piece of text uniquely identifying the receiver.

Access: read-only

Type: enumerated

Values: `Topcon`

Default: N/A

Usage Guidelines: N/A

Command Examples: Displays a producer of GNSS board.
`print,/par/rcv/vendor`
Or
`print,rcv/vendor`

Receiver Availability: This parameter is available for GR-3, NET-G3, and receivers based on Turbo boards.

History: This parameter was introduced in firmware 2.5.

See Also: N/A

Receiver model

`[/par/]rcv/model`

Description:	Receiver model from a predefined list.
Access:	read-only
Type:	string [0...128]
Values:	up to 128 alphanumeric characters
Default:	N/A
Usage Guidelines:	If your program configures its functionality according to the receiver model, use this parameter to identify the receiver model.
Command Examples:	Displays the receiver model. `print,/par/rcv/model` Or `print,rcv/model`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Power Management

This section describes the parameters for managing and monitoring a receiver's power sources, power distribution in the receiver, battery charging and more.

Turn receiver off

`[/par/]power`

Description: Turns the receiver off.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – turns the receiver off.
- `on` – has no any effect.

Default: `on`

Usage Guidelines:

- This parameter does not turn off certain components in the receiver completely. For example, power and modem boards (if available) will still consume power. Consider putting the receiver to Zero Power Mode, if you are going to store it for a long time.
- If the receiver is off, it can not be turned on with this parameter.
- After turning the receiver on, the parameter's default value is immediately restored.

Command Examples: Turn the receiver off.

```
set,/par/power,off
```

Or

```
set,power,off
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to Operator's Manual supplied with your receiver for details on how to put the receiver to Zero Power Mode and how to turn on the receiver.

Sleep mode

`[/par/]sleep`

Description: Puts the receiver into sleep mode.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – has no any effect.
- `on` – receiver goes to sleep mode.

Default: `off`

Option: wake-up time as `time_spec`

Usage Guidelines:

- In sleep mode, the receiver goes into a dormant state, but still consumes very low power. This might cause the internal batteries to deplete completely over time. Consider putting the receiver to Zero Power Mode, if you are going to store it for a long time.
- If a character is received on one of the serial ports while the receiver is in sleep mode, power will be immediately switched on.
- You can awaken the receiver from a sleep state at a specified GPS time using the wake-up option.
- After waking up the receiver, the parameter's default value is immediately restored.

Command Examples: Put the receiver into sleep.

```
set,/par/sleep,on
```

Or

```
set,sleep,on
```

Put the receiver into sleep mode and then wake up on next Monday (2d) at 23h3m55s GPS time.

```
set,/par/sleep,on:2d23h3m55s
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to Operator's Manual supplied with your receiver for details on how to put the receiver to Zero Power Mode.

Low power mode for processor

`[/par/]lpm`

Description: Manages the processor's low power mode (LPM).

Access: read & write

Type: boolean

Values: `off|on`

- `off` – processor will not go into LPM when it is idling.
- `on` – processor will go into LPM when it is idling.

Default: `on`

Usage Guidelines: The receiver will consume less power while the processor is in LPM.

Command Examples: Enable LPM.
`set,/par/lpm,on`
Or
`set,lpm,on`

Query the current state of LPM.
`print,/par/lpm`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reset receiver

`[/par/]reset`

Description: Executes a hardware reset of the GNSS board.

Access: write-only

Type: boolean

Values: `yes|no`

- `yes` – resets the receiver's GNSS board.
- `no` – has no any effect.

Default: `no`

Usage Guidelines:

- From a functional point of view, hardware reset is equivalent to turning the power off and then back on.
- This parameter resets only GNSS board; it does not reset any other boards in the receiver (i.e., power, modem boards).
- After resetting the receiver, the parameter's default value is immediately restored.
- If you record a raw data file and execute this reset, the file will be closed.
- After changing certain parameters, you are required to reset the receiver.

Command Examples: Execute reset.

```
set,/par/reset,yes
```

Or

```
set,reset,yes
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Voltage of external power supply

`[/par/]pwr/ext`

Description:	Displays the external power supply's voltage.
Access:	read-only
Type:	float
Values:	[`0.00...cur_value`]; expressed in volts
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Query the voltage of external power supply. `print,/par/pwr/ext` Or `print,pwr/ext`
Receiver Availability:	This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Power source

`[/par/]pwr/mode`

Description: With this parameter, you select a power source from which the receiver will consume power.

Access: read & write

Type: enumerated

Values: `auto|mix|ext|a|b`

- `auto` – the receiver will automatically select the most appropriate power source according to the following algorithm:
  - If an external power source is detected, the receiver will use this. Otherwise, the receiver will be powered by its batteries.
  - For the HiPer family, if both batteries are mounted and no external power is detected, the receiver will start with the battery having the higher voltage, run it down until a discharged state, and then switch to the second battery.
  - For the GR-3 receiver, if both batteries are mounted and no external power is detected, the receiver will start with the battery having the lower voltage and will constantly switch from one battery to the other, maintaining a difference of 0.4 V between the batteries.
- `mix` – the receiver will automatically identify and start using the power source with the highest voltage. It is not recommended to run the receiver in this mode for a long time unless it is powered by a high capacity external source. Note that in `mix` mode, power consumption is about 10% higher than in other modes.
  - For the GR-3 receiver, if both batteries are mounted and no external power is detected, the receiver will discharge the batteries almost simultaneously.
- `ext` – the receiver will consume power from an external power source.
- `a` – the receiver will consume power from battery A.
  - For the HiPer family and Odyssey-RS, the receiver will run until battery A is discharged.
  - For the GR-3 receiver, both batteries will discharge in sequence: first A, then B, A, B, and so on. In this case, battery B will last 1 to 2 hours longer.
- `b` – the receiver will consume power from battery B.
  - For the HiPer family and Odyssey-RS, the receiver will run until battery B is discharged.
  - For the GR-3 receiver, both batteries will discharge in sequence: first B, then A, B, A, and so on. In this case, battery A will last 1 to 2 hours longer.

Default: `auto`

Usage Guidelines:

- The value `auto` is recommended. This approach ensures that you will have enough time to attach a fresh battery without an interruption in receiver operation.
- The `print` command reports the value that has been applied with the last `set` command (not the actual power source being used by the receiver).

Command Examples: Use an external power source.

```
set,/par/pwr/mode,ext
```

`[/par/]pwr/mode` *(Continued)*

Receiver Availability:
- This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.
- The range of supported values might vary depending on the connected receiver.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the Actual power source parameter to find out which power source is currently in use by the receiver.

Actual power source

`[/par/]pwr/curmode`

Description: This parameter reports the current power source.

Access: read-only

Type: enumerated

Values: `mix|ext|a|b|extbat`

- `mix` – the receiver is using the power source with the highest voltage. It is not recommended to run the receiver in this mode for a long time unless it is powered by a high capacity external source. Note that in mix mode, power consumption is about 10% higher than in other modes.
  - For the GR-3 receiver, if both batteries are mounted and no external power is detected, the receiver will discharge the batteries almost simultaneously.
- `ext` – the receiver is consuming power from an external power source.
- `a` – the receiver is consuming power from battery A.
  - For the HiPer family and Odyssey-RS, the receiver will run until battery A is discharged.
  - For the GR-3 receiver, both batteries will discharge in sequence: first A, then B, A, B, and so on. In this case, battery B will last 1 to 2 hours longer.
- `b` – the receiver is consuming power from battery B.
  - For the HiPer family and Odyssey-RS, the receiver will run until battery B is discharged.
  - For the GR-3 receiver, both batteries will discharge in sequence: first B, then A, B, A, and so on. In this case, battery A will last 1 to 2 hours longer.
- `extbat` – available for the GR-3 receiver only; receiver is consuming power from the battery charging cradle or via pin #5 of the power connector.

Default: N/A

Usage Guidelines: The value `auto` is recommended. This approach ensures that you will have enough time to attach a fresh battery without an interruption in receiver operation.

Command Examples: Report the current power source.

```
print,/par/pwr/curmode
```

Receiver Availability:

- This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.
- The range of supported values might vary depending on the connected receiver.

History:

- This parameter was introduced in the first release of firmware.
- The `extbat` value was implemented in firmware 3.0.

See Also: N/A

Battery charge mode

`[/par/]pwr/charge/bat`

Description: Charges the corresponding internal battery or batteries in the receiver.

Access: read & write

Type: enumerated

Values: `auto|off|a|b`

- `auto` – the receiver will automatically control the course of battery charging. The receiver automatically detects all of the batteries attached to the receiver (A, B, or both) and, if both are mounted, charges them one after the other for the HiPer family and Odyssey-RS (beginning with battery A) or simultaneously for the GR-3 receiver. Once the batteries are fully charged, the receiver stops charging automatically.
- `off` – the receiver will not charge batteries.
- `a` – available for the HiPer family and Odyssey-RS; the receiver will charge only battery A. Once the battery is fully charged, the receiver stops charging automatically.
- `b` – available for the HiPer family and Odyssey-RS; the receiver will charge only battery B. Once the battery is fully charged, the receiver stops charging automatically.

Default: `auto`

Usage Guidelines:

- You can start battery charging only after the parameter `/par/pwr/mode` has been set to `ext` or `auto`.
- The receiver will charge the batteries even if it is turned off. In Zero Power Mode the receiver will not charge the batteries.
- The `print` command reports the value that has been applied with the last `set` command (not the actual charging mode being used by the receiver).

Command Examples: Disable battery charging.

```
set,/par/pwr/charge/bat,off
```

Or

```
set,pwr/charge/bat,off
```

Query the state of battery charging.

```
print,/par/pwr/charge/bat
```

Receiver Availability:

- This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.
- The range of supported values might vary depending on the connected receiver.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the Actual power source parameter to find out which power source is currently in use by the receiver.

Actual mode used for battery charging

`[/par/]pwr/charge/curbat`

Description: Reports which of the charging modes is effective at the time of query.

Access: read-only

Type: enumerated

Values: `off|a|b|ab`

- `off` – the receiver is not charging batteries.
- `a` – the receiver is charging only battery A.
- `b` – the receiver is charging only battery B.
- `ab` – for the GR-3 receiver only; both batteries are charging simultaneously.

Default: N/A

Usage Guidelines: N/A

Command Examples: Report the current charging mode.

```
print,/par/pwr/charge/curbat
```

Receiver Availability:

- This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.
- The range of supported values might vary depending on the connected receiver.

History:

- This parameter was introduced in the first release of firmware.
- The `ab` value was implemented in firmware 3.0.

See Also: N/A

Battery voltage

`[/par/]pwr/bat/<a|b>`

Description: Displays the voltage of the corresponding battery.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: The GR-3, HiPer family, and Odyssey-RS normally accommodate up to two batteries A and B. They can be either detachable or not removable depending on the receiver model. These parameters allow querying the voltage of batteries A and B, respectively.

Command Examples: Query the voltage of battery A.
`print,/par/pwr/bat/a`
Or
`print,pwr/bat/a`

Query the voltage of both batteries.
`print,pwr/bat`
Or
`print,pwr/bat:on`

Receiver Availability: This parameter is not available in the NET-G3, GB-1000, GB-500, and GMS-2.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GNSS board voltage

`[/par/]pwr/board`

Description: Displays the voltage that GNSS board draws.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: With this parameter, you can find out the *true* voltage the receiver board draws. This voltage is measured directly on the board excluding the voltage drop across the power switching circuitry.

Command Examples: Query the voltage of GNSS board.
`print,/par/pwr/board`
Or
`print,pwr/board`

Receiver Availability: This parameter is not available in the GMS-2.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Backup battery voltage

`[/par/]pwr/backup`

Description: Reports the voltage of a battery used to preserve the contents of the NVRAM when the receiver is turned off.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines:
- The backup batteries are rated for backup use for 10 years.
- Battery replacement is required if the voltage drops below 3.3 V.

Command Examples: Query the voltage of backup battery.
`print,/par/pwr/backup`
Or
`print,pwr/backup`

Receiver Availability: This parameter is not available in the GMS-2.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output voltage of DC-DC converter for charge

`[/par/]pwr/charger`

Description: Reports the output voltage of the receiver's DC-DC converter used for battery charging.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: The receiver displays a meaningful value only when batteries are being charged.

Command Examples: Query the voltage of the receiver's battery charger.
```
print,/par/pwr/charger
```
Or
```
print,pwr/charger
```

Receiver Availability: This parameter is available in the HiPer family, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output voltage of DC-DC converter for power distribution

`[/par/]pwr/conv`

Description: Reports the output voltage of the receiver's DC-DC converter used for supplying a constant voltage to various operational elements of the receiver.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines:
- The receiver displays a meaningful value only when an external power source is used.
- The converter outputs 9.8 V if an external power source supplies voltage above 11 V. For other accepted voltage levels, the converter will output voltage that is about 0.8 V lower than the supplied voltage.

Command Examples: Query the voltage of the converter.
```
print,/par/pwr/conv
```
Or
```
print,pwr/conv
```

Receiver Availability: This parameter is available in the HiPer family, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output voltage on serial ports

`[/par/]pwr/extports`

Description: Reports the output voltage supplied by the receiver to the corresponding pin of the serial port connectors.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines:
- The receiver displays meaningful value only for 7-pin ODU connectors.
- The standard voltage output is 12 V DC ± 5% at no load.

Command Examples: Query the voltage on serial ports.
```
print,/par/pwr/extports
```
Or
```
print,pwr/extports
```

Receiver Availability: This parameter is available in the HiPer family receivers based on the Turbo boards.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Voltage across the board's 3-volt power stabilizer (digital)

`[/par/]pwr/d3v`

Description: Reports the voltage across the digital part of the GNSS board.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the voltage across the digital part of the GNSS board.
```
print,/par/pwr/d3v
```
Or
```
print,pwr/d3v
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Voltage across the board's 3-volt power stabilizer (analog)

`[/par/]pwr/a3v`

Description: Reports the voltage across the analog part of the GNSS board.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the voltage across the analog part of the GNSS board.
```
print,/par/pwr/a3v
```
Or
```
print,pwr/a3v
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Voltage across the board's 5-volt power stabilizer (digital)

`[/par/]pwr/d5v`

Description: Reports the voltage across the digital part of the GNSS board.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the voltage across the digital part of the GNSS board.
```
print,/par/pwr/d5v
```
Or
```
print,pwr/d5v
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Voltage across the board's 5-volt power stabilizer (analog)

`[/par/]pwr/a5v`

Description: Reports the voltage across the analog part of the GNSS board.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the voltage across the analog part of the GNSS board.
`print,/par/pwr/a5v`
Or
`print,pwr/a5v`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

External antenna voltage

`[/par/]pwr/extant`

Description: Reports the voltage supplied by the receiver to an external antenna.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in volts

Default: N/A

Usage Guidelines:
- A meaningful value is displayed only when you configure the receiver for external antenna.
- The standard voltage output is 4.6 ± 0.1 V DC at a 50-Ohm load.

Command Examples: Query the voltage supplied to an external antenna.
`print,/par/pwr/extant`
Or
`print,pwr/extant`

Receiver Availability: This parameter is available in the HiPer family and Odyssey-RS receivers.

History: This parameter was introduced in the first release of firmware.

See Also:
- Refer to the parameters `/par/ant/rcv/inp` on page 3-84 and `/par/ant/rcv/curinp` on page 3-85 for information on how to select an external antenna and check the actual antenna the receiver uses.
- Refer to technical specifications for a specific receiver.

External antenna current

`[/par/]pwr/extantdc`

Description: Reports the current supplied by the receiver to an external antenna.

Access: read-only

Type: float

Values: [`0.00...cur_value`]; expressed in milliamperes

Default: N/A

Usage Guidelines:
- A meaningful value is displayed only when you connect the external antenna with the receiver and it is configured for external antenna.
- If the current is equal or greater than 100 mA, the receiver's overcurrent protection device cuts off the antenna power circuit.

Command Examples: Query the current for the connected external antenna.
`print,/par/pwr/extantdc`
Or
`print,pwr/extantdc`

Receiver Availability: This parameter is available in the HiPer family and Odyssey-RS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the parameters `/par/ant/rcv/inp` on page 3-84 and `/par/ant/rcv/curinp` on page page 3-85 for information on how to select an external antenna and check the actual antenna the receiver uses.

Status of external antenna connection

`[/par/]ant/dc`

Description: Reports if the antenna draws DC power.

Access: read-only

Type: enumerated

Values: `off | normal | overload`

- `off` – the external antenna draws no DC from the receiver.
- `normal` – the external antenna draws normal DC from the receiver.
- `overload` – the external antenna draws current higher that expected from the receiver.

Default: `cur_value`

Usage Guidelines: The `overload` status can be displayed if the receiver is subjected to a greater load than it was designed for or if a short circuit occurs in the antenna, antenna cable, or the receiver.

Command Examples: Check the antenna status.

```
print,/par/ant/dc
```

Or

```
print,ant/dc
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduce in the first release of firmware.

See Also: N/A

Output current of DC-DC converter for charge

`[/par/]pwr/chdc`

Description:	Reports the output current of the receiver's DC-DC converter used for battery charging.
Access:	read-only
Type:	float
Values:	[`0.00...cur_value`]; expressed in milliamperes
Default:	N/A
Usage Guidelines:	The receiver displays meaningful value only when batteries are being charged.
Command Examples:	Query the current of the receiver's battery charger. `print,/par/pwr/chdc` Or `print,pwr/chdc`
Receiver Availability:	This parameter is available in the HiPer family, Odyssey-RS, and GR-3 receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Power output on serial ports

`[/par/]pwr/ports`

Description: Configures power output on the serial ports.

Access: read & write

Type: enumerated

Values: `on|off|always`

- `on` – the power board will deliver voltage on pin one of all 7-pin ODU serial port connectors when the receiver is turned on.
- `off` – the power will be absent even if the receiver is turned on.
- `always` – the power board will deliver voltage on pin one of all 7-pin ODU serial port connectors even if the receiver is turned off.

Default: Defined by `/par/pwr/ports&def`.

Usage Guidelines: The standard voltage output is 12 V DC ± 5%.

Command Examples: Disable power output on the serial ports.
`set,/par/pwr/ports,off`
Or
`set,pwr/ports,off`

Query power output on the serial ports.
`print,/par/pwr/ports`
Or
`print,pwr/ports`

Receiver Availability: This parameter is available in the HiPer family and Odyssey-RS receivers based on the Turbo boards.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the parameter `/par/pwr/extports` on page 3-25 for how to check the actual voltage outputted on ports.

Default for power output on serial ports

`[/par/]pwr/ports&def`

Description: Configures the default value for power output on the serial ports.

Access: read & write

Type: enumerated

Values: `on|off|always`

- `on` – the power board will deliver voltage on pin one of all 7-pin ODU serial port connectors when the receiver is turned on.
- `off` – the power will be absent even if the receiver is turned on.
- `always` – the power board will deliver voltage on pin one of all 7-pin ODU serial port connectors even if the receiver is turned off.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set the default value to always.

```
set,/par/pwr/ports&def,always
```

Or

```
set,pwr/ports&def,always
```

Query the default value.

```
print,/par/pwr/ports&def
```

Or

```
print,pwr/ports&def
```

Receiver Availability: This parameter is available in the HiPer family and Odyssey-RS receivers based on the Turbo boards.

History: This parameter was introduced in firmware 2.5.

See Also: N/A

Power output to internal slots

`[/par/]pwr/slots`

Description: Governs power output to the internal slots.

Access: read & write

Type: enumerated

Values: `on|off|always`

- `on` – when the receiver is on, all internal slots will receiver power. If the receiver is turned off, no power will be sent to the slots.
- `off` – internal slots will not be powered, even if the receiver is turned on.
- `always` – all internal slots will be powered even if the receiver is turned off.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set power output to always.

```
set,/par/pwr/slots,always
```

Or

```
set,pwr/slots,always
```

Query power output to the internal slots.

```
print,/par/pwr/slots
```

Or

```
print,pwr/slots
```

Receiver Availability: This parameter is available in the HiPer family, Map-R/RT/HP, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Default for power output to internal slots

`[/par/]pwr/slots&def`

Description: Governs default power output to the internal slots.

Access: read & write

Type: enumerated

Values: `on|off|always`

- `on` – when the receiver is on, all internal slots will receiver power. If the receiver is turned off, no power will be sent to the slots.
- `off` – internal slots will not be powered, even if the receiver is turned on.
- `always` – all internal slots will be powered even if the receiver is turned off.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set default power output to always.

```
set,/par/pwr/slots&def,always
```

Or

```
set,pwr/slots&def,always
```

Query default power output to the internal slots.

```
print,/par/pwr/slots&def
```

Or

```
print,pwr/slots&def
```

Receiver Availability: This parameter is available in the HiPer family, Map-R/RT/HP, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in firmware 2.5.

See Also: N/A

Managing internal slots

`[/par/]pwr/switch`

Description: Manages the internal slots.

Access: read & write

Type: array of three enumerated values

Values: `{sw2,sw3,sw4}`, where each `sw` corresponds to the particular slot and can be `y | n | always`.

- `y` – when the receiver is on, the internal slot is turned on. If the receiver is turned off, the slot is turned off too.
- `n` – the slot is turned off, even if the receiver is turned on.
- `always` – the internal slot will be turned on even if the receiver is turned off.

Default: `{y,y,y}`

Usage Guidelines: The GR-3 has only one available slot (`sw2`).

Command Examples: Turn the slots as follows.

```
set,/par/pwr/switch,{always,y,n}
```

Or

```
set,pwr/switch,{always,y,n}
```

Query the slots status.

```
print,/par/pwr/switch
```

Or

```
print,pwr/switch
```

Receiver Availability: This parameter is available in the HiPer family, Map-R/RT/HP, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Managing specified internal slot

`[/par/]pwr/switch/<N>`

Description: Manages the specified internal slot.

Access: read & write

Type: enumerated

Values: `y | n | always`

- `y` – when the receiver is on, the specified internal slot is turned on. If the receiver is turned off, the slot is turned off too.
- `n` – the specified slot is turned off, even if the receiver is turned on.
- `always` – the specified internal slot will be turned on even if the receiver is turned off.

Default: `y`

Usage Guidelines:

- `N` stands for 2, 3, and 4.
- The GR-3 has only one available slot (`sw2`).
- Do not confuse the `/par/pwr/slots` and `/par/pwr/switch` parameters. With the `/par/pwr/slots` parameter, you instruct the receiver to apply/remove power to/from all internal slots of the receiver. With the `/par/pwr/switch` parameter, you instruct the receiver to use a control signal to turn the corresponding slot on/off.

Command Examples:

Set slot #2 to always.
```
set,/par/pwr/switch/2,always
```
Or
```
set,pwr/switch/2,always
```

Query the status of slot #3.
```
print,/par/pwr/switch/2
```
Or
```
print,pwr/switch/2
```

Receiver Availability: This parameter is available in the HiPer family, Map-R/RT/HP, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Default for specified internal slot

`[/par/]pwr/switch/<N>&def`

Description: Configure the default value for the specified internal slot.

Access: read & write

Type: enumerated

Values: `y | n | always`

- `y` – when the receiver is on, the specified internal slot is turned on. If the receiver is turned off, the slot is turned off too.
- `n` – the specified slot is turned off, even if the receiver is turned on.
- `always` – the specified internal slot will be turned on even if the receiver is turned off.

Default: `y`

Usage Guidelines:

- `N` stands for 2, 3, and 4.
- The GR-3 has only one available slot (`sw2`).

Command Examples: Set the default for slot #2 to always.

```
set,/par/pwr/switch/2&def,always
```

Or

```
set,pwr/switch/2&def,always
```

Query the default for slot #3.

```
print,/par/pwr/switch/3&def
```

Or

```
print,pwr/switch/3&def
```

Receiver Availability: This parameter is available in the HiPer family, Map-R/RT/HP, Odyssey-RS, and GR-3 receivers.

History: This parameter was introduced in firmware 2.5.

See Also: N/A

Parameters used only in discontinued receivers

Battery charging mode

`[/par/]pwr/charge/speed`

Description: This parameter is used to toggle the battery charge current between 1 ampere or 2 ampere.

Access: read & write

Type: enumerated

Values: `normal|fast`

- `normal` – the battery charge current is 1 ampere.
- `fast` – the battery charge current is 2 ampere.

Default: `normal`

Usage Guidelines:

- The normal mode (charge current = 1 ampere) is preferable. It is recommended to use normal mode when charging batteries overnight.
- This parameter returns the value set with the last `set,/par/pwr/charge/speed` command, not necessarily the actual charging mode effective at the time this command is issued.

Command Examples: Set the charging to fast mode.

```
set,/par/pwr/charge/speed,fast
```

Or

```
set,pwr/charge/speed,fast
```

Query charging mode.

```
print,/par/pwr/charge/speed
```

Or

```
print,pwr/charge/speed
```

Receiver Availability: This parameter is available in the Odyssey receivers based on the JPS_3 and JPS_4_x boards.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the parameter `/par/pwr/charge/curspeed` for actual charging mode.

Actual battery charging mode

`[/par/]pwr/charge/curspeed`

Description: Indicates which of the two charging modes (fast or normal) is currently effective.

Access: read-only

Type: enumerated

Values: `normal|fast`

- `normal` – the battery charge current is 1 ampere.
- `fast` – the battery charge current is 2 ampere.

Default: N/A

Usage Guidelines: When `auto` is used to charge the battery, the actual battery charge speed may differ from the value set with the `set,/par/pwr/charge/speed` command.

Command Examples: Query actual charging mode.
`print,/par/pwr/charge/curspeed`
Or
`print,pwr/charge/curspeed`

Receiver Availability: This parameter is available in the Odyssey receivers based on the JPS_3 and JPS_4_x boards.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Power output on ports A and B

`[/par/]pwr/out/ab`

Description: Turns power output on/off on ports A and B.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the power will be absent on the power pins of ports A and B.
- `on` – the power will be delivered to the power pins of ports A and B.

Default: `off`

Usage Guidelines:

- With this parameter, you can reduce the power consumption when the removable UHF radio is attached to the receiver but not activated. Recall that removable UHF radio is powered via the corresponding pins on port A or B (depending on which tray is used).
- When using a removable UHF radio with the Odyssey, ensure that the parameter `/par/pwr/out/ab` is set to `on`.
- This parameter makes sense only if the Odyssey has a modified interconnection board. For older models of the Odyssey receiver, the parameter cannot be used. An error message will be reported indicating that the receiver does not have the power consumption saving feature. In older Odysseys, power is always supplied on the power pins of ports A and B. Recall that these pins are only available through the receiver's right- and left-hand trays respectively.

Command Examples: Enable power output on ports A and B.

```
set,/par/pwr/out/ab,on
```

Or

```
set,pwr/out/ab,on
```

Receiver Availability: This parameter is available in the Odyssey receivers based on the JPS_3 and JPS_4_x boards.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Power output on port C

`[/par/]pwr/out/c`

Description: Turns power output on/off on port C.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the power will be absent on the power pin of port C.
- `on` – the power will be delivered to the power pin of port C.

Default: `off`

Usage Guidelines:

- With this parameter, you can reduce the power consumption when an external radio is attached to your Odyssey but is not activated. Recall that an external radio, which is normally connected to port C of the Odyssey receiver, is powered via the corresponding pin on this port.
- This parameter makes sense only if the Odyssey has a modified interconnection board. For older models of the Odyssey receiver, the parameter cannot be used. An error message will be reported indicating that the receiver does not have the power consumption saving feature. In older Odysseys power is always supplied on the power pin of port C.

Command Examples: Enable power output on port C.

```
set,/par/pwr/out/c,on
```

Or

```
set,pwr/out/c,on
```

Receiver Availability: This parameter is available in the Odyssey receivers based on the JPS_3 and JPS_4_x boards.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Memory and File Management

This section describes parameters for:

- Working with current and existing log files in the receiver memory.
- Displaying information about the memory module and file system in the receiver.
- Configuring automatic rotation of log files.
- Configuring output of two concurrent log files.

Memory

Size of receiver's RAM

`/par/rcv/mem`

Description:	Reports the size of RAM in the connected receiver.
Access:	read-only
Type:	integer
Values:	`cur_value`; kilobytes
Default:	`cur_value`
Usage Guidelines:	N/A
Command Examples:	Query the RAM size. `print,/par/rcv/mem` Or `print,rcv/mem`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Free memory in NVRAM

`/par/dev/nvm/a/free`

Description:	Reports the free memory in the receiver's NVRAM.
Access:	read-only
Type:	integer
Values:	`cur_value`; bytes
Default:	`cur_value`
Usage Guidelines:	N/A
Command Examples:	Query the free memory in NVRAM. `print,/par/dev/nvm/a/free` Or `print,dev/nvm/a/free`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Access time

`/par/rcv/flash/waitstates`

Description:	Reports how fast the receiver's memory card.
Access:	read-only
Type:	integer
Values:	`[0...9]`
Default:	`cur_value`
Usage Guidelines:	The smaller the returned value, the faster the memory. For most TPS receivers, this parameter is 1.
Command Examples:	Query the memory access time. `print,/par/rcv/flash/waitstates` Or `print,rcv/flash/waitstates`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Number of blocks available in the receiver's memory card

`/par/dev/blk/a&blocks`

Description: Reports the total amount of physical memory available on the receiver's data storage device.

Access: read-only

Type: integer

Values: `cur_value`; blocks

Default: `cur_value`

Usage Guidelines: Data storage device can be either a CompactFlash or Secure Digital card depending on the receiver used.

Command Examples: Query the number of blocks.
`print,/par/dev/blk/a&blocks`
Or
`print,dev/blk/a&blocks`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Block size in the receiver's memory card

`/par/dev/blk/a&block_size`

Description: Reports the block size of physical memory available on the receiver's data storage device.

Access: read-only

Type: integer

Values: `cur_value`; bytes

Default: `cur_value`

Usage Guidelines: Data storage device can be either a CompactFlash or Secure Digital card depending on the receiver used.

Command Examples: Query the block size.
`print,/par/dev/blk/a&block_size`
Or
`print,dev/blk/a&block_size`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Working with Log Files

Information about all existing log files

`/log`

Description:	Displays the list of raw data files stored in the receiver memory. Along with the file names, this list contains their sizes in bytes and last-modified times. Last-modified time is given is seconds since 00:00:00 January 1, 1988.
Access:	read-only
Type:	list
Values:	`{size,yyyyYmmMddDHHhMMmSSs}`, where size is in bytes; the lower case letters y, m, d and upper case letters HH, MM, SS will be numbers.
Default:	`{}`
Usage Guidelines:	N/A
Command Examples:	Query the list of existing log files. `print,/log` Or `print,/log:on`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Information about specific log file

`/log/<NAME>`

Description: Displays the specific raw data files stored in the receiver memory. Along with the file name, this list contains its size in bytes and last-modified time. Last-modified time is given is seconds since 00:00:00 January 1, 1988.

Access: read-only

Type: list of two values

Values: N/A

Default: N/A

Usage Guidelines: In this parameter, `NAME` must be replaced with the actual file name.

Command Examples: Query the information about existing log file `Static`.
`print,/log/Static`
Or
`print,/log/Static:on`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Size of specific log file

`/log/<NAME>&size`

Description: Displays the size of the specified raw data file.

Access: read-only

Type: integer

Values: `cur_value`; bytes

Default: N/A

Usage Guidelines: In this parameter, `NAME` must be replaced with the actual file name.

Command Examples: Query the size of log file `GNSS_file`.
`print,/log/GNSS_file&size`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Current log file

`/cur/file/<a|b>`

Description: Contains the name of the current log file, if such a file exists, or an empty string, otherwise.

Access: read & write

Type: string

Values: any existing file name or empty name

Default: empty name

Usage Guidelines:

- If there is no current log file, then setting this parameter to an existing file's name will instruct the receiver to open this file for data appending thus making it the new current file.
- If the current log file exists, then changing this parameter will instruct the receiver to close the existing current file and then open a new current file.
- If the command refers to the file that does not exist, the receiver will ignore the command.
- To create files, use the command `create` described on page 2-11.

Command Examples: Example with the `set` command.

```
set,/cur/file/b,static_session
```

Example with the `print` command.

```
print,/cur/file/a
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Size of current log file

`/cur/file/<a|b>&size`

Description:	Reports the size of log file being recorded.
Access:	read-only
Type:	integer
Values:	`cur_value`; bytes
Default:	`cur_value`
Usage Guidelines:	An error message will be reported if there is no current log file.
Command Examples:	Query the size of the current log file a. `print,/cur/file/a&size`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Recover deleted log files

`/par/dev/blk/a/removed`

Description: Recovers deleted log files.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the receiver's file system behaves as usual (i.e., the file recovering feature is unavailable).
- `on` – the receiver's file system allows you to view and restore the deleted files.

Default: `off`

Usage Guidelines: The receiver's file system will be remounted every time the command `set,/par/dev/blk/a/removed,on` is executed.
The alternative way to force the file system to remount is clearing the NVRAM, but this may be unacceptable in many cases.
Should you accidentally delete raw data file(s), make sure you strictly follow the pre-recovery guidelines below:

- Do not record any new raw data files to the internal memory of the receiver from which the file was deleted. Recording new files may overwrite the deleted file and make it permanently lost.
- Do not attempt to initialize the file system. This procedure permanently erases all files inside the receiver.
- Do not try to remove the receiver's memory card and read its contents with a cardreader.
- Typical steps you should take when recovering the deleted files are described in the PC-CDU User's Manual (Chapter 2, How to Recover Deleted Files).

Since this parameter is not stored in the receiver's NVRAM, it will always be set to `off` after power cycling the receiver.
When you enable this mode, only the deleted file(s) will display.

Command Examples: Enables file recovering.
`set,/par/dev/blk/a/removed,on`
Or
`set,dev/blk/a/removed,on`

Query the parameter's value.
`print,/par/dev/blk/a/removed`
Or
`print,dev/blk/a/removed`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File system

Available memory for raw data logging

`/par/dev/blk/a/size`

Description: Reports the total capacity of the internal memory card that is enabled for logging raw data files.

Access: read-only

Type: integer

Values: `cur_value`; bytes

Default: `cur_value`

Usage Guidelines: This parameter shows the capacity enabled by the option _MEM. The actual physical capacity of the memory card can be obtained by multiplying the values of `/dev/blk/a&blocks` and `/dev/blk/a&block_size`.

Command Examples: Query the available memory.
`print,/par/dev/blk/a/size`
Or
`print,dev/blk/a/size`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Free memory for raw data logging

`/par/dev/blk/a/free`

Description: Reports the amount of free space on the internal memory card used for logging raw data files.

Access: read-only

Type: integer

Values: `cur_value`; bytes

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the free memory.
`print,/par/dev/blk/a/free`
Or
`print,dev/blk/a/free`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File system block size

`/par/dev/blk/a/block_size`

Description: Reports the block size of file system in the connected receiver.

Access: read-only

Type: integer

Values: `cur_value`; bytes

Default: `cur_value`

Usage Guidelines: In essence, the block size is the smallest amount of receiver's memory that is allocated by the receiver's file system to hold the raw measurement data.

Command Examples: Query the block size.
`print,/par/dev/blk/a/block_size`
Or
`print,dev/blk/a/block_size`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of blocks in file system

`/par/dev/blk/a/blocks`

Description: Reports the number of blocks in the file system.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the number of blocks.
`print,/par/dev/blk/a/block`
Or
`print,dev/blk/a/block`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum number of files in the file system

`/par/dev/blk/a/max_files`

Description: Reports the maximum number of raw data files that the file system can create.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines:
- Although this number varies depending on the receiver used, normally it equals 512 or 1024.
- The file system will refuse to create a new file if the current number of files in the receiver is greater or equal to the value of this parameter.

Command Examples: Query the maximum number of files.
`print,/par/dev/blk/a/max_files`
Or
`print,dev/blk/a/max_files`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of files in the file system

`/par/dev/blk/a/files`

Description: Reports the number of raw data files the file system has already created and stored in the receiver memory.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the number of files.
`print,/par/dev/blk/a/files`
Or
`print,dev/blk/a/files`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of bad blocks

`/par/dev/blk/a/bad_blocks`

Description: Reports the current number of bad blocks (if any) in the receiver memory.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: It is recommended to initialize the file system regularly (e.g., once a month). This procedure checks for bad blocks. See “init” on page 2-15 for how to perform file system initialization.

Command Examples: Query the number of bad blocks.
`print,/par/dev/blk/a/bad_blocks`
Or
`print,dev/blk/a/bad_blocks`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Verification of block recording

`/par/dev/blk/a/verify`

Description: Reports what type of verification used to ensure that data has been correctly written.

Access: write & read

Type: enumerated

Values: `off | slow | fast`

Default: `cur_value`

Usage Guidelines: Although you could formally use the `set` command to change the default value, it is recommended to do so only under the guidance of Topcon technician.

Command Examples: Query the state of verification.
`print,/par/dev/blk/a/verify`
Or
`print,dev/blk/a/verify`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File system version

`/par/dev/blk/a/ver`

Description: Reports the version of file system as hexadecimal.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the file system version.
`print,/par/dev/blk/a/ver`
Or
`print,dev/blk/a/ver`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Time-tag of the last initialization of the file system

`/par/dev/blk/a/tfmt`

Description: Reports the time-tag to which the most recent file system initialization applies.

Access: read-only

Type: integer

Values: `cur_value`; seconds

Default: `cur_value`

Usage Guidelines: The time is measured since 00:00:00 January 1, 1988.

Command Examples: Query the time-tag of the last initialization.
`print,/par/dev/blk/a/tfmt`
Or
`print,dev/blk/a/tfmt`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File system initialization status

`/par/stat/fsinit`

Description: Reports the status of the file system initialization.

Access: read-only

Type: list of two integer values

Values: `{int1,int2}`

- `int1` – the total number of blocks used for file storage.
- `int2` – the number of blocks that are already processed.

Default: `cur_value`

Usage Guidelines:

- These two values allow monitoring of the file system initialization or remount progress.
- When initialization/remount is not in progress, these two values are the same.

Command Examples: Query the initialization status.
`print,/par/stat/fsinit`
Or
`print,stat/fsinit`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Implicit message output period for MINTER and AFRM

`[/par/]log/sc/period`

Description: This parameter specifies the interval to output messages to the log-file when data logging is activated with the MINTER FN button or through the automatic file rotation algorithm. In these cases, it is impossible to explicitly define the message output interval, thus the parameter name "implicit period".

Access: read & write

Type: float, seconds

Values: `[0...86400]`

Default: `1`

Usage Guidelines: N/A

Command Examples: Set the data logging interval (output messages to the log-file) to 10 seconds.

```
set,/par/log/sc/period,10.0
```

Or

```
set,log/sc/period,10.0
```

Query the current status

```
print,/par/log/sc/period
```

Or

```
print,log/sc/period
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File name prefix

`[/par/]log/cmd/create/prefix`

Description: This parameter determines the file name prefix used as part of the name of the new file the receiver will create when the MINTER FN button is pressed to start data recording.

Access: read & write

Type: string

Values: `up to 20 characters`

Default: `log`

Usage Guidelines: This parameter can be set to a string comprising up to 20 valid characters (for valid characters, please see the create command).

Command Examples: Set the log-file name prefix to roof.
`set,/par/cmd/create/prefix,roof`
Or
`set,cmd/create/prefix,roof`

Query the current status
`print,/par/cmd/create/prefix`
Or
`print,cmd/create/prefix`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “create” on page 2-11 for valid characters.

Automatic File Rotation Mode (AFRM)

TPS receivers are capable of automatically rotating log-files. The term *file rotation* means that the receiver closes the current log-file and opens a new one according to a user-defined schedule. The parameters *file rotation period* and *file rotation phase* specify this schedule.

File rotation launches the moment the receiver time modulo *period* is equal to *phase*. More precisely, a new log-file opens immediately before the scheduled epoch causing data tagged with this epoch to be recorded into the new log-file.

In addition, AFRM uses a counter that decrements on every file rotation until its value becomes zero, then file rotation automatically stops. This feature allows you to create as many log-files as necessary and then stop data logging. The counter initializes simultaneously with AFRM, and its initial value is set through the corresponding parameter described below.

A log-file opens immediately after turning AFRM on. This start-up file is not considered a file rotation event; the AFRM counter will not decrement.

When opening a new log-file, the receiver enables the default set of messages output with the default output period. Both the default set of messages and the default output period are programmable.

When no free memory is available to record data, the TPS receiver is capable of removing the earliest log-files. If this feature and AFRM are enabled, your receiver will erase the files with the earliest file creation time/date.

Enable AFRM

`[/par/]log/rot/mode`

Description: Activates or deactivates AFRM.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – activate AFRM.
- `on` – deactivate AFRM.

Default: `off`

Usage Guidelines:

- If this parameter is changed from `off` to `on`, the receiver creates a log-file with an automatically generated name and starts data logging to it.
- If this parameter is changed from `on` to `off`, the receiver will disable AFRM and will also close the current log-file.

Command Examples: Enable AFRM.

```
set,/par/log/rot/mode,on
```

Or

```
set,log/rot/mode,on
```

Check the AFRM state.

```
print,/par/log/rot/mode
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

`[/par/]log/rot/mode` *(Continued)*

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File rotation period

`[/par/]log/rot/sc/period`

Description: Specifies the time duration of each log-file created in AFRM.

Access: read & write

Type: integer

Values: [`60...86400`]; expressed in seconds

Default: `3600`

Usage Guidelines: N/A

Command Examples: Configure the period to 1800 seconds.
`set,/par/log/rot/sc/period,1800`
Or
`set,log/rot/sc/period,1800`

Check the current value for period.
`print,/par/log/rot/sc/period`
Or
`print,log/rot/sc/period`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File rotation phase

`[/par/]log/rot/sc/phase`

Description: Specifies the phase (constant time shift) of creating multiple log-files in AFRM.

Access: read & write

Type: integer

Values: [`0...86400`]; expressed in seconds

Default: `0`

Usage Guidelines:

- Apply the desired phase whenever you need to have a constant time shift or delay in the occurrence of file rotation from the moment when the receiver time is a multiple of period. For example, apply 14 seconds if you want to account for difference between GPS time and UTC time for log-files created in AFRM. In this case, all AFRM files will open with the 14-second delay.
- The phase must be less than the period. Otherwise, file rotation will not occur.

Command Examples: Configure the phase to 14 seconds.

```
set,/par/log/rot/sc/phase,14
```

Or

```
set,log/rot/sc/phase,14
```

Check the current value for period.

```
print,/par/log/rot/sc/phase
```

Or

```
print,log/rot/sc/phase
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to “Defining Message Generation” on page 1-18 for details on how phase and period work.

File rotation counter

`[/par/]log/rot/sc/count`

Description: Specifies how many log-files must be created in AFRM until this mode turns off.

Access: read & write

Type: integer

Values: [0...2^{31}-1]; expressed in seconds
0 – unlimited number of files.

Default: 0

Usage Guidelines: N/A

Command Examples: Configure the counter to 20.
`set,/par/log/rot/sc/count,20`
Or
`set,log/rot/sc/count,20`

Check the current value for period.
`print,/par/log/rot/sc/count`
Or
`print,log/rot/sc/count`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

File rotation running counter

`[/par/]log/rot/count`

Description: Reports the number of files remaining before AFRM mode turns off.

Access: read-only

Type: integer

Values: [0...$2^{31}-1$]; expressed in seconds

Default: 0

Usage Guidelines: When starting AFRM, `/par/log/rot/count` is automatically set equal to `/par/log/rot/sc/count`. If you reset `/par/log/rot/sc/count` while AFRM is on, the receiver will immediately update `/par/log/rot/count` to the same new count.

Command Examples: Check the number of files remaining.
```
print,/par/log/rot/count
```
Or
```
print,log/rot/count
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Automatically remove old files

`[/par/]log/rot/rmold`

Description: Automatically removes the earliest log file when no memory remains.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – The receiver will not erase the file with the earliest time/date.
- `on` – The receiver will erase the file with the earliest time/date.

Default: `off`

Usage Guidelines: N/A

Command Examples: Remove the earliest log file.

```
set,/par/log/rot/rmold,on
```

Or

```
set,log/rot/rmold,on
```

Check automatic file deletion.

```
print,/par/log/rot/rmold
```

Or

```
print,log/rot/rmold
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Force file rotation event

`[/par/]log/rot/force`

Description: Forces the receiver to execute file rotation immediately.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – ignored.
- `on` – The receiver will erase the file with the earliest time/date.

Default: `off`

Usage Guidelines:

- AFRM must be turned on.
- Setting this parameter to `off` while AFRM is on will have no effect.
- If AFRM is off, setting this parameter to either `on` or `off` will also do nothing. Whenever this parameter is queried, the response will be `off`.

Command Examples: Execute forced file rotation.

```
set,/par/log/rot/force,on
```

Or

```
set,log/rot/force,on
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Two Concurrent Log files

The *Two Concurrent Log files* feature allows you to handle two log files concurrently. Hereinafter these two files will be referred to as *first concurrent log file* (a) and *second concurrent log file* (b).

The following receiver parameters were normally used to control the (single) current log file:

/cur/log	/par/out/cur/log&count	/par/out/epochs/cur/log
/cur/log&size	/par/out/elm/cur/log	/par/log/sc/period
/par/out/cur/log	/par/out/minsvs/cur/log	/par/cmd/create/prefix

The above listed parameters are associated with the first concurrent log file. For better readability, these parameters have the aliases shown in the right-hand column of the table below.

Table 3-1. Mapping between old and new parameters

Parameter Name for Single File	Parameter Name (alias) for First Concurrent File
/cur/log	/cur/file/a
/cur/log&size	/cur/file/a&size
/par/out/cur/log	/par/out/cur/file/a
/par/out/cur/log&count	/par/out/cur/file/a&count
/par/out/elm/cur/log	/par/out/elm/cur/file/a
/par/out/minsvs/cur/log	/par/out/minsvs/cur/file/a
/par/out/epochs/cur/log	/par/out/epochs/cur/file/a
/par/log/sc/period	/par/log/a/sc/period
/par/cmd/create/prefix	/par/cmd/create/pre/a

Similar receiver parameters are available to handle the second concurrent log file:

/cur/file/b	/par/out/cur/file/b&count	/par/out/epochs/cur/file/b
/cur/file/b&size	/par/out/elm/cur/file/b	/par/log/b/sc/period
/par/out/cur/file/b	/par/out/minsvs/cur/file/b	/par/cmd/create/pre/b

Configuring AFRM and MINTER to control two log files

`[/par/]log/imp`

Description: Specifies whether only the first concurrent log file, or only the second concurrent log file, or both will be controllable via AFRM and MINTER.

Access: read & write

Type: array of two boolean values

Values: `{n | y,n | y}`

- `n` – the corresponding concurrent log file will be controllable through AFRM and MINTER.
- `y` – the corresponding log file cannot be created or closed via MINTER or AFRM.

Default: `{y,n}`

Usage Guidelines: N/A

Command Examples: Enable the receiver to work with two log files simultaneously. Once you have sent this command to the receiver, press and hold down the FN button for at least one second (but less than five seconds) to start recording raw data in both concurrent log files.

```
set,/par/log/imp,{y,y}
```

Or

```
set,log/imp,{y,y}
```

Check the value of parameter.

```
print,/par/log/imp
```

Or

```
print,log/imp
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Configuring AFRM and MINTER to control a specific concurrent log file

`[/par/]log/imp/<N>`

Description: Specifies whether only the first concurrent log file, or only the second concurrent log file will be controllable via AFRM and MINTER.

Access: read & write

Type: boolean

Values: `n | y`

- `n` – the corresponding concurrent log file will be controllable through AFRM and MINTER.
- `y` – the corresponding log file cannot be created or closed via MINTER or AFRM.

Default: `y` for N = 0
`n` for N = 1

Usage Guidelines: `N` can be 0 or 1. Zero stands for the log file specified by the parameter `/cur/file/a`. Unity stands for the log file specified by the parameter `/cur/file/b`.

Command Examples: Enable the receiver to work with the first concurrent log file. Once you have sent this command to the receiver, press and hold down the FN button for at least one second (but less than five seconds) to start recording raw data in the concurrent log file a.

```
set,/par/log/imp/0,y
```

Or

```
set,log/imp/0,y
```

Check the value of parameter.

```
print,/par/log/imp/0
```

Or

```
print,log/imp/0
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Frequency Source

Certain models of TPS receivers are capable of connecting and using an external frequency source as a reference frequency replacing an internal quartz oscillator.

This section describes parameters for configuring and monitoring the reference frequency; either a high-stability external frequency or the receiver's internal quartz oscillator.

Reference frequency source

`[/par/]frq/input`

Description: Specifies the reference frequency input that the receiver will use.

Access: read & write

Type: enumerated

Values: `int|ext|off`

- `int` – the receiver uses the internal quartz oscillator
- `ext` – the receiver uses an external frequency
- `off` – the receiver uses the internal quartz oscillator; external frequency detection is completely disabled

Default:
- `int` – for the Odyssey-RS, Legacy-E receivers and the E_GGDT_x, E_GGD boards.
- `off` – for the Net-G3 receiver and the EG3_x board.

Usage Guidelines:
- Before connecting an external frequency source, set this parameter to `off`, connect the external frequency source and set this parameter to `ext`.
- Switching from `int` to `ext` and vise versa may result in temporary loss of lock to satellites.
- For external frequency, the option _FRI must be enabled.

Command Examples: Set the source of reference frequency to `ext`.

```
set,/par/frq/input,ext
```

Or

```
set,frq/input,ext
```

Receiver Availability: This parameter may only be used with the receivers that have an external frequency connector and the enabled option _FRI.

History:
- This parameter was introduced in the first release of firmware.
- The `off` value was implemented in firmware 3.1.

See Also: N/A

External frequency value

`[/par/]frq/ext`

Description: Specifies the reference frequency supplied by the connected external frequency source.

Access: read & write

Type: integer

Values: `[5, 10...40]`; expressed in MHz

Default: `10`

Usage Guidelines: Ensure that you set the correct value or the receiver will not lock onto satellites.

Command Examples: Set the external frequency to 20 MHz.

```
set,/par/frq/ext,20
```

Or

```
set,frq/frq/ext,20
```

Query the external frequency value.

```
print,/par/frq/ext
```

Or

```
print,frq/frq/ext
```

Receiver Availability: This parameter may only be used with the receivers that have an external frequency connector and the enabled option _FRI.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

External frequency status

`[/par/]frq/stat`

Description: Shows whether or not the external frequency is being used.

Access: read-only

Type: enumerated

Values: `off | wait | locked`

- `off` – the receiver is using the internal quartz oscillator.
- `wait` – the receiver is waiting for the external frequency lock because the external frequency source is disconnected, the amplitude of input signal is too low, or the actual external frequency is different from that specified in `/par/frq/ext`.
- `locked` – the receiver is using the external frequency.

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the external frequency status.

```
print,/par/frq/stat
```

Or

```
print,frq/frq/stat
```

Receiver Availability: This parameter may only be used with the receivers that have an external frequency connector and the enabled option _FRI.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Amplitude of external frequency signal

`[/par/]frq/amp`

Description: Shows an estimate of the external frequency signal amplitude.

Access: read-only

Type: enumerated

Values: `off | low | ok`

- `off` – the receiver is using the internal quartz oscillator and amplitude estimate is unavailable.
- `low` – the external frequency signal's amplitude is lower than needed.
- `ok` – the external frequency signal's amplitude meets the specs.

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the amplitude.

```
print,/par/frq/amp
```

Or

```
print,frq/frq/amp
```

Receiver Availability: This parameter may only be used with the receivers that have an external frequency connector and the enabled option _FRI.

History: This parameter was introduced in the first release of firmware.

See Also: For voltage levels associated with each value, refer to technical specifications for the corresponding receiver or board.

Eurocard J502 connector I/O pin support

Configure pins for input/output

`[/par/]dev/gpio/out`

Description: Specifies which pins are configures for input and output.

Access: read & write

Type: list of eight boolean values

Values: `{[y|n],[y|n],[y|n],[y|n],[y|n],[y|n],[y|n],[y|n]}`

- `y` – indicates that the corresponding pin is configured for output.
- `n` – indicates that the corresponding pin is configured for input.

Default: `{y,y,y,y,y,y,y,y}`

Usage Guidelines: The first element in the list corresponds to pin# 0, the last element corresponds to pin# 7.
After you send the `print` command, the receiver will return a list of booleans, e.g., {y,y,n,n,y,n,y,y}.
You can configure all eight pins at one time by setting this parameter to an appropriate integer value. Your receiver will interpret such an integer as a binary code whose least significant bits (LSB) are mapped onto pins## 0 thru 7 (the least significant bit corresponds to pin#0). Bits other than the eight LSB are ignored. For example, setting this parameter to 0x3D (hexadecimal) is fully equivalent to setting it to the list {y,y,n,n,y,n,y,y}.
Note that you can also configure separate pins independently from each other. To configure a separate pin, use the corresponding one-bit parameter: `/par/dev/gpio/out/<m>`, where `m` stands for the pin number (0 thru 7). For example, the command `set,/par/dev/gpio/out/3,n` will configure pin# 3 for input.

`[/par/]dev/gpio/out` *(Continued)*

Command Examples: Set all pins for output.
`set,/par/dev/gpio/out,{y,y,y,y,y,y,y,y}`
Or
`set,dev/gpio/out,{y,y,y,y,y,y,y,y}`

More examples.
`set,/par/dev/gpio/out/0,y` – configure pin#0 for output.
`set,/par/dev/gpio/out/7,n` – configure pin#7 for input
`set,/par/dev/gpio/out,0xff` – configure all pins for output.
`set,/par/dev/gpio/out,0` – configure all pins for input.
`set,/par/dev/gpio/out,0xf` – configure pins ## 0 thru 3 for output and pins ## 4 thru 7 for input.

Query the current values.
`print,/par/dev/gpio/out`
Or
`print,dev/gpio/out`

Receiver Availability: This parameter is available for the Eurocard board only.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Configure pin logical levels

`[/par/]dev/gpio/value`

Description: Specifies the logical levels for pins.

Access: read & write

Type: list of eight boolean values

Values: `{[y|n],[y|n],[y|n],[y|n],[y|n],[y|n],[y|n],[y|n]}`

- `y` – indicates that the corresponding pin is set to logical level 1.
- `n` – indicates that the corresponding pin is set to logical level 0.

Default: `{n,n,n,n,n,n,n,n}`

Usage Guidelines: You can handle this parameter in the same way you do `/par/dev/gpio/out`, i.e., you can use either a list of booleans or an integer to specify it. In the latter case, only the eight LSB of the binary code are taken into consideration. Also, you can set the required logical signal level for any individual pin.
When setting this parameter to a specific value (a list of booleans or an integer), you specify individual logical levels only for the pins configured for output. For the pins configured for input, their logical levels remain unchanged.
The logical levels of input pins will depend on the actual voltage applied to them. When you query the pin logical values with the corresponding `print` command, the receiver will report the true logical levels for both input and output pins.

Command Examples: Set all pins to logical level 1.
`set,/par/dev/gpio/value,{y,y,y,y,y,y,y,y}`
Or
`set,dev/gpio/value,{y,y,y,y,y,y,y,y}`

More examples.
`set,/par/dev/gpio/value/0,y` – set pin#0 to logical level "1"
`set,/par/dev/gpio/value,0xff` – set all of the output pins to logical signal "1"

Query the current values.
`print,/par/dev/gpio/value`
Or
`print,dev/gpio/value`

Receiver Availability: This parameter is available for the Eurocard board only.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Advanced Input Mode (AIM)

Hereinafter in this section `port` denotes an input port that can function in `jps` mode (e.g., `dev/ser/a`, `cur/term`, etc).

The *advanced input mode* allows the receiver to decode the incoming >> messages and non-standard messages and to pass the data decoded from these messages to a specified decoder (e.g., commands interpreter) or to send the decoded data to a specified output port.



Recall that the primary purpose of the jps input mode is to receive, decode, and apply the TPS proprietary messages.

For information on the >> message, refer to page 4-126. For information on non-standard messages, refer to page 1-15.

To provide backward compatibility with the earlier firmware versions, advanced input mode is turned off by default. In this case the >> messages and non-standard messages are ignored.

For every `port`, there is a set of `/par/<port>/jps` parameters through which AIM for the selected `port` can be controlled. Each parameter has a value which is represented by an array of three (3) elements[3]. In turn, each element of the array is a specification that consists of the following four (4) fields:

- mode
- id
- skip
- port

These fields are used in AIM's *compare-execute process*. This process includes two stages: comparison or matching and execution.

Consider the compare-execute process step-by-step.

Matching stage. On this stage, the receiver compares the >> or non-standard message with the `mode` and `id` fields.

1. Comparison of the mode field:
   - If the `mode` field has the `none` value, the specification will never match.
   - If the `mode` field has one of the following values:

     –`cmd`, `echo`, `jps`, `rtcm`, `rtcm3`, `cmr`;

     the specification could match the >> message (depending on the value of `id` field) but never matches a non-standard message.

3. The number of elements is somewhat arbitrary and can be expanded in the future (if required).

- If the mode field has the `nscmd` or `nsecho` value, the specification could match a non-standard message (depending on `id`) but never matches the >> message.

2. Comparison of the `id` field:

- Once the value in the `mode` field matches the >> or non-standard message, the receiver will compare the value in `id` field with the first byte of the message in question.

  –For the >> message, the first byte is the `port` field.

  –For the non-standard message, the first byte is the message first character lying between "!" and "/" in ASCII (inclusive).

- The message matches the specification in two cases

  –If the message first byte is equal to the value of `id` field.

  –If the `id` field of the specification has a special value `-1`. This value matches any identifier byte of the message.

Execution stage. The execution stage activates only on condition that the received message matches the `mode` and `id` fields of the matching stage. On this stage, the receiver executes the contents of the message according to the `mode`, `skip`, and `port` fields.



The execution stage activates only on condition that the received message matches the mode and id fields of the matching stage.

- If the `skip` field is set to `y`, the first byte of the message is skipped before executing the message contents. Otherwise the first byte is considered as the part of the message contents.
- After optional skipping of the first byte, the message contents execute according to the value of the `mode` field in the matched specification.
- For the modes `jps`, `rtcm`, `rtcm3`, and `cmr`, the receiver will pass the contents to the corresponding type of decoder as if the message has been received through the port specified in the `port` field of the specification. The input mode of the corresponding port (`/par/<port>/imode`) should be set accordingly for execution to actually take place. The empty value of the `port` field denotes the current port (i.e., the port the initial message is received through).



The empty value of the port field for jps mode is meaningless (as well as explicit setting the port parameter to the port matching those the parameter is being set for) as the contents would pass to the same decoder the initial message came from. This, in turn, would break the decoder logic. However the TPS receiver allows you to set such meaningless combination of parameters, but any message matching it will be ignored.

- For the modes `cmd` and `nscmd`, the receiver will pass the contents to the command interpreter as if the command has been received through the port `port`. The empty value of `port` denotes the current port.
- For the modes `echo` and `nsecho`, the receiver will either send the contents to the output port specified in the `port` field of the matched specification or, if the `port` field of the specification is empty, pass the contents to the command interpreter.

AIM Parameters

This section describes parameters used for configuring AIM in the receiver.

Input specifications

`[/par/]<port>/jps`

Description: Reports the input specifications for the selected port.

Access: read-only

Type: array [0...2] of input specification

Values: input specification described in the parameters that follow

Default: `{{none,-1,y,""},{none,-1,y,""},{none,-1,y,""}}`

Usage Guidelines: N/A

Command Examples: Query the input specification for serial port A.
`print,/par/dev/ser/a/jps`
Or
`print,dev/ser/a/jps`

Receiver Availability: This parameter is available for all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Input specifications of a particular element

`[/par/]<port>/jps/<N>`

Description: Defines the input specifications of a particular element for the selected port.

Access: read & write

Type: list of four values

Values: described in the parameters that follow

Default: `{none,-1,y,""}`

Usage Guidelines: `N` ranges between 0, 1, and 2.

Command Examples: Set the specification 1 for serial port A.

```
set,/par/dev/ser/a/jps/1,{cmr,-1,n,/dev/ser/c}
```

Or

```
set,dev/ser/a/jps/1,{cmr,-1,n,/dev/ser/c}
```

Query the input specification 1 for serial port A.

```
print,/par/dev/ser/a/jps/1
```

Or

```
print,dev/ser/a/jps/1
```

Receiver Availability: This parameter is available for all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Operation mode

`[/par/]<port>/jps/<N>/mode`

Description: Defines the mode according to which the contents of the matching messages should execute.

Access: read & write

Type: enumerated

Values: `none | cmd | echo | jps | rtcm | rtcm3 | cmr | nscmd | nsecho`

Default: `none`

Usage Guidelines: `N` ranges between 0, 1, and 2.
In the matching stage, the values have the following meaning:

- none – the specification will never match any message
- cmd, echo, jps, rtcm, rtcm3, cmr – the specification may possibly match the >> message. The matching criterion is the product of comparison between the port field of the >> message and the id field of the specification.
- nscmd, nsecho – the specification may possibly match a non-standard message. The matching criterion is the product of comparison between the first byte of non-standard message and the id field of the specification.

In the execution stage, the values have the following meaning:

- none – never appears in the execution stage
- cmd – execute the message contents as command(s) received from the port specified in the port field
- echo – send the message contents to the port specified in the port field
- jps – process the message contents as data in TPS format received from the port specified in the port field
- rtcm – process the message contents as data in RTCM format received from the port specified in the port field
- rtcm3 – process the message contents as data in RTCM 3.0 format received from the port specified in the port field
- cmr – process the message contents as data in CMR format received from the port specified in the port field
- nscmd – the same as cmd
- nsecho – the same as echo

Command Examples: Set the mode to cmd of input specification 1 for serial port A.
`set,/par/dev/ser/a/jps/1/mode,cmd`
Or
`set,dev/ser/a/jps/1/mode,cmd`

Query the mode of input specification 1 for serial port A.
`print,/par/dev/ser/a/jps/1/mode`
Or
`print,dev/ser/a/jps/1/mode`

Receiver Availability: This parameter is available for all receivers.

`[/par/]<port>/jps/<N>/mode` *(Continued)*

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Message Identifier

`[/par/]<port>/jps/<N>/id`

Description: Defines the value of `id` field in the specification.

Access: read & write

Type: integer

Values: `[-1...255]`

Default: `-1`

Usage Guidelines: `N` ranges between 0, 1, and 2.

Command Examples: Set the message identifier to 5.

```
set,/par/dev/ser/a/jps/1/id,5
```

Or

```
set,dev/ser/a/jps/1/id,5
```

Query the message identifier of the input specification 1 for serial port A.

```
print,/par/dev/ser/a/jps/1/id
```

Or

```
print,dev/ser/a/jps/1/id
```

Receiver Availability: This parameter is available for all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to “Matching stage” on page 3-75 to find out when and how this field is used.

Value of skip field

`[/par/]<port>/jps/<N>/skip`

Description: Defines the value of `skip` field in the specification.

Access: read & write

Type: boolean

Values: `n | y`

- `n` – the first byte is considered as the part of the message contents and therefore is not skipped.
- `y` – the first byte of the message is skipped before executing the message contents.

Default: `y`

Usage Guidelines: `N` ranges between 0, 1, and 2.

Command Examples: Set the `skip` field to n.

```
set,/par/dev/ser/a/jps/1/skip,n
```

Or

```
set,dev/ser/a/jps/1/skip,n
```

Query the `skip` field.

```
print,/par/dev/ser/a/jps/1/skip
```

Or

```
print,dev/ser/a/jps/1/skip
```

Receiver Availability: This parameter is available for all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Using the port as a source or destination

`[/par/]<port>/jps/<N>/port`

Description: Defines whether the port will be a source or destination on execution stage.

Access: read & write

Type: enumerated

Values: `/port | ""`

Default: `""`

Usage Guidelines:

- `N` ranges between 0, 1, and 2.
- If the value of the `mode` field is set to `echo` or `nsecho` (see the parameter `/par/<port>/jps/N/mode`), this parameter specifies the output port to which the message contents should be sent.
- For all other values of the `mode` field, this parameter specifies the input port so that the message contents execute as if the data have been received through the specified port.
- The default value `""` (empty string) designates the current port, i.e., the port through which the initial message has been received.

Command Examples:

Set the port.
```
set,/par/dev/ser/a/jps/1/port,/dev/ser/c
```
Or
```
set,dev/ser/a/jps/1/port,/dev/ser/c
```

Query the port.
```
print,/par/dev/ser/a/jps/1/port
```
Or
```
print,dev/ser/a/jps/1/port
```

Receiver Availability: This parameter is available for all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Cinderella Mode

Cinderella sets all receiver options to their maximum allowable values for 24 hours every other Tuesday at GPS midnight.

Configuring Cinderella

`[/par/]opts/cind`

Description: Activates and deactivates the Cinderella mode.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – Cinderella is deactivated.
- `on` – Cinderella is activated.

Default: `off`

Usage Guidelines:

- This mode requires the enabled Cinderella option in the OAF.
- You can send the command `set,/par/opts/cind,on` anytime between the previous and the coming Cinderella Tuesday or during the Cinderella Tuesday.
- You will need to restart the receiver after sending the command to make this mode activated.
- The receiver will automatically set the parameter to `off` at midnight Cinderella Tuesday/Wednesday.
- Time means GPS time.

Command Examples:

Activate Cinderella.
`set,/par/opts/cind,on`
Or
`set,opts/cind,on`

Check Cinderella status.
`print,/par/opts/cind`
Or
`print,opts/cind`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Antenna Parameters

Antenna input mode

`[/par/]ant/rcv/inp`

Description: Specifies the antenna (whether internal or external) the receiver will use.

Access: read & write

Type: enumerated

Values: `int | ext | auto`

- `int` – the internal antenna used.
- `ext` – an external antenna used.
- `auto` – antenna selected automatically.

Default: receiver-dependant

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use an external antenna.

```
set,/par/ant/rcv/inp,ext
```

Or

```
set,ant/rcv/inp,ext
```

Check the parameter value.

```
print,/par/ant/rcv/inp
```

Or

```
print,ant/rcv/inp
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in firmware 3.2.
- Firmware versions 3.1p3 and older use the parameter `/par/ant/inp`.

See Also: N/A

Current antenna input

`[/par/]ant/rcv/curinp`

Description: Reports the actual antenna the receiver is using.

Access: read-only

Type: enumerated

Values: `int | ext`

- `int` – the internal antenna used.
- `ext` – an external antenna used.

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Check the current antenna being used.

```
print,/par/ant/rcv/curinp
```

Or

```
print,ant/rcv/curinp
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in firmware 3.2.
- Firmware versions 3.1p3 and older use the parameter `/par/ant/curinp`.

See Also: N/A

OmniSTAR antenna input mode

`[/par/]ant/omni/inp`

Description: Specifies the OmniSTAR antenna (whether internal or external) the receiver will use.

Access: read & write

Type: enumerated

Values: `int | ext`

- `int` – the internal antenna used.
- `ext` – an external antenna used.

Default: `int`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use an external antenna.

```
set,/par/ant/omni/inp,ext
```

Or

```
set,ant/omni/inp,ext
```

Check the parameter value.

```
print,/par/ant/omni/inp
```

Or

```
print,ant/omni/inp
```

Receiver Availability: This parameter is reserved for future versions of firmware.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Communication Link Parameters

TPS receivers provide a rich set of communication capabilities that allow versatility, reliability, and efficiency at the jobsite. Among common RF solutions operating at 400-470 MHz and 915 MHz, the receivers might use internal and external GSM/GPRS/EDGE and CDMA modules.

This section describes the parameters used for:

- Configuring GSM modules for a conventional single-base RTK.
- Configuring GPRS/EDGE and CDMA modules for real-time network services.
- Monitoring the connection status.

Modem connection mode

`[/par/]modem/<X>/mode`

Description: Specifies the mode to be used by the receiver's communication module for establishing a connection with a communication module or service at the other end of the data link.

Access: read & write

Type: enumerated

Values: `off | master | slave | gprs`

- `off` – Communication control program is off. The data link is disconnected.
- `master` – the receiver's GSM modem will try to dial in to the slave modem (a base GSM modem) using the phone number specified by the parameter `/par/mode/<X>/dial`.
- `slave` – the receiver's GSM modem will wait for incoming calls from a master modem (a rover GSM modem).
- `gprs` – the receiver's GSM modem will transfer data in GPRS/EDGE mode.
- `dialup` – an external modem connected to the receiver will use a dial-up connection.
- `cdma` – the receiver's modem will transfer data in CDMA mode.

Default: `off`

Usage Guidelines:

- TPS receivers support up to communication modules, called `a`, `b`, `c`, and `d`. The internal module is hardware-wise connected to port C in TPS receivers. As for external modules, it is common practice to connect them to the receiver's port D if available.
- `X` designates any one of the letters `a`, `b`, `c`, `d`.

`[/par/]modem/<X>/mode` *(Continued)*

Command Examples: Set the internal GSM modem on port C to master.
`set,/par/modem/c/mode,master`
Or
`set,modem/c/mode,master`

Confirm the connection mode on port C.
`print,/par/modem/c/mode`
Or
`print,modem/c/mode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History:
- This parameter was introduced in firmware 2.4.
- The `gprs` and `dialup` values were implemented in firmware 2.6.
- The `cdma` value was implemented in firmware 3.1p3.

See Also: N/A

Receiver's port GSM modem is connected with

`[/par/]modem/<X>/port`

Description: Reports the name of a serial communications port the GSM modem is connected with.

Access: read-only

Type: string

Values: `dev/ser/<X>`

Default: N/A

Usage Guidelines: `X` designates any one of the letters `a`, `b`, `c`, `d`.

Command Examples: Confirm the port.
`print,/par/modem/c/mode`
Or
`print,modem/c/mode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

PIN

`[/par/]modem/<X>/pin`

Description: Specifies a numeric password for access to the services on the GSM/GPRS/EDGE modem's SIM card.

Access: read & write

Type: string

Values: four decimal digits

Default: `"0000"`

Usage Guidelines: `X` designates any one of the letters `a`, `b`, `c`, `d`.

Command Examples: Set the PIN to 1234.

```
set,/par/modem/c/pin,1234
```

Or

```
set,modem/c/pin,1234
```

Check the PIN.

```
print,/par/modem/c/pin
```

Or

```
print,modem/c/pin
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Dial number

`[/par/]modem/<X>/dial`

Description:	Specifies the phone number of the base GSM modem.
Access:	read & write
Type:	string
Values:	up to 32 decimal digits
Default:	`""`
Usage Guidelines:	`X` designates any one of the letters `a`, `b`, `c`, `d`.
Command Examples:	Set the phone number to 9169357653. `set,/par/modem/c/dial,9169357653` Or `set,modem/c/dial,9169357653` Check the phone number. `print,/par/modem/c/dial` Or `print,modem/c/dial`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

Monitoring modem status

`[/par/]modem/<X>/state`

Description: Reports the status of GSM modem.

Access: read-only

Type: enumerated

Values: `off` | `detect` | `detected` | `registration` | `ready` | `ring` | `dialing` | `connect` | `discon` | `err`

- `off` – Modem control is turned off.
- `detect` – Searching for a modem on the corresponding port.
- `detected` – A modem has been detected. Modem initialization is in progress.
- `registration` – The modem is being registered in the network.
- `ready` – The modem has been initialized and registered in the GSM network. If the modem is in slave mode, it is ready to receive an incoming call. If the modem is in master mode, it is ready to dial in to the slave modem.
- `ring` – An incoming call is being received (in slave mode only).
- `dialing` – The modem is dialing the selected phone number as specified by the parameter `/par/modem/<X>/dial` (in master mode only).
- `connect` – A connection has been established.
- `discon` – The connection has been broken (disconnecting).
- `err` – A fatal error has occurred. In this case, the user will need to change the parameter `/par/modem/<X>/mode` to `off`, fix the problem and then retry setting the required modem mode. See the parameter `/par/modem/<X>/err` for what specifically might have caused the error.

Default: `off`

Usage Guidelines: `X` designates any one of the letters `a`, `b`, `c`, `d`.

Command Examples: Check the GSM status.

```
print,/par/modem/c/state
```

Or

```
print,modem/c/state
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Last detected modem error

[/par/]modem/<X>/err

Description: Reports the last of the errors identified by the GSM modem driver.

Access: read-only

Type: enumerated

Values: none | detect | init | pin | net | busy | no_carrier

- none – No errors have been detected.
- detect – Cannot find a modem on the port.
- init – An initialization error has occurred.
- pin – Wrong PIN.
- net – A network error has occurred.
- busy – The phone number is busy. To rectify this temporary problem, just call again at a later time.
- no_carrier – Cannot detect the carrier signal. This temporary error can occur if the second modem (at the other end of the radio link) has not been initialized or if there are some problems with the GSM network. The given GSM modem will continue to dial in to the modem on the other side of the radio link until the carrier signal is detected.

Default: none

Usage Guidelines:

- X designates any one of the letters a, b, c, d.
- The modem control will attempt to automatically fix a detected error in case the parameter /par/modem/<X>/state takes a value other than err or off). No user intervention is needed unless the parameter /par/modem/<X>/state turns out equal to err, which means that the modem control has not been able to fix the problem on its own.

Command Examples: Check the last detected error.

```
print,/par/modem/c/err
```

Or

```
print,modem/c/err
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Wait on data from modem timeout

`[/par/]modem/<X>/rcvtimeout`

Description: Specifies how long the GSM modem will wait for data.

Access: read & write

Type: integer

Values: `[0...100]`; expressed in seconds
`0` – the timeout will not be used in data transmission

Default: `5`

Usage Guidelines:
- `X` designates any one of the letters `a`, `b`, `c`, `d`.
- If the receiver has not received any data from the GSM modem for `rcvtimeout` seconds since the modem state changed to `connect`, the modem will be disconnected and then reinitialized.
- The parameter is used to control the connection established between the pair of GSM modems (master – slave).

Command Examples: Set the timeout to 6 seconds.
```
set,/par/modem/c/rcvtimeout,6
```
Or
```
set,modem/c/rcvtimeout,6
```

Check the timeout.
```
print,/par/modem/c/rcvtimeout
```
Or
```
print,modem/c/rcvtimeout
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: The `/par/mode/<X>/sndtime` parameter on page 3-94.

Service word repeat period

`[/par/]modem/<X>/sndtime`

Description: Specifies a time interval to ensure reliable communication between the pair of GSM modems (master – slave) and avoid unnecessary modem reinitialization.

Access: read & write

Type: integer

Values: `[0...1000]`; expressed in seconds
`0` – the service word will not be used in data transmission.

Default: `2`

Usage Guidelines:
- `X` designates any one of the letters `a`, `b`, `c`, `d`.
- The transmit modem will send the service word to the receive modem in every `sndtime` seconds. Note that the service word will not affect the differential corrections (RTCM, CMR, etc., messages) in any way.
- To ensure reliable and secure modem communication, the parameter `/par/modem/X/sndtime` must be larger then the period of transmitting differential corrections. Also, care should be taken that the timeout in the parameter `/par/modem/X/rcvtimeout` is greater than the service word period by 2 to 3 seconds.

Command Examples: Set the service word period to 3 seconds.
`set,/par/modem/c/sndtime,3`
Or
`set,modem/c/sndtime,3`

Check the service word period.
`print,/par/modem/c/sndtime`
Or
`print,modem/c/sndtime`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

GSM module

`[/par/]modem/<X>/model`

Description: Specifies the model of GSM module.

Access: read & write

Type: enumerated

Values: `generic | tps | satel | arwest`

- `generic` – an unknown GSM module.
- `tps` – the GSM module embedded on the Topcon modem board.
- `satel` – the GSM module embedded on the Satel modem board.
- `arwest` – the GSM module embedded on the ArWest modem board.

Default: `generic`

Usage Guidelines: `X` designates any one of the letters `a`, `b`, `c`, `d`.

Command Examples: Set the GSM module to tps.

```
set,/par/modem/c/model,tps
```

Or

```
set,modem/c/model,tps
```

Check the service word period.

```
print,/par/modem/c/model
```

Or

```
print,modem/c/model
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

GSM debug

`[/par/]modem/debug`

Description: Enables the debug mode for GSM.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the debug mode is enabled.
- `on` – the debug mode is disabled.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable the debug mode.

```
set,/par/modem/debug,on
```

Or

```
set,modem/debug,on
```

Check the service word period.

```
print,/par/modem/debug
```

Or

```
print,modem/debug
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of PPP parameters

`[/par/]ppp`

Description:	Reports a complete list of PPP parameters.
Access:	read-only
Type:	list
Values:	See parameters that follow for values.
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Output PPP parameters. `print,/par/ppp` Or `print,ppp`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

GSM/GPRS modem type used in PPP connection

`[/par/]ppp/modem`

Description: Specifies the GSM/GPRS modem type used in PPP connection.

Access: read & write

Type: enumerated

Values: `wavecom | tps | satel | arwest`

- `wavecom` – Wavecom modems.
- `tps` – FH/UHF TPS modems.
- `satel` – Satel modems.
- `arwest` – ArWest modems.

Default: `wavecom`

Usage Guidelines: N/A

Command Examples: Set the mode type to tps.

```
set,/par/ppp/modem,tps
```

Or

```
set,ppp/modem,tps
```

Check the modem type.

```
print,/par/ppp/modem
```

Or

```
print,ppp/modem
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.1p2.

See Also: N/A

PPP connection status

`[/par/]ppp/state`

Description:	Reports PPP connection status.
Access:	read-only
Type:	enumerated
Values:	`off` | `connecting` | `connected` | `disconnected`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Check PPP status. `print,/par/ppp/state` Or `print,ppp/state`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

PPP connection speed

`[/par/]ppp/speed`

Description: Specifies PPP connection speed.

Access: read & write

Type: enumerated

Values: `9600 | 19200 | 38400 | 57600 | 115200;` expressed in baud rate

Default: `9600`

Usage Guidelines: If you change this parameter before setting the modem mode to `gprs`, `cdma` or `dialup`, the firmware starts searching from the specified baud rate. It speeds up the modem search if the real baud rate is known.

Command Examples: Set the baud rate to 38400.

```
set,/par/ppp/speed,38400
```

Or

```
set,ppp/speed,38400
```

Check the baud rate.

```
print,/par/ppp/speed
```

Or

```
print,ppp/speed
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History:
- This parameter was introduced in firmware 2.6 as read-only.
- A write capability was introduced in firmware 3.1p3.

See Also: N/A

IP address of PPP peer

`[/par/]ppp/addr`

Description: Specifies IP address of a peer of the PPP connection.

Access: read & write

Type: string

Values: any valid IP address

Default: `0.0.0.0`

Usage Guidelines: If set to default, the peer will have to supply the local IP address during IPCP negotiations. When connected, the actual local IP address is accessible through the `/par/net/stat/if/ppp0/addr` parameter.

Command Examples: Set IP address to 213.87.12.147.
`set,/par/ppp/addr,213.87.12.147`
Or
`set,ppp/addr,213.87.12.147`

Check the IP address.
`print,/par/ppp/addr`
Or
`print,ppp/addr`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Debug PPP

`[/par/]ppp/debug`

Description: Enables output of debug PPP connection.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable debug.
`set,/par/ppp/debug,on`
Or
`set,ppp/debug,on`

Check the debug status.
`print,/par/ppp/debug`
Or
`print,ppp/debug`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Authentication parameters

`[/par/]ppp/auth`

Description: Reports authentication parameters for PPP connection.

Access: read-only

Type: list of two boolean values

Values: `{pap,chap}`, where `pap` and `chap` can be `off` or `on`.

Default: `{on,on}`

Usage Guidelines: N/A

Command Examples: Check the authentication.
`print,/par/ppp/auth`
Or
`print,ppp/auth`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Usage of unencrypted password (PAP)

`[/par/]ppp/auth/pap`

Description:	Enables PAP.
Access:	read & write
Type:	boolean
Values:	`off	on`
Default:	`on`
Usage Guidelines:	N/A
Command Examples:	Disable PAP. `set,/par/ppp/auth/pap,off` Or `set,ppp/auth/pap,off` Check PAP. `print,/par/ppp/auth/pap` Or `print,ppp/auth/pap`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6
See Also:	N/A

Usage of challenge handshake authentication protocol (CHAP)

`[/par/]ppp/auth/chap`

Description: Enables CHAP.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Disable CHAP.
`set,/par/ppp/auth/chap,off`
Or
`set,ppp/auth/chap,off`

Check CHAP.
`print,/par/ppp/auth/chap`
Or
`print,ppp/auth/chap`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

PPP compression parameters

`[/par/]ppp/comp`

Description: Reports PPP compression parameters.

Access: read-only

Type: boolean

Values: list of two boolean values

Default: `{vj,vjc}`, where `vj` and `vjc` can be `off` or `on`.

Usage Guidelines: N/A

Command Examples: Check compression parameters.
`print,/par/ppp/comp`
Or
`print,ppp/comp`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Van Jacobson compression

`[/par/]ppp/comp/vj`

Description: Enables Van Jacobson style TCP/IP header compression in both the transmit and receive directions.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable Van Jacobson compression.
`set,/par/ppp/comp/vj,on`
Or
`set,ppp/comp/vj,on`

Check VJ compression.
`print,/par/ppp/comp/vj`
Or
`print,ppp/comp/vj`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Connection-ID compression in Van Jacobson compression

`[/par/]ppp/comp/vjc`

Description: Enables the connection-ID compression in Van Jacobson style TCP/IP header compression.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable the connection-ID compression.
`set,/par/ppp/comp/vjc,on`
Or
`set,ppp/comp/vjc,on`

Check the connection-ID compression.
`print,/par/ppp/comp/vjc`
Or
`print,ppp/comp/vjc`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

List of GPRS parameters

`[/par/]ppp/gprs`

Description: Reports a complete list of GPRS parameters.

Access: read-only

Type: list

Values: See parameters that follow for values.

Default: N/A

Usage Guidelines: N/A

Command Examples: Output GPRS parameters.
`print,/par/ppp/gprs`
Or
`print,ppp/gprs`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

GPRS dial number

`[/par/]ppp/gprs/dial`

Description: Specifies the dial number for GPRS connection.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `"*99***1#"`

Usage Guidelines: N/A

Command Examples: Set the dial number to *99#.

```
set,/par/ppp/gprs/dial,*99#
```

Or

```
set,ppp/gprs/dial,*99#
```

Check the dial number.

```
print,/par/ppp/gprs/dial
```

Or

```
print,ppp/gprs/dial
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

GPRS user name

`[/par/]ppp/gprs/user`

Description: Specifies the user name for GPRS connection.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the user name to net.

```
set,/par/ppp/gprs/user,net
```

Or

```
set,ppp/gprs/user,net
```

Check the user name.

```
print,/par/ppp/gprs/user
```

Or

```
print,ppp/gprs/user
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

GPRS password

`[/par/]ppp/gprs/passwd`

Description: Specifies the password for GPRS connection.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the password to123abc.
`set,/par/ppp/gprs/passwd,123abc`
Or
`set,ppp/gprs/passwd,123abc`

Check the user name.
`print,/par/ppp/gprs/passwd`
Or
`print,ppp/gprs/passwd`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Packet Data Protocol (PDP) context parameters

`[/par/]ppp/gprs/pdp`

Description: Reports PDP context parameters.

Access: read-only

Type: list

Values: See parameters that follow for values.

Default: `{1,""}`

Usage Guidelines: N/A

Command Examples: Check PDP parameters.
`print,/par/ppp/gprs/pdp`
Or
`print,ppp/gprs/pdp`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

PDP context identifier

`[/par/]ppp/gprs/pdp/id`

Description: Specifies PDP context identifier.

Access: read & write

Type: integer

Values: `[1...4]`

Default: `1`

Usage Guidelines: N/A

Command Examples: Set PDP context identifier to 2.

```
set,/par/ppp/gprs/pdp/id,2
```

Or

```
set,ppp/gprs/pdp/id,2
```

Check the user name.

```
print,/par/ppp/gprs/pdp/id
```

Or

```
print,ppp/gprs/pdp/id
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Access point name

`[/par/]ppp/gprs/pdp/apn`

Description: Specifies access point name.

Access: read & write

Type: string[0...64]

Values: up to 64 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Set APN to net.

```
set,/par/ppp/gprs/pdp/apn,net
```

Or

```
set,ppp/gprs/pdp/apn,net
```

Check APN.

```
print,/par/ppp/gprs/pdp/apn
```

Or

```
print,ppp/gprs/pdp/apn
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Extra Initialization string

`[/par/]ppp/gprs/extras`

Description: Specifies extra modem initialization string to let the user add other AT command.

Access: read & write

Type: string[0...32]

Values: up to 32 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Set initialization string to AT+CGQMIN=1.

```
set,/par/ppp/gprs/extras,AT+CGQMIN=1
```

Or

```
set,ppp/gprsextras,AT+CGQMIN=1
```

Check initialization string.

```
print,/par/ppp/gprs/extras
```

Or

```
print,ppp/gprs/extras
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

List of dialup parameters

`[/par/]ppp/dialup`

Description: Reports the list of dialup parameters.

Access: read-only

Type: list

Values: See parameters that follow for values.

Default: `{"","","TIMEOUT|10||+++||ATH|OK|ATI|OK|ATE0|OK|"}`

Usage Guidelines: N/A

Command Examples: Check dialup parameters.

```
print,/par/ppp/dialup
```

Or

```
print,ppp/dialup
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Dial number for accessing an Internet service provider (ISP)

`[/par/]ppp/dialup/dial`

Description: Specifies the dial number used for establishing PPP connection with ISP.

Access: read & write

Type: string[0...32]

Values: up to 32 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the dial number to 1234567890.

```
set,/par/ppp/dialup/dial,1234567890
```

Or

```
set,ppp/dialup/dial,1234567890
```

Check the dial number.

```
print,/par/ppp/dialup/dial
```

Or

```
print,ppp/dialup/dial
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

User (login)

`[/par/]ppp/dialup/user`

Description: Specifies the user name (login).

Access: read & write

Type: string[0...32]

Values: up to 32 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the user name to topnet.

```
set,/par/ppp/dialup/user,topnet
```

Or

```
set,ppp/dialup/user,topnet
```

Check the user name.

```
print,/par/ppp/dialup/user
```

Or

```
print,ppp/dialup/user
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Dialup password

`[/par/]ppp/dialup/passwd`

Description: Specifies the dialup password.

Access: read & write

Type: string[0...32]

Values: up to 32 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the password to 123abc.

```
set,/par/ppp/dialup/passwd,123abc
```

Or

```
set,ppp/dialup/passwd,123abc
```

Check the password.

```
print,/par/ppp/dialup/passwd
```

Or

```
print,ppp/dialup/passwd
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

Dialup Initialization string

`[/par/]ppp/dialup/init`

Description: Specifies the chat script string to initialize the dialup modem.

Access: read & write

Type: string[0...64]

Values: up to 64 alphanumeric character

Default: `"TIMEOUT|10||+++||ATH|OK|ATI|OK|ATE0|OK|"`

Usage Guidelines: Use the (|) character instead of cartridge return to separate the commands.

Command Examples: Set the initialization string to default.

```
set,/par/ppp/dialup/
init,TIMEOUT|10||+++||ATH|OK|ATI|OK|ATE0|OK|
```

Or

```
set,ppp/dialup/
init,TIMEOUT|10||+++||ATH|OK|ATI|OK|ATE0|OK|
```

Check the initialization string.

```
print,/par/ppp/dialup/init
```

Or

```
print,ppp/dialup/init
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History:
- This parameter was introduced in firmware 2.6
- The new command separator (@) was changed to (|) in firmware 3.1p1.

See Also: N/A

PPP errors

`[/par/]ppp/error`

Description: Reports the PPP error messages.

Access: read-only

Type: string[0...256]

Values: up to 256 alphanumeric character

Default: `""`

Usage Guidelines: N/A

Command Examples: Check the errors.
`print,/par/ppp/dialup/init`
Or
`print,ppp/dialup/init`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6

See Also: N/A

MTU

`[/par/]ppp/mtu`

Description: Specifies the MTU for PPP connection.

Access: read & write

Type: integer

Values: `[1000...1500]`; expressed in bytes

Default: `1500`

Usage Guidelines: N/A

Command Examples: Set the MTU to 1522.
`set,/par/ppp/mtu,1522`
Or
`set,ppp/mtu,1522`

Check the MTU.
`print,/par/ppp/mtu`
Or
`print,ppp/mtu`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Ping

`[/par/]ppp/ping`

Description: Specifies the number of echo messages to send to a server for checking connectivity.

Access: read & write

Type: integer

Values: `[0...300]`

Default: `10`

Usage Guidelines:
- If no messages reach the server, the receiver terminates the connection.
- For CDMA, this parameter is usually set to zero.

Command Examples: Set the number of echo messages to 20.

```
set,/par/ppp/ping,20
```

Or

```
set,ppp/ping,20
```

Check the number of echo messages.

```
print,/par/ppp/ping
```

Or

```
print,ppp/ping
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Idle timeout

`[/par/]ppp/idle`

Description: Specifies the amount of time the receiver allows am inactive connection to remain open. After this time, the receiver terminates the unused connection.

Access: read & write

Type: integer

Values: `[0...7200]`; expressed in seconds

Default: `0`

Usage Guidelines:
- You can use this command to ensure that a connection is terminated if it has been idle for the specified amount of time. This ensures that there are no hanging connections.
- Primarily intended for CDMA.

Command Examples: Set the idle timeout to 300 seconds.

```
set,/par/ppp/idle,300
```

Or

```
set,ppp/idle,300
```

Check the idle timeout.

```
print,/par/ppp/idle
```

Or

```
print,ppp/idle
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of CDMA parameters

`[/par/]ppp/cdma`

Description: Reports the list of CDMA parameters.

Access: read & write

Type: list

Values: See the parameters below.

Default: `{"#777",""}`

Usage Guidelines: N/A

Command Examples: Check the CDMA parameters.
`print,/par/ppp/cdma`
Or
`print,ppp/cdma`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.1p3.

See Also: N/A

CDMA dial number

`[/par/]ppp/cdma/dial`

Description: Specifies the dial number for a CDMA connection.

Access: read & write

Type: string [0...31]

Values: up to 32 alphanumeric characters

Default: `"#777"`

Usage Guidelines: N/A

Command Examples: Set the dial number to 89035223734.
`set,/par/ppp/cdma/dial,89035223734`
Or
`set,ppp/cdma/dial,89035223734`

Check the CDMA dial number.
`print,/par/ppp/cdma/dial`
Or
`print,ppp/cdma/dial`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.1p3.

See Also: N/A

CDMA user name

`[/par/]ppp/cdma/user`

Description: Specifies the user name for a CDMA connection.

Access: read & write

Type: string [0...31]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the user name to surveyor.

```
set,/par/ppp/cdma/user,surveyor
```

Or

```
set,ppp/cdma/user,surveyor
```

Check the user name.

```
print,/par/ppp/cdma/user
```

Or

```
print,ppp/cdma/user
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.1p3.

See Also: N/A

CDMA password

`[/par/]ppp/cdma/passwd`

Description: Specifies the password for a CDMA connection.

Access: read & write

Type: string [0...31]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the password to constellation.

```
set,/par/ppp/cdma/passwd,constellation
```

Or

```
set,ppp/cdma/passwd,constellation
```

Check the user name.

```
print,/par/ppp/cdma/passwd
```

Or

```
print,ppp/cdma/passwd
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.1p3.

See Also: N/A

Ntrip Parameters

This section describes the parameters for configuring and monitoring Ntrip service.

List of Ntrip parameters

`[/par/]ntrip`

Description:	Reports the list of Ntrip parameters.
Access:	read-only
Type:	list
Values:	See values of parameters that follow for details.
Default:	`{off,off,{0.0.0.0,0,"","",0},{/dev/ser/d,cmd},"none"}`
Usage Guidelines:	N/A
Command Examples:	Check the current Ntrip parameters. `print,/par/ntrip` Or `print,ntrip`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6
See Also:	N/A

Ntrip mode

`[/par/]ntrip/mode`

Description: Enables or disables Ntrip service.

Access: read & write

Type: boolean

Values: `off | on | server`

- `off` – disable Ntrip.
- `on` – enable Ntrip client.
- `server` – enable Ntrip server.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable Ntrip.

```
set,/par/ntrip/mode,on
```

Or

```
set,ntrip/mode,on
```

Check Ntrip.

```
print,/par/ntrip/mode
```

Or

```
print,ntrip/mode
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History:

- This parameter was introduced in firmware 2.6.
- The `server` value was added in firmware 3.3.

See Also: N/A

Ntrip Server Parameters

This section describes the parameters for configuring and monitoring Ntrip server.

List of Ntrip server parameters

`[/par/]ntrip/server`

Description:	Reports the list of Ntrip server parameters.
Access:	read-only
Type:	list
Values:	see below parameters for description
Default:	see below parameters
Usage Guidelines:	N/A
Command Examples:	View the Ntrip server parameters. `print,/par/ntrip/server` Or `print,ntrip/server`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 3.3.
See Also:	N/A

IP address Ntrip server connects to

`[/par/]ntrip/server/addr`

Description: Specifies the IP address to which an Ntrip server will connect.

Access: read & write

Type: string

Values: Four numbers from 0 to 255 and separated by periods.

Default: `0.0.0.0`

Usage Guidelines: N/A

Command Examples: Set the IP address to 192.168.2.2.

```
set,/par/ntrip/server/addr,192.168.2.2
```

Or

```
set,ntrip/server/addr,192.168.2.2
```

Check the IP address.

```
print,/par/ntrip/server/addr
```

Or

```
print,ntrip/server/addr
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Port of Ntrip server

`[/par/]ntrip/server/port`

Description: Specifies the port of an Ntrip server.

Access: read & write

Type: integer

Values: `[0...65535]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the port to 8001.
`set,/par/ntrip/server/port,8001`
Or
`set,ntrip/server/port,8001`

Check the port.
`print,/par/ntrip/server/port`
Or
`print,ntrip/server/port`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

List of the Ntrip server mountpoint parameters

`[/par/]ntrip/server/mountpt`

Description: Specifies the list of the Ntrip server mountpoint parameters.

Access: read-only

Type: list

Values: see below parameters for description

Default: `{"","",}`

Usage Guidelines: N/A

Command Examples: Check the mountpoint parameters.
`print,/par/ntrip/server/mountpt`
Or
`print,ntrip/server/mountpt`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Mountpoint name

`[/par/]ntrip/server/mountpt/name`

Description:	Specifies the mountpoint name of a broadcast Ntrip server.
Access:	read & write
Type:	string [0...32]
Values:	up to 32 alphanumeric characters
Default:	""
Usage Guidelines:	N/A
Command Examples:	Set the mountpoint to NRC32. `set,/par/ntrip/server/mountpt,NRC32` Or `set,ntrip/server/mountpt,NRC32` Check the mountpoint. `print,/par/ntrip/server/mountpt` Or `print,ntrip/server/mountpt`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 3.3.
See Also:	N/A

Mountpoint description

`[/par/]ntrip/server/mountpt/descr`

Description: Specifies the mountpoint description of a broadcast Ntrip server.

Access: read & write

Type: string [0...128]

Values: up to 128 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the mountpoint description to `This_is_a_mountpoint`.

```
set,/par/ntrip/server/mountpt/descr,This_is_a_mountpoint
```

Or

```
set,ntrip/server/mountpt/descr,This_is_a_mountpoint
```

Check the mountpoint description.

```
print,/par/ntrip/server/mountpt/descr
```

Or

```
print,ntrip/server/mountpt/descr
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Password for an Ntrip server

`[/par/]ntrip/server/passwd`

Description: Specifies the password for the selected Ntrip server.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the password to alpha.
`set,/par/ntrip/server/passwd,alpha`
Or
`set,ntrip/server/passwd,alpha`

Check the password.
`print,/par/ntrip/server/passwd`
Or
`print,ntrip/server/passwd`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Reporting Ntrip data port and input mode

`[/par/]ntrip/data`

Description: Reports the Ntrip data port and input mode of this port.

Access: read-only

Type: list of two enumerated values

Values: See the parameters below for description.

Default: N/A

Usage Guidelines: N/A

Command Examples: Reports the current values.
`print,/par/ntrip/data`
Or
`print,ntrip/data`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

Ntrip data port

`[/par/]ntrip/data/port`

Description:	Specifies the Ntrip data port.
Access:	read & write
Type:	enumerated
Values:	`any_port`
Default:	`/dev/ser/d`
Usage Guidelines:	The data received from an Ntrip mountpoint will be passed to the appropriate decoder as if they were received from the specified port. The input mode of the specified port should be set by the user (via the `/par/<port>/imode` parameter) and to match the data format accepted from the mountpoint.
Command Examples:	Set the data port to /dev/ser/b. `set,/par/ntrip/data/port,/dev/ser/b` Or `set,ntrip/data/port,/dev/ser/b` Check the data port. `print,/par/ntrip/data/port` Or `print,ntrip/data/port`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

Input mode for the current Ntrip data port

`[/par/]ntrip/data/imode`

Description:	Reports the input mode for the current Ntrip data port.
Access:	read-only
Type:	enumerated
Values:	See the parameter `/par/<port>/imode`.
Default:	`cmd`
Usage Guidelines:	N/A
Command Examples:	Reports the input mode. `print,/par/ntrip/data/imode` Or `print,ntrip/data/imode`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	“Input mode” on page 3-167 for available values.

Output mode for the current Ntrip data port

`[/par/]ntrip/data/omode`

Description:	Reports the output mode for the current Ntrip data port.
Access:	read-only
Type:	enumerated
Values:	See the parameter `/par/<port>/omode`.
Default:	`cmd`
Usage Guidelines:	N/A
Command Examples:	Reports the output mode. `print,/par/ntrip/data/omode` Or `print,ntrip/data/omode`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 3.3.
See Also:	“Output mode” on page 3-168 for available values.

Ntrip errors

`[/par/]ntrip/error`

Description:	Reports the errors occurred while using Ntrip.
Access:	read-only
Type:	string [0...64]
Values:	up to 64 alphanumeric characters
Default:	`"none"`
Usage Guidelines:	N/A
Command Examples:	Reports the Ntrip errors, if any. `print,/par/ntrip/error` Or `print,ntrip/error`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

Ntrip Client Parameters

This section describes the parameters for configuring and monitoring Ntrip client.

Ntrip connection state

`[/par/]ntrip/state`

Description: Reports the connection state of Ntrip.

Access: read-only

Type: enumerated

Values: `off | wait | data | sleep`

- `off` – Ntrip is disabled.
- `wait` – the receiver is trying to connect with the caster.
- `data` – the receiver is getting correction data from Ntrip mountpoint.
- `sleep` – the receiver's Ntrip caster was unable to connect with the caster.

Default: `off`

Usage Guidelines: When you see the `sleep` state, check the parameter `/par/ntrip/error` for the reason.

Command Examples: Check the connection state.
`print,/par/ntrip/state`
Or
`print,ntrip/state`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

List of caster parameters

`[/par/]ntrip/caster`

Description:	Reports the list of caster parameters.
Access:	read-only
Type:	list
Values:	See values of parameters that follow for details.
Default:	`{0.0.0.0,0,"","",0}`
Usage Guidelines:	N/A
Command Examples:	Check the caster parameters. `print,/par/ntrip/caster` Or `print,ntrip/caster`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6
See Also:	N/A

IP of Ntrip caster

`[/par/]ntrip/caster/addr`

Description:	Specifies the IP address of an Ntrip caster.
Access:	read & write
Type:	string
Values:	Four numbers from 0 to 255 and separated by periods.
Default:	`0.0.0.0`
Usage Guidelines:	N/A
Command Examples:	Set the IP address to 192.168.2.1. `set,/par/ntrip/caster/addr,192.168.2.1` Or `set,ntrip/caster/addr,192.168.2.1` Check the IP address. `print,/par/ntrip/caster/addr` Or `print,ntrip/caster/addr`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

Port of Ntrip caster

`[/par/]ntrip/caster/port`

Description:	Specifies the port of an Ntrip caster.
Access:	read & write
Type:	integer
Values:	`[0...65535]`
Default:	`0`
Usage Guidelines:	N/A
Command Examples:	Set the port to 8001. `set,/par/ntrip/caster/port,8001` Or `set,ntrip/caster/port,8001` Check the port. `print,/par/ntrip/caster/port` Or `print,ntrip/caster/port`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

Mountpoint of Ntrip caster

`[/par/]ntrip/caster/mountpt`

Description:	Specifies the mountpoint of an Ntrip caster.
Access:	read & write
Type:	string [0...15]
Values:	up to 15 alphanumeric characters
Default:	""
Usage Guidelines:	N/A
Command Examples:	Set the mountpoint to NRC30. `set,/par/ntrip/caster/mountpt,NRC30` Or `set,ntrip/caster/mountpt,NRC30` Check the mountpoint. `print,/par/ntrip/caster/mountpt` Or `print,ntrip/caster/mountpt`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

User-ID for accessing data stream(s) of an Ntrip caster

`[/par/]ntrip/caster/user`

Description: Specifies the user-ID for accessing data stream(s) of the selected Ntrip caster.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: If user-ID is empty, the receiver will not send the user and password values to the NTRIP caster.

Command Examples: Set the user-ID to alpha.

```
set,/par/ntrip/caster/user,alpha
```

Or

```
set,ntrip/caster/user,alpha
```

Check the user-ID.

```
print,/par/ntrip/caster/user
```

Or

```
print,ntrip/caster/user
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

Password for accessing data stream(s) of an Ntrip caster

`[/par/]ntrip/caster/passwd`

Description: Specifies the password for accessing data stream(s) of the selected Ntrip caster.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: N/A

Command Examples: Set the password to omega.
`set,/par/ntrip/caster/passwd,omega`
Or
`set,ntrip/caster/passwd,omega`

Check the password.
`print,/par/ntrip/caster/passwd`
Or
`print,ntrip/caster/passwd`

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

Period of sending NMEA GGA to an Ntrip caster

`[/par/]ntrip/caster/nmea`

Description: Specifies the period of sending NMEA GGA sentence to the selected Ntrip caster.

Access: read & write

Type: integer

Values: `[-1...86400]`; expressed in seconds

- `-1` – the receiver will not send GGA to the Ntrip caster.
- `0` – the receiver will send GGA one time after connecting with the Ntrip caster.
- `[1...86400]` – the receiver will send GGA with the specified period.

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the period to 10 seconds.

```
set,/par/ntrip/caster/nmea,10
```

Or

```
set,ntrip/caster/nmea,10
```

Check the period.

```
print,/par/ntrip/caster/nmea
```

Or

```
print,ntrip/caster/nmea
```

Receiver Availability: This parameter is available for receivers with 4096 KB RAM.

History: This parameter was introduced in firmware 2.6.

See Also: N/A

Forcing sourcetable download

`[/par/]ntrip/caster/table`

Description:	Forces the receiver to retrieve a sourcetable from the selected Ntrip caster and reports the table in the reply messages.
Access:	read-only
Type:	string
Values:	A set of strings defined by the Ntrip protocol.
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Reports the sourcetable. `print,/par/ntrip/caster/table` Or `print,ntrip/caster/table`
Receiver Availability:	This parameter is available for receivers with 4096 KB RAM.
History:	This parameter was introduced in firmware 2.6.
See Also:	N/A

Accepting data over TCP/IP (Mobile mode)

This section describes the parameters used for obtaining data over a TCP/IP connection.

List of TCP/IP parameters

`[/par/]mobile`

Description:	Reports the list of TCP/IP parameters.
Access:	read-only
Type:	list
Values:	See parameters below for the description.
Default:	See parameters below for the description.
Usage Guidelines:	N/A
Command Examples:	Reports the TCP/IP parameters. `print,/par/mobile` Or `print,mobile`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Activating mobile mode

`[/par/]mobile/mode`

Description:	Turns the mobile mode on and off.
Access:	read & write
Type:	enumerated
Values:	`off | on` • `off` – disable the mobile mode. • `on` – enable the mobile mode.
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Enable the mobile mode. `set,/par/mobile/mode,on` Or `set,mobile/mode,on` Check the data port. `print,/par/mobile/mode` Or `print,mobile/mode`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Selecting mobile service

`[/par/]mobile/service`

Description: Selects the mobile service.

Access: read & write

Type: enumerated

Values: `rover | base | ip`

- `rover` – reserved for future use.
- `base` – reserved for future use.
- `ip` – the receiver will connect to the specified IP.

Default: `ip`

Usage Guidelines: N/A

Command Examples: Enable the mobile service to ip.

```
set,/par/mobile/service,ip
```

Or

```
set,mobile/service,ip
```

Check the mobile service.

```
print,/par/mobile/service
```

Or

```
print,mobile/service
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Mobile state

`[/par/]mobile/state`

Description: Reports the mobile state.

Access: read-only

Type: enumerated

Values: `off | available | unavailable | terminated | data | terminal`

- `off` – the mobile service is disabled.
- `available` – the selected mobile service is available.
- `unavailable` – the selected mobile service is unavailable.
- `terminated` – the selected mobile service is terminated.
- `data` – the receiver accepts data using the selected mobile service.
- `terminal` – the user can manage the remote source of data.

Default: `off`

Usage Guidelines: N/A

Command Examples: Check the mobile state.

```
print,/par/mobile/state
```

Or

```
print,mobile/state
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of IP parameters

`[/par/]mobile/ip`

Description: Reports the list of IP parameters.

Access: read-only

Type: list

Values: See the parameters below for description.

Default: See the parameters below for default values.

Usage Guidelines: N/A

Command Examples: Check the IP parameters.
`print,/par/mobile/ip`
Or
`print,mobile/ip`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

IP address

`[/par/]mobile/ip/addr`

Description: Specifies the IP address of a data stream.

Access: read & write

Type: string

Values: Four numbers from 0 to 255 and separated by periods.

Default: `0.0.0.0`

Usage Guidelines: N/A

Command Examples: Set the IP address to 192.168.2.1.
`set,/par/mobile/ip/addr,192.168.2.1`
Or
`set,mobile/ip/addr,192.168.2.1`

Check the IP address.
`print,/par/mobile/ip/addr`
Or
`print,mobile/ip/addr`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Port of data stream

`[/par/]mobile/ip/port`

Description:	Specifies the port of a data stream.
Access:	read & write
Type:	integer
Values:	`[0...65535]`
Default:	`0`
Usage Guidelines:	N/A
Command Examples:	Set the port to 8001. `set,/par/mobile/ip/port,8001` Or `set,mobile/ip/port,8001` Check the port. `print,/par/mobile/ip/port` Or `print,mobile/ip/port`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

User-ID for accessing data stream

`[/par/]mobile/ip/user`

Description: Specifies the user-ID for accessing a data stream.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: `""`

Usage Guidelines: This parameters is not used currently and is reserved for future updates.

Command Examples: Set the user-ID to alpha.

```
set,/par/mobile/ip/user,alpha
```

Or

```
set,mobile/ip/user,alpha
```

Check the user-ID.

```
print,/par/mobile/ip/user
```

Or

```
print,mobile/ip/user
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Password for accessing data stream

`[/par/]mobile/ip/passwd`

Description:	Specifies the password for accessing a data stream.
Access:	read & write
Type:	string [0...32]
Values:	up to 32 alphanumeric characters
Default:	""
Usage Guidelines:	This parameters is not used currently and is reserved for future updates.
Command Examples:	Set the password to omega. `set,/par/mobile/ip/passwd,omega` Or `set,mobile/ip/passwd,omega` Check the password. `print,/par/mobile/ip/passwd` Or `print,mobile/ip/passwd`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Period of sending NMEA GGA to a data source

`[/par/]mobile/ip/gga`

Description: Specifies the period of sending NMEA GGA sentence to the selected data source.

Access: read & write

Type: integer

Values: `[0...86400]`; expressed in seconds

- `0` – the receiver will not send GGA to the data source.
- `[1...86400]` – the receiver will send GGA with the specified period.

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the period to 10 seconds.

```
set,/par/mobile/ip/gga,10
```

Or

```
set,mobile/ip/gga,10
```

Check the period.

```
print,/par/mobile/ip/gga
```

Or

```
print,mobile/ip/gga
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of rover parameters

`[/par/]mobile/rover`

Description:	Reports the list of rover parameters.
Access:	read-only
Type:	list
Values:	See the parameters below for description.
Default:	See the parameters below for default values.
Usage Guidelines:	N/A
Command Examples:	Check the rover parameters. `print,/par/mobile/rover` Or `print,mobile/rover`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

IP address of the base to connect

`[/par/]mobile/rover/base`

Description:	Specifies the IP address of the base to connect.
Access:	read & write
Type:	string
Values:	Four numbers from 0 to 255 and separated by periods.
Default:	`0.0.0.0`
Usage Guidelines:	N/A
Command Examples:	Set the IP address to 192.168.2.1. `set,/par/mobile/rover/base,192.168.2.1` Or `set,mobile/rover/base,192.168.2.1` Check the IP address. `print,/par/mobile/rover/base` Or `print,mobile/rover/base`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Port of the base to connect

`[/par/]mobile/rover/port`

Description:	Specifies the port of the base to connect.
Access:	read & write
Type:	integer
Values:	`[0...65535]`
Default:	`0`
Usage Guidelines:	N/A
Command Examples:	Set the port to 8001. `set,/par/mobile/rover/port,8001` Or `set,mobile/rover/port,8001` Check the port. `print,/par/mobile/rover/port` Or `print,mobile/rover/port`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Password for the base to connect

`[/par/]mobile/rover/passwd`

Description: Specifies the password for the base to connect.

Access: read & write

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: ""

Usage Guidelines: N/A

Command Examples: Set the password to omega.
`set,/par/mobile/rover/passwd,omega`
Or
`set,mobile/rover/passwd,omega`

Check the password.
`print,/par/mobile/rover/passwd`
Or
`print,mobile/rover/passwd`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of base parameters

`[/par/]mobile/base`

Description: Reports the list of base parameters.

Access: read-only

Type: list

Values: See the parameters below for description.

Default: See the parameters below for default values.

Usage Guidelines: N/A

Command Examples: Check the base parameters.
`print,/par/mobile/base`
Or
`print,mobile/base`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

IP address of the base after establishing PPP connection

`[/par/]mobile/base/addr`

Description: Reports the IP address of the receiver obtained after PPP connection was established.

Access: read-only

Type: string

Values: Four numbers from 0 to 255 and separated by periods.

Default: `0.0.0.0`

Usage Guidelines: N/A

Command Examples: Check the IP address.
```
print,/par/mobile/base/addr
```
Or
```
print,mobile/base/addr
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Port of the base

`[/par/]mobile/base/port`

Description: Reports the port of the base.

Access: read-only

Type: integer

Values: `[0...65535]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Check the port.
```
print,/par/mobile/base/port
```
Or
```
print,mobile/base/port
```

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Password for the base

`[/par/]mobile/base/passwd`

Description: Reports the password for the base.

Access: read-only

Type: string [0...32]

Values: up to 32 alphanumeric characters

Default: ""

Usage Guidelines: N/A

Command Examples: Check the password.
`print,/par/mobile/base/passwd`
Or
`print,mobile/base/passwd`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

List of mobile data parameters

`[/par/]mobile/data`

Description: Reports the list of mobile data parameters.

Access: read-only

Type: list

Values: See the parameters below for description.

Default: See the parameters below for default values.

Usage Guidelines: N/A

Command Examples: Check the mobile parameters.
`print,/par/mobile/data`
Or
`print,mobile/data`

Receiver Availability: This parameter is unavailable for the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Mobile data port

`[/par/]mobile/data/port`

Description:	Specifies the mobile data port.
Access:	read & write
Type:	enumerated
Values:	`port`
Default:	`/dev/ser/d`
Usage Guidelines:	The data received from a data stream will be passed to the appropriate decoder as if they were received from the specified port. The input mode of the specified port should be set by the user (via the `/par/<port>/imode` parameter) and to match the data format accepted from the data source.
Command Examples:	Set the data port to /dev/ser/b. `set,/par/mobile/data/port,/dev/ser/b` Or `set,mobile/data/port,/dev/ser/b` Check the data port. `print,/par/mobile/data/port` Or `print,mobile/data/port`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Input mode for the current mobile data port

`[/par/]mobile/data/imode`

Description:	Reports the input mode for the current mobile data port.
Access:	read-only
Type:	enumerated
Values:	See the parameter `/par/<port>/imode`.
Default:	`cmd`
Usage Guidelines:	N/A
Command Examples:	Reports the input mode. `print,/par/mobile/data/imode` Or `print,mobile/data/imode`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Ntrip errors

`[/par/]mobile/error`

Description:	Reports the errors occurred while using mobile mode.
Access:	read-only
Type:	string [0...64]
Values:	up to 64 alphanumeric characters
Default:	`"No errors"`
Usage Guidelines:	N/A
Command Examples:	Reports the mobile mode errors, if any. `print,/par/mobile/error` Or `print,mobile/error`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Debug mobile mode

`[/par/]mobile/debug`

Description:	Enables output of data used in debugging mobile mode.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Enable debug. `set,/par/mobile/debug,on` Or `set,mobile/debug,on` Check the debug status. `print,/par/mobile/debug` Or `print,mobile/debug`
Receiver Availability:	This parameter is unavailable for the GMS-2.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Port Communication Parameters

Serial Port Parameters

Baud rate

`[/par/]<rs_port>/rate`

Description: Specifies the baud rate for the selected serial port.

Access: read & write

Type: enumerated

Values: `460800 | 230400 | 153600 | 115200 | 57600 | 38400 | 19200 | 9600 | 4800 | 2400 | 1200 | 600 | 300`

Default: see next parameter

Usage Guidelines: N/A

Command Examples: Set the baud rate of serial port C to 38400.
`set,/par/dev/ser/c/rate,38400`
Or
`set,dev/ser/c/rate,38400`

Check the baud rate on serial port D.
`print,/par/dev/ser/d/rate`
Or
`print,dev/ser/d/rate`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Default baud rate

`[/par/]<rs_port>/rate&def`

Description: Specifies which default baud rates will be enabled for the corresponding serial ports after clearing NVRAM or executing the `init,/par/` command.

Access: read & write

Type: enumerated

Values: `460800 | 230400 | 153600 | 115200 | 57600 | 38400 | 19200 | 9600 | 4800 | 2400 | 1200 | 600 | 300`

Default: `115200`

Usage Guidelines: You cannot update the default baud rates without using these parameters.

Command Examples: Set the default baud rate of serial port C to 38400.
`set,/par/dev/ser/c/rate&def,38400`
Or
`set,dev/ser/c/rate&def,38400`

Check the baud rate on serial port D.
`print,/par/dev/ser/d/rate&def`
Or
`print,dev/ser/d/rate&def`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Hardware handshaking

`[/par/]<rs_port>/rtscts`

Description:	Enables and disables hardware handshaking.
Access:	read & write
Type:	boolean
Values:	`on` | `off`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Enable handshaking on serial port C. `set,/par/dev/ser/c/rtscts,on` Or `set,dev/ser/c/rtscts,on` Check handshaking on serial port D. `print,/par/dev/ser/d/rtscts` Or `print,dev/ser/d/rtscts`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

RTS signal

`[/par/]<rs_port>/rts`

Description: Enables and disables the state of the RTS output signal.

Access: read & write

Type: boolean

Values: `on | off`

Default: `on`

Usage Guidelines: N/A

Command Examples: Set RTS to `on`.
```
set,/par/dev/ser/c/rts,on
```
Or
```
set,dev/ser/c/rts,on
```

Check RTS on serial port D.
```
print,/par/dev/ser/d/rts
```
Or
```
print,dev/ser/d/rts
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

CTS signal

`[/par/]<rs_port>/cts`

Description: Enables and disables the state of the CTS input signal.

Access: read-only

Type: boolean

Values: `on | off`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Check CTS on serial port D.
```
print,/par/dev/ser/d/cts
```
Or
```
print,dev/ser/d/cts
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of stop bits

`[/par/]<rs_port>/stops`

Description: Specifies the number of stop bits.

Access: read & write

Type: integer

Values: `1 | 2`

Default: `1`

Usage Guidelines: N/A

Command Examples: Set stop bits to `2`.
`set,/par/dev/ser/c/stops,2`
Or
`set,dev/ser/c/stops,2`

Check stop bits on serial port D.
`print,/par/dev/ser/d/stops`
Or
`print,dev/ser/d/stops`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Parity

`[/par/]<rs_port>/parity`

Description: Specifies the parity scheme.

Access: read & write

Type: enumerated

Values:
- `N` – no parity
- `odd` – odd parity
- `even` – even parity
- `fodd` – forced odd parity (logical 1)
- `feven` – forced even parity (logical 0)

Default: `N`

Usage Guidelines: N/A

Command Examples: Set parity to `odd`.

```
set,/par/dev/ser/c/parity,odd
```

Or

```
set,dev/ser/c/parity,odd
```

Check stop bits on serial port D.

```
print,/par/dev/ser/d/parity
```

Or

```
print,dev/ser/d/parity
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of data bits

`[/par/]<rs_port>/bits`

Description: Specifies the number of data bits.

Access: read & write

Type: integer

Values: `[5...8]`

Default: `8`

Usage Guidelines: N/A

Command Examples: Set data bits to `7`.
`set,/par/dev/ser/c/bits,7`
Or
`set,dev/ser/c/bits,7`

Check stop bits on serial port D.
`print,/par/dev/ser/d/bits`
Or
`print,dev/ser/d/bits`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Infrared mode

`[/par/]<rs_port>/ir`

Description: Specifies the infrared port.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: The TPS receiver may have either one infrared port, which is always port D, or no infrared port.
Only use this mode if the following conditions have been met:

- The external device is compatible with the receiver's infrared interface (for example, CDU-1).
- The receiver's hardware supports the infrared port.
- The Infrared Port option has been enabled in the receiver.

Command Examples: Set infrared port on serial port C.
`set,/par/dev/ser/c/ir,on`
Or
`set,dev/ser/c/ir,on`

Check infrared port on serial port D.
`print,/par/dev/ser/d/ir`
Or
`print,dev/ser/d/ir`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Common Communication Parameters

Input mode

`[/par/]<port>/imode`

Description: Specifies the type of data to input on the selected port.

Access: read & write

Type: enumerated

Values: `cmd | echo | rtcm | rtcm3 | cmr | jps | omni | dtp | term | none`

- `cmd` – the port is in command mode and will recognize the commands only.
- `echo` – the port is in echo mode and will redirect all incoming data to an output stream, either another port or the current log file.
- `rtcm` – the port recognizes and decodes accepted RTCM version 2.1, 2.2, and 2.3 messages.
- `rtcm3` – the port recognizes and decodes accepted RTCM version 3.0 messages.
- `cmr` – the port recognizes and decodes accepted CMR/CMR+ messages.
- `jps` – the port recognizes and decodes accepted TPS messages.
- `omni` – the port recognizes and decodes OmniSTAR VBS corrections.
- `dtp` – the port is in Data Transfer Protocol mode, which is used for data transfer to/from the receiver.
- `term` – the port is in PPP mode.
- `none` – the port will reject any incoming data.

Default: `cmd`

Usage Guidelines: The port through which you are about to connect with the receiver must be in the `cmd` mode. Otherwise, connection will not be established.

Command Examples: Set the receiver's port C to accept incoming RTCM 3.0 messages.

```
set,/par/dev/ser/c/imode,rtcm3
```

Or

```
set,dev/ser/c/imode,rtcm3
```

Check the input mode on the USB port.

```
print,/par/dev/usb/a/imode
```

Or

```
print,dev/usb/a/imode
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in the first release of firmware.
- The `rtcm3` value was added in firmware 2.4.
- The `term` value was added in firmware 2.6.

See Also: N/A

Output mode

`[/par/]<port>/omode`

Description: Specifies the type of data to output on the selected port.

Access: read-only

Type: enumerated

Values: `std | dtp | term`

- `std` – the port is in standard output mode.
- `dtp` – the port is in Data Transfer Protocol mode, which is used for data transfer to/from the receiver.
- `term` – the port is in PPP mode.

Default: `std`

Usage Guidelines: N/A

Command Examples: Check the output mode on the USB port.

```
print,/par/dev/usb/a/omode
```

Or

```
print,dev/usb/a/omode
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in the first release of firmware.
- The `term` value was added in firmware 2.6.

See Also: N/A

Stream duplication

`[/par/]<stream>/dup`

Description: Instructs the receiver to duplicate the output stream of a port to another port.

Access: read & write

Type: enumerated

Values: `stream | /dev/null`
`/dev/null` – duplication is disabled.

Default: `/dev/null`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to duplicate the data stream by outputting it not only to port C but also to the USB port.
`set,/par/dev/ser/c/dup,/dev/usb/a`
Or
`set,dev/ser/c/dup,/dev/usb/a`

Check if the duplication to a port is enabled on serial port D.
`print,/par/dev/ser/d/dup`
Or
`print,dev/ser/d/dup`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Echo mode

`[/par/]<stream>/echo`

Description: Defines the output stream into which you are going to redirect input data thus activating data echoing

Access: read & write

Type: enumerated

Values: `stream | /dev/null`
`/dev/null` – data echoing is disabled.

Default: `/dev/null`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to redirect data from serial port C to the USB port.
`set,/par/dev/ser/c/echo,/dev/usb/a`
Or
`set,dev/ser/c/echo,/dev/usb/a`

Check the echo mode on serial port D.
`print,/par/dev/ser/d/echo`
Or
`print,dev/ser/d/echo`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Echo-off sequence

`[/par/]<stream>/eoff`

Description: Defines an echo-off sequence for a specific receiver port.

Access: read & write

Type: string [4...32]

Values: up to 32 alphanumeric characters

Default: `#OFF#`

Usage Guidelines: N/A

Command Examples: Set the echo-off sequence to QUIT.

```
set,/par/dev/ser/c/eoff,QUIT
```

Or

```
set,dev/ser/c/eoff,QUIT
```

Check the echo-off sequence on serial port D.

```
print,/par/dev/ser/d/eoff
```

Or

```
print,dev/ser/d/eoff
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Wrapped echo

`[/par/]<stream>/ewrap`

Description: Defines an echo-off sequence for a specific receiver port.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – data echoing is carried out as specified by `/par/<stream>/echo`. This mode is called *normal data echoing* or *raw data echoing*.
- `on` – when `/par/[port]/echo` is other than `/dev/null`, data echoing is carried out in the *wrapped mode*, i.e. all of the characters received from `stream` are combined into a corresponding >> message before being directed to the output stream.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable wrapped mode on serial port C.
`set,/par/dev/ser/c/ewrap,on`
Or
`set,dev/ser/c/ewrap,on`

Check the wrapped mode on serial port D.
`print,/par/dev/ser/d/ewrap`
Or
`print,dev/ser/d/ewrap`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Wrapped data size

`[/par/]<stream>/wsize`

Description: Defines a threshold value for wrapping input data when in the wrapped mode.

Access: read & write

Type: integer

Values: `[0...128]`; bytes

Default: `128`

Usage Guidelines: N/A

Command Examples: Set the wrapped data size to 64 bytes on serial port C.

```
set,/par/dev/ser/c/wsize,64
```

Or

```
set,dev/ser/c/wsize,64
```

Check the wrapped data size on serial port D.

```
print,/par/dev/ser/d/wsize
```

Or

```
print,dev/ser/d/wsize
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

TCP/IP Communication Parameters

IP address

`[/par/]net/ip/addr`

Description: Identifies the receiver on a TCP/IP network by specifying its IP address.

Access: read & write

Type: string

Values: any valid IP address

Default: `192.168.2.2`

Usage Guidelines: N/A

Command Examples: Specify the IP address = 205.142.135.237.

```
set,/par/net/ip/addr,205.142.135.237
```

Or

```
set,net/ip/addr,205.142.135.237
```

Check the IP address.

```
print,/par/net/ip/addr
```

Or

```
print,net/ip/addr
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Network mask

`[/par/]net/ip/mask`

Description: Specifies the network mask.

Access: read & write

Type: string

Values: any valid IP address

Default: `255.255.255.192`

Usage Guidelines: N/A

Command Examples: Specify the network mask = 255.255.255.0.

```
set,/par/net/ip/mask,255.255.255.0
```

Or

```
set,net/ip/mask,255.255.255.0
```

Check the network mask.

```
print,/par/net/ip/mask
```

Or

```
print,net/ip/mask
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Gateway

`[/par/]net/ip/gw`

Description:	Specifies the gateway.
Access:	read & write
Type:	string
Values:	any valid IP address
Default:	`192.168.2.1`
Usage Guidelines:	N/A
Command Examples:	Specify the gateway = 192.168.2.5. `set,/par/net/ip/gw,192.168.2.5` Or `set,net/ip/gw,192.168.2.5` Check the gateway. `print,/par/net/ip/gw` Or `print,net/ip/gw`
Receiver Availability:	This parameter is unavailable in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

MAC address

`[/par/]net/mac/addr`

Description:	Specifies your receiver's unique hardware number on a network.
Access:	read & write
Type:	string
Values:	any valid MAC address
Default:	automatically generated unique value
Usage Guidelines:	High part of the MAC address is fixed and is equal to 00:04:07. Low part is generated automatically from the receiver ID. The user usually does not need to change MAC address.
Command Examples:	Check the MAC address. `print,/par/net/mac/addr` Or `print,net/mac/addr`
Receiver Availability:	This parameter is unavailable in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

TCP port

`[/par/]net/tcp/port`

Description: Specifies TCP port the receiver is listening on for telnet-like connections.

Access: read & write

Type: integer

Values: `[1...65535]`

Default: `8002`

Usage Guidelines:

- Up to 5 simultaneous connections are supported.
- After a client connected to the TCP port, receiver outputs login prompt. Explicit a, b,..., e reply followed by new line to the login prompt will make connection to corresponding TCP stream (`/dev/tcp/a`, ..., `/dev/tcp/e`). Entering just new line will make receiver to automatically select the first available TCP stream and make connection to it. After login prompt, the receiver will issue the password prompt. The string specified in `/par/net/passwd` followed by new line should be entered by the client in reply to the password prompt. If the string entered does not match, connection will be denied.

Command Examples:

Specify the TPC port = 8002.
```
set,/par/net/tcp/port,8002
```
Or
```
set,net/tcp/port,8002
```

Check the TCP port.
```
print,/par/net/tcp/port
```
Or
```
print,net/tcp/port
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

TCP connection timeout

`[/par/]net/tcp/timeout`

Description: Specifies a period of time that must elapse before an inactive connection will be terminated.

Access: read & write

Type: integer

Values: `[1...2^31-1]`; seconds

Default: `600`

Usage Guidelines: N/A

Command Examples: Set the timeout to 1200 seconds.
`set,/par/net/tcp/timeout,1200`
Or
`set,net/tcp/timeout,1200`

Check the timeout.
`print,/par/net/tcp/timeout`
Or
`print,net/tcp/timeout`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

FTP port

`[/par/]net/ftp/port`

Description: Specifies TCP port the receiver is listening on for FTP connection.

Access: read & write

Type: integer

Values: `[1...65535]`

Default: `21`

Usage Guidelines: Only one simultaneous connection is supported.

Command Examples: Specify the FTP port = 21.
`set,/par/net/ftp/port,21`
Or
`set,net/ftp/port,21`

Check the FTP port.
`print,/par/net/ftp/port`
Or
`print,net/ftp/port`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

FTP connection timeout

`[/par/]net/ftp/timeout`

Description: Specifies a period of time that must elapse before an inactive connection will be terminated.

Access: read & write

Type: integer

Values: `[1...2^31-1]`; seconds

Default: `600`

Usage Guidelines: N/A

Command Examples: Set the timeout to 1200 seconds.

```
set,/par/net/ftp/timeout,1200
```

Or

```
set,net/ftp/timeout,1200
```

Check the timeout.

```
print,/par/net/ftp/timeout
```

Or

```
print,net/ftp/timeout
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Telnet password

`[/par/]net/passwd`

Description:	Sets a password for telnet-like TCP connections.
Access:	read & write
Type:	string [0...15]
Values:	up to 15 alphanumeric characters
Default:	automatically generated value unique for each receiver
Usage Guidelines:	This protection will prevent unauthorized Telnet access to the receiver.
Command Examples:	Set the password to ABCXYZ123. `set,/par/net/passwd,ABCXYZ123` Or `set,net/passwd,ABCXYZ123` Check the password. `print,/par/net/passwd` Or `print,net/passwd`
Receiver Availability:	This parameter is unavailable in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

History Logger Parameters

History Logger provides statistical information on the raw data (receiver observables) being logged to a selected stream.

After History Logger has been associated with a specific output stream (see the parameter `/par/out/logh/stream` below) and data logging to this stream is enabled, History Logger will start collecting and recording the corresponding information. History Logger records one bit of information per satellite every N seconds (refer to the parameter `/par/out/logh/period` below). This bit is set to unity if and only if

- at least some of the observables have been acquired, and
- there have been no loss-of-lock events

for the given satellite in the last N seconds. If either or both of these conditions are not met, a zero bit is recorded to the logging history.

Note that the logging history has a limited capacity (a maximum of 32 satellites, 128 bits per satellite). Satellites that have only zero bits are excluded from the logging history.

All data collected by History Logger can be obtained from the LH message. For detailed information about LH, see page 4-127.

Logging period

`[/par/]out/logh/period`

Description: Specifies the period of logging history for the tracked satellites.

Access: read & write

Type: integer

Values: `[0...86400]`; expressed in seconds

Default: `30`

Usage Guidelines: One bit per satellite records to [LH] message with the specified period.

Command Examples: Set the history logging period to 60 seconds.
`set,/par/out/logh/period,60`
Or
`set,out/logh/period,60`

Query the external frequency value.
`print,/par/out/logh/period`
Or
`print,out/logh/period`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

`[/par/]out/logh/period` *(Continued)*

History: This parameter was introduced in the first release of firmware.

See Also: "LH – Logging history" on page 4-127.

Output stream

`[/par/]out/logh/stream`

Description: Specifies the output stream for which the history logger should gather information.

Access: read & write

Type: enumerated

Values: `<stream>`

Default: `/dev/null`

Usage Guidelines: Setting this parameter to `/dev/null` will disable history logging.

Command Examples: Set serial port D as the output stream.
`set,/par/out/logh/stream,/dev/ser/d`
Or
`set,out/logh/stream,/dev/ser/d`

Query the external frequency value.
`print,/par/out/logh/stream`
Or
`print,out/logh/stream`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also:
- "LH – Logging history" on page 4-127.
- "Parameter Specification" on page 3-3 for `<stream>` definition.

Output Time-Frame Parameters

The *output time-frame* mode configures your receiver for TDMA. It allows the receiver to share the same data link with other receivers without interfering using a time ordered access.

Time-frame is a periodic time interval of a specified length which defines a time slot when the receiver can output data to a port. A time-frame specification consists of three parameters: period, length, and delay. A time-frame begins `delay` seconds after the receiver time modulo `period` becomes equal to zero and lasts for the `length` seconds.

By enabling this mode, the receiver starts buffering the output data until a time-frame occurs. When it occurs, the receiver outputs buffered data to the given port until the time-frame ends. At the end of every time-frame the receiver cleares its internal buffers in order to avoid the data remaining in these buffers to be output in the subsequent time frames.

Time-frame mode

`[/par/]<port>/oframe/mode`

Description: Enables or disables output time-frames.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the corresponding port will output data all the time.
- `on` – the corresponding port will output data only at the specified time-frames.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable time-frame mode for serial port C.
`set,/par/dev/ser/c/oframe/mode,on`
Or
`set,dev/ser/c/oframe/mode,on`

Query time-frame mode for serial port D.
`print,/par/dev/ser/d/oframe/mode`
Or
`print,dev/ser/d/oframe/mode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “Multiple Reference Stations” on page 3-511 for how you might use time-frames.

Time-frame period

`[/par/]<port>/oframe/period`

Description: Specifies the frequency at which output time-frames occur.

Access: read & write

Type: float

Values: `[0.000...86400.000]`; expressed in seconds

Default: `1.000`

Usage Guidelines:
- Time-frames start at the moments when the receiver time modulo period is equal to zero.
- A time-frame actually starts with a delay specified in the parameter `/par/<port>/oframe/delay`.

Command Examples: Set time-frame period for serial port C to 0.5 second.

```
set,/par/dev/ser/c/oframe/period,0.5
```

Or

```
set,dev/ser/c/oframe/period,0.5
```

Query time-frame period for serial port D.

```
print,/par/dev/ser/d/oframe/period
```

Or

```
print,dev/ser/d/oframe/period
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “Multiple Reference Stations” on page 3-511 for how you might use time-frames.

Time-frame length

`[/par/]<port>/oframe/length`

Description: Specifies how long the data output will last after the time-frame has started.

Access: read & write

Type: float

Values: `[0.000...86400.000]`; expressed in seconds

Default: `1.000`

Usage Guidelines: N/A

Command Examples: Set time-frame length for serial port C to 2 seconds.
`set,/par/dev/ser/c/oframe/length,2`
Or
`set,dev/ser/c/oframe/length,2`

Query time-frame period for serial port D.
`print,/par/dev/ser/d/oframe/length`
Or
`print,dev/ser/d/oframe/length`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “Multiple Reference Stations” on page 3-511 for how you might use time-frames.

Time-frame delay

`[/par/]<port>/oframe/delay`

Description: Specifies the amount of time that the receiver will wait from the scheduled moment before it runs the time-frame.

Access: read & write

Type: float

Values: `[0.000...86400.000]`; expressed in seconds

Default: `0.000`

Usage Guidelines: N/A

Command Examples: Set time-frame delay for serial port C to 0.363 second.

```
set,/par/dev/ser/c/oframe/delay,0.363
```

Or

```
set,dev/ser/c/oframe/delay,0.363
```

Query time-frame delay for serial port D.

```
print,/par/dev/ser/d/oframe/delay
```

Or

```
print,dev/ser/d/oframe/delay
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “Multiple Reference Stations” on page 3-511 for how you might use time-frames.

Job Programming

Job programming allows you to specify a fixed number of jobs that can be performed by the receiver at the desired time. This will make the receiver to run multi-step operations automatically. A job might consist of one, two, or more commands. Each job has the following elements:

- a unique integer identifier – serves as the name of the job and also controls the order (priority) in which the job is performed. The highest priority is 0, the lowest 15. If two or more jobs are programmed to start simultaneously, the job with the higher priority will start first. For example, if the jobs with identifiers "0" and "1" are to be started at the same time, job "0" will be executed before job "1".
- an activity flag – indicates whether a job is a candidate for execution. A job with the activity flag set to unity is called *active*. The receiver will execute an active job as soon as the current time matches the value from the job processing definition. The receiver will ignore jobs with activity flags set to zero.
- processing definition – contains a set of user-defined fields which provide the receiver with instructions on how to process the job.
  - `spec` specifies job execution time.
  - `cmds` specifies the tag of a command set to be executed in the job.
  - `count` specifies how many times the job will be executed.
  - `port` specifies an output port.

After you configure the desired jobs, the receiver's job scheduler executes the jobs as follows:

- The scheduler takes a tag from the `cmds` field of the job processing definition and executes command(s) found at this tag in the array of command strings. Using the tag to identify command string allows multiple jobs to share the same command string.
- The string found is executed the same way if it were received through input port specified in the `port` field of the job. As a consequence of this rule, should the command generate some output, the output is sent to the port specified in the `port` field. Current input mode of the field `port` does not matter though; the command is executed as if the port is in the command mode.

Every active job also serves as a wake-up point for the receiver. Therefore if the job scheduler is enabled and the receiver is in sleep mode, the next job to be executed will first wake-up the receiver and only then corresponding commands will be executed. This feature also makes it useful to have empty command string to be assigned to a job definition. Such definition will just wake-up the receiver without execution of any commands.

When session scheduler is being turned on or is being restarted, the scheduler makes a copy of activity flag and `count` field of every job. Let us call these copies *stat/active* and *stat/count*, respectively. Every time the job is executed and *stat/count* is non-zero, the *stat/count* decrements. Should *stat/count* become zero as a result of decrement, the *stat/active* flag is turned off. The job is thus deactivated and stays inactive until scheduler mode is changed or scheduler is restarted. This allows you to limit number of executions of a job by setting its *count* field to a non-zero value. The *stat/mode*

and *stat/count* fields of every job could be read from the receiver for reference but could not be directly changed by user.

Execution time `spec` is a template containing four fields, namely day of week `day`, hour of day `hour`, minute of hour `minute`, and second of minute `second`. Each field can either be set to an integer value in corresponding range, or to a special value that serves as a wild-card that matches any value. In order for a job to be executed at a given time T, job `spec` should match the given time T. Scheduler compares current time against `spec` by first decomposing time into the set of corresponding fields, then comparing every field of decomposed time against corresponding field of `spec`. Time matches `spec` if and only if every field of time matches corresponding field of `spec`. Fields of `spec` having special value match any value of corresponding field of time. For example, specification "2d17h__m30s" ("__" is double underscore) matches "2:17:00:30", "2:17:01:30",..., "2:17:59:30". It means that job having such `spec` will be executed at the middle of every minute during one hour starting on Monday, 17:00:00, i.e., as week number could not be specified, it will be executed every Monday at 17:00:30, 17:01:30,..., and 17:59:30 (60 times). To limit execution of such job to single (next) Monday, job counter should be set to 60 before activation of the session.

In fact session scheduler does not check `spec` of every active job against current time every second. Instead it determines job that should be executed next (along with its next execution time) whenever scheduler mode is changed, scheduler is restarted, or job is executed. The identifier of the next job to be run and corresponding execution time along with current time broken into four fields could be read from the receiver for reference.

Parameters

This topic describes the parameters for configuring, executing, and monitoring the programmed jobs.

Managing job scheduler

`[/par/]sess/mode`

Description: Activates and deactivates the job scheduler.

Access: read & write

Type: enumerated

Values: `off | on | susp`

- off – the job scheduler is deactivated.
- on – the job scheduler is activated.
- susp – the job scheduler is activated without commands execution.

Default: `off`

Usage Guidelines: This parameter initiates job processing definitions.

`[/par/]sess/mode` *(Continued)*

Command Examples: Activate job scheduler.
`set,/par/sess/mode,on`
Or
`set,sess/mode,on`

Confirm status of job scheduler.
`print,/par/sess/mode`
Or
`print,sess/mode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Restarting job scheduler

`[/par/]sess/restart`

Description: Restarts the job scheduler.

Access: read & write

Type: boolean

Values: `off | on`

- off – has no any effect.
- on – the job scheduler restarts.

Default: `off`

Usage Guidelines:
- After restarting the scheduler, the parameter's default value is immediately restored.
- This parameter is similar to setting the parameter `/par/sess/mode` to `off` then to `on`.

Command Examples: Restart the job scheduler.
`set,/par/sess/restart,on`
Or
`set,sess/restart,on`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Activity flags for all jobs

`[/par/]sess/active`

Description: Enables and disables activity flags for all jobs.

Access: read & write

Type: array of 16 boolean values

Values: `{bool0,bool2,...bool15}`, where each `bool` corresponds to a particular job and can be `on` or `off`.

- `y` – activity flag is enabled.
- `n` – activity flag is disabled.

Default: `{n,...,n}`

Usage Guidelines: The values of this parameter is copied to the `/par/sess/stat/active` parameter whenever job scheduler mode is changed or scheduler is restarted.

Command Examples: Enable the activity flags for jobs 0 through 5.

```
set,/par/sess/active,{y,y,y,y,y,y,n,n,n,n,n,n,n,n,n,n}
```

Or

```
set,sess/active,{y,y,y,y,y,y,n,n,n,n,n,n,n,n,n,n}
```

Check job activity flags.

```
print,/par/sess/active
```

Or

```
print,sess/active
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Activity flag for specified job

`[/par/]sess/active/<N>`

Description: Enables and disables activity flag for a specified job.

Access: read & write

Type: boolean

Values:
- `y` – activity flag is enabled.
- `n` – activity flag is disabled.

Default: `n`

Usage Guidelines: `N` ranges from 0 to 15.

Command Examples: Enable the activity flag for job 3.
```
set,/par/sess/active/3,y
```
Or
```
set,sess/active/3,y
```

Check the activity flag for job 3.
```
print,/par/sess/active/3
```
Or
```
print,sess/active/3
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Processing definitions for all jobs

`[/par/]sess/job`

Description: Configures and retrieves processing definitions for all jobs.

Access: read-only

Type: array of 16 processing definitions

Values:
```
{{spec0,cmds0,count0,port0},
...
{spec16,cmds16,count16,port16}}
```

Default:
```
{{__d__h__m__s,0,0,""},
...
{__d__h__m__s,0,0,""}}
```

Usage Guidelines: N/A

Command Examples: Check processing definitions for all jobs.
```
print,/par/sess/job
```
Or
```
print,sess/job
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Processing definition for specified job

`[/par/]sess/job/<N>`

Description:	Configures and retrieves processing definition for a specified job.
Access:	read & write
Type:	list
Values:	`{spec,cmds,count,port}`
Default:	`{__d__h__m__s,0,0,""}`
Usage Guidelines:	`N` ranges from 0 to 15.
Command Examples:	Configure the processing definition for job 7. `set,/par/sess/job/7,{"_d_h_m00s",2,0,/cur/log}` Or `set,sess/job/7,{"_d_h_m00s",2,0,/cur/log}` Check the processing definition for job 7. `print,/par/sess/job/7` Or `print,sess/job/7`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Tag of a command set for specified job

`[/par/]sess/job/<N>/cmds`

Description: Defines the tag of a command set to be executed in a specified job.

Access: read & write

Type: integer

Values: `[0...7]`

Default: `0`

Usage Guidelines: `N` ranges from 0 to 15.

Command Examples: Set the tag to 2 for job 3.

```
set,/par/sess/job/3/cmds,2
```

Or

```
set,sess/job/2/cmds,2
```

Check the tag for job 7.

```
print,/par/sess/job/7/cmds
```

Or

```
print,sess/job/7/cmds
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of times to execute specified job

`[/par/]sess/job/<N>/count`

Description: Designates the number of times the job should be executed by the scheduler before deactivation.

Access: read & write

Type: integer

Values: `[0...2^31-1]`
`0` – means will be executed an unlimited number of times.

Default: `0`

Usage Guidelines:
- The value is copied to the parameter `/par/sess/stat/count/N` every time job scheduler mode is changed or scheduler is restarted.
- `N` ranges from 0 to 15.

Command Examples: Set the count to zero for job 3.
```
set,/par/sess/job/3/count,0
```
Or
```
set,sess/job/3/count,0
```

Check the count for job 3.
```
print,/par/sess/job/3/count
```
Or
```
print,sess/job/3/count
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output port for specified job

`[/par/]sess/job/<N>/port`

Description: Defines an output port for the specified job.

Access: read & write

Type: enumerated

Values: `/<stream>|""`

Default: `""`

Usage Guidelines:

- The commands assigned to the job are executed as if they were received from the port specified by this parameter.
- Should the commands generate replies, they will be sent to the port specified by this parameter.
- `N` ranges from 0 to 15.

Command Examples: Set the port to serial A for job 3.

```
set,/par/sess/job/3/port,/dev/ser/a
```

Or

```
set,sess/job/3/port,/dev/ser/a
```

Check the port for job 3.

```
print,/par/sess/job/3/port
```

Or

```
print,sess/job/3/port
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: “Parameter Specification” on page 3-3 for `<stream>` definition.

Command sets

`[/par/]sess/cmds`

Description:	Reports command sets.
Access:	read-only
Type:	array of eight strings
Values:	`{str0,str1,str2,str3,str4,str5,str6,str7}`, where each `str` corresponds to a particular command set and can be up to 64 characters.
Default:	`{"","","","","","","",""}`
Usage Guidelines:	N/A
Command Examples:	Check the command sets. `print,/par/sess/cmds` Or `print,sess/cmds`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Command set for specified job

`[/par/]sess/cmds/<N>`

Description: Specifies and reports a command set for the specified job.

Access: read & write

Type: string

Values: up to 64 characters

Default: `""`

Usage Guidelines: `N` ranges from 0 to 15.

Command Examples: Set a command set for job 3 consisting from a single command that will open the log file HiPer_diag.

```
%%set,/par/sess/cmds/3,"%%create,HiPer_diag"
```

Or

```
set,sess/cmds/3,"%%create,HiPer_diag"
```

Check a command set for job 3.

```
print,/par/sess/cmds/3
```

Or

```
print,sess/cmds/3
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Status for all jobs

`[/par/]sess/stat`

Description: Reports the status parameters for all jobs.

Access: read-only

Type: list

Values:
```
{{active0,...,active15},{count0,...,count15},nextjob,
{next,curr,delta}}
```

Default:
```
{{n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},-1,
{__d__h__m__s, current_time,__d__h__m__s}}
```

Usage Guidelines: N/A

Command Examples: Check status for jobs.
```
print,/par/sess/stat
```
Or
```
print,sess/stat
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Activity flags for jobs

`[/par/]sess/stat/active`

Description: Reports the activity flags for all jobs.

Access: read-only

Type: array of 16 boolean values

Values: `{active0,...,active15}`, where each `active` corresponds to a particular job and can be `on` or `off`.

- `y` – activity flag is enabled.
- `n` – activity flag is disabled.

Default: `{n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n}`

Usage Guidelines: This parameter is updated to the values of the parameter `/par/sess/active` whenever job scheduler mode is changed or job scheduler is restarted.

Command Examples: Check activity flags for jobs.

```
print,/par/sess/stat/active
```

Or

```
print,sess/stat/active
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Activity flag for specified job

`[/par/]sess/stat/active/<N>`

Description: Reports the activity flag for the specified job.

Access: read-only

Type: boolean

Values: `y | n`

- `y` – activity flag is enabled.
- `n` – activity flag is disabled.

Default: `n`

Usage Guidelines: `N` ranges from 0 to 15.

Command Examples: Check activity flag for job 4.
```
print,/par/sess/stat/active/4
```
Or
```
print,sess/stat/active/4
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of executions remaining for jobs

`[/par/]sess/stat/count`

Description: Reports the number of executions remaining before jobs turn off.

Access: read-only

Type: array of 16 integer values

Values: `{count0,...,count15}`, where each `count` corresponds to a particular job and can range from `0` to $2^{31}-1$.

Default: `{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}`

Usage Guidelines: N/A

Command Examples: Check remaining executions for jobs.
```
print,/par/sess/stat/count
```
Or
```
print,sess/stat/count
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of executions remaining for specified job

`[/par/]sess/stat/count/<N>`

Description: Reports the number of executions remaining before a job turns off.

Access: read-only

Type: integer

Values: `[0...2^31-1]`

Default: `0`

Usage Guidelines: This parameter decrements on every job execution until its value becomes zero, then the job stops. The value of this parameter initializes whenever the job scheduler is changed or restarted, and its initial value is set from the parameter `/par/sess/job/<N>/count`.

Command Examples: Check remaining executions for job 4.
```
print,/par/sess/stat/count/4
```
Or
```
print,sess/stat/count/4
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Next job to execute

`[/par/]sess/stat/job`

Description: Reports the name of the job to be executed next.

Access: read-only

Type: integer

Values: `[-1...15]`
`-1` – no job to execute.

Default: `-1`

Usage Guidelines: N/A

Command Examples: Query the name of the next job.
```
print,/par/sess/stat/job
```
Or
```
print,sess/stat/job
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Time information for current and next jobs

`[/par/]sess/stat/time`

Description: Reports the time when the next job will run, the current time, and how much time is left for the next job.

Access: read-only

Type: list

Values: `{next,curr,delta}`

Default: `{__d__h__m__s, DDdHHhMMmSSs,__d__h__m__s}`

Usage Guidelines: N/A

Command Examples: Query the time information.
`print,/par/sess/stat/time`
Or
`print,sess/stat/time`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Execution time for next job

`[/par/]sess/stat/time/next`

Description: Reports the execution time for the next job.

Access: read-only

Type: time_spec

Values: `DDdHHhMMmSSs`

Default: `__d__h__m__s`

Usage Guidelines: N/A

Command Examples: Query the execution time for the next job.
`print,/par/sess/stat/time/next`
Or
`print,sess/stat/time/next`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Current time

`[/par/]sess/stat/time/curr`

Description:	Reports the current time.
Access:	read-only
Type:	time_spec
Values:	`DDdHHhMMmSSs`
Default:	`N/A`
Usage Guidelines:	N/A
Command Examples:	Query the current time. `print,/par/sess/stat/time/curr` Or `print,sess/stat/time/curr`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Time left to next job

`[/par/]sess/stat/time/delta`

Description:	Reports the time left to the execution of the next job.
Access:	read-only
Type:	time_spec
Values:	`DDdHHhMMmSSs`
Default:	`N/A`
Usage Guidelines:	If there is no next job, the parameter will report `__d__h__m__s`.
Command Examples:	Query the time left fo the next job. `print,/par/sess/stat/time/delta` Or `print,sess/stat/time/delta`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Notebook Parameters

Notebook allows the user to store arbitrary information to the receiver and retrieve it later if necessary.

Notes

`[/par/]note`

Description:	Reports the contents and change flags of notes.
Access:	read-only
Type:	list
Values:	`{{str0,...,str7},{cf0,...,cf7}}`
Default:	`{{"","","","","","","",""},{y,y,y,y,y,y,y,y}}`
Usage Guidelines:	N/A
Command Examples:	Query the information about notes. `print,/par/note` Or `print,note`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Notes contents

`[/par/]note/str`

Description:	Reports the contents of notes.
Access:	read-only
Type:	array of seven strings
Values:	up to 64 characters in a string
Default:	`{"","","","","","","","","}`
Usage Guidelines:	N/A
Command Examples:	Query the contents of notes. `print,/par/note/str` Or `print,note/str`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Content of specified note

`[/par/]note/str/<N>`

Description: Reports the content of the specified note.

Access: read & write

Type: string

Values: up to 64 characters

Default: `{"","","","","","","","","}`

Usage Guidelines:

- When setting a string for note N, the corresponding flag `/par/note/nv/N` is cleared (set to "n") to indicate that this string is set by the user, not read from the NVRAM.
- `N` ranges from 0 to 7.

Command Examples:

Set the content of note 5.

```
%%set,/par/note/str/5,"HiPer"
```

Or

```
set,note/str/5,"HiPer"
```

Query the content of note 5.

```
print,/par/note/str/5
```

Or

```
print,note/str/5
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Change flags

`[/par/]note/nv`

Description:	Reports the change flags for notes.
Access:	read-only
Type:	array of seven boolean values
Values:	`{y|n,...y|n}`
Default:	`{y,y,y,y,y,y,y,y}`
Usage Guidelines:	When the receiver starts up, it reads all the notes from its NVRAM and sets all the values in this array to "y". When the user enters some contents to the strings, the corresponding NVRAM flag is cleared (set to "n").
Command Examples:	Query the change flags. `print,/par/note/nv` Or `print,note/nv`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Raw Data Management and Signal Processing Parameters

Raw measurement elevation masks

`[/par/]out/elm/<stream>`

Description:	Sets the elevation masks for TPS messages output to a stream.
Access:	read & write
Type:	integer
Values:	`[-90...90]`; degrees
Default:	`5`
Usage Guidelines:	Data will not be output to `stream` for the satellites whose elevations are less than the specified mask.
Command Examples:	Set the elevation mask to 15 degrees for the messages output to the current log file "a". `%%set,/par/out/elm/cur/file/a,15` Or `set,out/elm/cur/file/a,15` Query the content status. `print,/par/out/elm/cur/file/a` Or `print,out/elm/cur/file/a`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Number of epochs output to stream

`[/par/]out/epochs/<stream>`

Description: Reports the number of epochs output to the specified stream.

Access: read-only

Type: integer

Values: `cur_value`

Default: `0`

Usage Guidelines:
- The number resets to zero once the RT message starts outputting to `stream` where no RT message is already being output.
- The number increments every time a new RT message is output to `stream`.

Command Examples: Query the number of epochs output to the current log file "a".
```
print,/par/out/epochs/cur/file/a
```
Or
```
print,out/epochs/cur/file/a
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Number of messages output to stream

`[/par/]out/<stream>&count`

Description: Reports the number of messages being output to the specified stream.

Access: read-only

Type: integer

Values: `[0...49]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Query the number of messages output to the current log file "a".
```
print,/par/out/cur/file/a&count
```
Or
```
print,out/cur/file/a&count
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GPS satellites used for signal acquisition

`[/par/]lock/gps/sat`

Description: Enables and disables the receiver to track GPS satellites.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: After you issue the corresponding `print` command, the receiver will output a comma-delimited sequence of `y`'s and/or `n`'s whose positions in the sequence will indicate the PRN numbers of the GPS satellites allowed/not allowed for signal acquisition, respectively.

Command Examples: Enable tracking of GPS satellites.

```
%%set,/par/lock/gps/sat,y
```

Or

```
set,lock/gps/sat,y
```

Query the tracking status.

```
print,/par/lock/gps/sat
```

Or

```
print,lock/gps/sat
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specific GPS satellite used for signal acquisition

`[/par/]lock/gps/sat/<N>`

Description: Enables and disables the receiver to track a specific GPS satellite.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: `N` denotes an integer number from 1 to 32.

Command Examples: Enable tracking GPS satellite with PRN 17.

```
%%set,/par/lock/gps/sat/17,y
```

Or

```
set,lock/gps/sat/17,y
```

Query the tracking status of GPS PRN 17.

```
print,/par/lock/gps/sat/17
```

Or

```
print,lock/gps/sat/17
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GLONASS satellites used for signal acquisition by frequency channel numbers (FCN)

`[/par/]lock/glo/fcn`

Description: Enables and disables the receiver to track GLONASS satellites.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: After you issue the corresponding `print` command, the receiver will output a comma-delimited sequence of `y`'s and/or `n`'s whose positions in the sequence will indicate FCN of the GLONASS satellites allowed/not allowed for signal acquisition, respectively.

Command Examples: Enable tracking of GLONASS satellites.

```
%%set,/par/lock/glo/fcn,y
```

Or

```
set,lock/glo/fcn,y
```

Query the tracking status.

```
print,/par/lock/glo/fcn
```

Or

```
print,lock/glo/fcn
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specific GLONASS satellite used for signal acquisition by FCN

`[/par/]lock/glo/fcn/<N>`

Description: Enables and disables the receiver to track a specific GLONASS satellite.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: `N` denotes an integer number from 1 to 24 for firmware 3.1p3 or older and from -7 to +13 for firmware 3.2 or newer.

Command Examples: Enable tracking GLONASS satellite with FCN 6.

```
%%set,/par/lock/glo/fcn/6,y
```

Or

```
set,lock/glo/fcn/6,y
```

Query the tracking status of GLONASS FCN 6.

```
print,/par/lock/glo/fcn/6
```

Or

```
print,lock/glo/fcn/6
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in the first release of firmware.
- The range of values was changed starting with firmware 3.2. Refer to Usage Guidelines above.

See Also: N/A

Tracking GPS L2C-L and L2C-M codes

`[/par/]lock/gps/l2c`

Description: Configures the receiver to track GPS L2C-L and L2C-M codes.

Access: read & write

Type: array of two boolean values, where the first value is for L2C-L and the second value is for L2C-M

Values: `off | on`

- `off` – the receiver will not track GPS L2C-L and L2C-M codes.
- `on` – the receiver will track GPS L2C-L and L2C-M codes.

Default: `{off,off}`

Usage Guidelines: The values can be set to `{on,on}` or `{off,off}`.

Command Examples: Enable the receiver to track GPS L2C-L and L2C-M codes.

```
set,/par/lock/gps/l2c,{on,on}
```

Or

```
set,lock/gps/l2c,{on,on}
```

Query the current status.

```
print,/par/lock/gps/l2c
```

Or

```
print,lock/gps/l2c
```

Receiver Availability: This parameter is available for the G3-based receivers.

History:

- This parameter was introduced in firmware 3.1.
- The separate values for L2C-L and L2C-M tracking were implemented in firmware 3.2.

See Also: N/A

Tracking GLONASS C/A L2

`[/par/]lock/glo/l2c`

Description: Configures the receiver to track GLONASS C/A L2 signal.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not track GLONASS C/A L2.
- `on` – the receiver will track GLONASS C/A L2.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable the receiver to track GLONASS C/A L2.

```
set,/par/lock/glo/l2c,on
```

Or

```
set,lock/glo/l2c,on
```

Query the current status.

```
print,/par/lock/glo/l2c
```

Or

```
print,lock/glo/l2c
```

Receiver Availability: This parameter is available for the G3-based receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Managing GPS P/L1 and P/L2 channel tracking

`[/par/]lock/gps/pcode`

Description: Configures the receiver to track GPS P-codes on L1 and L2.

Access: read & write

Type: array of two boolean values, where the first value is for P/L1 and the second value is for P/L2

Values: `off | on`

- `off` – the receiver will track GPS C/A L1 data only.
- `on` – the receiver will track GPS C/A L1 and P L1/L2 signals.

Default: `{on,on}`

Usage Guidelines:

- The values can be set to `{on,on}` or `{off,off}`.
- Use this parameter as a replacement for the obsolete parameter `/par/lock/pcode`.

Command Examples: Disable the receiver to track GPS P/L1 and P/L2 signals.

```
set,/par/lock/gps/pcode,{off,off}
```

Or

```
set,lock/gps/pcode,{off,off}
```

Query the current status.

```
print,/par/lock/gps/pcode
```

Or

```
print,lock/gps/pcode
```

Receiver Availability: This parameter is unavailable in the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Managing GLONASS P/L1 and P/L2 channel tracking

`[/par/]lock/glo/pcode`

Description: Configures the receiver to track GLONASS P-codes on L1 and L2.

Access: read & write

Type: array of two boolean values, where the first value is for P/L1 and the second value is for P/L2

Values: `off | on`

- `off` – the receiver will track GLONASS C/A L1 data only.
- `on` – the receiver will track GLONASS C/A L1 and P L1/L2 signals.

Default: `{on,on}`

Usage Guidelines:

- The values can be set to `{on,on}` or `{off,off}`.
- Use this parameter as a replacement for the obsolete parameter `/par/lock/pcode`.

Command Examples: Disable the receiver to track GLONASS P/L1 and P/L2 signals.

```
set,/par/lock/glo/pcode,{off,off}
```

Or

```
set,lock/glo/pcode,{off,off}
```

Query the current status.

```
print,/par/lock/glo/pcode
```

Or

```
print,lock/glo/pcode
```

Receiver Availability: This parameter is unavailable in the GMS-2.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

GPS PRNs codes vs. GPS satellite numbers mapping

`[/par/]/gps/prn`

Description: Reports GPS PRNs.

Access: read-only

Type: array of 32 integer values

Values: `{prn1,prn2,...,prn32}`, where each `prn` ranges from 1 to 37

Default: `{1,2,...,32}`

Usage Guidelines: If you send the command `print,/par/gps/prn:on`, you will get a clear correspondence between GPS satellite numbers and PRNs codes.

Command Examples: Report GPS PRNs.
`print,/par/gps/prn`
Or
`print,gps/prn`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specific satellite number to GPS PRN code

`[/par/]/gps/prn/<N>`

Description: Assigns a GPS PRN code to a specific satellite number.

Access: write-only

Type: integer

Values: `[1...37]`

Default: N/A

Usage Guidelines:
- `N` denotes an integer number from 1 to 32.
- Use this parameter for acquiring a pseudolite's signals. The pseudolite must be associated with a GPS satellite number.

Command Examples: Instruct the receiver that the GPS satellite with number 32 will have PRN code 35.
`%%set,/par/gps/prn/32,35`
Or
`set,gps/prn/32,35`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Almanac status for all GPS satellites

`[/par/]alm/gps`

Description: Reports the almanac availability status for all GPS satellites.

Access: read-only

Type: array of 32 boolean values

Values: `{y|n,...,y|n}`

Default: N/A

Usage Guidelines: After you issue the `print` command, the receiver will output a comma-delimited sequence of `y`'s and/or `n`'s whose positions in the sequence will indicate the almanac data are available/unavailable for the corresponding GPS satellite.

Command Examples: Query the almanac status for GPS.
`print,/par/alm/gps`
Or
`print,alm/gps`

More informative variant with PRN numbers of the GPS satellites.
`print,/par/alm/gps:on`
Or
`print,alm/gps:on`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Almanac status for specific GPS satellite

`[/par/]alm/gps/<N>`

Description: Reports the almanac availability status for a specific GPS satellite.

Access: read-only

Type: boolean

Values: `y|n`

- `y` – the almanac data are available for the satellite.
- `n` – the almanac data are unavailable for the satellite.

Default: N/A

Usage Guidelines: In this parameter, `N` denotes a GPS PRN ranging from 1 to 32.

Command Examples: Query the almanac status for GPS PRN 19.

```
print,/par/alm/gps/19
```

Or

```
print,alm/gps/19
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Almanac status for all GLONASS satellites

`[/par/]alm/glo`

Description: Reports the almanac availability status for all GLONASS satellites.

Access: read-only

Type: array of 24 boolean values

Values: `{y|n,...,y|n}`

Default: N/A

Usage Guidelines: After you issue the `print` command, the receiver will output a comma-delimited sequence of `y`'s and/or `n`'s whose positions in the sequence will indicate the almanac data are available/unavailable for the corresponding GLONASS satellite.

Command Examples: Query the almanac status for GLONASS.
`print,/par/alm/glo`
Or
`print,alm/glo`

More informative variant with FCNs of the GLONASS satellites.
`print,/par/alm/glo:on`
Or
`print,alm/glo:on`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Almanac status for specific GLONASS satellite

`[/par/]alm/glo/<N>`

Description:	Reports the almanac availability status for a specific GLONASS satellite.
Access:	read-only
Type:	boolean
Values:	`y|n` • `y` – the almanac data are available for the satellite. • `n` – the almanac data are unavailable for the satellite.
Default:	N/A
Usage Guidelines:	In this parameter, `N` denotes a GLONASS FCN ranging from 1 to 24.
Command Examples:	Query the almanac status for GLONASS FCN 5. `print,/par/alm/glo/5` Or `print,alm/glo/5`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Minimum number of GNSS satellites for data output

`[/par/]out/minsvs/<stream>`

Description: Defines the minimum number of GNSS satellites used for data output to the given stream.

Access: read & write

Type: integer

Values: `[0...255]`

Default: `0`

Usage Guidelines: The receiver will not output data to the given stream as long as the number of GNSS satellites whose elevations exceed the specified elevation mask is fewer than this parameter.

Command Examples: Output data to the current log file "a" only if the receiver tracks at least four GNSS satellites.

```
%%set,/par/out/minsvs/cur/file/a,4
```

Or

```
set,lock/glo/fcn/6,y
```

Query the parameter's value for serial port C.

```
print,/par/out/minsvs/dev/ser/c
```

Or

```
print,out/minsvs/dev/ser/c
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Elevation mask for satellites tracking

`[/par/]lock/elm`

Description: Defines the minimum angle at which the receiver will track satellites.

Access: read & write

Type: integer

Values: `[-90...90]`; degrees

Default: `5`

Usage Guidelines: The receiver will not track satellites below this elevation mask, provided the parameter `/par/lock/notvis` is set to `off`.

Command Examples: Set the elevation mask for tracking to 15 degrees.

```
%%set,/par/lock/elm,15
```

Or

```
set,lock/elm,15
```

Query the parameter's value.

```
print,/par/lock/elm
```

Or

```
print,lock/elm
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver's reaction to unhealthy satellites and satellites with elevation angles smaller than the specified mask

`[/par/]lock/notvis`

Description: Programs the receiver's reaction to unhealthy satellites and satellites with elevation angles smaller than the specified mask in the parameter `/par/lock/elm`.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not acquire satellites whose elevations, as calculated by using the current receiver position and the almanac data, are smaller than the specified mask and will not acquire satellites whose health status flags, as follows from the almanac data, indicate that the satellites are unhealthy.
- `on` – the receiver will acquire satellites with elevations smaller than the specified mask and will acquire satellites whose health status flags indicate that the satellites are unhealthy.

Default: `off`

Usage Guidelines:

- The main purpose of the parameter is to reduce the receiver's processor load (to some extent). This is achieved by putting the free channels to idle mode when the number of visible satellites is less than the number of channels available in the receiver.
- If the parameter is set to `on`, the free channels will too be forced to keep searching for satellites even if these satellites are known to be below the horizon.

Command Examples: Set the parameter to `off`.

```
%%set,/par/lock/notvis,off
```

Or

```
set,lock/notvis,off
```

Query the parameter's value.

```
print,/par/lock/notvis
```

Or

```
print,lock/notvis
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in the first release of firmware.
- Starting with firmware 3.0, this parameter is always set to `off` for G3-based receivers.
- Starting with firmware 3.2, this parameter is always set to `on` for G3-based receivers and it is a read-only parameter for these receivers.

See Also: N/A

Raw measurement update rate

`[/par/]raw/msint`

Description: Determines the rate of receiver generating pseudoranges, carrier phases and some other GNSS observables.

Access: read & write

Type: integer

Values:
- `[50,100,150,...,5000]` for all receivers except TG3
- `[50,100,150,...,500000]` for TG3 with board revision older than 3.0
- `[10,20,30,...,500000]` for TG3 with board revision 3.0 or newer

expressed in milliseconds

Default: `cur_value`

Usage Guidelines: Special care should be taken when using the 50 ms recording interval. Unwary users often get bad tps files when logging receiver data at 20 Hz. In such files, a considerable percent of epochs may be either lost altogether or contain corrupt data. The main factors that can result in tps files having lost epochs and corrupt messages at 20 Hz are as follows:

- All or some of the options *Anti-jamming suppressor*, *Code multipath reduction*, *Carrier multipath reduction*, *Co-Op tracking* are on. At 20 Hz, using these options will lead to peak processor loading. If this occurs, the time allocated to the receiver to record the current data block into the flash memory may be insufficient. Thus lost epochs and corrupt messages in the tps file.
- Large number of satellites acquired. Amount of data recorded into the memory will be on the average proportional to the number of satellites tracked.
- Large number of message types enabled for the log file. There are 40 messages used in the default set of messages currently. Note that, in many cases, the user may significantly decimate this message set without any harm to the task accomplished (and add some new messages to the reduced set if required). Remember, however, that some of the message types are critical for Topcon Tools™ and other TPS post-processing software to be able to import and process tps files correctly. Great care should be exercised with respect to the *time messages* such as RD and RT. For your reference, below is the list of TPS messages that Topcon Tools™ will identify when parsing an imported tps file: JP, MF, PM, ==, RD, ~~, TO, UO, PO, VE, PV, DP, SI, NN, EL, RC, R1, R2, rc, r1, r2, 1R, 2R, 1r, 2r, CC, C1, C2, cc, c1, c2, PC, P1, P2, pc, p1, p2, CP, 1P, 2P, cp, 1p, 2p, DC, D2, TC, FC, F1, F2, EC, E1, E2, NE, GE, WE, GA, NA, IO, XA, XB, SS, SE, rM, rE,rV.

Command Examples: Set the parameter to 50 ms.

```
%%set,/par/raw/msint,50
```

Or

```
set,raw/msint,50
```

Query the parameter's value.

```
print,/par/raw/msint
```

Or

```
print,raw/msint
```

Receiver Availability: This parameter is available in all receivers.

`[/par/]raw/msint` *(Continued)*

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Position update rate

`[/par/]pos/msint`

Description: Determines the period of receiver updating the position.

Access: read & write

Type: integer

Values:
- `[50,100,150,...,5000]` for all receivers and TG3 with board revision older than 3.0
- `[10,20,30,...,5000]` for TG3 with board revision 3.0 or newer

expressed in milliseconds

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Set the parameter to 1000 ms.
```
%%set,/par/pos/msint,1000
```
Or
```
set,pos/msint,1000
```

Query the parameter's value.
```
print,/par/pos/msint
```
Or
```
print,pos/msint
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Current raw measurement update rate

`[/par/]raw/curmsint`

Description:	Reports the raw measurement update rate actually being used.
Access:	read-only
Type:	integer
Values:	`[10...500000]`; milliseconds
Default:	`cur_value`
Usage Guidelines:	Although you can formally set `/par/raw/msint` to arbitrary multiples of 10 or 50 (depending on the receiver), the receiver may need to adjust this original parameter in order to make it consistent, first, with the internal processing cyclogram (time grid) and, second, with the receiver options. The adjusted setting is stored to the described parameter `/par/raw/curmsint`.
Command Examples:	Query the parameter's value. `print,/par/raw/curmsint` Or `print,raw/curmsint`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Current position update rate

`[/par/]pos/curmsint`

Description:	Reports the current position update rate actually being used.
Access:	read-only
Type:	integer
Values:	`[10...5000]`; milliseconds
Default:	`cur_value`
Usage Guidelines:	Although you can formally set `/par/pos/msint` to arbitrary multiples of 10 or 50 (depending on the receiver), the receiver may need to adjust this original parameter in order to make it consistent, first, with the internal processing cyclogram (time grid) and, second, with the receiver options. The adjusted setting is stored to the described parameter `/par/pos/curmsint`.
Command Examples:	Query the parameter's value. `print,/par/pos/curmsint` Or `print,pos/curmsint`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Pseudorange smoothing interval

`[/par/]raw/smi`

Description: Specifies the interval over which pseudoranges are smoothed using the corresponding carrier phase measurements.

Access: read & write

Type: integer

Values: `[0...900]`; seconds

- `0` – the receiver will not use carrier phase measurements for smoothing pseudoranges. In this case the pseudorange noise error will depend only on the corresponding DLL bandwidth (see the parameter `/par/raw/gdl/band`).
- `[1...900]` – the receiver will smooth pseudoranges based on a Kalman filter whose time constant is set to the specified number of seconds.

Default: `300`

Usage Guidelines: N/A

Command Examples: Set the parameter to 350 seconds.

```
%%set,/par/raw/smi,350
```

Or

```
set,raw/smi,350
```

Query the parameter's value.

```
print,/par/raw/smi
```

Or

```
print,raw/smi
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Pseudorange smoothing filter type

`[/par/]raw/sm/filt`

Description: Specifies the filter type used for pseudorange smoothing.

Access: read & write

Type: integer

Values: `0 | 1`

Default: `1`

Usage Guidelines: It is not recommended that you change this parameter's default value unless DGPS mode provides poor accuracy.

Command Examples: Set the parameter to 1.

```
%%set,/par/raw/sm/filt,1
```

Or

```
set,raw/sm/filt,1
```

Query the parameter's value.

```
print,/par/raw/sm/filt
```

Or

```
print,raw/sm/filt
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Doppler smoothing interval

`[/par/]raw/dopp/smi`

Description: Governs the doppler computation algorithm.

Access: read & write

Type: integer

Values: `0 | 1 | 2`

- `0` – Receiver outputs raw (unsmoothed) Doppler. Instantaneous yet rather noisy Doppler measurements.
- `1` – Doppler is computed using two consecutive carrier phase measurements, CarPhase[i] and CarPhase[i-1], where i stands for the current epoch. Such Doppler measurements are less noisy than in the first case.
- `2` – Doppler is computed using only three consecutive carrier phase measurements, CarPhase[i], CarPhase[i-1] and CarPhase[i-2], where i stands for the current epoch. Doppler measurements obtained in this mode are least noisy.

Default: `2`

Usage Guidelines: N/A

Command Examples: Set the parameter to 1.

```
%%set,/par/raw/dopp/smi,1
```

Or

```
set,raw/dopp/smi,1
```

Query the parameter's value.

```
print,/par/raw/dopp/smi
```

Or

```
print,raw/dopp/smi
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Bandwidth of Doppler smoothing filter

`[/par/]raw/dopp/band`

Description: Governs the bandwidth of Doppler (velocity) smoothing filter.

Access: read & write

Type: float

Values: `[0.2...10.0]`; Hz

Default: `3.0`

Usage Guidelines: By narrowing the bandwidth of this filter, you reduce the noise error of velocity estimation and magnify the dynamic error of velocity estimation. By making the bandwidth wider, you, on the contrary, magnify the noise error of velocity estimation and reduce the dynamic error of velocity estimation.
The default value is an optimal compromise that will allow you to have minimized noise and dynamic errors at the same time. However, you may change the default value and get the optimized results for your application.

Command Examples: Set the parameter to 5 Hz.

```
%%set,/par/raw/dopp/band,5
```

Or

```
set,raw/dopp/band,5
```

Query the parameter's value.

```
print,/par/raw/dopp/band
```

Or

```
print,raw/dopp/band
```

Receiver Availability: This parameter is available in TG3- and G3-based receivers.

History: This parameter was introduced in firmware 3.0.

See Also: N/A

Smart Doppler smoothing

`[/par/]raw/dopp/smsm`

Description: Enables smart Doppler smoothing.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable smoothing.

```
%%set,/par/raw/dopp/smsm,on
```

Or

```
set,raw/dopp/smsm,on
```

Query the parameter's value.

```
print,/par/raw/dopp/smsm
```

Or

```
print,raw/dopp/smsm
```

Receiver Availability: This parameter is available in TG3- and G3-based receivers.

History: This parameter was introduced in firmware 3.0.

See Also: N/A

Nominal ionosphere correction smoothing interval

`[/par/]raw/iono/smi`

Description: Specifies the nominal interval over which raw ionospheric corrections are smoothed (assuming the receiver has been working for some time and has already obtained enough raw ionospheric corrections to perform such smoothing).

Access: read & write

Type: integer

Values: `[0...900]`; seconds

Default: `60`

Usage Guidelines:

- The current ionosphere smoothing interval will vary in time. After you switch your receiver on, the current smoothing interval will be growing from zero to the nominal value as new raw ionospheric corrections are computed. Once the nominal value is reached, this smoothing interval will be fixed.
- The smoothing filter is a simple n-point running average.

Command Examples: Set the parameter to 120 seconds.

```
%%set,/par/raw/iono/smi,120
```

Or

```
set,raw/iono/smi,120
```

Query the parameter's value.

```
print,/par/raw/iono/smi
```

Or

```
print,raw/iono/smi
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Minimum ionosphere correction smoothing interval

`[/par/]raw/iono/minsmi`

Description: Specifies the minimum smoothing interval for the receiver to filter raw ionospheric corrections before they can be used in position computation.

Access: read & write

Type: integer

Values: `[0...900]`; seconds

Default: `30`

Usage Guidelines: In practice, the minimum smoothing interval is always smaller than the nominal one.

Command Examples: Set the parameter to 60 seconds.

```
%%set,/par/raw/iono/minsmi,60
```

Or

```
set,raw/iono/minsmi,60
```

Query the parameter's value.

```
print,/par/raw/iono/minsmi
```

Or

```
print,raw/iono/minsmi
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

C/A code multipath reduction

`[/par/]raw/corr/ca/code`

Description: Allows the reduction of C/A code phase multipath.

Access: read & write

Type: enumerated

Values: `normal | mpnew`

- `normal` – the multipath reduction is disabled.
- `mpnew` – the multipath reduction is enabled.

Default: `normal`

Usage Guidelines: The value `mpnew` is recommended for receivers (Reference and Rover) in DGPS mode on jobsites that contain multipath.

Command Examples: Enable code multipath reduction.

```
%%set,/par/raw/corr/ca/code,mpnew
```

Or

```
set,raw/corr/ca/code,mpnew
```

Query the parameter's value.

```
print,/par/raw/corr/ca/code
```

Or

```
print,raw/corr/ca/code
```

Receiver Availability: This parameter is available in all receivers where the option _MPR is enabled.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

C/A carrier phase multipath reduction

`[/par/]raw/corr/ca/carrier`

Description: Allows the reduction of C/A carrier phase multipath.

Access: read & write

Type: enumerated

Values: `normal | mpnew`

- `normal` – the multipath reduction is disabled.
- `mpnew` – the multipath reduction is enabled.

Default: `normal`

Usage Guidelines: The value `mpnew` is recommended for receivers (Reference and Rover) in RTK mode on jobsites that contain multipath.

Command Examples: Enable carrier multipath reduction.

```
%%set,/par/raw/corr/ca/carrier,mpnew
```

Or

```
set,raw/corr/ca/carrier,mpnew
```

Query the parameter's value.

```
print,/par/raw/corr/ca/carrier
```

Or

```
print,raw/corr/ca/carrier
```

Receiver Availability: This parameter is available in all receivers where the option _MPR is enabled.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Anti-jamming

`[/par/]ajm/mode`

Description: Enables suppression of narrow-band interference.

Access: read & write

Type: enumerated

Values: `gps | glo | l1 | l2 | auto | off | manual`

- `gps` – Anti-jamming is effective only for GPS L1 and L2.
- `glo` – Anti-jamming is effective only for GLONASS L1 and L2.
- `l1` – Anti-jamming is effective only for GLONASS L1 and GPS L1.
- `l2` – Anti-jamming is effective only for GLONASS L2 and GPS L2.
- `auto` – The receiver automatically detects which bands are affected by interference and launches the jamming suppressor there.
- `off` – Anti-jamming is off.
- `manual` – You will select the band manually using the parameters `/par/ajm/chan...` described below.

Default: `off`

Usage Guidelines: Use the following assessments to determine the presence of interference:

- When the number of tracked satellites is fewer (by 2 or greater) than the number of satellites in view.
- When the signal-to-noise ratio (C/N0) in the C/A channel for the satellites having elevations above 30 degrees does not exceed 40 dB/Hz.
- When the information about interferences indicates that the Jamming Suppressor has detected interference signals within the specified band. Also, the strength of aggregate in-band interference is characterized as high or hard.
- Use the JI message to view this information.

Command Examples: Enable anti-jamming on GPS L1/L2.

```
%%set,/par/ajm/mode,gps
```

Or

```
set,ajm/mode,gps
```

Query the parameter's value.

```
print,/par/ajm/mode
```

Or

```
print,ajm/mode
```

Receiver Availability:

- This parameter is available for JPS_3, JPS_4_x, Euro_x, E_GGD, and EGGDT_x receiver board versions, where "x" stands for the board revision number.
- This parameter is also available for EG3_x boards and NET-G3 receivers starting with firmware 3.3.
- The option _AJM must be enabled. The EG3_x boards and NET-G3 receivers may also require hardware modification.

`[/par/]ajm/mode` *(Continued)*

History:
- This parameter was introduced in the first release of firmware.
- EG3_x boards and NET-G3-based receivers support this feature starting with firmware 3.3.
- The `manual` value was added in firmware 3.3.

See Also: N/A

Configuring anti-jamming for GPS in manual mode

`[/par/]ajm/chan/gps/<l1|l2|l5>`

Description: Enables suppression of narrow-band interference for GPS L1/L2/L5.

Access: write-only

Type: boolean

Values: `off | on`
- `off` – Anti-jamming is deactivated for the corresponding channel.
- `on` – Anti-jamming is activated the corresponding channel.

Default: `off`

Usage Guidelines: This parameter is effective only if the parameter `/par/ajm/mode` is set to `manual`.

Command Examples: Enable anti-jamming on GPS L1.
```
%%set,/par/ajm/chan/gps/l1,on
```
Or
```
set,ajm/chan/gps/l1,on
```

Enable anti-jamming on GPS L5.
```
%%set,/par/ajm/chan/gps/l5,on
```
Or
```
set,ajm/chan/gps/l5,on
```

Query the parameter's value on all GPS channels.
```
print,/par/ajm/chan/gps
```
Or
```
print,ajm/chan/gps
```

Query the parameter's value on GPS L2.
```
print,/par/ajm/chan/gps/l2
```
Or
```
print,ajm/chan/gps/l2
```

Receiver Availability:
- This parameter is available in the EG3_x board and NET-G3 receiver.
- The option _AJM must be enabled in the above receivers.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Configuring anti-jamming for GLONASS in manual mode

`[/par/]ajm/chan/gln/<l1|l2>`

Description: Enables suppression of narrow-band interference for GLONASS L1/L2.

Access: write-only

Type: boolean

Values: `off | on`

- `off` – Anti-jamming is deactivated for the corresponding channel.
- `on` – Anti-jamming is activated the corresponding channel.

Default: `off`

Usage Guidelines: This parameter is effective only if the parameter `/par/ajm/mode` is set to `manual`.

Command Examples: Enable anti-jamming on GLONASS L1.

```
%%set,/par/ajm/chan/gln/l1,on
```

Or

```
set,ajm/chan/gln/l1,on
```

Enable anti-jamming on GLONASS L2.

```
%%set,/par/ajm/chan/gln/l2,on
```

Or

```
set,ajm/chan/gln/l2,on
```

Query the parameter's value on all GLONASS channels.

```
print,/par/ajm/chan/gln
```

Or

```
print,ajm/chan/gln
```

Query the parameter's value on GLONASS L2.

```
print,/par/ajm/chan/gln/l2
```

Or

```
print,ajm/chan/gln/l2
```

Receiver Availability:

- This parameter is available in the EG3_x board and NET-G3 receiver.
- The option _AJM must be enabled in the above receivers.

History: This parameter was introduced in firmware 3.3.

See Also: N/A

Oscillator Offset Reduction

`[/par/]osc/mode`

Description: Governs the behavior of the internal crystal oscillator.

Access: read & write

Type: enumerated

Values: `off | locked | tied`

- `off` – the receiver will not adjust the internal oscillator.
- `locked` – the receiver will adjust the internal oscillator's frequency until the measured frequency offset is reduced to zero. By using the incoming satellite signals, the receiver will force the internal oscillator to generate a very stable 20 MHz frequency signal. This frequency output is available via the corresponding receiver output pin.
- `tied` – the receiver will adjust both the internal oscillator's frequency until the measured frequency offset is reduced to zero and the internal clock until it gets fully synchronized with the specified reference time scale (for example, GPS time).

Default: `off`

Usage Guidelines:

- After you have issued the command `set,osc/mode,locked`, it may take the receiver up to a minute to adjust the internal oscillator frequency to the true (nominal) value. The transient period is even longer when running the command `set,osc/mode,tied`.
- If you change this parameter from `locked` or `tied` to `off`, the internal oscillator frequency will rebound to its quiescent value abruptly, which may result in temporary loss of lock to satellites.
- Setting this parameter to `locked` only guarantees that the receiver's 20 MHz output will have high long-term stability, not necessarily high short-term stability (the latter depends on the internal oscillator type). However, there is a way to guarantee that both of these characteristics are good even when using the internal crystal oscillator. It can be done by switching the receiver to *common loop* mode.

Command Examples:

Set the oscillator reduction mode to `tied`.

```
%%set,/par/osc/mode,tied
```

Or

```
set,osc/mode,tied
```

Query the parameter's value.

```
print,/par/osc/mode
```

Or

```
print,osc/mode
```

Receiver Availability:

- This parameter is available in the receivers based on E_GGD_x, EGGDT_x, and EG3_x boards.
- The option _FRO must be enabled in the above receives.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Loop Parameters

This section contains the parameters for configuring the receiver's capability of searching, acquiring and tracking the GNSS satellite signals.

You should distinguish between *guiding* and *guided* loops. In TPS receivers, C/A PLLs are referred to as guiding loops whereas the other lock loops (specifically, C/A DLLs, P/L1 PLLs, P/L1 DLLs, P/L2 PLLs, P/L2 DLLs) are called guided loops.

C/A PLL bandwidth

`[/par/]raw/pll/band`

Description: Specifies C/A PLL bandwidths in guiding loops and serves both GPS and GLONASS satellites.

Access: read & write

Type: float

Values: `[2.0...50.0]`; Hz

Default: `25.0`

Usage Guidelines: Care should be taken that the setting used does not result in receiver malfunction. In most scenarios, values set in the range from 10 to 50 Hz are usually safe.
The following conditions prevents you from using more narrow (less than 5 Hz) PLL bandwidths in guiding loops.

- ionosphere fluctuations
- quick multipath
- the internal oscillator's noise often does not allow the use of bandwidths less than 5 Hz

In many kinematic applications you may have to use wider PLL bandwidths to account for the receiver dynamic.

Command Examples: Set PLL bandwidths to 35 Hz.
`set,/par/raw/pll/band,35`
Or
`set,raw/pll/band,35`

Query the current status.
`print,/par/raw/pll/band`
Or
`print,raw/pll/band`

Receiver Availability: This parameter is available in all TPS receivers.

History: This parameter was introduced in the first release of firmware.

`[/par/]raw/pll/band` *(Continued)*

See Also: N/A

C/A delay loop bandwidth

`[/par/]raw/cagdl/band`

Description: Specifies the bandwidth of the receiver's C/A delay lock loop.

Access: read & write

Type: float

Values: `[0.1...50.0]`; Hz

Default: `0.8`

Usage Guidelines: Care should be taken that the setting used doesn't result in receiver malfunction. You do not usually experience any problems with this parameter as long as its value lies between 0.2 Hz and 5 Hz.
The following limitations prevents you from using more narrow bandwidths in guided loops:

- ionosphere fluctuations
- quick multipath

Command Examples: Set the receiver's C/A delay lock loop to 1.0 Hz.
`set,/par/raw/cagdl/band,1.0`
Or
`set,raw/cagdl/band,1.0`

Query the current status.
`print,/par/raw/cagdl/band`
Or
`print,raw/cagdl/band`

Receiver Availability: This parameter is available in all TPS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P/L1 and P/L2 guided loops bandwidth

`[/par/]raw/gdl/band`

Description: Governs all of the guided loops but the C/A DLLs, specifically, P/L1 PLLs, P/L1 DLLs, P/L2 PLLs and P/L2 DLLs.

Access: read & write

Type: float

Values: `[0.1...10.0]`; Hz

Default: `0.5`

Usage Guidelines: Care should be taken when using values other than the default (see *Usage Guidelines* to the previous parameter).

Command Examples: Set the P/L1 and P/L2 guided loops bandwidth to 0.9 Hz.
`set,/par/raw/gdl/band,0.9`
Or
`set,raw/gdl/band,0.9`

Query the current status.
`print,/par/raw/gdl/band`
Or
`print,raw/gdl/band`

Receiver Availability: This parameter is available in all TPS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Guiding and common lock loops order

`[/par/]raw/pll/order`

Description: Specifies the order of PLLs.

Access: read & write

Type: integer

Values: `2 | 3`

Default: `2` – for TG3 board and GMS-2 receiver.
`3` – for all other boards and receivers.

Usage Guidelines: Care should be exercised when changing this parameter from the default value to 2. It is not recommended that you use a 2nd order PLL unless you are certain that the receiver's trajectory is far from being a uniformly accelerated motion, and therefore using a lower-order PLL might make a difference. Imagine you are a field operator carrying a GNSS receiver in your backpack and you need to collect raw data for your RTK project. In this and in similar cases, using a 2nd order PLL may pay off.

Command Examples: Set to use a 2nd order PLL:

```
set,/par/raw/pll/order,2
```

Or

```
set,raw/pll/order,2
```

Query the current status.

```
print,/par/raw/pll/order
```

Or

```
print,raw/pll/order
```

Receiver Availability: This parameter is available in all TPS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Order of GPS L2C and GLONASS C/A L2 PLL

`[/par/]raw/l2pll/order`

Description: Specifies the order of GPS L2C and GLONASS C/A L2 PLL.

Access: read & write

Type: integer

Values: `1 | 2`

Default: `1`

Usage Guidelines: N/A

Command Examples: Set to use a 2nd order PLL:

```
set,/par/raw/l2pll/order,2
```

Or

```
set,raw/l2pll/order,2
```

Query the current status.

```
print,/par/raw/l2pll/order
```

Or

```
print,raw/l2pll/order
```

Receiver Availability: This parameter is available for the G3-based receivers only.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Bandwidth of GPS L2C and GLONASS C/A L2 PLL

`[/par/]raw/l2pll/band`

Description:	Specifies the bandwidth of GPS L2C and GLONASS C/A L2 PLL.
Access:	read & write
Type:	float
Values:	`[0.000001...50.0]`; Hz
Default:	`2.5`
Usage Guidelines:	N/A
Command Examples:	Set the parameter to 3.0 Hz: `set,/par/raw/l2pll/band,3` Or `set,raw/l2pll/band,3` Query the current status. `print,/par/raw/l2pll/band` Or `print,raw/l2pll/band`
Receiver Availability:	This parameter is available for the G3-based receivers only.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Order of GPS L2C and GLONASS C/A L2 DLL

`[/par/]raw/l2cll/order`

Description: Specifies the order of GPS L2C and GLONASS C/A L2 DLL.

Access: read & write

Type: integer

Values: `1 | 2`

Default: `1`

Usage Guidelines: N/A

Command Examples: Set to use a 2nd order DLL:

```
set,/par/raw/l2cll/order,2
```

Or

```
set,raw/l2cll/order,2
```

Query the current status.

```
print,/par/raw/l2cll/order
```

Or

```
print,raw/l2cll/order
```

Receiver Availability: This parameter is available for the G3-based receivers only.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Bandwidth of GPS L2C and GLONASS C/A L2 DLL

`[/par/]raw/l2cll/band`

Description: Specifies the bandwidth of GPS L2C and GLONASS C/A L2 DLL.

Access: read & write

Type: float

Values: `[0.000001...10.0]`; Hz

Default: `0.4`

Usage Guidelines: N/A

Command Examples: Set the parameter to 0.8 Hz:

```
set,/par/raw/l2cll/band,0.8
```

Or

```
set,raw/l2cll/band,0.8
```

Query the current status.

```
print,/par/raw/l2cll/band
```

Or

```
print,raw/l2cll/band
```

Receiver Availability: This parameter is available for the G3-based receivers only.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Common loops mode

`[/par/]raw/clp/loops`

Description: Turns on and off the common loops mode.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – common loops mode is turned off.
- `on` – common loops mode is turned on.

Default: `off`

Usage Guidelines: When common loops mode is turned on, you may want to change either or both of the common loops bandwidth and individual loops bandwidth.

Command Examples: Turn on common loops mode.

```
set,/par/raw/clp/loops,on
```

Or

```
set,raw/clp/loops,on
```

Query the current status.

```
print,/par/raw/clp/loops
```

Or

```
print,raw/clp/loops
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Individual loops bandwidth

`[/par/]raw/clp/indband`

Description: Specifies the bandwidth of the individual loops.

Access: read & write

Type: float

Values: `[2.0…20.0]`; Hz

Default: `5.0`

Usage Guidelines: N/A

Command Examples: Set the individual loops bandwidth to 8.0 Hz.

```
set,/par/raw/clp/inband,8.0
```

Or

```
set,raw/clp/inband,8.0
```

Query the current status.

```
print,/par/raw/clp/inband
```

Or

```
print,raw/clp/inband
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Common loops bandwidth

`[/par/]raw/clp/comband`

Description: Specifies the bandwidth of the common loops.

Access: read & write

Type: float

Values: `[2.0…50.0]`; Hz

Default: `25.0`

Usage Guidelines: N/A

Command Examples: Set the common loops bandwidth to 30.0 Hz.

```
set,/par/raw/clp/comband,30.0
```

Or

```
set,raw/clp/comband,30.0
```

Query the current status.

```
print,/par/raw/clp/comband
```

Or

```
print,raw/clp/comband
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Static Co-Op tracking

`[/par/]raw/clp/static`

Description: Runs static Co-Op tracking.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receive will not use static Co-Op tracking.
- `on` – the receive will not use static Co-Op tracking.

Default: `off`

Usage Guidelines:

- After you set this parameter to `on`, thus instructing the receiver to start static Co-Op tracking, you will be able to acquire satellites with lower signal-to-noise ratios. This is achieved by reducing the number of unknowns estimated (specifically, from 4 down to 1), which is possible only for a static receiver.
- This parameter is effective only when Co-Op tracking is enabled with the command `set,raw/clp/loops,on`.
- Only enable this parameter if the receiver's antenna remains completely stationary throughout the survey. Any movement may result in losing the satellite lock.

Command Examples: Enable static Co-Op.

```
set,/par/raw/clp/static,on
```

Or

```
set,raw/clp/static,on
```

Query the current status.

```
print,/par/raw/clp/static
```

Or

```
print,raw/clp/static
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Adaptive guided loops

`[/par/]raw/gdl/adapt`

Description: Instructs the receiver to enable/disable adaptive guided loops.

Access: read & write

Type: boolean

Values: `off | on`

- off – no adaptation will be performed.
- on – the receiver will adjust the guided loops bandwidths depending on the actual strengths of the signals tracked. The weaker the signals, the narrower the bandwidths.

Default: `on`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to disable adaptive guided loops.

```
set,/par/raw/gdl/adapt,off
```

Or

```
set,raw/gdl/adapt,off
```

Query the current status.

```
print,/par/raw/gdl/adapt
```

Or

```
print,raw/gdl/adapt
```

Receiver Availability: This parameter is available in all TPS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Additional (L2-specific) common loop

`[/par/]raw/clp2/loop`

Description: Instructs the receiver to use/not to use additional common loop.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not use common loop to track L2 carrier phases.
- `on` – the receiver will use common loop to track L2 carrier phases.

Default: `on`

Usage Guidelines: In dynamic applications the measured carrier phase will also depend on antenna rotation around its axis. Abrupt antenna rotation can bring about significant additional carrier phase dynamics, which may result in poor L2 phase tracking. By setting this parameter to `on`, you ensure that the receiver effectively tracks L2 carrier phases in all channels.

Command Examples: Instruct the receiver not to use additional (L2-specific) common loop:

```
set,/par/raw/clp2/loop,off
```

Or

```
set,raw/clp2/loop,off
```

Query the current status.

```
print,/par/raw/clp2/loop
```

Or

```
print,raw/clp2/loop
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Common PLL order

`[/par/]raw/clp/pll/order`

Description: Selects between common PLL orders.

Access: read & write

Type: integer

Values: `2 | 3`

Default: `3`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use a 2nd order:

```
set,/par/raw/clp/pll/order,2
```

Or

```
set,raw/clp/pll/order,2
```

Query the current status.

```
print,/par/raw/clp/pll/order
```

Or

```
print,raw/clp/pll/order
```

Receiver Availability:

- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Time to extrapolate missing observations

`[/par/]raw/clp/wait`

Description: Specifies the amount of time during which the receiver will extrapolate observations should the loss of lock on the signal occur.

Access: read & write

Type: integer

Values: `[1...100]`; seconds

Default: `10`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to extrapolate observations for 15 seconds:
`set,/par/raw/clp/wait,15`
Or
`set,raw/clp/wait,15`

Query the current status.
`print,/par/raw/clp/wait`
Or
`print,raw/clp/wait`

Receiver Availability:
- This parameter is not available in the GMS-2 receiver.
- Although you can formally set this parameter in the G3-based receivers, it will not be applied.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P-code lock loop bandwidth scale factor

`[/par/]raw/pcll/scale`

Description: Establishes the ratio between P-code PLL bandwidth and P-code CLL bandwidth.

Access: read & write

Type: float

Values: `[0.1...100.0]`

Default: `2.0`

Usage Guidelines: N/A

Command Examples: Use the default value.

```
set,/par/raw/pcll/scale,2.0
```

Or

```
set,raw/pcll/scale,2.0
```

Query the current status.

```
print,/par/raw/pcll/scale
```

Or

```
print,raw/pcll/scale
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Positioning Parameters

Satellite elevation mask for position computation

`[/par/]pos/elm`

Description:	Defines the minimum angle at which the receiver will use satellites for the position computation.
Access:	read & write
Type:	integer
Values:	`[0...90]`; degrees
Default:	`5`
Usage Guidelines:	Satellites with elevations angles (angle above the horizon) lower than this mask angle will be excluded from the position computation.
Command Examples:	Set the elevation mask angle to 15 degrees. `set,/par/pos/elm,15` Or `set,pos/elm,15` Query the current status. `print,/par/pos/elm` Or `print,pos/elm`
Receiver Availability:	This parameter is available for all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Minimum allowed SNR for satellites used in position computation

`[/par/]pos/minsnr`

Description: Specifies the minimum SNR at which the receiver will use satellites for the position computation.

Access: read & write

Type: integer

Values: `[0...50]`; dB*Hz

Default: `30`

Usage Guidelines: Only those satellites with SNRs in excess of the specified threshold value will be used in the position computation.

Command Examples: Set the minimum SNR to 32 dB*Hz.

```
set,/par/pos/minsnr,32
```

Or

```
set,pos/minsnr,32
```

Query current status.

```
print,/par/pos/minsnr
```

Or

```
print,pos/minsnr
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Position computation mode

`[/par/]pos/mode/cur`

Description: Specifies the receiver operation mode.

Access: read & write

Type: enumerated

Values: `sp|cd|pf|pd`

- `sp` – single point positioning (also known as *absolute*, *stand alone* or *point positioning*)
- `cd` – code differential
- `pf` – RTK with float ambiguities
- `pd` – RTK with fixed ambiguities

Default: `sp`

Usage Guidelines: N/A

Command Examples: Set the receiver to operate in *RTK with fixed ambiguities* mode.

```
set,/par/pos/mode/cur,pd
```

Or

```
set,pos/mode/cur,pd
```

Query the current status.

```
print,/par/pos/mode/cur
```

Or

```
print,pos/mode/cur
```

Receiver Availability: This parameter is available for all receivers capable of RTK operations.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output of single point position

`[/par/]pos/mode/sp`

Description: Specifies to output the current standalone position if a DGPS (or RTK) position is unavailable in the current epoch.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: If the receiver is operating in a differential mode and is unable to obtain a differential solution position at the current epoch, the receiver will output the current single point position for the unavailable differential solution unless this parameter has been switched off.

Command Examples: Disable the output of a single point position when operating in differential mode, if the receiver is unable to obtain a differential solution position at the current epoch.

```
set,/par/pos/mode/sp,off
```

Or

```
set,pos/mode/sp,off
```

Query current status.

```
print,/par/pos/mode/sp
```

Or

```
print,pos/mode/sp
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Output of code differential position

`[/par/]pos/mode/cd`

Description: Specifies to output the current code differential position if a RTK position is unavailable in the current epoch.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines:

- If this parameter is set to `on` and a code differential solution cannot be obtained, you can instruct the receiver to output single-point position for the unavailable differential. For details, see the `/par/pos/mode/sp` parameter above.
- Code differential mode requires broadcasting the corresponding DGPS (not RTK) messages from the reference receiver and accepting them on the rover receiver. If any of these requirements are not met, then enabling this parameter will not have any effect.

Command Examples: Enable the output of a code differential position if the receiver is unable to obtain a RTK solution position at the current epoch.

```
set,/par/pos/mode/cd,on
```

Or

```
set,pos/mode/cd,on
```

Query current status.

```
print,/par/pos/mode/cd
```

Or

```
print,pos/mode/cd
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Measurement types used for position computation

`[/par/]pos/sp/meas`

Description: Specifies that the receiver will use either an ionosphere-free combination or original psuedoranges from the corresponding slot in the position computation.

Access: read & write

Type: enumerated

Values: `ca|p1|p2|ionofree`

- `ca` – C/A L1 code
- `p1` – P/L1 code
- `p2` – PL2 code
- `ionofree` – Ionosphere-free combination

Default: `ca`

Usage Guidelines: This parameter applies only to the single point positioning mode.

Command Examples: Enable the receiver to use ionosphere-free combination for the position computation.

```
set,/par/pos/sp/meas,ionofree
```

Or

```
set,pos/sp/meas,ionofree
```

Query current status

```
print,/par/pos/sp/meas
```

Or

```
print,pos/sp/meas
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GPS constellation for position computation

`[/par/]pos/gps/sat`

Description: Enables or disables the use of the GPS satellite constellation.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: The `print` command will cause the receiver to output a comma-delimited sequence of `y`'s and `n`'s – the position of each indicating the PRN numbers of the enabled, or disabled satellites respectively. By disabling the entire constellation, all `n`'s will be returned.

Command Examples: Enable the use of GPS satellites.
`set,/par/pos/gps/sat,on`
Or
`set,pos/gps/sat,on`

Query the current status.
`print,/par/pos/gps/sat`
Or
`print,pos/gps/sat`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specific GPS satellites used for position computation

`[/par/]pos/gps/sat/<N>`

Description: Enables or disables specific GPS satellites.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: In this parameter, `N` denotes an integer number between 1 and 32 representing the GPS satellites' PRN number.

Command Examples: Disable the use of GPS satellite PRN 13.

```
set,/par/pos/gps/sat/13,off
```

Or

```
set,pos/gps/sat/13,off
```

Query the current status.

```
print,/par/pos/gps/sat/13
```

Or

```
print,pos/gps/sat/13
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GLONASS constellation (orbital slot) for position computation

`[/par/]pos/glo/sat`

Description: Enables or disables the use of the GLONASS satellite constellation.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: The `print` command will cause the receiver to output a comma-delimited sequence of `y`'s and `n`'s – the position of each indicating the orbital slot numbers of the GLONASS satellites used for position computation. By disabling the entire constellation, all `n`'s will be returned.

Command Examples: Enable the use of GLONASS satellites.
`set,/par/pos/glo/sat,on`
Or
`set,pos/glo/sat,on`

Query the current status.
`print,/par/pos/glo/sat`
Or
`print,pos/glo/sat`

Receiver Availability: This parameter is available in receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GLONASS satellites (orbital slot) for position computation

`[/par/]pos/glo/sat/<N>`

Description: Enables or disables specific GLONASS satellites to be used.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: In this parameter, `N` denotes an integer number between 1 and 24 and corresponds to the GLONASS satellite's orbital slot number.

Command Examples: Disable the use of GLONASS SV#6 from position computation.

```
set,/par/pos/glo/sat/6,off
```

Or

```
set,pos/glo/sat/6,off
```

Query the current status.

```
print,/par/pos/glo/sat/6
```

Or

```
print,pos/glo/sat/6
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GLONASS constellation (frequency channel #) for position computation

`[/par/]pos/glo/fcn`

Description: Enables or disables the use of the GLONASS satellite constellation.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines:

- The `print` command will cause the receiver to output a comma-delimited sequence of `y`'s and `n`'s – the position of each indicating the frequency channel numbers of the GLONASS satellites used for position computation. By disabling the entire constellation, all `n`'s will be returned.
- From a functional point of view, the command `set,/par/pos/glo/fcn,on` is the equivalent to `set,/par/pos/glo/sat,on` and the command `set,/par/pos/glo/fcn,off` is equivalent to `set,/par/pos/glo/sat,off`.

Command Examples: Enable the use of GLONASS satellites.

```
set,/par/pos/glo/fcn,on
```

Or

```
set,pos/glo/fcn,on
```

Query the current status.

```
print,/par/pos/glo/fcn
```

Or

```
print,pos/glo/fcn
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

GLONASS satellites (frequency channel #) for position computation

`[/par/]pos/glo/fcn/<N>`

Description: Enables or disables specific GLONASS satellites to be used.

Access: read & write

Type: boolean, integer

Values: `off|on`

Default: `on`

Usage Guidelines:

- In this parameter, `N` denotes an integer number from 1 to 24 for firmware 3.1p3 or older and from -7 to +13 for firmware 3.2 or newer.
- From a functional point of view, the command `set,/par/pos/glo/fcn/<N>,off` is the equivalent to `set,/par/pos/glo/sat/<N>,off`.

Command Examples: Disable the use of the GLONASS satellite with frequency number 5.

```
set,/par/pos/glo/fcn/5,off
```

Or

```
set,pos/glo/fcn/5,off
```

Query the current status

```
print,/par/pos/glo/fcn/5
```

Or

```
print,pos/glo/fcn/5
```

Receiver Availability: This parameter is available in all receivers.

History:

- This parameter was introduced in the first release of firmware.
- The range of values was changed starting with firmware 3.2. Refer to *Usage Guidelines* above.

See Also: N/A

Select satellite constellation for position computation

`[/par/]pos/sys`

Description: Selects a satellite constellation to be used for position computation.

Access: read & write

Type: list of two boolean values, where the first and second values correspond to GPS and GLONASS respectively.

Values: `{off|on,off|on}`

Default: `{on,on}`

Usage Guidelines: N/A

Command Examples: Enable the receiver to use the GPS constellation only, no GLONASS.
`set,/par/pos/sys,{on,off}`
Or
`set,pos/sys,{on,off}`

Query current status
`print,/par/pos/sys`
Or
`print,pos/sys`

Receiver Availability: This parameter is available on all dual constellation GNSS receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Correct pseudoranges for ionosheric errors when computing point position

`[/par/]pos/sp/iono`

Description: Corrects the measured pseudoranges for ionospheric delay errors before computing the point position.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines:
- For the ionospheric model used, refer to ICD-GPS-200C, Revision IRN-200C-004 April 12, 2000.
- Ionospheric corrections are computed using data from the IO message.
- These corrections are used exclusively for computing single point position.

Command Examples: Enable the receiver to correct the measured pseudoranges for ionospheric delay errors before computing the point position.
`set,/par/pos/sp/iono,on`
Or
`set,pos/sp/iono,on`

Query current status.
`print,/pos/sp/iono`
Or
`print,pos/sp/iono`

Receiver Availability: This parameter is available on all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: “LH – Logging history” on page 4-127.

Correct pseudoranges for tropospheric errors when computing point position

`[/par/]pos/sp/tropo`

Description: Corrects the measured pseudoranges for tropospheric delay errors when computing the point position.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines:
- For the tropospheric model applied, refer to [2].
- The tropospheric model is used by the receiver exclusively for single point positioning.

Command Examples: Enable the receiver to correct the measured pseudoranges for tropospheric delay errors when computing the receiver single point position.

```
set,/par/pos/sp/tropo,on
```

Or

```
set,pos/sp/tropo,on
```

Query current status

```
print,/par/sp/tropo
```

Or

```
print,sp/tropo
```

Receiver Availability: This parameter is available on all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PDOP mask for position computation

`[/par/]pos/pdop`

Description:	Specifies the PDOP mask used in position computation.
Access:	read & write
Type:	float
Values:	`[0...500]`
Default:	`30`
Usage Guidelines:	If the computed PDOP exceeds the specified threshold value, the receiver will not compute the position.
Command Examples:	Set the PDOP mask to a value of 8. `set,/par/pos/pdop,8` Or `set,pos/pdop,8` Query current status. `print,/par/pos/pdop` Or `print,pos/pdop`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Altitude used in position computation

`[/par/]pos/alt`

Description: Describes the ellipsoidal height in the currently used datum.

Access: read & write

Type: float

Values: `[-1000.0000...+10000.0000]`; meters

Default: `0.0000`

Usage Guidelines: This command serves two purposes:

- First, using an a priori ellipsoidal height will allow the receiver to get a position fix in critical situations when there are few satellites in sight and when it is impossible to derive the point solution using the current measurements only.
- Second, using a precise a priori ellipsoidal height estimate will allow you to have more precise position fixes.

Care should be taken that this parameter is specified before you issue the `set,/par/pos/fix/alt,on` command described below.

Command Examples: Set the ellipsoid height to be used in the position computation to be +15.72 meters above the currently specified datum.

```
set,/par/pos/alt,+15.72
```

Or

```
set,pos/alt,+15.72
```

Query current status

```
print,/par/pos/alt
```

Or

```
print,pos/alt
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of a fixed altitude in the position computation

`[/par/]pos/fix/alt`

Description: Enables the receiver to use in position computation the fixed ellipsoidal height entered with a `set,/par/pos/alt,...` command.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines: This setting is effective only when the receiver runs in either "sp" or "cd" mode.

Command Examples: Enable the receiver to use the fixed ellipsoidal height entered with the previously described `set,/par/pos/alt,...` command.
`set,/par/pos/fix/alt,on`
Or
`set,pos/fix/alt,on`

Query current status
`print,/par/pos/fix/alt`
Or
`print,pos/fix/alt`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of a user-defined geoid separation in the position computation

`[/par/]pos/fix/geoidh`

Description: Instructs the receiver to use an alternative geoidal separation from the parameter `/par/pos/geoidh` (see below) for the geoidal separation value computed through the built-in model.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines:

- The receiver incorporates the global geoid model from "*Technical characteristics of the NAVSTAR GPS (June 1991). Reprinted October, 1993 by Navtech Seminars & Navtech Book and Software Store Inc., from U.S. Government document. Specifically, Appendix 6, Standard Mean Sea Level Algorithm*".
- The receiver will use this model if the receiver option _GEO is enabled.

Command Examples: Enable the receiver to use an alternative geoidal separation from the parameter `/par/pos/geoidh`.

```
set,/par/pos/fix/geoidh,on
```

Or

```
set,pos/fix/geoidh,on
```

Query current status.

```
print,/par/pos/fix/geoidh
```

Or

```
print,pos/fix/geoidh
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

User-defined geoid separation in the position computation

`[/par/]pos/geoidh`

Description: Specifies the user-defined geoidal separation.

Access: read & write

Type: float

Values: `[-1000.0000...+1000.0000]`; meters

Default: `0.0000`

Usage Guidelines: If the parameter `/par/pos/fix/geoidh` is set to `on`, the user-defined geoidal separation value is used for orthometric height computation (overriding its counterpart from the built-in geoidal model if this model is enabled in the receiver.

Command Examples: Set the user-defined geoid separation to a value of -5.83

```
set,/par/pos/geoidh,-5.83
```

Or

```
set,pos/geoidh,-5.83
```

Query current status

```
print,/par/pos/geoidh
```

Or

```
print,pos/geoidh
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Previous parameter.

Filtering Position Estimates

Position filter mode

`[/par/]pos/filt/mode`

Description:	Turns on and off the position filter mode.
Access:	read & write
Type:	boolean
Values:	`off|on`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Turn on the position filter mode. `set,/par/pos/filt/mode,on` Or `set,pos/filt/mode,on` Query current status `print,/par/pos/filt/mode` Or `print,pos/filt/mode`
Receiver Availability:	This parameter is unavailable on G3-based receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Position filter type

`[/par/]pos/filt/type`

Description: Specifies the type of position filter to use.

Access: read & write

Type: string

Values: `stat`

Default: `stat`

Usage Guidelines: If this parameter is set to `stat`, a simple N-point weighted moving average is used to smooth the current position. Note that this type of position filter normally applies only to static (or nearly static) receivers in sp or cd mode. It is especially useful when the number of tracked satellites changes abruptly; in this case the position accuracy may drop dramatically unless the position filter is on.
For moving receivers, using this type of position filter may adversely affect the receiver trajectory's accuracy.

Command Examples: Set the position filter type to `stat`.
`set,/par/pos/filt/type,stat`
Or
`set,pos/filt/type,stat`

Query current status
`print,/par/pos/filt/type`
Or
`print,pos/filt/type`

Receiver Availability: This parameter is unavailable on G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Position filter length

`[/par/]pos/filt/num`

Description: Designates the number of the preceding pre-filtered position measurements used by the least-squares estimator to obtain the current smooth position.

Access: read & write

Type: integer

Values: `[1...10000]`

Default: `30`

Usage Guidelines: If `/par/pos/filt/type` = `stat`, this estimator is just an N-point weighted running average.

Command Examples: Set the number of the preceding pre-filtered position measurements to a value of 50.

```
set,/par/pos/filt/num,50
```

Or

```
set,pos/filt/num,50
```

Query current status

```
print,/par/pos/filt/num
```

Or

```
print,pos/filt/num
```

Receiver Availability: This parameter is unavailable on G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum allowed time gap between adjacent position estimates

`[/par/]pos/filt/maxdel`

Description: Specifies the maximum allowed time gap mask for two successive position estimates.

Access: read & write

Type: integer

Values: `[1...3600]`; seconds

Default: `10`

Usage Guidelines: If the current position estimate is obtained in more than maximum allowed time gap seconds after the previous one, the position filter is reset (functionally, this is similar to executing a `set,/par/pos/filt/reset,on` command).

Command Examples: Set the maximum allowed time gap mask for two successive position estimates to a value of 120 seconds.
`set,/par/pos/filt/maxdel,120`
Or
`set,pos/filt/maxdel,120`

Query current status
`print,/par/pos/filt/maxdel`
Or
`print,pos/filt/maxdel`

Receiver Availability: This parameter is unavailable on G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reset the position filter

`[/par/]pos/filt/reset`

Description: Resets the position.

Access: write-only

Type: boolean

Values: `off|on`

- `off` – ignored.
- `on` – the receiver will reset the position filter.

Default: `off`

Usage Guidelines: Immediately after the reset, the parameter will automatically be set to `off`.

Command Examples: Reset the position filter.
`set,/par/pos/filt/reset,on`
Or
`set,pos/filt/reset,on`

Receiver Availability: This parameter is unavailable on G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Weighting mode for the position filter

`[/par/]pos/filt/weight`

Description: Specifies the weighting mode to be used in the position filter.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the position filter treats all of the position estimates as uniformly weighted measurements.
- `on` – each position estimate will have its own weight depending on the corresponding rms error.

Default: `off`

Usage Guidelines: N/A

Command Examples: Set the position weighting so that each position estimate will have its own weight depending on the corresponding rms error.

```
set,/par/pos/filt/weight,on
```

Or

```
set,pos/filt/weight,on
```

Query current status

```
print,/par/pos/filt/weight
```

Or

```
print,pos/filt/weight
```

Receiver Availability: This parameter is unavailable on G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Timing Parameters

Fixed time offset between GPS and GLONASS time scales

`[/par/]pos/gpsglo`

Description: Determines the a priori known (constant) time offset between the GLONASS and GPS time scales.

Access: read & write

Type: float

Values: `[-300000.0000...+300000.0000]`; meters

Default: `0.0000`

Usage Guidelines: This offset is entered in meters, not in time units (just divide this value by the speed of light to get the offset in seconds).
This parameter serves two purposes:

- Using an a priori time offset will allow the receiver to get a position fix in critical situations when there are few satellites in sight and when it is impossible to derive the point solution using the current measurements only. For example, if there are only three GPS satellites and one GLONASS satellite in view, the receiver will not be able to get a position fix unless the user enters a GLONASS vs. GPS time offset or some other a priori data, thus reducing the number of unknowns in the corresponding set of equations.
- Using a precise a priori time offset will allow you to have more precise position fixes.

Care should be taken that this parameter is correctly specified before you issue the `set,/par/pos/fix/gpsglo,on` command.

Command Examples: Set the fixed time offset between GLONASS and GPS to a value of 0.0.
`set,/par/pos/gpsglo,0.0`
Or
`set,pos/gpsglo,0.0`

Query current status
`print,/par/pos/gpsglo`
Or
`print,pos/gpsglo`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Following parameter, “Use of a priori (fixed) offset between GPS and GLONASS time scales.”

Use of a priori (fixed) offset between GPS and GLONASS time scales

`[/par/]pos/fix/gpsglo`

Description: Applies the constant time offset in position computation.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines: This parameter is effective only when the receiver runs in either "sp" or "cd" mode.

Command Examples: Instruct the receiver to use the time offset between GPS and GLONASS specified in the `set,/par/pos/gpsglo,...` command.
`set,/par/pos/fix/gpsglo,on`
Or
`set,pos/fix/gpsglo,on`

Query current status
`print,/par/pos/fix/gpsglo`
Or
`print,pos/fix/gpsglo`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Preceding parameter, “Fixed time offset between GPS and GLONASS time scales.”

Local time zone (offset)

`[/par/]pos/ltz`

Description: Specifies the local time zone offset.

Access: read & write

Type: list of two integer values

Values: `{[-13...+13],[00...+59]}`
The first value in the list describes the local zone hour offset from the UTC time.
The second value in the list describes the local zone minute offset from the UTC time.

Default: `{+0,00}`

Usage Guidelines: These local zone hour and minute offsets will be used in NMEA ZDA sentence.

Command Examples: Set the local time zone offset to a value 0.0.
`set,/par/pos/ltz,{0,0}`
Or
`set,pos/ltz,{0,0}`

Query current status.
`print,/par/pos/ltz`
Or
`print,pos/ltz`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Improved timing mode for a static receiver

`[/par/]pos/clk/fixpos`

Description: Uses the a priori known coordinates from the `/par/ref/pos` parameter to solve for the unknown time offsets *GPS system time vs. receiver time* and *GLONASS system time vs. receiver time*.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines: Running the receiver in the Improved Timing mode serves two main purposes:

- Enables you to synchronize your receiver with the GPS and/or GLONASS time scales even when you have only one GPS satellite and/or one GLONASS satellite in sight.
- Most important, this mode provides better synchronization precisions as compared with the general case when the receiver has to solve for both coordinates and time offsets.

When is the Improved Timing mode worth running?
There are static applications where the receiver is used as an external high-precision timer. In such applications, it is of great importance to synchronize the PPS and EVENT signals to the global time scales with the highest possible accuracy and precision.
Before using this mode, make sure that the priori coordinates you have specified through the `/par/ref/pos` parameter are accurate enough. This mode is intended for static applications only.

Command Examples: Enable improved timing mode.
`set,/par/pos/clk/fixpos,on`
Or
`set,pos/clk/fixpos,on`

Query current status
`print,/par/pos/clk/fixpos`
Or
`print,pos/clk/fixpos`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A.

Getting date and time from the receiver

The parameters below report time information expressed in various time scales. The times are updated every millisecond. Since these time updates are asynchronous with respect to the `print` commands, retrieving of different time parameters corresponding to the same moment in time with sequential commands cannot be achieved. To avoid this problem, it is recommended to use a single `print` command to retrieve multiple values as will be shown below. Such an approach ensures that the retrieved time information will correspond to the same moment in time.

Current receiver time

`[/par/]time/rcv`

Description:	Reports the current time in local (i.e., receiver) time scale.
Access:	read-only
Type:	integer
Values:	`[0...86400000)`; milliseconds
Default:	`off`
Usage Guidelines:	This parameter is effective only when the receiver runs in either "sp" or "cd" mode.
Command Examples:	Query the current receiver time. `print,/par/time/rcv` Or `print,time/rcv`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

Current UTC date

`[/par/]time/utc/date`

Description: Reports the current UTC date.

Access: read-only

Type: string

Values: `YYYY-MM-DD`

- `YYYY` – the year between 0001 and 9999
- `MM` – the month of the year between 01 (January) and 12 (December)
- `DD` – the day of the month between 01 and 31

Default: `cur_value`

Usage Guidelines: If the date information is unavailable, the receiver reports an empty string.

Command Examples: Query the current UTC date.

```
print,/par/time/utc/date
```

Or

```
print,time/utc/date
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Current UTC time of day

`[/par/]time/utc/clock`

Description: Reports the current time of day in UTC.

Access: read-only

Type: string

Values: `HH:MM:SS.SSS`

- `HH` – hours [00-23]
- `MM` – minutes [00-59]
- `SS` – seconds [00-60[a]]
- `SSS` – milliseconds [000-999]

Default: `cur_value`

Usage Guidelines:

- If the time information is unavailable, the receiver reports an empty string.
- With the `print,/par/time/utc` command, you will instruct the receiver to output the UTC date and time referred to the same moment in time.

Command Examples: Query the current UTC time of day.

```
print,/par/time/utc/clock
```

Or

```
print,time/utc/clock
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

a. 60 only if UTC leap second.

Current GPS week number

`[/par/]time/gps/week`

Description: Reports the current GPS week number.

Access: read-only

Type: integer

Values: `[-1...1023]`
`-1` – GPS week number is unavailable.

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the current GPS week number.
`print,/par/time/gps/week`
Or
`print,time/gps/week`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Current GPS time of week

`[/par/]time/gps/ms`

Description: Reports the current GPS time of week.

Access: read-only

Type: integer

Values: `[-1...604800000)`; milliseconds
`-1` – GPS time of week is unavailable.

Default: `cur_value`

Usage Guidelines: With the `print,/par/time/gps` command, you will instruct the receiver to output the GPS week number and the GPS time of week referred to the same moment in time.

Command Examples: Query the current GPS time of week.
`print,/par/time/gps/ms`
Or
`print,time/gps/ms`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Current GLONASS day number

`[/par/]time/glo/day`

Description: Reports the current GLONASS day number within the 4-year period beginning with the leap year.

Access: read-only

Type: integer

Values: `-1, [1...1461)`
`-1` – GLONASS day number is unavailable.

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the current GLONASS day number.
`print,/par/time/glo/day`
Or
`print,time/glo/day`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Current GLONASS time of day

`[/par/]time/glo/ms`

Description: Reports the current time of day in GLONASS system time.

Access: read-only

Type: integer

Values: `[-1...86400000)`; milliseconds
`-1` – GLONASS time of day is unavailable.

Default: `cur_value`

Usage Guidelines: With the `print,/par/time/glo` command, you will instruct the receiver to output the GLONASS day number and the GLONASS time of day referred to the same moment in time.

Command Examples: Query the current GLONASS time of day.
`print,/par/time/glo/ms`
Or
`print,time/glo/ms`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Date and time from real-time clock

`[/par/]dev/rtc/time`

Description: Reports the current date and time from the receiver's real-time clock.

Access: read-only

Type: list of six integer values

Values: `{sec,min,hour,day of month,month,year}`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the date and time.
`print,/par/dev/rtc/time`
Or
`print,dev/rtc/time`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Time since last reboot

`[/par/]rcv/uptime`

Description: Reports the time the receiver has been up and running without rebooting.

Access: read-only

Type: string

Values: `DdHHhMMmSSs`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the time since the last reboot.
`print,/par/rcv/uptime`
Or
`print,rcv/uptime`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Datum Parameters

This section uses the following notations:

- *datum* is a string of up to 5 upper case characters. This designates the geodesic datum in which the receiver's reference position is specified.
- *x,y,z* designates the receiver's Cartesian coordinates in the specified datum.
- *lat,lon,alt* designates the receiver's latitude, longitude and ellipsoidal height in the specified datum, respectively.

To enter Cartesian coordinates, you can use any of the float point formats the C programming language allows for the types ***float*** and ***double***.

The `print` command currently supports the following format for geodetic coordinates:

- for latitude – `[N|S]DDdMMmSS.SSSSSSs`
- for longitude – `[E|W]DDDdMMmSS.SSSSSSs`

  where

  – the direction signs `N`, `S`, `E`, and `W` designate Northern, Southern, Eastern, and Western hemisphere, respectively,

  – `DD` and `DDD` stand for integer degrees,

  – `MM` stands for integer minutes,

  – `SS.SSSSSS` designates float seconds,

  – `d`, `m` and `s` serve as delimiters.

Examples.

1. `N83d42m47.556000s` means 83 degrees 42 minutes and 47.556 seconds Northern latitude.
2. `E083d42m47.556000s` means 83 degrees 42 minutes and 47.556 seconds Eastern longitude.



Whether an updated print command with programmable format ever appears or not, GRIL will support the current fixed format version in either case (to provide backward compatibility with the earlier firmware versions).

The corresponding `set` command supports the following formats:

Latitudes and longitudes you enter can range within [-90, 90] and [-180, 180] degrees, respectively. Negative longitude corresponds to the Western hemisphere; negative latitude corresponds to the Southern hemisphere.

The `set` command supports two different formats for latitude and longitude. These are the *general format* and the *almost fixed format* discussed below.

General Format

As the name of this format implies, this is a very flexible format enabling you to specify latitude and longitude in a number of different ways. You can use various angular units (specifically, degrees, minutes, seconds, and radians) and their combinations.

Angle representation may comprise one or more floating point numbers. Every float number in the angle representation except the right-most one must have a delimiter after it. Allowed delimiters are "d", "m", "s", or "r", which denote degrees, minutes, seconds, and radians, respectively. Using a delimiter after the right-most float number in the angle representation is optional. If you omit the delimiter after the right-most float number, the receiver will first identify the preceding delimiter and then retrieve the omitted one by using the following decision rule:

Preceding delimiter	Omitted delimiter is assumed equal to
d	m
m	s
s	s
r	r
none	d

An angle representation may or may not have a direction sign. Direction signs are "E" and "W" to denote Eastern and Western longitude, "N" and "S" to denote Northern and Southern latitude, respectively. A direction sign may be placed either at the very beginning or at the very end of the angle representation (see the examples below).

Each of the separate floats describing "degrees", "minutes", "seconds" and "radians" is multiplied by an appropriate factor and the resulting products are summed. Also, if there is either "W" or "S" in the angle representation, the resulting sum's sign is inverted.

If there is no direction sign in the angle representation, such notation is still valid. Lastly, you can enclose your angle representation in double quotes (this is optional).

Examples.

1. `37.87` and `37.87d`

   These angle representations are equivalent. Either means 37.87 degrees Eastern longitude or 37.87 degrees Northern latitude.

2. `37.87W`, `W37.87`, `37.87dW`, and `W37.87d`

   These four representations are all equivalent meaning 37.87 degrees Western longitude.

3. `27d37m20.45sE` and `27d37m20.45E`

   Equivalent representations meaning 27 degrees 37 minutes and 20.45 seconds Eastern longitude.

4. `-0.85r`

   This means either 0.85 radians Western longitude or 0.85 radians Southern latitude.

5. `27.13d-12.6s34dW`

   This means 27.13 degrees minus 12.6 seconds plus 34 degrees Western longitude. It is equivalent to 61.13 degrees minus 12.6 seconds Western longitude or simply 61.1265 degrees Western longitude.

Almost Fixed Format

This format comprises the following fields (from left to right):

- letter “x” (in lower case),
- one or more decimal digits before the decimal point,
- decimal point “.” (optional),
- one or more decimal digits (if any) after the decimal point (optional),
- direction sign (“E” or “W” for longitude, “N” or “S” for latitude).

NOTICE *This format is close to how latitude and longitude are presented in NMEA sentences.*

This format uses integer degrees and float minutes as follows:

- *Integer minutes* are described either by the two decimal digits immediately before the decimal point or, if there is no decimal point in the angle notation, by the two decimal digits immediately before the direction sign.
- *Fractional minutes* (if any) are described by the decimal digits after the decimal point.
- The remaining decimal digits (the left-most ones) describe degrees (note that this format uses only integer degrees)

Again, you can optionally enclose your angle representation in double quotes.

Examples.

1. `x12023.234E` means 120 degrees and 23.234 minutes Eastern longitude
2. `x1202E` means 12 degrees and 2 minutes Eastern longitude.

Parameters

Receiver reference position in Cartesian coordinate system

`[/par/]ref/pos/<gps|glo>/xyz`

Description: Specifies reference position in Cartesian coordinate system.

Access: read & write

Type: list of four values

Values: `{datum,x,y,z}`
The first argument, *datum*, indicates the datum in which the base station's reference coordinates are specified. For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.
x,y,z are Cartesian coordinates in meters (float numbers).

Default: `{W84,+6378137.000,+0.000,+0.000}`

Usage Guidelines:

- If, for some reason, you want to have independent reference position estimates for GPS and GLONASS, then you will have to set the parameters `/par/ref/pos/gps/xyz` and `/par/ref/pos/glo/xyz` separately, one after the other. For example, you may want to explicitly specify the reference position in WGS-84 for GPS measurements and the reference position in PZ-90 for GLONASS. In most cases, however, it is recommended to use one reference position estimate for both GPS and GLONASS. With the command `set,/par/ref/pos//xyz,...`, you can specify both of the parameters `/par/ref/pos/gps/xyz` and `/par/ref/pos/glo/xyz` at one time.
- The receiver will not recognize the command `print,/par/ref/pos//xyz`.

Command Examples: Set the receiver reference position to the following cartesian coordinate values in the WGS84 datum.
x = 2850717.2067
y = 2199375.5841
z = 5247224.5240

```
set,/par/ref/pos/gps/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
set,/par/ref/pos/glo/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
set,/par/ref/pos//
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Query current status.

```
print,/par/ref/pos/gps/xyz
print,/par/ref/pos/glo/xyz
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Receiver reference position in geodetic coordinate system

`[/par/]ref/pos/<gps|glo>/geo`

Description: Specifies reference position in geodetic coordinate system.

Access: read & write

Type: list of four values

Values: `{datum,lat,lon,alt}`
The first argument, *datum*, indicates the datum in which the base station's reference coordinates are specified. For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.
lat,lon,alt are geodesic coordinates (latitude, longitude and ellipsoidal height, respectively).

Default: `{W84,N00d00m00.00000s,E00d00m00.00000s,+00.0000}`

Usage Guidelines:

- If, for some reason, you want to have independent reference position estimates for GPS and GLONASS, you will have to set the parameters `/par/ref/pos/gps/geo` and `/par/ref/pos/glo/geo` separately, one after the other. For example, you may want to explicitly specify the receiver's reference position in WGS84 for GPS measurements and its reference position in PZ-90 for GLONASS. As a rule, it is sufficient to use one and the same reference position estimate to specify both `/par/ref/pos/gps/geo` and `/par/ref/pos/glo/geo`. In addition, with the command `set,/par/ref/pos//geo,...`, you can specify both of the parameters at one time.
- The receiver will not recognize the command `print,/par/ref/pos//geo`.
- The objects `/par/ref/pos/gps/geo` and `/par/ref/pos/gps/xyz` are mutually dependant since they both represent a physical point's position in one and the same datum. In fact, these two parameters are just two different representations of the same physical object in the receiver's memory. The objects `/par/ref/pos/glo/geo` and `/par/ref/pos/glo/xyz` are logically dependent on each other too. They are but two different representations of the same physical object in the receiver memory.

Command Examples: Set the receiver reference position to the following Geodetic coordinate values in the WGS84 datum.
Latitude = N 55 degrees 43 minutes and 20.0301 seconds
Longitude = E 37 degrees 39 minutes and 02.8007 seconds
Height above ellipsoid = 165.863 meters

```
set,/par/ref/pos/gps/
geo,{W84,N55d43m20.0301s,E37d39m02.8007s,+165.8630}
set,/par/ref/pos/glo/
geo,{W84,N55d43m20.0301s,E37d39m02.8007s,+165.8630}
set,/par/ref/pos//
geo,{W84,N55d43m20.0301s,E37d39m02.8007s,+165.8630}
```

Query current status.

```
print,/par/ref/pos/gps/geo
print,/par/ref/pos/glo/geo
```

Receiver Availability: This parameter is available in all receivers.

`[/par/]ref/pos/<gps|glo>/geo`

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

The receiver's reference position in WGS-84 (for GPS) or PE-90 (for GLONASS) - Cartesian and geodetic coordinates

`[/par/]ref/syspos/<gps|glo>/<xyz|geo>`

Description: Reports the receiver's reference position in Cartesian or geodetic coordinate systems.

Access: read-only

Type: enumerated

Values:
```
{datum,x,y,z}
{datum,lat,lon,alt}
```

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the receiver's reference position in cartesian coordinate system.
```
print,/par/ref/syspos/gps/xyz
print,/par/ref/syspos/glo/xyz
```

Query the receiver's reference position in geodetic coordinate system.
```
print,/par/ref/syspos/gps/geo
print,/par/ref/syspos/glo/geo
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Current datum for position computation

`[/par/]ref/pos/datum/cur`

Description: Specifies the identifier of the datum used for position computation.

Access: read & write

Type: enumerated

Values: Up to 5 upper case alphanumeric characters.

Default: `W84`

Usage Guidelines: Your receiver will support, first, all of the datums whose identifiers are listed in the Department of Defense. World Geodetic System 1984 — Its Definitions and Relationships with Local Geodetic Systems —TR8305.2 and, second, the following four datums:

- W84 (this stands for the WGS-84 datum),
- P90, (this stands for the PZ-90 datum; recall that it is PZ-90 that GLONASS is based on),
- W72 (stands for the WGS-72 datum),
- USER, which designates the user-defined datum (see the command `set,pos/datum/USER` below). You can have only one user-defined datum in current use.

The reference source, the Department of Defense. World Geodetic System 1984 — Its Definitions and Relationships with Local Geodetic Systems — TR8305.2 does not give aliases for the datums WGS-84, PZ-90, WGS-72. Note that datum identifiers W84, P90 and W72 are used in NMEA sentences and other related sources.

Command Examples: Use WGS-84 for position computation.

```
set,/par/pos/datum/cur,W84
```

Or

```
set,pos/datum/cur,W84
```

Query current status.

```
print,/par/pos/datum/cur
```

Or

```
print,pos/datum/cur
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

User defined datum

`[/par/]/pos/datum/USER`

Description: Defines the datum parameters (user-defined).

Access: read & write

Type: list of 12 values

Values: `{{U-,Axis,InvFlat},{USER,R,dX,dY,dZ,Rx,Ry,Rz,S}}`

- `U-` – A flag designating the user-defined ellipsoid. You can either enter this flag exactly as shown above (the letter "U" in the upper case and the "minus" sign followed by comma) or, alternatively, enter nothing in this field (note that the delimiting comma will still exist).
- `Axis` – Float parameter designating the reference ellipsoid's major semi-axis. Ranges from 6300000.0 meters to 6500000.0 meters.
- `InvFlat` – Float parameter designating the reference ellipsoid's inverse flattening. This dimensionless parameter ranges within 280.0…310.0.
- `USER` – A flag designating the user-defined datum. You can either enter this flag as shown above (string "USER" in upper case) or simply omit these four characters (the following delimiting comma must be retained)
- `R` – Integer parameter specifies which of the two "principal" datums your user-defined datum is related to. Specify "0" for WGS-84 and "1" for PZ-90.
- `dX`,`dY`,`dZ` – Components of the translation vector with respect to the reference datum as specified by the "R" flag (see above). Each component can range within -10000.0...+10000.0 meters.
- `Rx`,`Ry`,`Rz` – Components of the rotation vector (in seconds of arc). Each component can range within -60...+60.
- `S` – scale in ppm (actual scale= S*10-6). It ranges within -100...+100.

Default: The user-defined datum will coincide with WGS84.

Usage Guidelines: The above 12 values specify a coordinate transformation from the user-defined datum to WGS-84 (or PZ-90) according to the following equations:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{WGS-84} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + (1 + S \cdot 10^{-6}) \cdot \begin{bmatrix} 1 & R_z & -R_y \\ -R_z & 1 & R_x \\ R_y & -R_x & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{local}$$

Command Examples: N/A

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PE-90 datum parameters

`[/par/]/pos/datum/P90/par`

Description: Specifies the PZ-90 datum parameters.

Access: read & write

Type: list of nine values

Values: `{P90,0,dX,dY,dZ,Rx,Ry,Rz,S}`

- `P90` – This flag designates your PZ-90 datum. You can either enter this flag as shown above (string "P90" in the upper case) or just omit these three characters (the following delimiting comma will still exist).
- `0` – This flag shows which datum your P90 transformation parameters are related to. You can either enter zero as shown above or, alternatively, just omit this zero character (the corresponding delimiting comma will still exist). This zero flag indicates that your version of the PZ-90 datum is related to the WGS-84 datum (which is usually the case in theory and practice).
- `dX,dY,dZ` – Components of the translation vector. Components can take float values in the range of -10000.0...+10000.0 meters.
- `Rx,Ry,Rz` – Components of the rotation vector. Components take float values in the range of -60...+60 arc seconds.
- `S` – Scale factor in ppm (actual scale= S*10-6). This factor can take float values between -100...+100.

Default: `{P90,0,+0.0000,+0.0000,+1.0000,+0.00000,+0.00000,-0.20626,+0.00000}`
This set was taken from [4]. These parameters specify a transformation from PZ-90 (in full, Parameters of Earth, 1990) to WGS-84.

Usage Guidelines: There are no universal Helmert transformation parameters for the PZ-90 datum so far. This explains why the user is allowed to define various versions of the transformation parameter set.

Command Examples: Set the PZ-90 datum parameters as follows:

dX = +0.0
dY = +0.0
dZ = +1.0
Rotation x = +0.0
Rotation y = +0.0
Rotation z = -0.20626
Scale factor = +0.0

```
set,/par/pos/datum/P90/
par,{P90,0,+0.0000,+0.0000,+1.0000,+0.00000,+0.00000,-0.20626,+0.00000}
```

Query the receiver's user defined datum parameters.

```
print,/par/pos/datum/p90/par
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

`[/par/]/pos/datum/P90/par`

See Also: N/A

Switching between dynamic constants of PZ-90 and PZ-90.02

`[/par/]/pos/datum/glo/dyn`

Description: Switches between dynamic constants used in the definition of PZ-90 and PZ-90.02 datum.

Access: read & write

Type: enumerated/par/pos/datum/glo/dyn

Values: `P90 | P90-I`

- `P90` – the receiver will use dynamic constants from PZ-90.
- `P90-I` – the receiver will use dynamic constants from PZ-90.02.

Default: `P90`

Usage Guidelines: Use this parameter to maintain compatibility with previous versions of firmware. when working with RTCM message types 20 and 21.

Command Examples: Use dynamic constants from PZ-90.02.
`set,/par/pos/datum/glo/dyn,P90-I`
Or
`set,pos/datum/glo/dyn,P90-I`

Query the parameter's value.
`print,/par/pos/datum/glo/dyn`
Or
`print,pos/datum/glo/dyn`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Pre-defined datum's parameters

`[/par/]/pos/datum/<D>`

Description: Reports pre-defined datum's parameters.

Access: read-only

Type: list of 12 values

Values: `{{EL,Axis,InvFlat},{DATUM,R,dX,dY,dZ,Rx,Ry,Rz,S}}`

- `EL` – The reference ellipsoid's ID code. Recall that for the user-defined ellipsoid EL="U-".
- `Axis` – The reference ellipsoid's major semi-axis in meters (ranges between 6300000.0...6500000.0).
- `InvFlat` – The reference ellipsoid's inverse flattening (dimensionless value ranging between 280.0...310.0)
- `DATUM` – The datum's ID code. Recall that for the user-defined datum this field will contain "USER".
- `R` – This flag indicates whether the datum's transformation parameters are specified with respect to WGS-84 (R=0) or PE-90 (R=1).
- `dX,dY,dZ` – Translations in X-, Y- and Z-direction, respectively. Each component ranges between -10000.0...+10000.0 meters.
- `Rx,Ry,Rz` – Rotations around X-, Y- and Z-axis, respectively. Each component ranges between -60...+60 seconds of arc.
- `S` – Scale in ppm (true scale= S*10-6) ranging within -100...+100.

Default: N/A

Usage Guidelines: In this parameter, `D` denotes either one of the valid datum identifiers (see the note to the parameter `/par/pos/datum/cur` above) or `INUSE`. The latter designates the current datum you have specified with a `set,pos/datum/cur,…` command.

Command Examples: Query the datum parameters of the datum currently in use.

```
print,/par/pos/datum/INUSE
```

Query the datum parameters of the datum identified as W72.

```
print,/par/pos/datum/W72
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Full transformation matrix for datum conversation

`[/par/]/pos/datum/ft`

Description: Applies the full transformation matrix for datum conversation.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – do not apply the full transformation matrix.
- `on` – apply the full transformation matrix.

Default: `off`

Usage Guidelines: It is recommended to set this parameter to `on` only when the rotation parameters used to relate the orientation of datums are big enough (tens seconds of arc or more). In all other cases, leave the default parameter's value (`off`).

Command Examples: Enable the use of full transformation matrix.

```
set,/par/pos/datum/ft,on
```

Or

```
set,pos/datum/ft,on
```

Query the parameter's value.

```
print,/par/pos/datum/ft
```

Or

```
print,pos/datum/ft
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.3p4.

See Also: N/A

Reference ellipsoid parameters

`[/par/]/pos/datum/<NAME>/ell`

Description: Reports the ellipsoid parameters of a specific datum.

Access: read-only

Type: list of three values

Values: `{EL,Axis,InvFlat}`

- `EL` – The reference ellipsoid's ID code. Recall that for the user-defined ellipsoid EL="U-".
- `Axis` – The reference ellipsoid's major semi-axis in meters (ranges between 6300000.0...6500000.0).
- `InvFlat` – The reference ellipsoid's inverse flattening (dimensionless value ranging between 280.0...310.0)

Default: N/A

Usage Guidelines: In this parameter, `NAME` designates the name of a datum.

Command Examples: Query the parameters of the ellipsoid currently in use.

```
print,/par/pos/datum/INUSE/ell
```

Query the parameters of the ellipsoid named ZAN.

```
print,/par/pos/datum/ZAN/ell
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

7 parameter transformation coefficients

`[/par/]/pos/datum/NAME/par`

Description: Reports the 7 parameter transformation coefficients of a specific datum.

Access: read-only

Type: list of nine values

Values: `{DATUM,R,dX,dY,dZ,Rx,Ry,Rz,S}`

- `DATUM` – The datum's ID code. Recall that for the user-defined datum this field will contain "USER".
- `R` – This flag indicates whether the datum's transformation parameters are specified with respect to WGS-84 (R=0) or PE-90 (R=1).
- `dX,dY,dZ` – Translations in X-, Y- and Z-direction, respectively. Each component ranges between -10000.0...+10000.0 meters.
- `Rx,Ry,Rz` – Rotations around X-, Y- and Z-axis, respectively. Each component ranges between -60...+60 seconds of arc.
- `S` – Scale in ppm (true scale= S*10-6) ranging within -100...+100.

Default: N/A

Usage Guidelines: In this parameter, `NAME` designates the name of a datum.

Command Examples: Query the 7 parameter transformation coefficients of the datum named W72.

```
print,/par/pos/datum/W72/par
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference ellipsoid and 7 parameter transformation data

`[/par/]/pos/datum/NAME`

Description: Reports both reference ellipsoid and 7-parameter transformation data of a specific datum.

Access: read-only

Type: list of 12 values

Values: {{EL,Axis,InvFlat},{DATUM,R,dX,dY,dZ,Rx,Ry,Rz,S}}

- `EL` – The reference ellipsoid's ID code. Recall that for the user-defined ellipsoid EL="U-".
- `Axis` – The reference ellipsoid's major semi-axis in meters (ranges between 6300000.0...6500000.0).
- `InvFlat` – The reference ellipsoid's inverse flattening (dimensionless value ranging between 280.0...310.0)
- `DATUM` – The datum's ID code. Recall that for the user-defined datum this field will contain "USER".
- `R` – This flag indicates whether the datum's transformation parameters are specified with respect to WGS-84 (R=0) or PE-90 (R=1).
- `dX`,`dY`,`dZ` – Translations in X-, Y- and Z-direction, respectively. Each component ranges between -10000.0...+10000.0 meters.
- `Rx`,`Ry`,`Rz` – Rotations around X-, Y- and Z-axis, respectively. Each component ranges between -60...+60 seconds of arc.
- `S` – Scale in ppm (true scale= S*10-6) ranging within -100...+100.

Default: N/A

Usage Guidelines: In this parameter, `NAME` designates the name of a datum.

Command Examples: Query the reference ellipsoid and 7-parameter transformation data for the datum SCK.
`print,/par/pos/datum/SCK`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Current grid system

`[/par/]/pos/grid/cur`

Description: Specifies the type of grid coordinate system to use.

Access: read & write

Type: enumerated

Values: `NONE | UTM | USER | LOC`

- `NONE` – the receiver will not compute grid coordinates (this mode allows some reduction of the processor's computation load).
- `UTM` – Universal Transverse Mercator (automatic selection of the right zone). 6-degree zones with the scale = 0.9996.
- `USER` – User-defined grid system.
- `LOC` – Local coordinates (as a result of the localization procedure).

Default: `UTM`

Usage Guidelines: The computation of grid and local coordinates depends on the parameter `/par/pos/datum/cur`.
Ensure that this parameter is specified properly.

Command Examples: Instruct the receiver to use the UTM coordinate system when grid coordinates are required.

```
set,/par/pos/grid/cur,UTM
```

View the current grid system.

```
print,/par/pos/grid/cur
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Parameters of user-defined grid system

`[/par/]/pos/grid/USER`

Description: Specifies a user-defined grid coordinate system.

Access: read & write

Type: list of 6 values

Values: `{MapPrj,Lat0,Lon0,Scale,FalseN,FalseE}`

- `MapPrj` – Map projection identifier:
  TM – Transverse Mercator projection.
  STER – Stereographic projection.
- `Lat0` – Latitude of the origin of the grid projection.
- `Lon0` – Longitude of the central meridian of the projection.
- `Scale` – Scale (ranges within 0.1...10.0).
- `FalseN` – False Northing (ranges between -10000000… +10000000 meters).
- `FalseE` – False Easting (ranges between -10000000… +10000000 meters).

Default: `{TM,N00d00m00.000000s,E000d00m00.000000s,1.000000000,0.0000,0.0000}`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to employ a user-defined grid coordinate system as follows:
Projection = Transverse Mercator
Latitude of Origin = N00 degrees, 00 minutes and 00.000 seconds
Longitude of Origin = E081 degrees, 00 minutes and 00.000 seconds
Scale factor = 1.0
False Northing = 1000.0
False Easting = 2000.0

```
set,/par/pos/grid/
USER,{TM,N00d00m00.0s,E081d00m00.0s,1.0,1000.0,2000.0}
```

Query user-defined system

```
print,/par/pos/grid/USER
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Parameters for transformation to local grid coordinate system

`[/par/]/pos/local/par`

Description: Specifies transformation parameters to a local grid coordinate system.

Access: read & write

Type: list of 12 values

Values: `{Lat0,Lon0,ScalePrj,FalseN,FalseE,delN,delE,Scale, Rotation,Hx,Hy,Ho}`

- Stereographic projection parameters:

  `Lat0` – latitude of the origin of the projection.
  `Lon0` – longitude of the origin of the projection.
  `ScalePrj` – Scale factor of the projection ranges within 0.1...10.0.
  `FalseN` – False Northing (ranges -10000000… +10000000 meters).
  `FalseE` – False Easting (ranges -10000000… +10000000 meters).

- Parameters obtained from the localization procedure:

  `delN` – offset in North direction (ranges -10000000… +10000000 meters).
  `delE` – offset in East direction (ranges between -10000000… +10000000 meters).
  `Scale` – Scale factor ranges within 0.1...10.0.
  `Rotation` – Rotation angle in degrees (0-360).
  `Hx`, `Hy`, `Ho` – parameters that relate to the height transformation; Hx and Hy are dimensionless (both range between ±1). Ho ranges between ±10000 meters.

  $n = scale \cdot (x_{stereo} \cdot \cos alpha - y_{stereo} \cdot \sin alpha) + ?N$

  $e = scale \cdot (x_{stereo} \cdot \sin alpha + y_{stereo} \cdot \cos alpha) + ?E$

  $h = H_o + H_x \cdot x_{stereo} + H_y \cdot y_{stereo}$

Default: `{N00d00m00.000000s,E000d00m00.000000s,1.0000000000,0.0000 ,0.0000,0.0000,0.0000,1.0000000000,000d00m00.000000s,0.00 0000000e+00,0.000000000e+00,0.0000}`

Usage Guidelines: N/A

`[/par/]/pos/local/par`

Command Examples:

Instruct (set) the receiver to define the transformation parameters to a local grid coordinate system as follows:
Latitude of Origin = N70 degrees, 41 minutes and 56.6108 seconds
Longitude of Origin = E023 degrees, 37 minutes and 03.9022 seconds
Scale factor = 1.0
False Northing = 1132.592
False Easting = 4163.613
delN = 2642.1273
delE = 300.6369
Scale factor = 0.9997139637
Hx = 0.00001119
Hy = 0.00002827
Ho = 69.4020

```
set,/par/pos/local/
par,{N70d41m56.6108s,E23d37m3.9022s,1,1132.592,4163.613,2
642.1273,300.6369,0.9997139637,3,0.00001119,0.00002827,
69.4020}
```

Query user-defined system.

```
print,/par/pos/local/par
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specific map projection for computing grid coordinates

`[/par/]/pos/grid/spc`

Description: Selects a specific map projection for computing grid coordinates.

Access: read & write

Type: enumerated

Values: `NONE|S34J|S34S|S34B`

- `NONE` – no specific projection is selected.
- `S34J` – System 34 Jutland is selected.
- `S34S` – System 34 Seeland is selected.
- `S34B` – System 34 Bornholm is selected.

Default: `NONE`

Usage Guidelines:

- There are map projections that require additional computations, for example, with the use of polynomials. This parameter allows you to select such a specific map projection. Currently the receiver supports three such projections which are used on Denmark maps.
- For the correct use of a projection from this list, you should set up the parameters of the desired projection using the `set,/par/pos/grid/USER,{«TM»,...}` command and specify the correct datum transformation parameters using the `set,/par/pos/datum/USER,{...}` command.

Command Examples: Select the *System 34 Jutland* projection.

```
set,/par/pos/grid/spc,S34J
```

Or

```
set,pos/grid/spc,S34J
```

View the parameter's value.

```
print,/par/pos/grid/spc
```

Or

```
print,pos/grid/spc
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna Database

TPS receivers contain embedded antenna database. This database provides identification and measurement information for more than 200 antennas. More precisely, for each antenna in this database, the following entries are given:

- Antenna identifier used in RTCM standard version 3.0. This identifier or ID (as denoted below) is a string comprising up to 20 characters.
- Antenna identifier used in CMR standard. This identifier or CMR ID (as denoted below) is an integer value in the range of 0...255.
- Vector offset from the ARP to L1 phase center (PC).
- Vector offset from L1 PC to L2 PC.

In addition, the database includes corrections for L1 and L2 antenna phase center variations depending on the satellites elevation angle. Although you cannot retrieve these PCVs with a command, they are automatically used by the RTK engine on condition that the reference and rover antenna identifiers are specified.

Antenna type definitions

`[/par/]antdb`

Description:	Reports the list of antenna definitions included in the antenna database.
Access:	read-only
Type:	list {ver,id,cmr,ids}
Values:	`ver|id|cmr|ids` • ver – for details, see the `/par/antdb/ver` parameter. • id – for details, see the `/par/antdb/id` parameter. • cmr – for details, see the `/par/antdb/cmr` parameter. • ids – for details, see the `/par/antdb/ids` parameter.
Default:	list of antenna definitions
Usage Guidelines:	N/A
Command Examples:	Instruct the receiver to report the list of the antenna definitions in the current antenna database. `list,/par/antdb` `print,/par/antdb`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

Antenna database versions

`[/par/]antdb/ver`

Description: Reports the antenna database version in the M.N.K. format where:
- M - database major version as decimal
- N - database minor version as decimal
- K - patch level as decimal

Access: read-only

Type: string

Values: up to 32 alphanumeric characters

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the list of the antenna definitions in the current antenna database.
```
print,/par/antdb/ver
```
Or
```
print,antdb/ver
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna database entries

`[/par/]antdb/id`

Description: Reports all the antenna ID's included in the antenna database along with their characteristics.

Access: read-only

Type: list

Values: list of ID's with characteristics

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report all the antenna ID's included in the antenna database along with their characteristics.
```
print,/par/antdb/id
list,/par/antdb/id
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna information for a specific antenna

`[/par/]antdb/id/<ID>`

Description: Reports numerical information for the selected antenna:

Access: read-only

Type: list of values

Values: `{cmr,l1_arp,l2_l1}`

- `cmr` – Antenna ID adopted in the CMR format.
- `l1_arp` – Vector offset from the ARP (Antenna Reference Point) to the L1 phase center represented as {east,north,height} in meters.
- `l2_l1` – Vector offset from the L1 phase center to L2 phase center represented as {east,north,height} in meters.

Default: N/A

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the antenna ID adopted in CMR format, the vector offset from the Antenna Reference Point (ARP) to the L1 phase center and the vector offset from the L1 phase center to the L2 phase center for the Topcon CR3 antenna.

```
print,/par/antdb/id/TPSCR3_GGD
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

CMR identifier for a specific antenna

`[/par/]/antdb/id/<ID>/cmr`

Description: Reports the CMR antenna identifier for the selected antenna ID. If the selected antenna ID does not have the corresponding CMR identifier, this parameter will return an empty string.

Access: read-only

Type: integer

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the CMR antenna identifier for the Topcon PG-A1 antenna.
`print,/par/antdb/id/TPSPG_A1/cmr`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Vector offset from ARP to L1 phase center for a specific antenna

`[/par/]/antdb/id/<ID>/l1_arp`

Description: Reports the vector offset in meters from the Antenna Reference Point (ARP) to the L1 antenna phase center (APC) for a specified antenna.

Access: read-only

Type: list (east,north,height} of float

Values: `{[-100.0...100.0],00],[-100.0...100.0],[-100.0...100.0]};` meters

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the vector offset in meters from the Antenna Reference Point (ARP) to the L1 antenna phase center (APC) for a Topcon PG-A1 antenna.
`print,/par/antdb/id/TPSPG_A1/l1_arp`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Vector offset from L1 to L2 phase centers for a specific antenna

`[/par/]/antdb/id/<ID>/l2_l1`

Description: Reports the vector offset in meters from the L1 phase center to the L2 phase center for a specified antenna.

Access: read-only

Type: list (east,north,height} of float

Values: `{[-100.0...100.0],00],[-100.0...100.0],[-100.0...100.0]};` meters

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the vector offset in meters from the L1 phase center to the L2 phase center for a Topcon PG-A1 antenna.
`print,/par/antdb/id/TPSPG_A1/l2_l1`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

ID,l1_arp and l2_l1 for antennas with known CMR ID's

`[/par/]/antdb/cmr`

Description: Reports ID, arp_l1, and l2_l1 for all antennas with known CMR ID's. The antennas for which their CMR IDs are unknown will not be included in this list.

Access: read-only

Type: list {[CMR_ID]}

Values: list of `cur_value`'s

Default: list of `cur_value`'s

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the ID, arp_l1, and l2_l1 for all antennas with known CMR ID's.
`print,/par/antdb/cmr`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna ID for a specific CMR ID

`[/par/]/antdb/cmr/<CMR_ID>/id`

Description: Reports the antenna ID for the selected CMR ID. If the selected CMR antenna ID does not have the corresponding ID, this parameter will return an error message.

Access: read-only

Type: string

Values: `cur_value`

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the antenna ID for the CMR antenna '214'.
`print,/par/antdb/cmr/214/id`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Vector offset from ARP to L1 phase center for a specific antenna with CMR ID

`[/par/]/antdb/cmr/<CMR_ID>/l1_arp`

Description: Reports the vector offset in meters from the Antenna Reference Point (ARP) to the L1 phase center for a specific antenna with CMR ID.

Access: read-only

Type: list (east,north,height} of float

Values: `{[-100.0...100.0],00],[-100.0...100.0],[-100.0...100.0]},` meters

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the vector offset in meters from the ARP to the L1 phase center of an antenna with CMR ID of 214.
`print,/par/antdb/cmr/214/l1_arp`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Vector offset from L1 to L2 phase centers for a specific antenna with CMR ID

`[/par/]/antdb/cmr/<CMR_ID>/l2_l1`

Description: Reports the vector offset in meters from the L1 phase center to the L2 phase center for a specified antenna with CMR ID.

Access: read-only

Type: list (east,north,height} of float

Values: `{[-100.0...100.0],00],[-100.0...100.0],[-100.0...100.0]}`, meters

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the vector offset in meters from the L1 phase center to the L2 phase center for the antenna with CMR ID value of 214.
`print,/par/antdb/cmr/214/l2_l1`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Mapping between antenna ID's and CMR ID's

`[/par/]/antdb/ids`

Description: Reports the antenna IDs together with associated CMR IDs (if both exist for the same antenna).

Access: read-only

Type: list {id,cmr}

Values: list of `cur_value`'s

Default: list of `cur_value`'s

Usage Guidelines: N/A

Command Examples: Instruct the receiver to display the antenna ID's and their associated CMR ID's.
`print,/par/antdb/ids`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

ID and CMR ID for each antenna in the database

`[/par/]/antdb/ids/id`

Description: Reports a complete list of antenna IDs together with the corresponding CMR IDs in the {ID,CMR_ID} format.

Access: read-only

Type: list {[ID]}

Values: list of `cur_value`'s

Default: list of `cur_value`'s

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report a complete list of antenna IDs together with the corresponding CMR IDs in the {ID,CMR_ID} format.

```
print,/par/antdb/ids/id
list,/par/antdb/ids/id
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

ID and CMR ID for a specific antenna

`[/par/]/antdb/ids/id/<ID>`

Description: Reports the specific antenna ID together with the corresponding CMR ID in the {ID,CMR_ID} format.

Access: read-only

Type: list of two values

Values: list of `cur_value`'s

Default: list of `cur_value`'s

Usage Guidelines: N/A

Command Examples: Instruct the receiver to report the specific antenna ID together with the corresponding CMR ID in the {ID,CMR_ID} format for a Topcon CR4 antenna.

```
print,/par/antdb/ids/id/TPSCR4
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna CMR ID's

`[/par/]/antdb/ids/cmr`

Description:	Reports only those antenna IDs for which CMR IDs are known.
Access:	read-only
Type:	list {[CMR_ID]}
Values:	list of `cur_value`'s
Default:	list of `cur_value`'s
Usage Guidelines:	N/A
Command Examples:	Instruct the receiver to display/list only those antenna IDs for which CMR IDs are known. `print,/par/antdb/ids/id/cmr` `list,/par/antdb/ids/cmr`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

ID and CMR ID for a specific antenna CMR ID

`[/par/]/antdb/ids/cmr/<CMR_ID>`

Description:	Reports the specific antenna ID together with CMR ID in the {ID,CMR_ID} format.
Access:	read-only
Type:	list of two values
Values:	list of `cur_value`'s
Default:	list of `cur_value`'s
Usage Guidelines:	N/A
Command Examples:	Instruct the receiver to report the antenna ID together with CMR ID in the {ID,CMR_ID} format for the antenna with the CMR value of 214. `print,/par/antdb/ids/cmr/214`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

MINTER Parameters

The following parameters allow you to configure the functionality of the MINTER FN button.

Appending data to a specific file

`[/par/]button/file`

Description: Designates a specific, existing file to which the receiver will append new data when data recording is commenced with the FN button.

Access: read & write

Type: string

Values: up to 20 alphanumeric characters

Default: empty string

Usage Guidelines: If you have specified an empty name, the receiver will assign the current log-file an "automatically created name" every time you use MINTER to start data recording. See “create” on page 2-11 for methods used to generate automatic file names.
Alternatively, suppose you have specified a non-empty file name, say NAME. If there is no log file with this name in the receiver memory, pushing the FN key will instruct the receiver to create a new file named `/log/NAME`. Otherwise, the receiver will be using the existing file `/log/NAME` for appending new data.

Command Examples: Log (append) new data to the existing file named `static`.

```
set,/par/button/file,static
```

Or

```
set,button/file,static
```

Query the current status.

```
print,/par/button/file
```

Or

```
print,button/file
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

FN key mode

`[/par/]button/action`

Description: Programs the receiver's reaction after pressing the FN key.

Access: read & write

Type: enumerated

Values: `led | dyn`

- `led` – pressing and holding down the FN key for less than a second, you switch the receiver between standard and extended information modes. Refer to the receiver's Operator's Manual for details about these modes.
- `dyn` – pressing and holding down the FN key for less than a second, you switches the receiver between static and kinematic survey modes when recording a file. The receiver will output an appropriate free-form event to the current log file every time you switches the survey mode.

Default: `led`

Usage Guidelines: You can easily distinguish between static and dynamic visually. If the REC LED blinks green, the current mode is dynamic, if it blinks yellow, you have static on.

Command Examples: Set the receiver's FN button to toggle between the static and dynamic receiver modes.

```
set,/par/button/action,dyn
```

Or

```
set,button/action,dyn
```

Query the current status.

```
print,/par/button/action
```

Or

```
print,button/action
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Initial dynamic mode

`[/par/]button/dyn`

Description: Specifies the initial dynamic mode for all of the new files opened through MINTER when the parameter `/par/button/action` is set to the `dyn` mode.

Access: read & write

Type: enumerated

Values: `static | dynamic`

Default: `static`

Usage Guidelines: N/A

Command Examples: Set the initial dynamic mode for all new log-files created.
`set,/par/button/dyn,dynamic`
Or
`set,button/dyn,dynamic`

Query the current status.
`print,/par/button/dyn`
Or
`print,button/dyn`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Automatic File Rotation Mode (AFRM) via MINTER

`[/par/]button/rot`

Description: Determines whether the MINTER FN button is enabled to turn data logging on and off, or to turn the AFRM function on and off.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the MINTER FN button turns data logging on and off.
- `on` – the MINTER FN button turns AFRM on and off.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable the MINTER FN button to turn the AFRM on.

```
set,/par/button/rot,on
```

Or

```
set,button/rot,on
```

Query the current status.

```
print,/par/button/rot
```

Or

```
print,button/rot
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver behavior and data recording on start-up

`[/par/]button/auto`

Description: Programs the receiver's behavior in the event of a power failure.

Access: read & write

Type: enumerated

Values: `off | on | always`

- `off` – If this parameter is set to this value, the receiver will not resume data logging on start-up.
- `on` – Should a power failure occur when the receiver is logging data, the receiver will automatically open a new file and resume data recording when power has been returned. From a functional point of view, this is equivalent to pushing the FN button once the receiver is powered on again.
- `always` – This case is similar to the previous one except that the auto-start mechanism will be launched at the receiver start time irrespective of whether the power failure occurred while data recording or not.

Default: `off`

Usage Guidelines: Setting this parameter to either `on` or `always` will not make the receiver automatically start when power is restored after a power failure.

Command Examples: Set the receiver to automatically open a new file and resume data recording when power has been returned to the receiver.

```
set,/par/button/auto,on
```

Or

```
set,button/auto,on
```

Query the current status.

```
print,/par/button/auto
```

Or

```
print,button/auto
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Governing Optimized Real Time Application (ORTA) Messages

These parameters are used to tailor ORTA messages to your particular needs.

C/A measurements in rM message

`[/par/]raw/rtm/meas/ca`

Description: Specifies whether the rM message will contain C/A measurements or not.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the rM message will not contain C/A measurements.
- `on` – the rM message will contain C/A measurements.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set the rM message so that it does not contain C/A measurements.

```
set,/par/raw/rtm/meas/ca,off
```

Or

```
set,raw/rtm/meas/ca,off
```

Query the current status.

```
print,/par/raw/rtm/meas/ca
```

Or

```
print,raw/rtm/meas/ca
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P/L1 measurements in rM message

`[/par/]raw/rtm/meas/p1`

Description: Specifies whether the rM message will contain P/L1 measurements or not.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the rM message will not contain P/L1 measurements.
- `on` – the rM message will contain P/L1 measurements.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set the rM message so that it does not contain P/L1 measurements.

```
set,/par/raw/rtm/meas/p1,off
```

Or

```
set,raw/rtm/meas/p1,off
```

Query the current status.

```
print,/par/raw/rtm/meas/p1
```

Or

```
print,raw/rtm/meas/p1
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P/L2 measurements in rM message

`[/par/]raw/rtm/meas/p2`

Description: Specifies whether the rM message will contain P/L2 measurements or not.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the rM message will not contain P/L2 measurements.
- `on` – the rM message will contain P/L2 measurements.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set the rM message so that it does not contain P/L2 measurements.

```
set,/par/raw/rtm/meas/p2,off
```

Or

```
set,raw/rtm/meas/p2,off
```

Query the current status.

```
print,/par/raw/rtm/meas/p2
```

Or

```
print,raw/rtm/meas/p2
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Version of rM message

`[/par/]raw/rtm/ver`

Description: Switches between various versions of the rM messages.

Access: read & write

Type: integer

Values: `[0...2]`

- `0` – the rM message is output as is (i.e, in its original form).
- `1` – the rM message size will be reduced by excluding `u4 word2` (i.e. Doppler) from the structure `SlotRec`.
- `2` – the rM message size will be further reduced by excluding the fields `u2 flags`, `u2 lock` and `u4 word2`. Also, the `i2 clock` field might not contain the clock offset.

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the rM message so that it excludes the Doppler frequency data.

```
set,/par/raw/rtm/ver,1
```

Or

```
set,raw/rtm/ver,1
```

Query the current status.

```
print,/par/raw/rtm/ver
```

Or

```
print,raw/rtm/ver
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.
The `2` value was implemented in firmware 2.4.

See Also: N/A

Time scale for ORTA messages

`[/par/]raw/rtm/tscale`

Description: Specifies the time scale to be referenced in the rE, rM, rV and rT messages.

Access: read & write

Type: enumerated

Values: `gps | glo | utc`

Default: `gps`

Usage Guidelines: N/A

Command Examples: Set the time reference scale to UTC for the ORTA messages.
`set,/par/raw/rtm/tscale,utc`
Or
`set,raw/rtm/tscale,utc`

Query the current status.
`print,/par/raw/rtm/tscale`
Or
`print,raw/rtm/tscale`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For more information about the ORTA messages, see “Optimized Real-Time Application (ORTA) Messages” on page 4-111.

Type of coordinates in rV message

`[/par/]raw/rtm/coord`

Description: Specifies the type of coordinates output in rV message.

Access: read & write

Type: enumerated

Values: `xyz | geo | grid`

- `xyz` – Cartesian coordinates.
- `geo` – Geodetic coordinates.
- `grid` – Grid coordinates.

Default: `xyz`

Usage Guidelines: N/A

Command Examples: Output geodetic coordinates in rV message.

```
set,/par/raw/rtm/coord,geo
```

Or

```
set,raw/rtm/coord,geo
```

Query the current status.

```
print,/par/raw/rtm/coord
```

Or

```
print,raw/rtm/coord
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: For more information about the ORTA messages, see “Optimized Real-Time Application (ORTA) Messages” on page 4-111.

Reference Station Parameters

Reference station output modes

`[/par/]base/mode`

Description: Defines which of the supported differential correction formats will be enabled to output messages to the receiver's ports.

Access: read & write

Type: list {rtcm:boolean,cmr:boolean,jps:boolean,rtcm3:boolean}

Values: `{on|off,on|off,on|off,on|off}`

Default: `{off,off,off,off}`

Usage Guidelines:
- At present, TPS receivers support RTCM 2.x, CMR, TPS and RTCM3 formats. As future differential correction formats are supported, additional boolean parameters will be added.
- All formats can be turned off using the command `set,/par/base/mode/,off`.

Command Examples: Turn off all supported differential correction formats.
```
set,/par/base/mode/,off
```
Or
```
set,base/mode/,off
```

Query the current status.
```
print,/par/base/mode
```
Or
```
print,base/mode
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History:
- This parameter was introduced in the first release of firmware.
- The value `rtcm3` was implemented in firmware 2.4.

See Also: Parameters for `rtcm`, `rtcm3`, `cmr` and `jps`
`/par/base/mode/rtcm`
`/par/base/mode/rtcm3`
`/par/base/mode/cmr`
`/par/base/mode/jps`

Offsets from the marker to the L1 APC

`[/par/]ref/ant/offs`

Description: Specifies the vector components measured from a surveyed point (land mark) to the L1 APC (L1 antenna phase center).

Access: read & write

Type: list [eastOffset:float,northOffset:float,height:float]
eastOffset – East component
northOffset – North component
height – Up component

Values: `[-100.0...100.0,-100.0...100.0,-100.0...100.0]`; meters

Default: `[0.0000,0.0000,0.0000]`

Usage Guidelines: N/A

Command Examples: Set the offset from the marker to the L1 APC of zero (0) in the east and north components and a height offset of 1.72 meters.

```
set,/par/ref/ant/offs,{0.0,0.0,1.72}
```

Or

```
set,ref/ant/offs,{0.0,0.0,1.72}
```

Query the current status.

```
print,/par/ref/ant/offs
```

Or

```
print,ref/ant/offs
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Offsets from the marker to the ARP

`[/par/]ref/ant/arpoffs`

Description: Specifies the vector components measured from a surveyed point (land mark) to the ARP (Antenna Reference Point).

Access: read & write

Type: list [east:float,north:float,height:float]
east – East component
north – North component
height – Up component

Values: `[-100.0...100.0,-100.0...100.0,-100.0...100.0]`; meters

Default: `[0.0000,0.0000,0.0000]`

Usage Guidelines: N/A

Command Examples: Set the vector components measured from a surveyed point to the ARP (Antenna Reference Point) with an east and north component of zero (0) and a height offset of 1.75 meters.
`set,/par/ref/ant/arpoffs,{0.0,0.0,1.75}`
Or
`set,ref/ant/arpoffs,{0.0,0.0,1.75}`

Query the current status.
`print,/par/ref/ant/arpoffs`
Or
`print,ref/ant/arpoffs`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Offsets from L1 APC to L2 APC

`[/par/]ref/ant/l2_l1`

Description: Specifies the vector components measured between the L1 APC (L1 antenna phase center) and the L2 APC (L2 antenna phase center).

Access: read & write

Type: list [eastOffset:float,northOffset:float,height:float]
eastOffset – East component
northOffset – North component
height – Up component

Values: `[-0.1...0.1,-0.1...0.1,-0.1...0.1]`; meters

Default: `[0.0000,0.0000,0.0000]`

Usage Guidelines: This parameter is used at both the base/reference and rover receivers.

Command Examples: Set the vector components measured between the L1 antenna phase center (L1 APC) and the L2 antenna phase center (L2 APC) to zero (0) in east and north, and 0.005 m in height.

```
set,/par/ref/ant/l2_l1,{0.0,0.0,0.005}
```

Or

```
set,ref/ant/l2_l1,{0.0,0.0,0.005}
```

Query the current status.

```
print,/par/ref/ant/l2_l1
```

Or

```
print,ref/ant/l2_l1
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Cartesian reference receiver position for GPS

`[/par/]ref/pos/gps/xyz`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for GPS in Cartesian coordinate systems.

Access: read & write

Type: list [datum,x:float,y:float,z:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *x,y,z* are Cartesian coordinates in meters (float numbers).

Default: `[W84,+6378137.0000,+0.0000,+0.0000]`

Usage Guidelines: Use this parameter if, for some reason, you want to have independent reference position estimates for GPS. For example, you may want to explicitly specify the reference position in WGS-84 for GPS measurements.

Command Examples: Set the GPS Cartesian coordinates to values of x = 2850717.2067
y = 2199375.5841
z = 5247224.5240

```
set,/par/ref/pos/gps/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Or

```
set,ref/pos/gps/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Query the current values set.

```
print,/par/ref/pos/gps/xyz
```

Or

```
print,ref/pos/gps/xyz
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Cartesian reference receiver position for GLONASS

`[/par/]ref/pos/glo/xyz`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for GLONASS in Cartesian coordinate systems.

Access: read & write

Type: list [datum,x:float,y:float,z:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *x,y,z* are Cartesian coordinates in meters (float numbers).

Default: `[W84,+6378137.0000,+0.0000,+0.0000]`

Usage Guidelines: Use this parameter if, for some reason, you want to have independent reference position estimates for GLONASS. For example, you may want to explicitly specify the reference position in PE-90 for GLONASS.

Command Examples: Set the GLONASS Cartesian coordinates to values of x = 2850717.2067
y = 2199375.5841
z = 5247224.5240

```
set,/par/ref/pos/glo/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Or

```
set,ref/pos/glo/
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Query the current values set.

```
print,/par/ref/pos/glo/xyz
```

Or

```
print,ref/pos/glo/xyz
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Cartesian reference receiver position for all satellite systems

`[/par/]ref/pos//xyz`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for both GPS and GLONASS in Cartesian coordinate systems.

Access: write-only

Type: list [datum,x:float,y:float,z:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *x,y,z* are Cartesian coordinates in meters (float numbers).

Default: `[W84,+6378137.0000,+0.0000,+0.0000]`

Usage Guidelines: In most cases, it is recommended to use this parameter as it provides one reference position estimate for both GPS and GLONASS.

Command Examples: Set the GPS and GLONASS Cartesian coordinates to values of
x = 2850717.2067
y = 2199375.5841
z = 5247224.5240

```
set,/par/ref/pos//
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Or

```
set,ref/pos//
xyz,{W84,2850717.2067,2199375.5841,5247224.5240}
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Geodetic reference receiver position for GPS

`[/par/]ref/pos/gps/geo`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for GPS in geodetic coordinate systems.

Access: read & write

Type: list [datum,lat:integer,float,long:integer,float,alt:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *Lat* (latitude) and *long* (longitude) are geodetic coordinates in degrees (*d*), minutes (*m*) and seconds (*s*). *Alt* refers to ellipsoidal height in meters.

Default: `[W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000]`

Usage Guidelines: Use this parameter if, for some reason, you want to have independent reference position estimates for GPS. For example, you may want to explicitly specify the reference position in WGS-84 for GPS measurements.

Command Examples: Set the GPS geodetic reference coordinates to the WGS84 datum values of
Latitude = 55 ^{o} 43‘ 20.0301”North
Longitude = 37 ^{o} 39‘ 02.8007” East
Altitude = +165.8630 meters

```
set,/par/ref/pos/gps/geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Or

```
set,ref/pos/gps/geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Query the current values set.

```
print,/par/ref/pos/gps/geo
```

Or

```
print,ref/pos/gps/geo
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Geodetic reference receiver position for GLONASS

`[/par/]ref/pos/glo/geo`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for GLONASS in geodetic coordinate systems.

Access: read & write

Type: list [datum,lat:integer,float,long:integer,float,alt:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *Lat* (latitude) and *long* (longitude) are geodetic coordinates in degrees (*d*), minutes (*m*) and seconds (*s*). *Alt* refers to ellipsoidal height in meters.

Default: `[W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000]`

Usage Guidelines: Use this parameter if, for some reason, you want to have independent reference position estimates for GLONASS.

Command Examples: Set the GLONASS geodetic reference coordinates to the WGS84 datum values of
Latitude = 55 ^{o} 43‘ 20.0301”North
Longitude = 37 ^{o} 39‘ 02.8007” East
Altitude = +165.8630 meters

```
set,/par/ref/pos/glo/geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Or

```
set,ref/pos/glo/geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Query the current values set.

```
print,/par/ref/pos/glo/geo
```

Or

```
print,ref/pos/glo/geo
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

Geodetic reference receiver position for all satellite systems

`[/par/]ref/pos//geo`

Description: Defines the position of the L1 antenna phase center (APC) of the reference receiver for both GPS and GLONASS in geodetic coordinate systems.

Access: write-only

Type: list [datum,lat:integer,float,long:integer,float,alt:float]
The first argument, *datum*, indicates the datum in which the reference coordinates are specified. *Lat* (latitude) and *long* (longitude) are geodetic coordinates in degrees (*d*), minutes (*m*) and seconds (*s*). *Alt* refers to ellipsoidal height in meters.

Default: `[W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000]`

Usage Guidelines: In most cases, it is recommended to use this parameter as it provides one reference position estimate for both GPS and GLONASS.

Command Examples: Set the GPS and GLONASS geodetic reference coordinates to the WGS84 datum values of

Latitude = 55 ^{o} 43‘ 20.0301”North

Longitude = 37 ^{o} 39‘ 02.8007” East
Altitude = +165.8630 meters

```
set,/par/ref/pos//geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Or

```
set,ref/pos//geo,{W84,N55d43m20.0301s,
E37d,39m,02.8007s,+165.8630}
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: For the list of valid datum identifiers, please see comments to `/par/pos/datum/cur`.

WGS84 Cartesian reference receiver position (for GPS)

`[/par/]ref/syspos/gps/xyz`

Description: Queries the Cartesian position of the L1 antenna phase center (APC) of the reference receiver in WGS84 Datum, the default datum for GPS.

Access: read-only

Type: list [datum,x:float,y:float,z:float]
datum - WGS84 datum, default for GPS.
x,y,z are Cartesian coordinates in meters (float numbers).

Default: `[W84,+6378137.0000,+0.0000,+0.0000]`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/ref/syspos/gps/xyz`
Or
`print,ref/syspos/gps/xyz`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

WGS84 geodetic reference receiver position (for GPS)

`[/par/]ref/syspos/gps/geo`

Description: Queries the geodetic position of the L1 antenna phase center (APC) of the reference receiver in WGS84 datum, the default datum for GPS.

Access: read-only

Type: list [datum,lat:integer,float,long:integer,float,alt:float]
datum - WGS84 datum, default for GPS.
Lat (latitude) and *long* (longitude) are geodetic coordinates in degrees (*d*), minutes (*m*) and seconds (*s*).
Alt refers to ellipsoidal height in meters.

Default: `[W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000]`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/ref/syspos/gps/geo`
Or
`print,ref/syspos/gps/geo`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PZ90 Cartesian reference receiver position (for GLONASS)

`[/par/]ref/syspos/glo/xyz`

Description: Queries the Cartesian position of the L1 antenna phase center (APC) of the reference receiver in PZ90 Datum, the default datum for GLONASS.

Access: read-only

Type: list [datum,x:float,y:float,z:float]
datum - PZ90 datum, default for GLONASS.
x,y,z are Cartesian coordinates in meters (float numbers).

Default: `{P90,+6378137.0000,-6.3781,-1.0000}`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/ref/syspos/glo/xyz`
Or
`print,ref/syspos/glo/xyz`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PZ90 geodetic reference receiver position (for GLONASS)

`[/par/]ref/syspos/glo/geo`

Description: Queries the geodetic position of the L1 antenna phase center (APC) of the reference receiver in PZ90 datum, the default datum for GLONASS.

Access: read-only

Type: list [datum,lat:integer,float,long:integer,float,alt:float]
datum - PZ90 datum, default for GLONASS.
Lat (latitude) and *long* (longitude) are geodetic coordinates in degrees (*d*), minutes (*m*) and seconds (*s*).
Alt refers to ellipsoidal height in meters.

Default: `{P90,S00d00m00.032557s,W000d00m00.206265s,+1.0000}`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/ref/syspos/glo/geo`
Or
`print,ref/syspos/glo/geo`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Coordinates of the ARP

`[/par/]ref/arp/<gps|glo>/<xyz|geo>`

Description: Specifies the geodetic or Cartersian coordinates of the ARP.

Access: read & write

Type: list {datum,x,y,z}
list {datum,lat,lon,alt}

Default: see the parameter `/par/ref/pos`

Usage Guidelines: With these parameters you specify the location of the ARP. This location is then transmitted using RTCM message 24 and RTCM 3.0 messages and is needed at the rover side in order to compute the RTK solution.

Command Examples: Set the Cartesian coordinates.

```
set,/par/ref/arp/gps/
xyz,{W84,3738358.7936,1148173.7076,5021816.7009}
set,/par/ref/arp/glo/
xyz,{W84,3738358.7936,1148173.7076,5021816.7009}
```

Set the geodetic coordinates.

```
set,/par/ref/arp/gps/
geo,{W84,N52d16m37.054502s,E017d04m24.437753s,+125.2958}
set,/par/ref/arp/glo/
geo,{W84,N52d16m37.054502s,E017d04m24.437753s,+125.2958}
```

Query the current values.

```
print,/par/ref/arp/gps/xyz
print,/par/ref/arp/glo/xyz
print,/par/ref/arp/gps/geo
print,/par/ref/arp/glo/geo
```

Receiver Availability: These parameters are not available in the GMS-2 receiver.

History: These parameters were introduced in firmware 2.4.

See Also: N/A

View coordinates of the ARP

`[/par/]ref/sysarp/<gps|glo>/<xyz|geo>`

Description: Reports the geodetic or Cartersian coordinates of the ARP in WGS84 (for GPS) or PZ-90 (for GLONASS).

Access: read-only

Type: list {datum,x,y,z}
list {datum,lat,lon,alt}

Default: see the parameter `/par/ref/pos`

Usage Guidelines: N/A

Command Examples: Query the current values.

```
print,/par/ref/sysarp/gps/xyz
print,/par/ref/sysarp/glo/xyz
print,/par/ref/sysarp/gps/geo
print,/par/ref/sysarp/glo/geo
```

Receiver Availability: These parameters are not available in the GMS-2 receiver.

History: These parameters were introduced in firmware 2.4.

See Also: N/A

Maximum tolerance level for the discrepancy between computed and user-defined position of reference station

`[/par/]ref/limit`

Description: Defines the maximum acceptable difference between the reference receiver own calculated position and the user-specified position.

Access: read & write

Type: float

Values: `[0...10000.0]`; meters

Default: `1000.0`

Usage Guidelines: With this parameter, the user can check the entered reference coordinates for errors.

Should the difference[a] between the currently estimated position and the position defined for the L1 APC by the user exceed the level specified by this parameter, the reference station will stop transmitting any RTK or DGPS messages. Specifically, the receiver will stop transmitting RTCM messages types 1, 3, 9, 18, 19, 20, 21, 22, 31, 32, 34 and CMR messages types 0, 1 and 10 (for more details on CMR message type 10 see "CMR Messages" on page 4-206.)

Command Examples: Set the maximum acceptable difference between the receiver's currently estimated position and the user-specified position to 500.0 meters.

```
set,/par/ref/limit,500
```

Or

```
set,ref/limit,500
```

Query the current status.

```
print,/par/ref/limit
```

Or

```
print,ref/limit
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

a. more precisely, the length of the vector connecting the current receiver position with that specified by the user

Reference position averaging mode

`[/par/]ref/avg/mode`

Description: Enables the receiver to compute its smoothed coordinates by averaging stand-alone position estimates during a specified interval (separate command).

Access: read & write

Type: boolean

Values: `on | off`

Default: `off`

Usage Guidelines: Assuming the reference receiver is set on a point whose precise coordinates are not known, using this command, the receiver will set some "provisional" base reference coordinates, which are obtained by averaging stand-alone position estimates over the specified interval (see command `set,ref/avg/span,...` following).
The absolute coordinates thus estimated will then be used as the receiver's reference position. So, if you run this receiver as RTK base, these coordinates will be transmitted to the rover receivers through the corresponding RTCM messages.
After the rover's (RTK) trajectory is computed with respect to the provisional base reference position, you should take the following steps:

- Refine the base reference coordinates. To do this, you may want to use appropriate measurements from a third receiver with precisely known coordinates. Use post-processing software to compute the baseline "third receiver" - "reference receiver". You normally get quite accurate base coordinates this way.
- Compute the offset between the provisional base coordinates and the refined base coordinates. Use this offset vector to adjust the original RTK trajectory.

Command Examples: Enable the reference position averaging mode.
`set,/par/ref/avg/mode,on`
Or
`set,ref/avg/mode,on`

Query the current status.
`print,/par/ref/avg/mode`
Or
`print,ref/avg/mode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Stand-alone position averaging interval

`[/par/]ref/avg/span`

Description: Specifies the interval over which stand-alone position estimates will be averaging.

Access: read & write

Type: integer

Values: `[0...32000]`; seconds

Default: `180`

Usage Guidelines: Set the amount of time for the receiver to average its stand-alone position to use as the receivers reference position.

Command Examples: Enable the reference position averaging time to 720 seconds (12 minutes).
`set,/par/ref/avg/span,720`
Or
`set,ref/avg/span,720`

Query the current status.
`print,/par/ref/avg/span`
Or
`print,ref/avg/span`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Checking maximum distance between previous reference position and the current averaged position

`[/par/]ref/avg/check`

Description: Enables and disables checking of the maximum distance between the previous reference coordinates of the receiver and the current averaged coordinates of the receiver.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – no checking is performed.
- `on` – after finishing the averaging of stand-alone position, the receiver will compare the distance between the previous reference coordinates of the receiver and the current averaged coordinates of the receiver with a pre-defined limit (specified by the parameter `/par/ref/avg/limit`). If the distance exceeds the limit, the receiver will substitute the previous reference coordinates with the new ones. Otherwise, the previous reference coordinates of the receiver will remain in use.

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable distance checking.
`set,/par/ref/avg/check,on`
Or
`set,ref/avg/check,on`

Query the current status.
`print,/par/ref/avg/check`
Or
`print,ref/avg/check`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced firmware 3.1.

See Also: N/A

Maximum distance between previous reference position and the current averaged position

`[/par/]ref/avg/limit`

Description:	Specifies the maximum distance between the previous reference coordinates of the receiver and the current averaged coordinates of the receiver.
Access:	read & write
Type:	float
Values:	`[0.1...10000.0]`; meters
Default:	`10.0`
Usage Guidelines:	If the distance does not exceed the limit, the receiver will use the previous reference coordinates as the current ones (provided the parameter /par/ref/avg/check is set to `on`).
Command Examples:	Set the maximum distance to 5 meters. `set,/par/ref/avg/limit,5` Or `set,ref/avg/limit,5` Query the current status. `print,/par/ref/avg/limit` Or `print,ref/avg/limit`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced firmware 3.1.
See Also:	N/A

Antenna type descriptor for RTCM 2.3 and 3.0

`[/par/]ref/ant/id`

Description: Specifies the antenna type descriptor for RTCM 2.3 and 3.0.

Access: read & write

Type: string [0...31]

Values: up to 31 alphanumeric characters

Default: empty value

Usage Guidelines: N/A

Command Examples: Specify the antenna descriptor as TOP_CR3_GGD.

```
set,/par/ref/ant/id,TOP_CR3_GGD
```

Or

```
set,ref/ant/id,TOP_CR3_GGD
```

Query the current status.

```
print,/par/ref/ant/id
```

Or

```
print,ref/ant/id
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna setup ID

`[/par/]ref/ant/setup`

Description: Specifies antenna setup ID.

Access: read & write

Type: integer

Values: `[0...255]`

Default: `0`

Usage Guidelines: This parameter is typically used by the differential service provider to inform the user about any change at the reference station that affects the antenna phase center variations.

Command Examples: Set the antenna setup ID to 1.
`set,/par/ref/ant/setup,1`
Or
`set,ref/ant/setup,1`

Query the current status.
`print,/par/ref/ant/setup`
Or
`print,ref/ant/setup`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna serial number

`[/par/]ref/ant/sernum`

Description:	Specifies the individual antenna serial number.
Access:	read & write
Type:	string [0...31]
Values:	up to 31 alphanumeric characters
Default:	empty value
Usage Guidelines:	N/A
Command Examples:	Set the antenna serial number to 0212. `set,/par/ref/ant/sernum,0212` Or `set,ref/ant/sernum,0212` Query the current status. `print,/par/ref/ant/sernum` Or `print,ref/ant/sernum`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in firmware 2.4.
See Also:	N/A

RTCM 2.x-related Reference Station Parameters

RTCM 2.x output mode

`[/par/]base/mode/rtcm`

Description: Configures the output of the RTCM 2.x messages for all of the receivers ports.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – you will disable all of the RTCM 2.x messages for all of the ports.
- `on` – you cannot set this value directly. The receiver will automatically set the parameter to `on` as soon as RTCM 2.x messages are enabled.

Default: `off`

Usage Guidelines: N/A

Command Examples: Disable all RTCM 2.x messages for all receiver ports.
`set,/par/base/mode/rtcm,off`
Or
`set,base/mode/rtcm,off`

Query the current status.
`print,/par/base/mode/rtcm`
Or
`print,base/mode/rtcm`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of local datum for referencing differential corrections from RTCM messages 1,9,20,21,31,34

`[/par/]base/rtcm/base/locdtm`

Description: Allows the reference receiver to transmit differential corrections referenced to a local datum rather than the RTCM standard WGS84 (GPS) and PZ90 (GLONASS).

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: The RTCM standard recommends using WGS84 and PZ90 for referencing GPS and GLONASS differential corrections, respectively [6]. In some cases, however, it can be desirable to transmit corrections referenced to a local datum (recall that the local datum is specified with the parameter `/par/pos/datum/cur`). For example, in code differential, coordinates of the base station can be given in a local datum. In this case, corrections referenced to this local datum may be transmitted to the rover. At the rover side, provided that the same local datum is chosen, a user can obtain the position expressed in the same local datum. Thus, it is possible to obtain the coordinates, expressed in a local datum, without any transformations from local datum to, say, WGS84 datum prior to transmitting the corrections. This procedure provides a comfortable method for obtaining coordinates, expressed in a local datum. However, some limitations of this procedure should be mentioned:

- the rover should know that differential corrections are referenced to a local datum. If a base station serves as a reference for many rovers, each of those rovers should use the local datum, specified at the base station, otherwise the rover coordinates can be distorted;
- on long baselines, referencing the differential corrections to a local datum may introduce an additional error in the coordinates (for more details see [6]).

Be sure that both the base station and the rover use the same setting of this parameter (see the parameters `/par/rtcm/base/locdtm` and `/par/rtcm/rover/locdtm` respectively). If use of local datum is enabled, the same local datum should be specified at both the base station and the rover (see the parameter `/par/pos/datum/cur`). Also, under use of the combined GPS/GLONASS receiver with local datum being turned off, be sure that the same transformation matrix from PZ90 to WGS84 datum (see the parameter `/par/pos/datum/P90/par`) is used at both the base station and the rover.

Command Examples: Enable the use of a local datum as the reference for broadcast differential corrections.

```
set,/par/rtcm/base/locdtm,on
```

Or

```
set,rtcm/base/locdtm,on
```

Query the current status.

```
print,/par/rtcm/base/locdtm
```

Or

```
print,rtcm/base/locdtm
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

`[/par/]base/rtcm/base/locdtm` *(Continued)*

History: This parameter was introduced in the first release of firmware.

See Also: Refer to the parameters `/par/rtcm/base/locdtm`, `/par/rtcm/rover/locdtm` and `/par/pos/datum/P90/par`.

Satellite constellation for RTCM message types 18, 19, 20, 21

`[/par/]rtcm/base/sys`

Description: Instructs the base receiver to include in RTCM message types 18,19,20 and 21 only data associated with the specified satellite constellation.

Access: read & write

Type: boolean

Values: `{on | off,on | off}`
The first and the second values correspond to GPS and GLONASS, respectively. By default, all of available GPS and GLONASS satellites will be taken into account when generating these four message types.

Default: `{on,on}`

Usage Guidelines: N/A

Command Examples: Include RTCM message types 18, 19, 20 and 21 for GPS only.
`set,/par/rtcm/base/sys,{on,off}`
Or
`set,rtcm/base/sys,{on,off}`

Query the current status.
`print,/par/rtcm/base/sys`
Or
`print,rtcm/base/sys`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum number of satellites for RTCM message types 18, 19, 20 and 21

`[/par/]rtcm/base/svm`

Description: Controls the maximum number of satellites to be included in the RTCM message types 18, 19, 20 and 21.

Access: read & write

Type: integer

Values: `[0...127]`
If this mask is set to zero, which is the default value, all of the available satellites will be included in RTCM messages.
If this mask is set to N>0 and the actual number of satellites in sight exceeds this value, only N satellites with highest elevations will be included in RTCM messages.

Default: `0`

Usage Guidelines: N/A

Command Examples: Include RTCM message types 18, 19, 20 and 21 for only the 12 highest satellites in view.
`set,/par/rtcm/base/svm,12`
Or
`set,rtcm/base/svm,12`

Query the current status.
`print,/par/rtcm/base/svm`
Or
`print,rtcm/base/svm`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference station health

`[/par/]rtcm/base/health`

Description: Specifies the health of the reference station.

Access: read & write

Type: enumerated

Values: `good | bad | unknown`

- `good` – normal performance
- `bad` – reference station not working
- `unknown` – reference station transmission not monitored

Default: `good`

Usage Guidelines: N/A

Command Examples: Set the reference station health message to unknown (reference station is not monitored).

```
set,/par/rtcm/base/health,unknown
```

Or

```
set,rtcm/base/health,unknown
```

Query the current status.

```
print,/par/rtcm/base/health
```

Or

```
print,rtcm/base/health
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference station identifier

`[/par/]rtcm/base/stid`

Description: Sets the reference station identification number to aid in distinguishing between reference stations transmitting in the same area.

Access: read & write

Type: integer

Values: `[0...1023]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the reference station identifier (ID#) to 23.
`set,/par/rtcm/base/stid,23`
Or
`set,rtcm/base/stid,23`

Query the current status
`print,/par/rtcm/base/stid`
Or
`print,rtcm/base/stid`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

User-defined text for RTCM message types 16 and 36

`[/par/]rtcm/base/text/<gps|glo>`

Description: Allows for the broadcast of an arbitrary string of as many as 90 characters in RTCM message types 16 and 36, for GPS and GLONASS.

Access: read & write

Type: string

Values: maximum 90 arbitrary characters

Default: `{" "," "}`

Usage Guidelines: The `print` command will return the same string except for the enclosing double quotes, which are omitted.

Command Examples: Specify the message test message to be contained in the GPS RTCM message types 16 and 36

```
set,/par/rtcm/base/text/gps,test message
```

Or

```
set,rtcm/base/text/gps,test message
```

Query the current status

```
print,/par/rtcm/base/text
```

Or

```
print,rtcm/base/text
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Include C/A measurements in RTCM message types 18, 19, 20 and 21

`[/par/]rtcm/base/meas/ca`

Description: Allows for the inclusion of C/A measurements in RTCM message types 18, 19, 20 and 21.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Include C/A measurements in RTCM message types 18, 19, 20 and 21
`set,/par/rtcm/base/meas/ca,on`
Or
`set,rtcm/base/meas/ca,on`

Query the current status.
`print,/par/rtcm/base/meas/ca`
Or
`print,rtcm/base/meas/ca`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Include P/L1 measurements in RTCM message types 18, 19, 20 and 21

`[/par/]rtcm/base/meas/p1`

Description:	Allows for the inclusion of P/L1 measurements in RTCM message types 18, 19, 20 and 21.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Include P/L1 measurements in RTCM message types 18, 19, 20 and 21 `set,/par/rtcm/base/meas/p1,on` Or `set,rtcm/base/meas/p1,on` Query the current status. `print,/par/rtcm/base/meas/p1` Or `print,rtcm/base/meas/p1`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Include P/L2 measurements in RTCM message types 18, 19, 20 and 21

`[/par/]rtcm/base/meas/p2`

Description: Allows for the inclusion of P/L2 measurements in RTCM message types 18, 19, 20 and 21.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Include P/L2 measurements in RTCM message types 18, 19, 20 and 21.
`set,/par/rtcm/base/meas/p2,on`
Or
`set,rtcm/base/meas/p2,on`

Query the current status.
`print,/par/rtcm/base/meas/p2`
Or
`print,rtcm/base/meas/p2`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use smoothed psuedoranges in RTCM message types 18, 19, 20 and 21

`[/par/]rtcm/base/smooth`

Description: Specifies to use smoothed pseudoranges in RTCM message types 18, 19, 20 and 21.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Use smoothed pseudoranges in RTCM message types 18, 19, 20 and 21.
`set,/par/rtcm/base/smooth,on`
Or
`set,rtcm/base/smooth,on`

Query the current status.
`print,/par/rtcm/base/smooth`
Or
`print,rtcm/base/smooth`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Enable delimiting characters at the end of RTCM messages

`[/par/]rtcm/base/end`

Description: Permits the use of delimiting characters at the end of every RTCM message.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: If this parameter is set to `on`, the receiver will automatically insert up to two delimiting characters at the end of every RTCM message (these characters are specified by the value of `/par/rtcm/base/es` parameter, see below).

Command Examples: Enable the use of delimiting characters at the end of every RTCM message.

```
set,/par/rtcm/base/end,on
```

Or

```
set,rtcm/base/end,on
```

Query the current status.

```
print,/par/rtcm/base/end
```

Or

```
print,rtcm/base/end
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Specify delimiting character(s) at the end of RTCM message

`[/par/]rtcm/base/es`

Description: Determines up to two delimiting characters that will be added to the end of every RTCM message.

Access: read & write

Type: list {integer,integer} (ASCII table)

Values: `{[-1...127],[-1...127]}`

Default: `{0x0D,0x0A}`

Usage Guidelines: The values are drawn from the ASCII table. "-1" means that no delimiting character is associated with the corresponding argument. For example, {65,-1} will instruct the receiver to insert only one ASCII character (specifically, the letter "A") after every RTCM message. Similarly, if both *s1* and *s2* are equal to "-1", no delimiting characters will be used. The default: {0xD,0xA}.

Command Examples: **#1**: The following command instructs the receiver to insert a single delimiting character, the ASCII letter A, at the end of every RTCM message

```
set,/par/rtcm/base/es,{65,-1}
```

Or

```
set,rtcm/base/es,{65,-1}
```

#2: This commands specifies that no delimiting characters will be used

```
set,/par/rtcm/base/es,{-1,-1}
```

Or

```
set,rtcm/base/es,{-1,-1}
```

Query the current status

```
print,/par/rtcm/base/es
```

Or

```
print,rtcm/base/es
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RTCM version used at the reference station

`[/par/]rtcm/base/ver`

Description: Specifies which of the supported versions of RTCM messages will be used to broadcast corrections.

Access: read & write

Type: enumerated

Values: `v2.1 | v2.2 | v2.3 | v.3.0`

Default: `v2.3`

Usage Guidelines:
- The parameter is intended to maintain back-compatibility.
- RTCM version 2.x and RTCM version 3.0, although these versions have both been developed by RTCM SC-104 for supporting differential GNSS service, they have remarkable differences in the messaging structure.

Command Examples: Assume you are going to use your receiver as RTK base. If you set the parameter `/par/rtcm/base/ver` to `v2.2`, the receiver will be transmitting messages according to the RTCM version 2.2 standard.

```
set,/par/rtcm/base/ver,v2.2
```

Or

```
set,rtcm/base/ver,v2.2
```

Query the current status.

```
print,/par/rtcm/base/ver
```

Or

```
print,rtcm/base/ver
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Range rate correction to zero in RTCM messages

`[/par/]rtcm/base/zrate/<gps|glo>`

Description: Instructs the receiver to set the rate of change of the pseudorange corrections to zero in RTCM messages 1, 9 and 31, 34.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: N/A

Command Examples: Set the range rate correction for GPS to zero.

```
set,/par/rtcm/base/zrate/gps,y
```

Or

```
set,rtcm/base/zrate/gps,y
```

Query the current status.

```
print,/par/rtcm/base/zrate/glo
```

Or

```
print,rtcm/base/zrate/glo
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History:
- This parameter was introduced in firmware 2.4.
- The default value `n` was replaced with `y` starting with firmware 3.1.

See Also: N/A

CMR-related Reference Station Parameters

CMR output mode

`[/par/]base/mode/cmr`

Description:	Configures the output of the CMR messages for all of the receivers ports.
Access:	read & write
Type:	boolean
Values:	`off	on` By setting this parameter to "off" you will disable all of the CMR messages for all of the ports. This parameter cannot be set to "on". The receiver will automatically set this parameter to "on" as soon as CMR messages are enabled.
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Disable all CMR messages for all receiver ports. `set,/par/base/mode/cmr,off` Or `set,base/mode/cmr,off` Query the current status. `print,/par/base/mode/cmr` Or `print,base/mode/cmr`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Reference receiver's motion state (this flag is used in CMR message type1)

`[/par/]cmr/base/motion`

Description:	Specifies whether the reference receiver is moving (kinematic), fixed (static) or its motion state is undefined (unknown).
Access:	read & write
Type:	enumerated
Values:	`unknown	static	kinematic`
Default:	`unknown`
Usage Guidelines:	If the parameter is set to "kinematic", CMR message type 1 (this is entitled *Reference Station Coordinates*) will use the current position estimate computed by the receiver (it can be an RTK, DGPS or single-point position estimate depending on which positioning mode is on).
Command Examples:	Command to specify the reference receiver is fixed (static). `set,/par/cmr/base/motion,static` Or `set,cmr/base/motion,static` Query the current status. `print,/par/cmr/base/motion` Or `print,cmr/base/motion`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Short station ID, COGO core, long station ID (used in CMR message type 2)

`[/par/]cmr/base/desc`

Description: Defines the short station ID, COGO code and long station ID descriptors used in CMR message type 2.

Access: read & write

Type: list

Values: `{string1,string2,string3}`

Default: `{"","",""}`

Usage Guidelines: Three strings specifying “short station ID”, “COGO code”, and “long station ID” are in this order:

- string1 - up to 8 characters long,
- string2 - up to 16 characters long,
- string3 - up to 50 characters long.

Command Examples: Set CMR message type 2 descriptors to the following:
Short station ID – *Msc*
COGO Code – *Field*
Long station ID – *""*

```
set,/par/cmr/base/desc,{Msc,Field,""}
```

Or

```
set,cmr/base/desc,{Msc,Field,""}
```

Query the current status.

```
print,/par/cmr/base/desc
```

Or

```
print,cmr/base/desc
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference station identifier

`[/par/]cmr/base/stid`

Description: Sets the reference station identification number to aid in distinguishing between reference stations transmitting in the same area.

Access: read & write

Type: integer

Values: `[0...31]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the reference station identifier (ID#) to 24.

```
set,/par/cmr/base/stid,24
```

Or

```
set,cmr/base/stid,24
```

Query the current status.

```
print,/par/cmr/base/stid
```

Or

```
print,cmr/base/stid
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Include P/L2 measurements in CMR observation messages

`[/par/]cmr/base/meas/p2`

Description: Allows for the inclusion of P/L2 measurements in CMR observation messages.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Include P/L2 measurements in CMR observation messages.
`set,/par/cmr/base/meas/p2,on`
Or
`set,cmr/base/meas/p2,on`

Query the current status.
`print,/par/cmr/base/meas/p2`
Or
`print,cmr/base/meas/p2`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Include P/L1 and C/A/L1 measurements in CMR observation messages

`[/par/]cmr/base/meas/pcode`

Description: Allows for the inclusion of P/L1 and C/A measurements in CMR observation messages.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Include P/L1 and C/A measurements in CMR observation messages.
`set,/par/cmr/base/meas/pcode,on`
Or
`set,cmr/base/meas/pcode,on`

Query the current status.
`print,/par/cmr/base/meas/pcode`
Or
`print,cmr/base/meas/pcode`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Message type used to transmit GLONASS measurements from the reference station

`[/par/]cmr/base/glo/type`

Description: Specifies which message type will be used to transmit GLONASS measurements from the reference station.

Access: read & write

Type: integer

Values: `[3...7]`

Default: `3`

Usage Guidelines: Since the CMR format does not allow for any predefined message type for GLONASS measurements, you must specify a message type for GLONASS raw data on your own. This is the purpose that the parameter serves.
Note: Since some new CMR message types may appear in the future, be sure that the message type assigned to GLONASS measurements is different from all the other CMR message types. Should a conflict due to ambiguous message types occur, you may need to re-define the message type associated with GLONASS measurements (just choose any unused number within a range of 3…7).
In addition, ensure that both the reference station and the rover receiver use the same message type for GLONASS data (see `/par/cmr/rover/glo/type` below).

Command Examples: Set message type to transmit GLONASS measurements from the reference station.

```
set,/par/cmr/base/glo/type,5
```

Or

```
set,cmr/base/glo/type,5
```

Query the current status.

```
print,/par/cmr/base/glo/type
```

Or

```
print,cmr/base/glo/type
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Antenna type used

`[/par/]cmr/base/ant/type`

Description: Assigns a specific CMR antenna numeric identifier to the type of antenna being used at the reference station.

Access: read & write

Type: integer

Values: `[0...255]`

Default: `0`

Usage Guidelines: For a relationship between your antenna type and the corresponding CMR antenna ID, use, for example, the parameter `/par/antdb/ids`.

Command Examples: Set the antenna ID to 217.
`set,/par/cmr/base/ant/type,217`
Or
`set,cmr/base/ant/type,217`

Query the current status.
`print,/par/cmr/base/ant/type`
Or
`print,cmr/base/ant/type`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Receiver type used

`[/par/]cmr/base/rcv/type`

Description: Assigns a specific CMR antenna numeric identifier to the type of receiver being used at the reference station.

Access: read & write

Type: integer

Values: `[0...255]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the receiver type to 15.

```
set,/par/cmr/base/rcv/type,15
```

Or

```
set,cmr/base/rcv/type,15
```

Query the current status.

```
print,/par/cmr/base/rcv/type
```

Or

```
print,cmr/base/rcv/type
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

RTCM 3-related Reference Station Parameters

Reference station ID for RTCM 3 messages

`[/par/]rtcm3/base/stid`

Description: Specifies the reference station ID for RTCM 3 messages.

Access: read & write

Type: integer

Values: `[0...4095]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set ID to 99.

```
set,/par/rtcm3/base/stid,99
```

Or

```
set,rtcm3/base/stid,99
```

Query the current status.

```
print,/par/rtcm3/base/stid
```

Or

```
print,rtcm3/base/stid
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

String for proprietary RTCM 3.0 text message

`[/par/]rtcm3/base/text`

Description: Specifies a string that will be included into proprietary RTCM 3.0 text message (message type 4091).

Access: read & write

Type: string [0...127]

Values: up to 127 alphanumeric characters

Default: `empty string`

Usage Guidelines: N/A

Command Examples: Use the word `test` in message 4091.

```
set,/par/rtcm3/base/text,test
```

Or

```
set,rtcm3/base/text,test
```

Query the current status.

```
print,/par/rtcm3/base/text
```

Or

```
print,rtcm3/base/text
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

TPS-related Reference Station Parameters

Currently, there is only one parameter in this topic.

TPS output mode

`[/par/]base/mode/jps`

Description: Restricts the receiver by disallowing the output of any TPS messages to the ports that have been enabled for the output of jps/Bl.

Access: read & write

Type: boolean

Values: `off | on`
By setting this parameter to `off`, you will forbid the receiver to output any TPS messages to the ports that have been enabled for output of jps/BI.
You cannot set this parameter to `on`. The receiver will automatically set this parameter to `on` as soon as you enable TPS messages.

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the reference receiver not to output any TPS messages to the ports that have been enabled for output of jps/BI.
`set,/par/base/mode/jps,off`
Or
`set,base/mode/jps,off`

Query the current status.
`print,/par/base/mode/jps`
Or
`print,base/mode/jps`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Rover Station Parameters

Input mode

`[/par/]rover/mode`

Description: Specifies a list of the differential correction formats enabled in the receiver.

Access: read & write

Type: list of four boolean values corresponding to RTCM 2.x, CMR, TPS, and RTCM 3

Values: `{[on|off],[on|off],[on|off],[on|off]}`

Default: `{off,off,off,off}`

Usage Guidelines: You can switch all of the ports running in rtcm, cmr, tps and rtcm3 modes back to command mode by using the command `set,/par/rover/mode/,off`. This will set all of the booleans in the list to `off`.

Command Examples: Instruct the reference receiver not to output any TPS messages to the ports that have been enabled for output of jps/BI.

```
set,/par/base/mode/jps,off
```

Or

```
set,base/mode/jps,off
```

Query the current status.

```
print,/par/base/mode/jps
```

Or

```
print,base/mode/jps
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: Parameters `/par/rover/mode/rtcm`, `/par/rover/mode/cmr`, `/par/rover/mode/jps` and `/par/rover/mode/rtcm3`.

Receive status of the reference station position on the rover side

`[/par/]rover/base/pos/got/<gps|glo>/valid`

Description:	Reports whether the reference station position was received on the rover side.
Access:	read-only
Type:	boolean
Values:	`off | on`
Default:	`cur_value`
Usage Guidelines:	• The reference station's GPS position are retrieved from RTCM message type 3 or CMR message type 1. Once these data are received from the base via the radio link, the parameter `/par/rover/base/pos/got/gps/valid` is set to `on`. • The reference station's GLONASS position are retrieved from RTCM message type 32. Once these data are received from the base via the radio link, the parameter `/par/rover/base/pos/got/glo/valid` is set to `on`.
Command Examples:	Query the current status. `print,/par/rover/base/pos/got/gps/valid` Or `print,rover/base/pos/got/glo/valid`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Received position of the reference station

`[/par/]rover/base/pos/got/<gps|glo>/<xyz|geo>`

Description: Reports the reference station position received on the rover side.

Access: read-only

Type:
`{datum,x,y,z}`
`{datum,lat,lon,alt}`

Values: see `/par/ref/pos...` parameter

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/rover/base/pos/got/gps/xyz`
Or
`print,rover/base/pos/got/glo/geo`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Marker to ARP offsets

`[/par/]rover/base/ant/got/offs`

Description: Reports the marker to ARP offsets obtained via radio data link.

Access: read-only

Type: `{EastOffset,NorthOffset,Height}`

Values: N/A

Default: `cur_value`

Usage Guidelines: N/A

Command Examples: Query the current values.
`print,/par/rover/base/ant/got/offs`
Or
`print,rover/base/ant/got/offs`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

L1 antenna phase center to L2 antenna phase center vector offset

`[/par/]rover/base/ant/got/l2_l1`

Description:	Reports the L1 antenna phase center to L2 antenna phase center vector offset obtained via radio data link.
Access:	read-only
Type:	`{EastOffset,NorthOffset,Height}`
Values:	N/A
Default:	`cur_value`
Usage Guidelines:	N/A
Command Examples:	Query the current values. `print,/par/rover/base/ant/got/l2_l1` Or `print,rover/base/ant/got/l2_l1`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Alternative position for the reference station

`[/par/]rover/base/pos/fix/<xyz|geo>`

Description: Specifies an alternative position for the reference station.

Access: read & write

Type:
`{datum,x,y,z}`
`{datum,lat,lon,alt}`

Values: see `/par/ref/pos...` parameter

Default: `cur_value`

Usage Guidelines:
- You can, if necessary, discard the reference station's coordinates received via the radio data link and start using an alternative coordinate set instead.
- Currently only two datums, WGS84 and PZ-90, can be specified by means of this parameter.

Command Examples: Specify alternative geodesic or Cartesian coordinates for the reference station.

```
set,/par/rover/base/pos/fix/geo,...
```
Or
```
set,/par/rover/base/pos/fix/xyz,...
```

Query the current status.

```
print,/par/rover/base/pos/fix/geo
```
Or
```
print,/par/rover/base/pos/fix/xyz
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Alternative marker to ARP offsets for the reference station's antenna

`[/par/]rover/base/ant/fix/offs`

Description: Specifies an alternative marker to ARP offsets for the reference station's antenna.

Access: read & write

Type: `{EastOffset,NorthOffset,Height}`

Values:
- `EastOffset` – East component
- `NorthOffset` – North component
- `Height` – Up component

Each component may vary between -100 and +100 meters.

Default: `cur_value`

Usage Guidelines: If you for some reason do not want to use the marker to ARP offsets obtained via the radio data link, you can use this parameter to specify alternative values.

Command Examples: Query the current values.
`print,/par/rover/base/ant/fix/offs`
Or
`print,/par/rover/base/ant/fix/offs`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Alternative L1 antenna phase center to L2 antenna phase center vector offset

`[/par/]rover/base/ant/fix/l2_l1`

Description: Specifies an alternative L1 antenna phase center to L2 antenna phase center vector offset on the rover side.

Access: read & write

Type: `{EastOffset,NorthOffset,Height}`

Values:
- `EastOffset` – East component
- `NorthOffset` – North component
- `Height` – Up component

Each component may vary between -0.1 and +0.1 meters.

Default: `cur_value`

Usage Guidelines: If you for some reason do not want to use the L1 antenna phase center to L2 antenna phase center vector offset obtained via the radio data link, you can use this parameter to specify alternative values.

Command Examples: Query the current values.
```
print,/par/rover/base/ant/fix/l2_l1
```
Or
```
print,/par/rover/base/ant/fix/l2_l1
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Source of reference station position

`[/par/]rover/base/pos/cur`

Description: Defines the source from which the reference station position is taken.

Access: read & write

Type: enumerated

Values: `got|fix`

- `got` – the RTK engine will use one of the reference station's position sets received via a radio data link. Recall that generally speaking the reference station may be transmitting GPS only position, GLONASS only position, or both GPS and GLONASS position data.
- `fix` – the RTK engine will use the reference station's alternative coordinates (those explicitly entered on the rover side) and accessible through either `/par/rover/base/pos/fix/xyz` or `/par/rover/base/pos/fix/geo`.

Default: `got`

Usage Guidelines: N/A

Command Examples: Set the parameter to `got`.

```
set,/par/rover/base/pos/cur,got
```

Or

```
set,rover/base/pos/cur,got
```

Query the current value.

```
print,/par/rover/base/pos/cur
```

Or

```
print,rover/base/pos/cur
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Source of marker to ARP offsets

`[/par/]rover/base/ant/cur`

Description: Defines the source from which the marker to ARP offsets are obtained.

Access: read & write

Type: enumerated

Values: `got|fix`

- `got` – the RTK engine will use the Marker to ARP offsets received from the reference station via the radio data link.
- `fix` – the RTK engine will use the alternative offsets that you can specify with the command `set,/par/rover/base/ant/fix/offs,...`.

Default: `got`

Usage Guidelines: N/A

Command Examples: Set the parameter to `got`.

```
set,/par/rover/base/ant/cur,got
```

Or

```
set,rover/base/ant/cur,got
```

Query the current value.

```
print,/par/rover/base/ant/cur
```

Or

```
print,rover/base/ant/cur
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference station position used in RTK

`[/par/]pos/pd/ref/pos/<xyz|geo>`

Description: Reports explicit coordinates, in Cartesian and geodetic format, used in RTK as the reference station position.

Access: read-only

Type:
`{datum,x,y,z}`
`{datum,lat,lon,alt}`

Values: see `/par/ref/pos...` parameter

Default: `cur_value`

Usage Guidelines: If the reference coordinates are undefined (which is the case if you have just cleaned the receiver NVRAM or turned power off and on after having set the parameter `/par/pos/pd/ref/keep` to `off`), the corresponding two commands will return the following values:
`{UNDEF,+6378137.0000,+0.0000,+0.0000}`
`{UNDEF,N00d00m00.000000s,E000d00m00.000000s,+0.0000}`
The datum name `UNDEF` indicates that the truth reference station coordinates are undefined or unavailable.

Command Examples: Query the current value.
`print,/par/pos/pd/ref/pos/xyz`
Or
`print,/par/pos/pd/ref/pos/geo`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use GPS or GLONASS reference position

`[/par/]ref/src`

Description: Instructs the rover which of the position sets received from the reference station will be used in RTK.

Access: read & write

Type: enumerated

Values: `gps|glo|any`

- `gps` – the receiver will not start RTK until a GPS position is received from the reference station.
- `glo` – the receiver will not start RTK until a GLONASS position is received from the reference station.
- `any` – thereceiver will use the very first of the position sets received from the reference station, which currently can be either a GPS or GLONASS position set.

Default: `any`

Usage Guidelines: N/A

Command Examples: Set the parameter to `got`.

```
set,/par/rover/base/ant/cur,got
```

Or

```
set,rover/base/ant/cur,got
```

Query the current value.

```
print,/par/rover/base/ant/cur
```

Or

```
print,rover/base/ant/cur
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Antenna ID transmitted in RTCM message 23, RTCM 3.0, or CMR messages

`[/par/]rover/base/ant/got/id`

Description: Informs you about the antenna ID used at the reference station.

Access: read-only

Type: string [0...31]

Values: up to 31 alphanumeric characters

Default: empty string

Usage Guidelines: If the antenna ID begins with a non-digit character, it is an RTCM antenna descriptor. Otherwise it is a CMR numerical antenna ID formatted as decimal.

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/id`
Or
`print,rover/base/ant/got/id`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Antenna serial number transmitted in RTCM message 23 and RTCM 3.0

`[/par/]rover/base/ant/got/sernum`

Description: Informs you about the antenna serial number used at the reference station.

Access: read-only

Type: string [0...31]

Values: up to 31 alphanumeric characters

Default: empty string

Usage Guidelines: N/A

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/sernum`
Or
`print,rover/base/ant/got/sernum`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Reference antenna setup ID transmitted in RTCM message 23 and RTCM 3.0

`[/par/]rover/base/ant/got/setup`

Description: Informs you about the antenna setup ID used at the reference station.

Access: read-only

Type: integer

Values: `[0...255]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/setup`
Or
`print,rover/base/ant/got/setup`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Reference antenna offset information transmitted in RTCM message 23 and RTCM 3.0

`[/par/]rover/base/ant/got/m_offs`

Description: Informs you about the reference antenna offset type and value used at the reference station.

Access: read-only

Type: list of two values; see parameters that follow

Values: see parameters that follow

Default: `{l1pc,{0.0000,0.0000,0.0000}}`

Usage Guidelines: N/A

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/m_offs`
Or
`print,rover/base/ant/got/m_offs`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: `/par/rover/base/ant/got/m_offs/type`
`/par/rover/base/ant/got/m_offs/val`

Reference antenna offset type transmitted in RTCM message 23 and RTCM 3.0

`[/par/]rover/base/ant/got/m_offs/type`

Description: Informs you about the reference antenna offset type used at the reference station.

Access: read-only

Type: enumerated

Values: `l1pc | arp`

- `l1pc` – indicates that the corresponding offset values in the `/par/rover/base/ant/got/m_offs/val` parameter are represented as the vector components (east, north, height) from the land mark to the L1 phase center.
- `arp` – indicates that the corresponding offset values in the `/par/rover/base/ant/got/m_offs/val` parameter are represented as the vector components (east, north, height) from the land mark to the ARP.

Default: `l1pc`

Usage Guidelines: N/A

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/m_offs/type`
Or
`print,rover/base/ant/got/m_offs/type`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Reference antenna vector offset transmitted in RTCM message 23 and RTCM 3.0

`[/par/]rover/base/ant/got/m_offs/val`

Description: Informs you about the reference antenna vector offset from the land mark to `l1pc` or `arp` depending on the offset type used at the reference antenna side.

Access: read-only

Type: list {east,north,height} of float

Values: `{[-100.0...100.0],[-100.0...100.0],[-100.0...100.0]}`; meters

Default: `{0.0,0.0,0.0}`

Usage Guidelines: N/A

Command Examples: Query the current value.
`print,/par/rover/base/ant/got/m_offs/val`
Or
`print,rover/base/ant/got/m_offs/val`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

RTCM 2.x-related Rover Station Parameters

RTCM input mode

`[/par/]rover/mode/rtcm`

Description: Reports the status of RTCM input mode in the receiver.

Access: read & write

Type: boolean

Values: `on|off`

- By setting this parameter to `off`, you will switch all of the ports running in rtcm mode back to cmd mode.
- Note, however, that you need not run the `set,/par/rover/mode/rtcm,on` command to set this parameter to `on`. As soon as you set some of the ports to rtcm mode (using `set,/par/dev/ser/a/imode,rtcm` and the like commands), your receiver will automatically set this parameter to `on`.

Default: `off`

Usage Guidelines: N/A

Command Examples: Query the current status.
`print,/par/rover/mode/rtcm`
Or
`print,rover/mode/rtcm`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Parameters `/par/rover/mode`, `/par/rover/mode/cmr`, `/par/rover/mode/jps` and `/par/rover/mode/rtcm3`.

Use the local datum for referencing differential corrections transmitted in RTCM messages types 1, 9, 20, 21, 31, 34

`[/par/]rtcm/rover/locdtm`

Description: Enables and disables the rover to use the local datum for referencing differential corrections transmitted in RTCM messages types 1,9,20,21,31,34.

Access: read & write

Type: boolean

Values: `off|on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use local datum.

```
set,/par/rtcm/rover/locdtm,on
```

Or

```
set,rtcm/rover/locdtm,on
```

Query the current status.

```
print,/par/rtcm/rover/locdtm
```

Or

```
print,rtcm/rover/locdtm
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use data from not monitored reference station

`[/par/]rtcm/rover/usenm`

Description: Uses data received from a reference station whose health status code is set to 110. According to the RTCM standard, this code indicates that data transmitted by this station is *not monitored*.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Instruct the receiver not to use data from not monitored reference stations.
`set,/par/rtcm/rover/usenm,off`
Or
`set,rtcm/rover/usenm,off`

Query the current status.
`print,/par/rtcm/rover/usenm`
Or
`print,rtcm/rover/usenm`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Verify the field sequence number from the RTCM message header

`[/par/]rtcm/rover/seqnum`

Description: Verifies the field sequence number from the RTCM message header when calculating data link quality.

Access: read & write

Type: boolean

Values: `off|on`

Default: `on`

Usage Guidelines: An RTCM message has a data field called *sequence number*. Sequence numbers allow the receiver to check whether any messages have been lost when receiving RTCM data.
Such checking is enabled by default. If the receiver detects that a message is lost (i.e. the difference between the current and the previous sequence numbers is not equal to unity), the *bad message counter* will be incremented.

Command Examples: Instruct the receiver to check for lost messages in RTCM stream.
`set,/par/rtcm/rover/seqnum,on`
Or
`set,rtcm/rover/seqnum,on`

Query the current status.
`print,/par/rtcm/rover/seqnum`
Or
`print,rtcm/rover/seqnum`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Multiple message indicator for RTCM messages

`[/par/]rtcm/rover/mmi`

Description: Instructs the receiver to use the *multiple message indicator* flag to identify the last message in a group of RTCM messages 18/19 and 20/21 referenced to the same time.

Access: read & write

Type: enumerated

Values: `off|on|def`

- `off` – the rover receiver will have to wait for RTCM data corresponding to the next epoch to arrive.
- `on` – the rover receiver will always verify the flag. This is expected to reduce the latency time since in this case the rover receiver needn't wait for arriving RTCM messages referenced to the next epoch. Note, however, that this will be possible only on condition that the Multiple Message Indicator behaves exactly as it is specified in version 2.3. Otherwise the data received may be interpreted incorrectly.
- `def` – the rover receiver will interpret the flag depending on the RTCM version used, i.e. `off` for versions 2.1 and 2.2, `on` for version 2.3.

Default: `off`

Usage Guidelines:

- Different manufacturers have interpreted this flag differently, which resulted in incompatibility between the formats used by different developers.
- The RTCM standard version 2.3, unlike the version 2.2, defines this flag explicitly and unambiguously.
- This flag will allow a TPS receiver configured as a rover to be capable of using RTCM messages transmitted by other manufacturers' reference receivers.
- It is recommended to set the mode to `on` if TPS receivers are used on both ends or if it is known in advance that the reference receiver supports RTCM version 2.3. Otherwise selecting this mode can cause malfunction of the rover receiver in RTK.

Command Examples: Instruct the receiver to set this parameter to `on`.

```
set,/par/rtcm/rover/mmi,on
```

Or

```
set,rtcm/rover/mmi,on
```

Query the current status.

```
print,/par/rtcm/rover/mmi
```

Or

```
print,rtcm/rover/mmi
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Complete epoch received logic for RTCM messages

`[/par/]rtcm/rover/full`

Description: Specifies if the *complete epoch received* logic is applied while the receiver decoding RTCM data for RTK.

Access: read & write

Type: enumerated

Values: `off|on|def`

- `off` – the logic is turned off.
- `on` – the logic is turned on.
- `def` – the logic is either turned on or off depending on the parameter `/par/rtcm/rover/ver`, i.e. it is turned off for the RTCM versions earlier than 2.3, and turned on for version 2.3 and the later ones.

Default: `def`

Usage Guidelines:

- The term *complete epoch received* means the receiver obtaining all the data necessary for the given epoch. For example, for a GNSS dual frequency receiver the term means the receiver having acquired the code and phase measurements on both frequencies for at least one satellite for either constellation.
- In RTCM versions earlier than 2.3, due to different interpretation of the RTCM standard by different manufacturers, it is not always possible to identify the end of epoch based on the RTCM format itself. In this case the receiver can either wait when an RTCM message with a different time arrives (so that to extract the current epoch from this message), or apply the *complete epoch received* logic. The latter can decrease the RTK corrections' latency by eliminating the delay required for receiving the first message referenced to the next epoch (typically 1 second).

Command Examples: Instruct the receiver to set this parameter to `on`.

```
set,/par/rtcm/rover/full,on
```

Or

```
set,rtcm/rover/full,on
```

Query the current status.

```
print,/par/rtcm/rover/full
```

Or

```
print,rtcm/rover/full
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RTCM 2.x version used at the rover

`[/par/]rtcm/rover/ver`

Description: Specifies the RTCM 2.x version used at the rover.

Access: read & write

Type: enumerated

Values: `v2.1|v2.2|v2.3`

Default: `v2.3`

Usage Guidelines:

- This parameter is intended to maintain back-compatibility. It allows you to use TPS receivers together with third-party receivers in your RTK projects.
- Assume you are going to use your receiver as RTK rover. If you set the parameter `/par/rtcm/rover/ver` to `v2.1`, this will enable the rover to receive and use RTCM version 2.1 messages.

Command Examples: Instruct the receiver to use RTCM version 2.1.

```
set,/par/rtcm/rover/ver,v2.1
```

Or

```
set,rtcm/rover/ver,v2.1
```

Query the current status.

```
print,/par/rtcm/rover/ver
```

Or

```
print,rtcm/rover/ver
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reference point for antenna coordinates in RTCM 2.x messages

`[/par/]rtcm/rover/refsrc`

Description: Specifies the antenna measurement point to which the reference antenna coordinates transmitted in RTCM 2.x messages refer.

Access: read & write

Type: enumerated

Values: `auto|l1pc|arp`

- `auto` – If both message sets (3/32/22 and 23/24) are transmitted in the same data stream, the rover receiver will use the Antenna Reference Point (ARP) coordinates (message type 24). If only one of the message sets is transmitted, the rover receiver will automatically extract the antenna coordinates available in the given data stream and applies them to the RTK engine.
- `l1pc` – the rover receiver will use the coordinates of the antenna's L1 phase center extracted from message types 3 or 32.
- `arp` – the rover receiver will use the coordinates of the ARP extracted from message type 24.

Default: `auto`

Usage Guidelines:

- The RTCM standard version 2.3 supports message type 24 which provides the exact location of the reference station and the antenna height as the distance to the ARP. Recall that the previous versions of the standard use the message types 3 and 32 to broadcast the coordinates of the reference antenna and these messages contain the coordinates of the antenna's L1 phase center.
- With this parameter you maintain compatibility between various ways of expressing the reference antenna position.
- It is recommended that you use the default value unless you are completely sure of the message set being transmitted by the reference station.

Command Examples: Instruct the receiver to use the default value.

```
set,/par/rtcm/rover/refsrc,auto
```

Or

```
set,rtcm/rover/refsrc,auto
```

Query the current status.

```
print,/par/rtcm/rover/refsrc
```

Or

```
print,rtcm/rover/refsrc
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

Reset RTCM 2.x decoders

`[/par/]rtcm/rover/reset`

Description: Instructs the receiver to reset RTCM 2.x decoders.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – ignored.
- `on` – the receiver resets RTCM decoders.

Default: `off`

Usage Guidelines:

- After executing the `set` command, the parameter's default value `off` is immediately restored.
- You may use this parameter to reset the logic associated with the value used in the `/par/rtcm/rover/refsrc` parameter.

Command Examples: Instruct the receiver to reset RTCM decoders.

```
set,/par/rtcm/rover/reset,on
```

Or

```
set,rtcm/rover/reset,on
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: N/A

CMR-related Rover Station Parameters

CMR input mode

`[/par/]rover/mode/cmr`

Description: Reports the status of CMR input mode in the receiver.

Access: read & write

Type: boolean

Values: `off|on`

- By setting this parameter to `off`, you will switch all of the ports running in cmr mode back to cmd mode.
- Do not use the `set,/par/rover/mode/cmr,on` command. As soon as you set some of the ports to cmr mode (e.g., `set,/par/dev/ser/a/imode,cmr`), the receiver will automatically set this parameter to `on`.

Default: `off`

Usage Guidelines: N/A

Command Examples: Query the current status.
`print,/par/rover/mode/cmr`
Or
`print,rover/mode/cmr`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: Parameters `/par/rover/mode`, `/par/rover/mode/rtcm`, `/par/rover/mode/jps` and `/par/rover/mode/rtcm3`.

Message type for GLONASS measurements

`[/par/]cmr/rover/glo/type`

Description: Specifies the message type for GLONASS measurements on the rover side.

Access: read & write

Type: integer

Values: `[3...7]`

Default: `3`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use message type 5 for GLONASS raw data.

```
set,/par/cmr/rover/glo/type,5
```

Or

```
set,cmr/rover/glo/type,5
```

Query the current status.

```
print,/par/cmr/rover/glo/type
```

Or

```
print,cmr/rover/glo/type
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

CMR version used at the rover

`[/par/]cmr/rover/idxset`

Description: Specifies the version message type for GLONASS measurements on the rover side.

Access: read & write

Type: integer

Values: `[3...7]`

Default: `3`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to use message type 5 for GLONASS raw data.

```
set,/par/cmr/rover/glo/type,5
```

Or

```
set,cmr/rover/glo/type,5
```

Query the current status.

```
print,/par/cmr/rover/glo/type
```

Or

```
print,cmr/rover/glo/type
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RTCM 3-related Rover Station Parameters

RTCM 3 input mode

`[/par/]rover/mode/rtcm3`

Description: Reports the status of TPS input mode in the receiver.

Access: read & write

Type: boolean

Values: `off|on`

- By setting this parameter to `off`, you will switch all of the ports running in rtcm3 mode back to cmd mode.
- Do not use the `set,/par/rover/mode/rtcm3,on` command. As soon as you set some of the ports to rtcm3 mode (e.g., `set,/par/dev/ser/a/imode,rtcm3`), the receiver will automatically set this parameter to `on`.

Default: `off`

Usage Guidelines: N/A

Command Examples: Query the current status.
`print,/par/rover/mode/rtcm3`
Or
`print,rover/mode/rtcm3`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 2.4.

See Also: Parameters `/par/rover/mode`, `/par/rover/mode/rtcm`, `/par/rover/mode/cmr` and `/par/rover/mode/jps`.

TPS-related Rover Station Parameters

TPS input mode

`[/par/]rover/mode/jps`

Description: Reports the status of TPS input mode in the receiver.

Access: read & write

Type: boolean

Values: `off|on`

- By setting this parameter to `off`, you will switch all of the ports running in jps mode back to cmd mode.
- Do not use the `set,/par/rover/mode/jps,on` command. As soon as you set some of the ports to jps mode (e.g., `set,/par/dev/ser/a/imode,jps`), the receiver will automatically set this parameter to `on`.

Default: `off`

Usage Guidelines: N/A

Command Examples: Query the current status.
`print,/par/rover/mode/jps`
Or
`print,rover/mode/jps`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Parameters `/par/rover/mode`, `/par/rover/mode/rtcm`, `/par/rover/mode/cmr` and `/par/rover/mode/rtcm3`.

RAIM Parameters

RAIM stands for Receiver Autonomous Integrity Monitoring. When this feature is enabled, the receiver continuously checks whether the signals received from satellites are usable or not. If a fault (measurement outlier) in satellite data has been detected based on the current alarm limit, RAIM excludes this satellite(s) from the positioning calculations, thereby allowing the receiver to continue providing correct position information without an interruption in the service.

Turn RAIM on and off

`[/par/]pos/raim/mode`

Description:	Turns the RAIM function on or off.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`on`
Usage Guidelines:	N/A
Command Examples:	Turns RAIM on. `set,/par/pos/raim/mode,on` Or `set,pos/raim/mode,on` Displays current RAIM status. `print,/par/pos/raim/mode` Or `print,pos/raim/mode`
Receiver Availability:	This parameter is not available in G3-based receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

RAIM alarm limit mode

`[/par/]pos/raim/al/mode`

Description: Governs the *alarm limit mode*.

Access: read & write

Type: enumerated

Values: `manual | npa | ter | enr`

- `manual` – user-defined alarm limit.
- `npa` – non-precision approach.
- `ter` – terminal.
- `enr` – en route.

Default:
- `npa`

Usage Guidelines:
- The alarm limit specifies the threshold value for the horizontal radial error; RAIM will detect errors equal to or greater than the entered value.
- The alarm limit is selected according to Table 3-2, "Alarm limit specifications," on page 418.
- `nmi` - stands for International Nautical Mile (=1852 m).
- If `manual` alarm limit mode has been selected by the user, RAIM will use the contents of the parameter `/par/pos/raim/al/manual` as the alarm limit.

Command Examples: Set the alarm limit to manual.

```
set,/par/pos/raim/al/mode,manual
```

Or

```
set,pos/raim/al/mode,manual
```

Display current RAIM alarm status.

```
print,/par/pos/raim/al/mode
```

Or

```
print,pos/raim/al/mode
```

Receiver Availability: This parameter is not available in G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Table 3-2. Alarm limit specifications

Phase of flight	Phase of flight ID	Alarm limit
Non-precision approach	Npa	0.3 nmi
Terminal	Ter	1.0 nmi
En route	Enr	2.0 nmi

Manual RAIM alarm limit

`[/par/]pos/raim/al/manual`

Description: Specifies the alarm limit values other than the pre-defined defaults.

Access: read & write

Type: float, meters

Values: `[10.0...10000.0]`

Default: `555.6` (it corresponds to "npa" mode)

Usage Guidelines: This parameter is used only if you select `manual` in the parameter `/par/pos/raim/al/mode`.

Command Examples: Set RAIM alarm range to 630 meters.

```
set,/par/pos/raim/al/manual,630.0
```

Or

```
set,pos/raim/al/manual,630.0
```

Display current RAIM alarm range.

```
print,/par/pos/raim/al/manual
```

Or

```
print,pos/raim/al/manual
```

Receiver Availability: This parameter is not available in G3-based receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Parameters Related to NMEA Sentences

NMEA standard version

`[/par/]nmea/ver`

Description: Instructs the receiver to generate NMEA sentences according to the specified NMEA standard version.

Access: read & write

Type: enumerated

Values: `v2.2 | v2.3 | v3.0`

- `v2.2` – designates version 2.20 (January 1, 1997).
- `v2.3` – designates version 2.30 (March 1, 1998).
- `v3.0` – designates version 3.0 (July 1, 2000).

Default: `v3.0`

Usage Guidelines: Use this parameter to specify which version of NMEA sentences to generate.

Command Examples: Set receiver to use NMEA version 3.0.
`set,/par/nmea/ver,v3.0`
Or
`set,nmea/ver,v3.0`

Query NMEA version currently set.
`print,/par/nmea/ver`
Or
`print,nmea/ver`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

True Heading offset from HDT sentence

`[/par/]nmea/head/offset`

Description: Allows you to get the True Heading with respect to the specified direction (which is offset degrees clockwise from True North).

Access: read & write

Type: float, degrees

Values: `[-360.0...360.0]`

Default: `0.0`

Usage Guidelines: N/A

Command Examples: Set 0 degree offset.
`set,/par/nmea/head/offset,0.0`
Or
`set,nmea/head/offset,0.0`

Query current offset.
`print,/par/nmea/head/offset`
Or
`print,nmea/head/offset`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

TalkerID in NMEA sentences

`[/par/]nmea/gp`

Description: Instructs the receiver to use "GP" as the Talker ID in NMEA sentences.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: This mode is implemented for compatibility with older equipment that may not be capable of recognizing "GN" or "GL" as Talker IDs.

Command Examples: Set "GP" as receiver Talker ID in NMEA sentences generated.
`set,/par/nmea/gp,on`
Or
`set,nmea/gp,on`

Query GP Talker ID status.
`print,/par/nmea/gp`
Or
`print,nmea/gp`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Limit the total number of satellites in GGA sentence to 12

`[/par/]nmea/ggalim`

Description: Limits the total number of satellites in a GGA sentence to 12.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: Although there may be more than 12 GPS satellites in sight, the NMEA-0183 standard limits the total number of satellites in a GGA sentence to 12. This parameter allows any software to adhere to the NMEA-0183 standard.

Command Examples: Set the NMEA GGA sentence limit to 12 satellites.
`set,/par/nmea/ggalim,on`
Or
`set,nmea/ggalim,on`

Query the current GGA sentence satellite limit status.
`print,/par/nmea/ggalim`
Or
`print,nmea/ggalim`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Resolution Parameters for Fractional Data Fields in NMEA Sentences

Mantissa length for fractional seconds of UTC time

`[/par/]nmea/frac/sec`

Description: Specifies the number of digits in the fractional part of the seconds of UTC time.

Access: read & write

Type: integer

Values: `0 | 1 | 2`

Default: `2`

Usage Guidelines: N/A

Command Examples: Set the number of digits in the fractional part of the seconds of UTC time to 1.

```
set,/par/nmea/frac/sec,1
```

Or

```
set,nmea/frac/sec,1
```

Query the current status.

```
print,/par/nmea/frac/sec
```

Or

```
print,nmea/frac/sec
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional minutes of latitude/longitude

`[/par/]nmea/frac/min`

Description: Specifies the number of digits in the fractional part of the minutes of latitude/longitude.

Access: read & write

Type: integer

Values: `[1...7]`

Default: 7

Usage Guidelines:
- This parameter relates to all of the NMEA sentences having data fields in fractional minutes format.
- There are three additional parameters allowing the user to set the format of fractional minutes for the individual NMEA sentences GGA, GLL and GNS.

Command Examples: Set the number of digits in the fractional part of the minutes of latitude/longitude to 6.

```
set,/par/nmea/frac/min,6
```

Or

```
set,nmea/frac/min,6
```

Query the current status – Note: This query command will return the values for all three (GGS,GLL,GNS) messages.

```
print,/par/nmea/frac/min
```

Or

```
print,nmea/frac/min
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional minutes for GGA sentence

`[/par/]nmea/frac/min/GGA`

Description: Specifies the number of digits in the fractional part of the minutes of latitude/longitude for GGA.

Access: read & write

Type: integer

Values: `[1...7]`

Default: `7`

Usage Guidelines: N/A

Command Examples: Set/query the mantissa length of fractional minutes for GGA sentence.

```
set,/par/nmea/frac/min/GGA,6
```

Or

```
set,nmea/frac/min/GGA,6
```

Query the mantissa length.

```
print,/par/nmea/frac/min/GGA
```

Or

```
print,nmea/frac/min/GGA
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional minutes for GLL sentence

`[/par/]nmea/frac/min/GLL`

Description: Specifies the number of digits in the fractional part of the minutes of latitude/longitude for GLL.

Access: read & write

Type: integer

Values: `[1...7]`

Default: `7`

Usage Guidelines: N/A

Command Examples: Set/query the mantissa length of fractional minutes for GGA sentence.
`set,/par/nmea/frac/min/GLL,6`
Or
`set,nmea/frac/min/GLL,6`

Query the mantissa length.
`print,/par/nmea/frac/min/GLL`
Or
`print,nmea/frac/min/GLL`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional minutes for GNS sentence

`[/par/]nmea/frac/min/GNS`

Description: Specifies the number of digits in the fractional part of the minutes of latitude/longitude for GNS.

Access: read & write

Type: integer

Values: `[1...7]`

Default: `7`

Usage Guidelines: N/A

Command Examples: Set/query the mantissa length of fractional minutes for GNS sentence.

```
set,/par/nmea/frac/min/GNS,6
```

Or

```
set,nmea/frac/min/GNS,6
```

Query the mantissa length.

```
print,/par/nmea/frac/min/GNS
```

Or

```
print,nmea/frac/min/GNS
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional meters of geoidal separation and orthometric height

`[/par/]nmea/frac/alt`

Description: Specifies the number of digits in the fractional meters for both geoidal separation and orthometric height.

Access: read & write

Type: integer

Values: `[1...4]`

Default: `4`

Usage Guidelines: The term *altitude above the geoid*, which is used in the description of NMEA sentences, is synonymous with *orthometric height*.

Command Examples: Set the number of digits in the fractional meters for both geoidal separation and orthometric height to a value of 3 meters.
`set,/par/nmea/frac/alt,3`
Or
`set,nmea/frac/alt,3`

Query the current status.
`print,/par/nmea/frac/alt`
Or
`print,nmea/frac/alt`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length for fractional part of speed in VTG sentence

`[/par/]nmea/frac/speed/VTG`

Description: Specifies the number of digits for the fractional part of the velocity in the VTG sentence.

Access: read & write

Type: integer

Values: `[1...4]`

Default: `4`

Usage Guidelines: N/A

Command Examples: Set the number of digits for the fractional part of the velocity in the VTG sentence to 3.

```
set,/par/nmea/frac/speed/VTG,3
```

Or

```
set,nmea/frac/speed/VTG,3
```

Query the current status.

```
print,/par/nmea/frac/speed/VTG
```

Or

```
print,nmea/frac/speed/VTG
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Mantissa length of fractional residuals for GRS sentence

`[/par/]nmea/frac/res/GRS`

Description: Specifies the number of digits of fractional residuals in the GRS sentence.

Access: read & write

Type: integer

Values: `[0...4]`

Default: `3`

Usage Guidelines: N/A

Command Examples: Set the number of digits of fractional residuals for the GRS sentence to 4.

```
set,/par/nmea/frac/res/GRS,4
```

Or

```
set,nmea/frac/res/GRS,4
```

Query the current status.

```
print,/par/nmea/frac/res/GRS
```

Or

```
print,nmea/frac/res/GRS
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Pulse Per Second (PPS) Parameters

TPS receivers can generate precise PPS signals with programmable time reference, period and offset. PPS signals are available via the corresponding output connector pins (for description of TPS receiver pin-out, please refer to your operator's manual). You can synchronize either the falling or rising edge of these trigger signals with the specified time reference. Also, you can use the offset parameter to have the PPS reference shifted with respect to the time reference.

The equation describing the PPS reference is as follows:
reference_time%period=offset, (i.e., receiver time reference modulo period is equal to offset).

TPS receivers support up to two PPS outputs; a (PPS A) and b (PPS B). Both PPS outputs can be used concurrently.

Due to a hardware limitation, PPS signals are discrete with a resolution of 25 ns or 5 ns depending on the board. TPS receivers, however, allow compensation for this discreteness error. You can force the receiver to generate for each PPS pulse a message containing the offset between the scheduled PPS time and the actual pulse edge's arrival time.

PPS signal generation

`[/par/]dev/pps/<a|b>/output`

Description: Enables or disables PPS signal generation.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – enable PPS signal.
- `on` – disable PPS signal.

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable PPS signal generation, output A.

```
set,/par/dev/pps/a/out,on
```

Or

```
set,dev/pps/a/out,on
```

Query the current status.

```
print,/par/dev/pps/a/out
```

Or

```
print,dev/pps/a/out
```

`[/par/]dev/pps/<a|b>/output` *(Continued)*

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal period

`[/par/]dev/pps/<a|b>/per/ms`

Description: Specifies the period of the signal.

Access: read & write

Type: integer

Values: `[10...1000000000]`; milliseconds

Default: `1000`

Usage Guidelines: N/A

Command Examples: **# 1**: Set the period of 1PPS A to 2 seconds.
`set,/par/dev/pps/a/per/ms,2000`
Or
`set,dev/pps/a/per/ms,2000`

2: Set the period of 1PPS B to 500 ms.
`set,/par/dev/pps/b/per/ms,500`
Or
`set,dev/pps/b/per/ms,500`

Query the current status.
`print,/par/dev/pps/a/per/ms`
Or
`print,dev/pps/a/per/ms`

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal offset (milliseconds)

`[/par/]dev/pps/<a|b>/offs/ms`

Description: Specifies the offset of the signal.

Access: read & write

Type: integer

Values: `[-500000000…500000000]`; milliseconds

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the offset of PPS A to 0 milliseconds
`set,/par/dev/pps/a/offs/ms,0`
Or
`set,dev/pps/a/offs/ms,0`

Query the current status
`print,/par/dev/pps/a/offs/ms`
Or
`print,dev/pps/a/offs/ms`

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal offset (nanoseconds)

`[/par/]dev/pps/<a|b>/offs/ns`

Description: Specifies the offset of the signal.

Access: read & write

Type: integer

Values: `[-500000…500000]`; nanoseconds

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the offset of PPS B to 0 nanoseconds

```
set,/par/dev/pps/b/offs/ns,0
```

Or

```
set,dev/pps/b/offs/ns,0
```

Query the current status

```
print,/par/dev/pps/b/offs/ns
```

Or

```
print,dev/pps/b/offs/ns
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal synchronization

`[/par/]dev/pps/<a|b>/edge`

Description: Specifies the synchronization with the specified time reference using either the rising or falling edge of the signal.

Access: read & write

Type: enumerated

Values: `rise | fall`

Default: `rise`

Usage Guidelines: N/A

Command Examples: Synchronize the specified time reference with the falling edge of the signal PPS A

```
set,/par/dev/pps/a/edge,rise
```

Or

```
set,dev/pps/a/edge,rise
```

Query the current status

```
print,/par/dev/pps/b/edge
```

Or

```
print,dev/pps/b/edge
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal reference time

`[/par/]dev/pps/<a|b>/time`

Description: Specifies the time reference to use when synchronizing the 1PPS signal.

Access: read & write

Type: enumerated

Values: `gps | glo | utcusno | utcsu`

- `gps` – GPS time
- `glo` – GLONASS time
- `utcusno` – UTC(USNO)
- `utcsu` – UTC(SU)

Default: `utcusno`

Usage Guidelines: If you are going to set this parameter to `glo` or `utcsu`, make sure that one or more GLONASS satellites are being locked. Similarly, if you are going to set this parameter to `gps` or `utcusno`, ensure that one or more GPS satellites are being locked.

Command Examples: Synchronize the PPS A signal to the GPS time reference.

```
set,/par/dev/pps/a/time,gps
```

Or

```
set,dev/pps/a/time,gps
```

Query the current status.

```
print,/par/dev/pps/b/time
```

Or

```
print,dev/pps/b/time
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Period of marked PPS signals

`[/par/]dev/pps/<a|b>/mper`

Description: Specifies the period of the marked PPS signal.

Access: read & write

Type: integer

Values: `[20...1000000000]`; milliseconds

Default: `0`

Usage Guidelines:

- The width of the marked pulse is 0.8192*6 ms whereas the normal pulse's width is 0.8192*4 ms.
- If this parameter is set to zero, which is the default value, the receiver will generate no marked pulses at all. If this parameter is set to a non-zero value ($N2 \neq 0$) and the parameter governing the period of normal pulses is set to N1, then the receiver will generate both normal and marked pulses, but marked pulses will be output only every (N1, N2) milliseconds, where (N1, N2) designates the least common multiple of these two integers.

Command Examples: **# 1**: For the receiver to output onto the PPSA pin ten pulses per second with every tenth pulse a marked one (marked pulses will correspond to integer seconds in the reference time system and all the other pulses will be normal), send the following commands

```
set,dev/pps/a/per/ms,100
set,dev/pps/a/mper,1000
```

2: The following commands will cause the receiver to generate two pulses per second through the PPSB pin with every seventh pulse a marked one.

```
set,dev/pps/b/per/ms,500
set,dev/pps/b/mper,35
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

PPS signal synchronization to the selected reference time

`[/par/]dev/pps/<a|b>/tied`

Description: Specifies whether the PPS signals are synchronized with the selected reference time frame or with the receiver's internal clock/external frequency.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: There are applications where the user needs to synchronize PPS signals with either the receiver's internal clock or an external frequency, not with the selected reference time.
If this parameter is set to `on`, PPS signals are synchronized with the selected reference time, which is the common practice (thus PPS signals are *tied* to the selected reference time).
If this parameter is set to `off`, PPS signals are synchronized either with the receiver's internal clock or, if the parameter `/par/frq/input` has been set to `ext`, with the corresponding external frequency.

Command Examples: Synchronize the PPSA signal with the selected reference timeframe.
`set,/par/dev/pps/a/tied,on`
Or
`set,dev/pps/a/tied,on`

Query the current status.
`print,/par/dev/pps/a/tied`
Or
`print,dev/pps/a/tied`

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the PPS functionality.
- The _PPS option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Event Signal Parameters

TPS receivers are capable of measuring/recording event times in the specified reference time system with high accuracy and it is possible to measure the time of either the rising edge or falling edge of the input event signal. Most of the TPS receivers may accommodate up to two external event pins, *a* (Event A) and *b* (Event B).

Event Marker signal

`[/par/]dev/event/<a|b>/in`

Description: Enables or disables the event marker signal.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable event marking for event A.

```
set,/par/dev/event/a/in,on
```

Or

```
set,dev/event/a/in,on
```

Query the current status.

```
print,/par/dev/event/b/in
```

Or

```
print,dev/event/b/in
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Falling or rising edge of Event Marker signal

`[/par/]dev/event/<a|b>/edge`

Description: Specifies whether the receiver will measure the time of the rising edge or falling edge of the input event signal.

Access: read & write

Type: enumerated

Values: `rise | fall`

Default: `rise`

Usage Guidelines: N/A

Command Examples: Enable the measurement of the falling edge of input event signal B.
`set,/par/dev/event/b/edge,fall`
Or
`set,dev/event/b/edge,fall`

Query the current status
`print,/par/dev/event/b/edge`
Or
`print,dev/event/b/edge`

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Event Marker's reference time

`[/par/]dev/event/<a|b>/time`

Description: Specifies the time reference to use when synchronizing the Event Marker signal.

Access: read & write

Type: enumerated

Values: `gps | glo | utcusno | utcsu`

- `gps` – GPS time
- `glo` – GLONASS time
- `utcusno` – UTC(USNO)
- `utcsu` – UTC(SU)

Default: `utcusno`

Usage Guidelines: If you are going to set this parameter to `glo` or `utcsu`, make sure that one or more GLONASS satellites are being locked. Similarly, if you are going to set this parameter to `gps` or `utcusno`, ensure that one or more GPS satellites are being locked.

Command Examples: Synchronize the Event Marker A signal to the GPS time reference.

```
set,/par/dev/event/a/time,gps
```

Or

```
set,dev/event/a/time,gps
```

Query the current status.

```
print,/par/dev/event/a/time
```

Or

```
print,dev/event/a/time
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of the computed receiver clock offset when measuring XA, XB events in the selected reference time

`[/par/]dev/event/<a|b>/tied`

Description: With this parameter, the receiver is instructed to measure the event reception time in the *selected time reference* with or without consideration for the computed receiver clock offset.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: If the parameter is set to `off`, the event time is measured in the receiver time scale that will differ from the *selected time reference* by the computed clock offset.
If the parameter is set to `on`, the event time is measured in the selected reference time properly. Thus the name of the parameter, *tied* (figuratively speaking, we *tie up* event signals rigidly with the selected time reference).

Command Examples: Synchronize the Event marker signal A in the selected reference time properly.
`set,/par/dev/event/a/tied,on`
Or
`set,dev/event/a/tied,on`

Query the current status.
`print,/par/dev/event/a/tied`
Or
`print,dev/event/a/tied`

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Synchronize the receiver clock offset with an event signal

`[/par/]dev/event/<a|b>/lock`

Description: Instructs the receiver to synchronize its one-millisecond cycle grid with the corresponding edge of the event signal arrived immediately after the command execution.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: The `set` command may be required to be executed repeatedly to ensure that the receiver maintains synchronization with the time scale governing the external event signals.

Command Examples: Synchronize the receivers clock with Event Marker signal A.

```
set,/par/dev/event/a/lock,on
```

Or

```
set,dev/event/a/lock,on
```

Query the current status.

```
print,/par/dev/event/a/lock
```

Or

```
print,dev/event/a/lock
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Status of the receiver clock synchronization

`[/par/]dev/event/<a|b>/locked`

Description: Returns a value indicating whether the receiver clock is actually being synchronized with the event signals or not.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Query the status of the receiver clock synchronization for Event Marker signal A.

```
print,/par/dev/event/a/locked
```

Or

```
print,dev/event/a/locked
```

Receiver Availability: This parameter may only be used if the following conditions have been met:

- The receiver's hardware supports the Event Marker functionality.
- The EVNT option has been enabled in the receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Code Differential Parameters

This section describes the parameters used for configuring your receiver in DGPS mode.

Source of differential corrections

`[/par/]pos/cd/port`

Description: Specifies the port from which the receiver will use differential corrections for position computation.

Access: read & write

Type: enumerated

Values: `any | /port`

- `any` – the receiver will use differential corrections from whichever port.
- `/port` – the receiver will only use differential corrections from the corresponding port.

Default: `any`

Usage Guidelines: This parameter is used by your TPS receiver only if multi-base DGPS is off.

Command Examples: Use differential corrections from serial port C.

```
set,/par/pos/cd/port,/dev/ser/c
```

Or

```
set,pos/cd/port,/dev/ser/c
```

Query the current status.

```
print,/par/pos/cd/port
```

Or

```
print,pos/cd/port
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum age of code differential corrections used for position computation

`[/par/]pos/cd/maxage`

Description:	Sets the maximum age of code differential corrections used for position computation.
Access:	read & write
Type:	integer
Values:	`[1...1200]`; seconds
Default:	`30`
Usage Guidelines:	If the age of corrections exceeds the value specified, the rover will compute a stand-alone position.
Command Examples:	Set the maximum age to 10 seconds. `set,/par/pos/cd/maxage,10` Or `set,pos/cd/maxage,10` Query the current status. `print,/par/pos/cd/maxage` Or `print,pos/cd/maxage`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Maximum age of ionosphere corrections used for position computation

`[/par/]pos/cd/iono/maxage`

Description: Specifies the maximum age of the ionosphere corrections used for position computation.

Access: read & write

Type: integer

Values: `[1...1800]`; seconds

Default: `300`

Usage Guidelines: N/A

Command Examples: Set the maximum age to 420 seconds.

```
set,/par/pos/cd/iono/maxage,420
```

Or

```
set,pos/cd/iono/maxage,420
```

Query the current status.

```
print,/par/pos/cd/iono/maxage
```

Or

```
print,pos/cd/iono/maxage
```

Receiver Availability: This parameter is not available in GMS-2.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use P-code measurements in DGPS

`[/par/]pos/cd/allmeas`

Description:	Enables the use of P-code measurements in DGPS.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`off`
Usage Guidelines:	N/A
Command Examples:	Do not use P-code measurements in DGPS. `set,/par/pos/cd/allmeas,off` Or `set,pos/cd/allmeas,off` Query the current status. `print,/par/pos/cd/allmeas` Or `print,pos/cd/allmeas`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Ionosphere-free differential correction mode

`[/par/]pos/cd/ionofree`

Description: Applies ionosphere-free differential correction mode.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the rover receiver will not use ionosphere corrections from RTCM message type 15.
- `on` – the rover receiver will use both ionosphere corrections from RTCM message type 15 and regular differential corrections from RTCM messages types 1 and 31 (or 9 and 34) when computing position.

Default: `off`

Usage Guidelines: N/A

Command Examples: Set the parameter value to on.

```
set,/par/pos/cd/ionofree,on
```

Or

```
set,pos/cd/ionofree,on
```

Query the current status.

```
print,/par/pos/cd/ionofree
```

Or

```
print,pos/cd/ionofree
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Range residual limit

`[/par/]pos/cd/rlim`

Description: Sets the range residual limit effective for code differential.

Access: read & write

Type: float

Values: `[1.0...100.0]`; meters

- `off` – the rover receiver will not use ionosphere corrections from RTCM message type 15.
- `on` – the rover receiver will use both ionosphere corrections from RTCM message type 15 and regular differential corrections from RTCM messages types 1 and 31 (or 9 and 34) when computing position.

Default: `5.0`

Usage Guidelines:

- Satellites whose range residuals are greater than this limit, will be discarded from code differential positioning.
- This parameter is used only if the parameter `/par/pos/raim/mode` has been set to `on`.

Command Examples: Set the range residual limit to 6 meters.

```
set,/par/pos/cd/rlim,6
```

Or

```
set,pos/cd/rlim,6
```

Query the current status.

```
print,/par/pos/cd/rlim
```

Or

```
print,pos/cd/rlim
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of external source of DGNSS data for code differential solution

`[/par/]pos/cd/src/ext`

Description: Disables and enables the use of DGNSS data obtained from an external source for computing code differential solution.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Use DGNSS data from an external source.

```
set,/par/pos/cd/src/ext,on
```

Or

```
set,pos/cd/src/ext,on
```

Query the current status.

```
print,/par/pos/cd/src/ext
```

Or

```
print,pos/cd/src/ext
```

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Use DGNSS data from SBAS for code differential solution

`[/par/]pos/cd/src/sbas`

Description:	Disables and enables the use of DGNSS data obtained from SBAS for computing code differential solution.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`on`
Usage Guidelines:	N/A
Command Examples:	Use DGNSS data from SBAS. `set,/par/pos/cd/src/sbas,on` Or `set,pos/cd/src/sbas,on` Query the current status. `print,/par/pos/cd/src/sbas` Or `print,pos/cd/src/sbas`
Receiver Availability:	This parameter is unavailable in the GMS-2 receiver.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Use DGNSS data from VBS for code differential solution

`[/par/]pos/cd/src/vbs`

Description: Disables and enables the use of DGNSS data obtained from VBS service for computing code differential solution.

Access: write-only

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Use DGNSS data from VBS.
`set,/par/pos/cd/src/vbs,on`
Or
`set,pos/cd/src/vbs,on`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Use DGNSS data from any source for code differential solution

`[/par/]pos/cd/src/`

Description: Disables and enables the use of DGNSS data obtained from any source for computing code differential solution.

Access: write-only

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Use DGNSS data from any source.
`set,/par/pos/cd/src/,on`
Or
`set,pos/cd/src/,on`

Receiver Availability: This parameter is unavailable in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Multi-base DGPS parameters

Multi-base mode

`[/par/]pos/cd/mult/mode`

Description: Activates multi-base mode.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the mode is deactivated. The receiver will use differential corrections from the specific source(s) defined by the parameter `/par/pos/cd/port`.
- `on` – the mode is activated. The rover will use differential corrections from all available sources (whatever the value of `/par/pos/cd/port`).

Default: `on`

Usage Guidelines: N/A

Command Examples: Set multi-base mode to off.
`set,/par/pos/cd/mult/mode,off`
Or
`set,pos/cd/mult/mode,off`

Query the current status.
`print,/par/pos/cd/mult/mode`
Or
`print,pos/cd/mult/mode`

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use differential corrections from the nearest reference station

`[/par/]pos/cd/mult/nearest`

Description: Instructs the receiver to use differential corrections transmitted by the nearest reference station.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines:

- This parameter will only be effective if the receiver can compute the distances to all of the relevant reference stations. This is possible only if the reference stations transmit their coordinates.
- The receiver will use differential data exclusively from the nearest reference station only when the mix and best parameters (description of these parameters follows) are set to `off` and this parameter (nearest) is set to `on`.

Command Examples: Set the multi-base mode to use the differential corrections transmitted by the nearest reference station.

```
set,/par/pos/cd/mult/nearest,on
```

Or

```
set,pos/cd/mult/nearest,on
```

Query the current status.

```
print,/par/pos/cd/mult/nearest
```

Or

```
print,pos/cd/mult/nearest
```

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Run rover in *mix* mode

`[/par/]pos/cd/mult/mix`

Description: Runs the receiver in the mix mode by averaging either differential corrections, or individual position estimates received from different reference stations.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the mix mode is disabled.
- `on` – the mix mode is enabled and mixing is performed as defined by the parameter `/par/pos/cd/mult/mixmode` (description follows).

Default: `off`

Usage Guidelines: If the parameter `/par/pos/cd/mult/mix` is set to `on` and the parameter `/par/pos/cd/mult/best` is set to `off`, the receiver will always mix differential corrections from different reference stations when computing its position.

Command Examples: Enable the mix mode.
`set,/par/pos/cd/mult/mix,on`
Or
`set,pos/cd/mult/mix,on`

Query the current status.
`print,/par/pos/cd/mult/mix`
Or
`print,pos/cd/mult/mix`

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Averaging[4] of individual position estimates or differential corrections from the different reference stations

`[/par/]pos/cd/mult/mixmode`

Description: Determines whether the individual position estimates or the differential corrections received from the different reference stations will be averaged to obtain more precise and reliable position estimates.

Access: read & write

Type: enumerated

Values: `cor | pos`

- `cor` – the rover will average, using appropriate weights, differential corrections from different reference stations in order to obtain more precise and reliable position estimates.
- `pos` – the rover will average, using appropriate weights, individual position estimates from different reference stations in order to obtain more precise and reliable position estimates.

Default: `cor`

Usage Guidelines:

- The receiver will take this parameter into account only if the above-described parameter (`/par/pos/cd/mult/mix`) is set to `on`.
- Since the processor time is proportional to the number of the differential correction sets received by the rover, this mode may prove quite time consuming. Therefore, when using this parameter, make sure that the receiver position is updated once a second (you may need to run the command `set,/par/raw/msint,1000`).

Command Examples: Set the parameter to `cor` to average the differential corrections when computing the position estimates.

```
set,/par/pos/cd/mult/mixmode,cor
```

Or

```
set,pos/cd/mult/mixmode,cor
```

Query the current status.

```
print,/par/pos/cd/mult/mixmode
```

Or

```
print,pos/cd/mult/mixmode
```

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

4. Averaging is synonymous with mixing in the context of this section.

Use the best differential corrections

`[/par/]pos/cd/mult/best`

Description: Automatically identifies, and uses, the differential correction sets that will yield the most precise position estimate.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not use the best differential corrections.
- `on` – the receiver will automatically identify which of the differential correction data sets yields the most precise position estimate.

Default: `off`

Usage Guidelines:

- If `/par/pos/cd/mult/mix` is set to `on`, mixed differential corrections will also be taken into account when selecting the best solution.
- *Best solution* means the solution having the least RMS error.
- When using this parameter, make sure that the receiver position is updated once a second (you may need to run the command `set,/par/raw/msint,1000`).

Command Examples: Set the parameter to `on` for the receiver to automatically identify which of the differential correction data sets yields the most precise position estimate.

```
set,/par/pos/cd/mult/best,on
```

Or

```
set,pos/cd/mult/best,on
```

Query the current status.

```
print,/par/pos/cd/mult/best
```

Or

```
print,pos/cd/mult/best
```

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Corrections (offsets) to the transmitted coordinates of the reference stations used in multi-base mode

`[/par/]rover/base/pos/par/<N>`

Description: Compensates for known coordinate offsets of up to five reference stations used in multi-base code differential.

Access: read & write

Type: `{valid,port,basedID,delNorth,delEast,delUp}`,
where:

- `valid` is a boolean value `on` or `off`. If `valid=off`, the parameter is associated with no reference station.
- `port` is a field that takes same values as, for example, the parameter `/par/pos/cd/port`. It specifies the port to which the offsets are referenced.
- `baseID` is an integer identification of the reference station for which the offsets are specified. It varies between -1 and 1023.
  If `baseID` = -1, the offsets are applicable to all reference stations used in multi-base mode except the stations for which offsets are already specified with the preceding parameters (see the examples below).
- `delNorth,delEast,delUp` is float numbers specifying the north-, east- and up- components of the vector correction to the reference station position [meters].

Usage Guidelines:

- This correction mechanism is especially important when differential corrections from different reference stations are used together to compute the rover position in the *mixed solution* mode (for details, see the parameter `/par/pos/cd/mult/mixpos`). Should the transmitted reference coordinates of the base stations be conspicuously different from their truth coordinates in this mode, the estimated rover position may prove corrupt unless the user enters appropriate coordinate offsets for these stations on the rover side.
- `N` ranges from a to e.
- These parameters are not applicable to RTK.

`[/par/]rover/base/pos/par/<N>` *(Continued)*

Command Examples:

1: With these commands you instruct the receiver to use coordinate corrections (offsets) for the reference stations with IDs 100, 101 and 102. Differential corrections may be received on any port.

```
set,rover/base/pos/par/a,{on,any,100,-0.02,0.033,-0.05}
set,rover/base/pos/par/b,{on,any,101,0.01,-0.034,-0.22}
set,rover/base/pos/par/c,{on,any,102,0.002,0.023,-0.011}
```

2: In this case the specified offsets will be applied to the reference stations whose differential corrections are received on ports A, B and C (irrespective of the stations' IDs).

```
set,rover/base/pos/par/a,{on,dev/ser/a,-1,-0.02,-0.03,
-0.05}
set,rover/base/pos/par/b,{on,dev/ser/b,-1,0.01,-0.04,0.0}
set,rover/base/pos/par/c,{on,dev/ser/c,-1,0.02,0.12,
-0.003}
```

3: First, the receiver will apply the offsets *(-0.02,0.03,-0.05)* to the transmitted coordinates of the reference station whose differential corrections are received on port A (whatever the station ID).

Second, the receiver will apply the offsets *(0.01,0.04,0.0)* to the reference station 101 if its differential corrections are received on port B (but not on port A). If port B is used to receive differential corrections from a station other than 101, the offsets *(-0.01,-0.04,0.1)* will be applied to this station.

Third, the offsets *(0.02,0.02,-0.03)* will be applied to any reference station whose differential corrections are received on port C (but not on port A or port B).

```
set,rover/base/pos/par/a,{on,dev/ser/a,-1,-0.02,0.03,
-0.05}
set,rover/base/pos/par/b,{on,dev/ser/b,101,0.01,0.04,0.0}
set,rover/base/pos/par/c,{on,dev/ser/b,-1,-0.01,
-0.04,0.1}
set,rover/base/pos/par/d,{off,dev/ser/b,101,0.1,
-0.04,0.01}
set,rover/base/pos/par/e,{on,dev/ser/c,-1,0.02,0.02,
-0.03}
```

Conflict Resolution – The general rule to resolve possible conflicts is as follows:
If multiple offsets are specified for the reference station, the receiver will be using those given in the quintet first (counting from "a" to "e").

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Numerical offset used to distinguish between single and multiple reference stations in solution

`[/par/]pos/cd/mult/offset`

Description: Allows you to distinguish whether DGPS solution has been obtained with single or multiple reference stations.

Access: read & write

Type: integer

Values: `[0...1023]`

Default: `1000`

Usage Guidelines: In the multi-base mode the rover receiver can accept the differential corrections from multiple reference stations simultaneously. In this case the reference station ID, which is available, for example, in the NMEA GGA message contains the total number of reference stations used to compute the rover position. In some cases, however, it can be desirable to distinguish, for example, DGPS solution that uses the differential corrections from the reference station having ID = 2 from DGPS solution computed with the use of two reference stations. For this, the user should specify a numerical offset that will be added to the total number of the reference stations used. Suppose the parameter is set to 1000 (default) and two reference stations are used to compute DGPS solution. In this way, the GGA message will display 1002 in the field corresponding to the reference station ID.

Command Examples: Set the parameter to 1000 (default).
`set,/par/pos/cd/mult/offset,1000`
Or
`set,pos/cd/mult/offset,1000`

Query the current status.
`print,/par/pos/cd/mult/offset`
Or
`print,pos/cd/mult/offset`

Receiver Availability: This parameter is not available in G3-based and GMS-2 receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Carrier Phase Differential Parameters

Position output mode

`[/par/]pos/pd/mode`

Description: Controls which output mode is enabled, *extrapolate* or *delay*.

Access: read & write

Type: enumerated

Values: `extrap | delay`

- `extrap` – the RTK engine will extrapolate the base station's carrier phase measurements to the current epoch (note that the *truth* carrier phases measured at the base cannot be transmitted and received at the rover instantly). The final positioning accuracy may be somewhat lower due to additional extrapolation errors, which may be up to a few millimeters vertical and horizontal for a one-second extrapolation time. Note that *extrapolate* mode could be used only when the reference station is static (i.e., not moving).
- `delay` – the RTK engine does not extrapolate the base station's carrier phases in position computation. Instead, the engine will either compute a delayed position or simply output the current stand-alone position (while waiting for new RTCM/CMR messages from the base station). Note that the delayed position is computed for the time (epoch) to which the last received base station's carrier phase measurements correspond. Accuracies achievable in delay mode are normally on a level with those of post-processing kinematic.

Default: `delay`

Usage Guidelines: N/A

Command Examples: Enable the receiver to use the output mode `extrapolate`.

```
set,/par/pos/pd/mode,extrap
```

Or

```
set,pos/pd/mode,extrap
```

Query the current status.

```
print,/par/pos/pd/mode
```

Or

```
print,pos/pd/mode
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Position computation technique used in delay mode

`[/par/]pos/pd/delay`

Description: Specifies position computation scenario that will be used in *delay* mode.

Access: read & write

Type: enumerated

Values: `last | every`

- `last` – the RTK engine will process the last received set of carrier phase differential data.
- `every` – the RTK engine will attempt to process all sequentially received sets of carrier phase differential data.

Default: `last`

Usage Guidelines: N/A

Command Examples: Enable the RTK engine to attempt to process all (*every*) sequentially received sets of carrier phase differential data.

```
set,/par/pos/pd/delay,every
```

Or

```
set,pos/pd/delay,every
```

Query the current status.

```
print,/par/pos/pd/delay
```

Or

```
print,pos/pd/delay
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Ambiguity fixing level

`[/par/]pos/pd/aflevel`

Description: Specifies a level of confidence when fixing integer ambiguities.

Access: read & write

Type: enumerated

Values: `low | medium | high`

Default: `medium`

Usage Guidelines: The RTK engine has 3 levels of confidence when fixing integer ambiguities, *low, medium* and *high* which correspond to the indicator's 95%, 99.5% and 99.9% thresholds, respectively.
The higher the confidence level specified, the longer the integer ambiguity search time. This is the price we have to pay for the higher reliability of the ambiguity fixed solution.
Assume you have set this parameter to `high`. The receiver's RTK engine will constantly update the confidence level indicator as new measurements arrive. Once this parameter exceeds the 99.9% threshold, the engine will fix up all or some of the integer ambiguities. The corresponding position estimate will be marked as fixed RTK solution although some of the ambiguities (usually a small part of them) may remain float.

Command Examples: Enable the RTK engine to use the *high* ambiguity fixing level corresponding to a confidence level of 99.9%.

```
set,/par/pos/pd/aflevel,high
```

Or

```
set,pos/pd/aflevel,high
```

Query the current status.

```
print,/par/pos/pd/aflevel
```

Or

```
print,pos/pd/aflevel
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RTK initialization

`[/par/]pos/pd/fixpos`

Description: Instructs the RTK engine to use the rover's precise position for ambiguity resolution.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: While this parameter allows the engine to fix ambiguities much faster, the rover's precise coordinates must be specified as described in “Datum Parameters” on page 3-299. Care should be taken that this parameter is set back to `off` once the RTK initialization is over and the antenna starts moving. Otherwise the rover's position will be computed incorrectly.

Command Examples: Instruct the RTK engine to use the rover's precise position for ambiguity resolution.

```
set,/par/pos/pd/fixpos,on
```

Or

```
set,pos/pd/fixpos,on
```

Query the current status.

```
print,/par/pos/pd/fixpos
```

Or

```
print,pos/pd/fixpos
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum number of satellites used in RTK

`[/par/]pos/pd/maxsat`

Description: Specifies the maximum number of satellites to use in the RTK position computation.

Access: read & write

Type: integer

Values: `[4...20]`

Default:
- `12` – for receivers and boards having the processor's clock rate lower than 56 MHz.
- `20` – for receivers and boards having the processor's clock rate greater than 56 MHz.

Usage Guidelines: If the actual number of satellites in sight exceeds the current parameter's value, the RTK engine will utilize data only from the satellites with the highest C/A SNRs and the number of satellites used will not be greater than that specified by the parameter.

Command Examples: Instruct the RTK engine to use a maximum of 12 satellites in the RTK position computation.

```
set,/par/pos/pd/maxsat,12
```

Or

```
set,pos/pd/maxsat,12
```

Query the current status

```
print,/par/pos/pd/maxsat
```

Or

```
print,pos/pd/maxsat
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History:
- This parameter was introduced in the first release of firmware.
- The default value `12` was added in firmware 3.3.

See Also: "Maintenance Parameters" on page G-1 for the parameter used for checking the processor's clock rate with the `print` command.

C/A measurements used by the rover in the position computation

`[/par/]pos/pd/meas/ca`

Description: Enables or disables the use of C/A measurements to computes the rover's position.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine not to use C/A measurements to compute the rover's position.
`set,/par/pos/pd/meas/ca,off`
Or
`set,pos/pd/meas/ca,off`

Query the current status.
`print,/par/pos/pd/meas/ca`
Or
`print,pos/pd/meas/ca`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P/L1 measurements used by the rover in the position computation

`[/par/]pos/pd/meas/p1`

Description: Enables or disables the use of P/L1 measurements to computes the rover's position.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: This command is unavailable for single frequency receivers.

Command Examples: Instruct the RTK engine not to use P/L1 measurements to compute the rover's position.

```
set,/par/pos/pd/meas/p1,off
```

Or

```
set,pos/pd/meas/p1,off
```

Query the current status.

```
print,/par/pos/pd/meas/p1
```

Or

```
print,pos/pd/meas/p1
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

P/L2 measurements used by the rover in the position computation

`[/par/]pos/pd/meas/p2`

Description: Enables or disables the use of P/L2 measurements to computes the rover's position.

Access: read & write

Type: boolean

Values: `off | on`

Default: `on`

Usage Guidelines: This command is unavailable for single frequency receivers.

Command Examples: Instruct the RTK engine not to use P/L2 measurements to compute the rover's position.
`set,/par/pos/pd/meas/p2,off`
Or
`set,pos/pd/meas/p2,off`

Query the current status.
`print,/par/pos/pd/meas/p2`
Or
`print,pos/pd/meas/p2`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Single difference carrier phase weighting scheme

`[/par/]pos/pd/scale`

Description: Specifies the weights that the RTK engine will apply to single-differenced (SD) carrier phase observables.

Access: read & write

Type: integer

Values: `[0...2]`

- `0` – The simplest weighting scheme will be used: all SD carrier phases will be used with a priori weight equal to *1/*(*sigma*sigma*), where *sigma* = 0.05 cycle. This scheme is recommended when running RTK in favorable environment conditions and allows a shorter ambiguity fixing time than the other two (see below).
- `1` – The first adaptive weighting scheme will be used: SD carrier phases for the given satellite will be used with a priori weight equal to *max(0.2, sin(elev))/ (sigma_est*sigma_est))*, where *elev* designates the satellite's elevation angle, *sigma_est* designates the RMS single-differenced carrier phase error.
- `2` – The second adaptive weighting scheme will be used. This is similar to scheme 1, but different estimator to get *sigma_est* is used. This scheme is strongly recommended in scenarios where strong multipath is expected.

Default: `1`

Usage Guidelines: This command is only effective in the *RTK Delay* mode.

Command Examples: Instruct the RTK engine not to apply the second (*2*) adaptive weighting scheme to single-difference carrier phase observables.

```
set,/par/pos/pd/scale,2
```

Or

```
set,pos/pd/scale,2
```

Query the current status.

```
print,/par/pos/pd/scale
```

Or

```
print,pos/pd/scale
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Differential correction update interval (period) for reference station

`[/par/]pos/pd/period`

Description: Instructs the rover receiver to output the RTK position (not the single point position) at the same rate at which differential corrections are updated.

Access: read & write

Type: float

Values: `[0.05...100.0]`; seconds

Default: `1.0`

Usage Guidelines:
- This command is only effective in the *RTK Delay* mode.
- You should know the exact rate at which the base station transmits its differential correction data.

Command Examples: Instruct the receiver to output the RTK position at a 5 second rate.
`set,/par/pos/pd/period,5.0`
Or
`set,pos/pd/period,5.0`

Query the current status.
`print,/par/pos/pd/period`
Or
`print,pos/pd/period`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Turn on/off the single-differenced ionospheric model

`[/par/]pos/pd/ion`

Description: Instructs the rover receiver to enable or disable the single-differenced ionospheric model.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the single-differenced ionospheric model is disabled.
- `on` – the single-differenced ionospheric model is enabled.

Default: `on`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to disable the single-differenced ionospheric model
```
set,/par/pos/pd/ion,off
```
Or
```
set,pos/pd/ion,off
```

Query the current status
```
print,/par/pos/pd/ion
```
Or
```
print,pos/pd/ion
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

The threshold value for the RTK ionospheric modeling

`[/par/]pos/pd/ionr`

Description: The ionospheric delay is modeled by RTK only if the baseline length is greater or equal to the specified value.

Access: read & write

Type: float

Values: `[0.0…999999.9]`; meters

Default: `4000.0`

Usage Guidelines: N/A

Command Examples: Instruct the receiver to set the threshold value for RTK modeling to 4,500 meters (4.5 km).

```
set,/par/pos/pd/ionr,4500.0
```

Or

```
set,pos/pd/ionr,4500.0
```

Query the current status.

```
print,/par/pos/pd/ionr
```

Or

```
print,pos/pd/ionr
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

The memory factor f for the float ambiguity filter

`[/par/]pos/pd/mem`

Description: Defines the memory factor for ambiguity estimates.

Access: read & write

Type: float

Values: `[0.5...1.0]`

Default: `0.9997`

Usage Guidelines: The smaller the filter memory factor, the less important are the older ambiguity estimates for the RTK engine estimating the current ambiguities.
On shorter vectors (up to 8 km), it is recommended to set the memory factor to 0.998 when running the receiver under tree.

Command Examples: Instruct the receiver to set the memory factor to 0.998.
`set,/par/pos/pd/mem,0.998`
Or
`set,pos/pd/mem,0.998`

Query the current status
`print,/par/pos/pd/mem`
Or
`print,pos/pd/mem`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Extrapolate missing carrier phase measurements in delay mode

`[/par/]pos/pd/se`

Description: Instructs the RTK engine to extrapolate the missing carrier phase measurements for the currently processed epoch (provided that the code phase measurements have been successfully received for this epoch).

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to extrapolate the missing carrier phase measurements for the currently processed epoch.

```
set,/par/pos/pd/se,on
```

Or

```
set,pos/pd/se,on
```

Query the current status.

```
print,/par/pos/pd/se
```

Or

```
print,pos/pd/se
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Force the RTK engine to verify fixed ambiguities in regular intervals

`[/par/]pos/pd/check`

Description: Specifies the period for the RTK engine to check fixed ambiguities for errors.

Access: read & write

Type: integer

Values: `[-1...32767]`

- `-1` – the ambiguity verification is off.
- `0,1` – the RTK engine will verify ambiguities every epoch.
- `[2...215-1]` – the RTK engine will verify ambiguities every specified number of epochs.

Default: `7`

Usage Guidelines: The engine will be forced to recompute/check the ambiguity vector every specified number of epochs. The estimated *forced ambiguities* will be compared against the current ones. If the forced ambiguity vector differs from the current ambiguity vector, the float solution mode will be enabled at once.

Command Examples: Instruct the RTK engine to re-compute/check the ambiguity vector every 10 epochs.
`set,/par/pos/pd/check,10`
Or
`set,pos/pd/check,10`

Query the current status.
`print,/par/pos/pd/check`
Or
`print,pos/pd/check`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Period of extrapolation for reference station carrier phase measurements

`[/par/]pos/pd/experiod`

Description: Instructs the RTK engine to extrapolate the received base station's carrier phase measurements if the time (epoch) to which these carrier phase measurements correspond is divisible by `experiod`.

Access: read & write

Type: float

Values: `[0.0...5.0]`

Default: `0.1`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to extrapolate the base station's carrier phase measurements if the time period (epoch) to which these carrier phase measurements correspond is divisible by 2.

```
set,/par/pos/pd/experiod,2.0
```

Or

```
set,pos/pd/experiod,2.0
```

Query the current status.

```
print,/par/pos/pd/experiod
```

Or

```
print,pos/pd/experiod
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Doppler in RTK processing

`[/par/]pos/pd/usebasedop`

Description: Enables or disables the use of Doppler measurements either received from the base or obtained by the base measurements extrapolator, in RTK processing.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: N/A

Command Examples: Instruct the rover receiver to use Doppler measurements in RTK processing.
`set,/par/pos/pd/usebasedop,y`
Or
`set,pos/pd/usebasedop,y`

Query the current status.
`print,/par/pos/pd/usebasedop`
Or
`print,pos/pd/usebasedop`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Period of ambiguities estimation

`[/par/]pos/pd/afperiod`

Description: Defines the time period at which the ambiguity estimation procedure is performed.

Access: read & write

Type: float

Values: `[0.05...5.0]`; seconds

Default: `1.0`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to perform the ambiguity estimation every 2 seconds.
`set,/par/pos/pd/afperiod,2.0`
Or
`set,pos/pd/afperiod,2.0`

Query the current status.
`print,/par/pos/pd/afperiod`
Or
`print,pos/pd/afperiod`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Enable measurements quality indicators

`[/par/]pos/pd/qcheck`

Description: Enables or disables the RTK engine's ability to take into account the measurement quality indicators.

Access: read & write

Type: boolean

Values: `n | y`

Default: `y`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to take into account the measurement quality indicators.
`set,/par/pos/pd/qcheck,y`
Or
`set,pos/pd/qcheck,y`

Query the current status.
`print,/par/pos/pd/qcheck`
Or
`print,pos/pd/qcheck`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Correlation time for estimating the residual ionosphere

`[/par/]pos/pd/iont`

Description: Sets the a priori correlation time for the RTK engine when estimating the residual ionosphere.

Access: read & write

Type: float

Values: `[0.0...1800.0]`; seconds

Default: `0.0`

Usage Guidelines: N/A

Command Examples: Specify the a priori correlation time for estimating the residual ionosphere to a value of zero (0.0) seconds.

```
set,/par/pos/pd/iont,0.0
```

Or

```
set,pos/pd/iont,0.0
```

Query the current status.

```
print,/par/pos/pd/iont
```

Or

```
print,pos/pd/iont
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Minimum C/A signal-to-noise ratio

`[/par/]pos/pd/minpot`

Description: Specifies the minimum C/A signal-to-noise ratio for satellites used in RTK position computation.

Access: read & write

Type: integer

Values: `[1...60]`; dB*Hz

Default: `31`

Usage Guidelines: Only those satellites will be used in RTK position computation whose CA/L1 signal-to-noise ratios exceed the specified threshold value.

Command Examples: Specify the minimum acceptable C/A signal-to-noise ratio for a satellite to be used in computing the RTK position of the receiver to a value of 33.
`set,/par/pos/pd/minpot,33`
Or
`set,pos/pd/minpot,33`

Query the current status.
`print,/par/pos/pd/minpot`
Or
`print,pos/pd/minpot`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RMS for pseudorange measurements noise

`[/par/]pos/pd/rmsc`

Description:	Sets the a priori RMS value for the pseudorange measurements noise.
Access:	read & write
Type:	float
Values:	`[0.001...100.0]`; meters
Default:	`5.0`
Usage Guidelines:	N/A
Command Examples:	Set the a priori RMS value for the pseudorange measurements noise to 7.0 meters. `set,/par/pos/pd/rmsc,7.0` Or `set,pos/pd/rmsc,7.0` Query the current status. `print,/par/pos/pd/rmsc` Or `print,pos/pd/rmsc`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

RMS for carrier phase measurements noise

`[/par/]pos/pd/rmsp`

Description:	Sets the a priori RMS value for the carrier phase measurements noise.
Access:	read & write
Type:	float
Values:	`[0.001...1.0]`; cycles
Default:	`0.05`
Usage Guidelines:	N/A
Command Examples:	Set the a priori RMS value for the carrier phase measurements noise to 0.05 cycles. `set,/par/pos/pd/rmsp,0.05` Or `set,pos/pd/rmsp,0.05` Query the current status. `print,/par/pos/pd/rmsp` Or `print,pos/pd/rmsp`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Maximum distance between Reference station and Rover

`[/par/]pos/pd/range`

Description: Specifies the maximum allowed distance between the reference and rover stations. If this distance exceeds the specified limit, the rover receiver will not provide the RTK position.

Access: read & write

Type: float

Values: `[0.0...1000000.0]`; meters

Default: `1000000.0`

Usage Guidelines: N/A

Command Examples: Set the maximum distance the rover may be from the reference station to 10,000 meters (10km).

```
set,/par/pos/pd/range,10000.0
```

Or

```
set,pos/pd/range,10000.0
```

Query the current status.

```
print,/par/pos/pd/range
```

Or

```
print,pos/pd/range
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Timeout to wait for data from multiple reference stations

`[/par/]pos/pd/mbtimeout`

Description: Specifies how many epochs the RTK engine will wait for a complete data set, for example, from three reference stations before processing, for example, the data from only two of them.

Access: read & write

Type: integer

Values: `[1...6]`

Default: `3`

Usage Guidelines: N/A

Command Examples: Set the number of epochs the RTK engine will wait for a complete data set to 3.
`set,/par/pos/pd/mbtimeout,3`
Or
`set,pos/pd/mbtimeout,3`

Query the current status.
`print,/par/pos/pd/mbtimeout`
Or
`print,pos/pd/mbtimeout`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Minimum number of satellites for fixing integer ambiguities

`[/par/]pos/pd/minsat`

Description: Specifies the minimum number of satellite the RTK engine will use when fixing integer ambiguities.

Access: read & write

Type: integer

Values: `[4...20]`

Default: `5`

Usage Guidelines: If fewer than the specified number of satellites are usable, then the RTK engine cannot process and resolve integer ambiguities.

Command Examples: Instruct the RTK engine that it must use the minimum eight (8) satellites when fixing integer ambiguities.
`set,/par/pos/pd/minsat,8`
Or
`set,pos/pd/minsat,8`

Query the current status.
`print,/par/pos/pd/minsat`
Or
`print,pos/pd/minsat`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

User defined rover dynamics

`[/par/]pos/pd/dyn`

Description: Toggles the RTK engine between static and kinematic.

Access: read & write

Type: float

Values: `[0.0000...1.0000]`

- `[0.0000...1.0000)` – static; when static is on, the engine uses a running average over a few consecutive raw estimates to decrease the resulting position's noise error.
- `1.0000` – kinematic.

Default: `1.0000`

Usage Guidelines: The RTK engine also provides a *static watchdog mechanism*. When in static, the receiver will automatically monitor estimated coordinates X, Y, Z. Should the position change over 4 centimeters in one of the coordinates, the receiver will immediately switch to *kinematic* running in this mode for the next 30 seconds (whereas the `print` command issued in this time returns with the `static` current status.)

Command Examples: Instruct the RTK engine to run in static mode.
`set,/par/pos/pd/dyn,0`
Or
`set,pos/pd/dyn,0`

Query the current status.
`print,/par/pos/pd/dyn`
Or
`print,pos/pd/dyn`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

RTK computational scheme version

`[/par/]pos/pd/ver`

Description: Specifies the RTK computational scheme version.

Access: read & write

Type: integer

Values: `2`

Default: `2`

Usage Guidelines: N/A

Command Examples: Specify version 2 for the RTK computational scheme.
`set,/par/pos/pd/ver,2`
Or
`set,pos/pd/ver,2`

Query the current status.
`print,/par/pos/pd/ver`
Or
`print,pos/pd/ver`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Reset RTK engine

`[/par/]pos/pd/reset`

Description: Resets the RTK engine. As soon as the command is issued, the receiver will automatically set this parameter back to `off`.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to be reset.
`set,/par/pos/pd/reset,on`
Or
`set,pos/pd/reset,on`

Query the current status.
`print,/par/pos/pd/reset`
Or
`print,pos/pd/reset`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Factor (f) for standard deviation of the residual ionosphere

`[/par/]pos/pd/ionf`

Description: Specifies the value of factor 'f' in the following equation used for the residual ionosphere standard deviation (measured in meters):
stdev_iono_delay = f * 1.e-6 * base_line_length

Access: read & write

Type: float

Values: `[0.0...999999.8]`

Default: `4.0`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to use the factor of 4.0 for the standard deviation of the modeled residual ionoshere.
`set,/par/pos/pd/ionf,4.0`
Or
`set,pos/pd/ionf,4.0`

Query the current status.
`print,/par/pos/pd/inof`
Or
`print,pos/pd/inof`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Environmental condition factor

`[/par/]pos/pd/env`

Description: Sets either normal or less rigid thresholds for filtering out measurement outliers.

Access: read & write

Type: enumerated

Values: `open | forest`

- `open` – the RTK engine will use normal thresholds when searching for measurement outliers. This setting is used if the environment conditions are considered favorable for RTK (many satellites in sight, few obstructions and low multipath).
- `forest` – the RTK engine will use less rigid thresholds when filtering out measurement outliers. This mode is recommended when working under tree canopy or in other cases of high multipath.

Default: `open`

Usage Guidelines: These thresholds are firmware embedded and cannot be changed via any GRIL commands.

Command Examples: Instruct the RTK engine to use less rigid thresholds when filtering out measurement outliers.

```
set,/par/pos/pd/env,forest
```

Or

```
set,pos/pd/env,forest
```

Query the current status.

```
print,/par/pos/pd/env
```

Or

```
print,pos/pd/env
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Invalidate reference station coordinates

`[/par/]pos/pd/ref/clean`

Description: Clears up the currently effective coordinates of the reference station.

Access: write-only

Type: boolean

Values: `off | on`

- `off` – the receiver ignores this value.
- `on` – the RTK engine will disable differential positioning and will be generating single-point positions until updated reference coordinates appear.

Default: `off`

Usage Guidelines: The updated reference coordinates will be received via the radio link (as new RTCM or CMR messages arrive from the base) or can be entered manually at the rover.

Command Examples: Instruct the RTK engine to clear up the currently effective coordinates of the base station

```
set,/par/pos/pd/ref/clean,on
```

Or

```
set,pos/pd/ref/clean,on
```

Query the current status.

```
print,/par/pos/pd/ref/clean
```

Or

```
print,pos/pd/ref/clean
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Pseudorange smoothing in RTK

`[/par/]pos/pd/smr`

Description: Instructs the RTK engine will use the smoothed pseudorange measurements.

Access: read & write

Type: boolean

Values: `off|on`

- `off` – the RTK engine will not use the smoothed pseudorange measurements.
- `on` – the RTK engine will use the smoothed pseudorange measurements.

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to use the smoothed pseudorange measurements.
`set,/par/pos/pd/smr,on`
Or
`set,pos/pd/smr,on`

Query the current status.
`print,/par/pos/pd/smr`
Or
`print,pos/pd/smr`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum extrapolation time

`[/par/]pos/pd/textr`

Description: In extrapolation mode, if the time elapsed since differential corrections were last received exceeds this value, RTK will stop calculating positions.

Access: read & write

Type: float

Values: `[0.0...60.0]`; seconds

Default: `30.0`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to stop calculating positions if more than 25 seconds pass since differential corrections were last received.

```
set,/par/pos/pd/textr,25.0
```

Or

```
set,pos/pd/textr,25.0
```

Query the current status.

```
print,/par/pos/pd/textr
```

Or

```
print,pos/pd/textr
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum number of iterations for float ambiguity

`[/par/]pos/pd/maxit`

Description: Specifies the number of iterations the RTK engine makes when estimating float ambiguities.

Access: read & write

Type: integer

Values: `[0...10]`

Default: `1`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to make 2 iterations when estimating float ambiguities.
`set,/par/pos/pd/maxit,2`
Or
`set,pos/pd/maxit,2`

Query the current status.
`print,/par/pos/pd/maxit`
Or
`print,pos/pd/maxit`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum number of iterations for fixed ambiguity

`[/par/]pos/pd/maxitf`

Description: Specifies the number of iterations the RTK engine makes when estimating the residuals of fixed ambiguities.

Access: read & write

Type: integer

Values: `[0...10]`

Default: `1`

Usage Guidelines: N/A

Command Examples: Instruct the RTK engine to make 2 iterations when estimating the residuals of the fixed ambiguities.

```
set,/par/pos/pd/maxitf,2
```

Or

```
set,pos/pd/maxitf,2
```

Query the current status.

```
print,/par/pos/pd/maxitf
```

Or

```
print,pos/pd/maxitf
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Track C/A L1 only

`[/par/]pos/pd/l1only`

Description: Forces the RTK engine to calculate the final position using only C/A measurements even if the ambiguities have been resolved using P L2 also.

Access: read & write

Type: boolean

Values: `n | y`

Default: `n`

Usage Guidelines: This parameter allows the receiver to stabilize the fixed position in case of poor L2 phase tracking.

Command Examples: Instruct the RTK engine to calculate the final position using only C/A measurements even if the ambiguities have been resolved using P L2 also.

```
set,/par/pos/pd/l1only,y
```

Or

```
set,pos/pd/l1only,y
```

Query the current status.

```
print,/par/pos/pd/l1only
```

Or

```
print,pos/pd/l1only
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Penalty parameter

`[/par/]pos/pd/pen`

Description:	The penalty parameter is used for the known point initialization function.
Access:	read & write
Type:	float
Values:	`[0.0...1000.0]`
Default:	`20.0`
Usage Guidelines:	N/A
Command Examples:	Set the penalty parameter used for known point initializations to a value of 20.0 `set,/par/pos/pd/pen,20.0` Or `set,pos/pd/pen,20.0` Query the current status. `print,/par/pos/pd/pen` Or `print,pos/pd/pen`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Use of FKP data from RTCM message 59

`[/par/]pos/pd/fkp`

Description: Instructs the rover receiver to use ionospheric and geometric corrections from RTCM messages type 59 when computing position.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – ignore FKP data.
- `on` – use FKP data.

Default: `off`

Usage Guidelines: N/A

Command Examples: Instruct the rover receiver to use ionospheric and geometric corrections from RTCM messages type 59 when computing position.

```
set,/par/pos/pd/fkp,on
```

Or

```
set,pos/pd/fkp,on
```

Query the current status.

```
print,/par/pos/pd/fkp
```

Or

```
print,pos/pd/fkp
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

VRS mode

`[/par/]pos/pd/vrs`

Description: Enables and disables the receiver VRS mode.

Access: read & write

Type: boolean

Values: `n | y`

- `y` – RTK will assume that the reference station in use is Virtual Reference Station (VRS).
- `n` – RTK will assume that the reference station is real one.

Default: `n`

Usage Guidelines: N/A

Command Examples: Enable the receivers VRS mode.

```
set,/par/pos/pd/vrs,y
```

Or

```
set,pos/pd/vrs,y
```

Query the current status.

```
print,/par/pos/pd/vrs
```

Or

```
print,pos/pd/vrs
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Halt RTK engine

`[/par/]pos/pd/engine`

Description: Disables and enables the RTK engine.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – disable the RTK engine.
- `on` – enable the RTK engine.

Default: `on`

Usage Guidelines: This parameter is intended to avoid interference with other engines that can produce RTK-level solution.

Command Examples: Disable the RTK engine.
`set,/par/pos/pd/engine,off`
Or
`set,pos/pd/engine,off`

Query the current status.
`print,/par/pos/pd/engine`
Or
`print,pos/pd/engine`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Use of RTK data from external source

`[/par/]pos/pd/scr/ext`

Description:	Enables and disables the receiver to use RTK data obtained from an external source for computing RTK solution.
Access:	write-only
Type:	boolean
Values:	`off | on`
Default:	`on`
Usage Guidelines:	An external source can be, for example, a reference station.
Command Examples:	Enable the use of RTK data from an external source. `set,/par/pos/pd/scr/ext,on` Or `set,pos/pd/scr/ext,on`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Use of RTK data from HP service

`[/par/]pos/pd/scr/hp`

Description:	Enables and disables the receiver to use RTK data obtained from HP service for computing RTK solution.
Access:	read & write
Type:	boolean
Values:	`off | on`
Default:	`on`
Usage Guidelines:	N/A
Command Examples:	Disable the use of HP data for RTK engine. `set,/par/pos/pd/scr/hp,off` Or `set,pos/pd/scr/hp,off` Query the current status. `print,/par/pos/pd/scr/hp` Or `print,pos/pd/scr/hp`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Use of any RTK data

`[/par/]pos/pd/scr/`

Description:	Enables and disables the receiver to use any RTK data for computing RTK solution.
Access:	write-only
Type:	boolean
Values:	`off | on`
Default:	`on`
Usage Guidelines:	N/A
Command Examples:	Enable the use of any RTK data for computing RTK solution. `set,/par/pos/pd/scr/,on` Or `set,pos/pd/scr/,on`
Receiver Availability:	This parameter is not available in the GMS-2 receiver.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

Heading mode

`[/par/]pos/pd/hd/mode`

Description: Enables and disables the heading mode.

Access: read & write

Type: boolean

Values: `off | on`
If set to `on`, this parameter indicates that the mutual distance between the base and the rover antennas remain fixed during the RTK session. This restriction allows the RTK engine to fix ambiguities faster and more accurately.

Default: `off`

Usage Guidelines: N/A

Command Examples: Disable the use of heading mode.
```
set,/par/pos/pd/hd/mode,off
```
Or
```
set,pos/pd/hd/mode,off
```

Query the current status.
```
print,/par/pos/pd/hd/mode
```
Or
```
print,pos/pd/hd/mode
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of fixed length for the baseline measured

`[/par/]pos/pd/hd/uselen`

Description: Enables and disables the use of a priori (fixed) length for the baseline measured.

Access: read & write

Type: boolean

Values: `off | on`
If set to `on`, the RTK engine will use in the heading mode the a priori baseline length from `/par/pos/pd/hd/len/0` (unless this value is equal to zero).
If set to `off`, the RTK engine will compute its own baseline length estimate, which is obtained by averaging instantaneous baseline estimates over a 30-second interval after the first ambiguity fix. Thus, a derived empirical estimate is then used by the RTK engine to improve ambiguity fixing. Also note that this empirical estimate is constantly refined by the receiver as new measurements arrive.

Default: `off`

Usage Guidelines: N/A

Command Examples: Disable the use of a fixed length.
`set,/par/pos/pd/hd/uselen,off`
Or
`set,pos/pd/hd/uselen,off`

Query the current status.
`print,/par/pos/pd/hd/uselen`
Or
`print,pos/pd/hd/uselen`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Value of fixed length for the baseline measured

`[/par/]pos/pd/hd/len/<N>`

Description: Specifies the value of a priori (fixed) length between the reference and rover antennas for the RTK engine #0, 1, and 2, respectively.

Access: read & write

Type: float

Values: `[0.0...10000.0]`; meters
If `/par/pos/pd/hd/len/<N>` = `0` and `/par/pos/pd/hd/uselen` = `on`, the corresponding RTK engine will discard these settings and will start computing a baseline length estimate on its own (as if `/par/pos/pd/hd/uselen` has been set to `off`).

Default: `0.0`

Usage Guidelines: In this parameter, `N` ranges from 0 to 2 and denotes the baseline for the corresponding RTK engine.

Command Examples: Specify the fixed length = 20 meters for RTK engine #0.
`set,/par/pos/pd/hd/len/0,20`
Or
`set,pos/pd/hd/len/0,20`

Query the current status.
`print,/par/pos/pd/hd/len/0`
Or
`print,pos/pd/hd/len/0`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Memory factor for smoothing heading angles

`[/par/]pos/pd/hd/memf`

Description: Specifies the memory factor for smoothing heading angles.

Access: read & write

Type: float

Values: `[0.0...1.0]`

Default: `0.0`

Usage Guidelines: The bigger the memory factor value, the more conservative is smoothing.

Command Examples: Set the parameter to 0.1.
`set,/par/pos/pd/hd/memf,0.1`
Or
`set,pos/pd/hd/memf,0.1`

Query the current status.
`print,/par/pos/pd/hd/memf`
Or
`print,pos/pd/hd/memf`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Multiple Reference Stations

TPS receivers support *multi-base mode* in which the rover receiver is able to obtain RTK data from more than one reference station. Running in this mode, the reference stations broadcast their RTK data using TDMA (Time Division Multiple Access) method. This method allows the reference stations to use a single frequency for transmitting their data. It is achieved by setting a transmission delay for each station.



In order to work in this mode, the receiver must use only CMR Plus messages.

Reference station ID

`[/par/]pos/pd/inuse`

Description: Identifies the reference station (by specifying its ID) you want to use to compute the RTK position.

Access: read & write

Type: integer

Values: `[-1...4095]`

- `-1` – the rover receiver will use RTK corrections from whichever station.
- `[0...4095]` – the rover receiver will use RTK corrections only from the reference station having the same ID. Data from all the other reference stations will be ignored. Thereby it guarantees that the rover will work properly if there are two or more reference stations transmitting RTK data on the same frequency.

Default: `-1`

Usage Guidelines: Setting the parameter to `-1` whereas two or more sources of RTK data are available simultaneously may result in inability to get the RTK solution since RTK data received from several reference stations will be mixed with each other.

Command Examples: Specify reference station ID = 25.

```
set,/par/pos/pd/inuse,25
```

Or

```
set,pos/pd/inuse,25
```

Query the current status.

```
print,/par/pos/pd/inuse
```

Or

```
print,pos/pd/inuse
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Selecting the nearest reference station automatically

`[/par/]pos/pd/nrs/mode`

Description: Instructs the rover receiver to use RTK data broadcasted by the nearest reference station.

Access: read & write

Type: boolean

Values: `n | y`

- `n` – the rover receiver will use RTK corrections from whichever reference station.
- `y` – the rover receiver will use RTK corrections from the nearest reference station.

Default: `n`

Usage Guidelines: The receiver will use this parameter only if `/par/pos/pd/inuse` is set to `-1`.

Command Examples: Use the nearest reference station.

```
set,/par/pos/pd/nrs/mode,y
```

Or

```
set,pos/pd/nrs/mode,y
```

Query the current status.

```
print,/par/pos/pd/nrs/mode
```

Or

```
print,pos/pd/nrs/mode
```

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Switching between reference stations

`[/par/]pos/pd/nrs/lim`

Description: Specifies the minimum difference in distances between the rover and reference stations for switching from the previous nearest reference station to the current one.

Access: read & write

Type: float

Values: `[1.0...10000.0]`; meters

Default: `25.0`

Usage Guidelines: This parameter can be useful if the receiver is located on the border of two or more reference stations. Assuming that one of these reference stations has been selected as the nearest one. If the receiver is either stationary or moving along the border, the selection of the nearest station may be confused. To avoid conflicts with the selection of the nearest station, you should use this parameter. Switching to the new reference station will occur only if the difference between the distance to the new reference station and the distance to the current reference station exceeds the specified limit.

Command Examples: Set the threshold value to 100 meters.
`set,/par/pos/pd/nrs/lim,100`
Or
`set,pos/pd/nrs/lim,100`

Query the current status.
`print,/par/pos/pd/nrs/lim`
Or
`print,pos/pd/nrs/lim`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum counter of attempts to stay the nearest reference station

`[/par/]pos/pd/nrs/cnt`

Description: Defines the counter that serves for setting up a total number of attempts during which the new nearest reference station will remain the nearest one among the other reference stations available. If during those attempts, at least one of the other reference stations is detected as the nearest one, the counter will be reset and started again.

Access: read & write

Type: integer

Values: `[1...30000]`

Default: `10`

Usage Guidelines: This parameter guarantees that a new nearest reference station will not be selected by mistake.
Similarly to the parameter `/par/pos/pd/nrs/lim`, this parameter can be used to avoid some kind of confusion when the receiver selects the nearest reference station.
Switching to a new reference station will take place if:

- during the number of attempts specified with `/par/pos/pd/nrs/cnt` the new reference station is nearer than the current reference station and
- the difference between the distance to the new reference station and the distance to the current reference station exceeds the limit specified with the parameter `/par/pos/pd/nrs/lim`.

The rover estimates the distances to the reference stations each time when it receives CMR message type 0.

Command Examples: Set the counter of attempts to 20.
`set,/par/pos/pd/nrs/cnt,20`
Or
`set,pos/pd/nrs/cnt,20`

Query the current status.
`print,/par/pos/pd/nrs/cnt`
Or
`print,pos/pd/nrs/cnt`

Receiver Availability: This parameter is not available in the GMS-2 receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

SBAS Parameters

The TPS receiver has a number of independent *channels* that can be allocated to SBAS satellites, including the Wide Area Augmentation System (WAAS), the European Geostationary Navigation Overlay Service (EGNOS), and the MTSAT Satellite-based Augmentation System (MSAS). Either channel can track any one of the SBAS satellites.

The WAAS signal and GPS corrections are available 24 hours a day, seven days a week throughout the majority of the continental U.S. and large parts of Canada and Alaska. For more information on WAAS, visit the FAA website (http://gps.faa.gov/).

EGNOS is a similar satellite system under development of the European Space Agency (ESA), the European Commission (EC), Eurocontrol, and the European Organization for the Safety of Air Navigation. For more information on EGNOS, visit the ESA website (http://www.esa.int/esaNA/egnos.html).

MTSAT is the wide area augmentation system being developed and operated by the Japan Civil Aviation Bureau (JCAB) for civil Aviation and Meteorological Agency.

SBAS Parameters for G3- and TG3-based Receivers

Tracking SBAS satellites

`[/par/]lock/waas/sat`

Description: Enables and disables the receiver to track SBAS satellites.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not track SBAS satellites.
- `on` – the receiver will track SBAS satellites.

Default: `off`

Usage Guidelines:

- The `print` command returns a list of enabled and disabled satellites.
- Fast acquisition/reacquisition for SBAS satellites is not very fast actually. Some parameters (such as ionospheric delays) are not stored in NVRAM so that TTFF (time to first fix) may be greater than that for GPS SVs.

`[/par/]lock/waas/sat` *(Continued)*

Command Examples: Enable tracking.
`set,/par/lock/waas/sat,on`
Or
`set,lock/waas/sat,on`

Query the tracking status.
`print,/par/lock/waas/sat`
Or
`print,lock/waas/sat`

Receiver Availability: This parameter is available in G3- and TG3-based receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Tracking specified SBAS satellite

`[/par/]lock/waas/sat/<prn>`

Description: Enables and disables the receiver to track the specified SBAS satellite.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will not track the specified SBAS satellite.
- `on` – the receiver will track the specified SBAS satellite.

Default: `off`

Usage Guidelines:
- `prn` ranges from 120 to 138.
- Fast acquisition/reacquisition for SBAS satellites is not very fast actually. Some parameters (such as ionospheric delays) are not stored in NV RAM so that TTFF (time to first fix) may be greater than that for GPS SVs.

Command Examples: Enable tracking of SBAS satellite with PRN 138.
`set,/par/lock/waas/sat/138,on`
Or
`set,lock/waas/sat/138,on`

Query the tracking status of SBAS satellite with PRN 138.
`print,/par/lock/waas/sat/138`
Or
`print,lock/waas/sat/138`

Receiver Availability: This parameter is available in G3- and TG3-based receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Smoothing range rate corrections

`[/par/]/waas/1/alp`

Description: Enables and disables the receiver to apply smoothing to the range rate corrections.

Access: read & write

Type: float

Values: `[0.10...1.00]`

- `[0.10...0.99]` – the receiver will apply smoothing to RRCs using the selected filtering time constant. This constant defines the degree of smoothing.
- `1.00` – the receiver will not apply smoothing to RRCs.

Default: `0.50`

Usage Guidelines: N/A

Command Examples: Apply smoothing with the filter constant set to 0.1.

```
set,/par/waas/1/alp,0.1
```

Or

```
set,lock/waas/1/alp,0.1
```

Confirm the status of smoothing.

```
print,/par/waas/1/alp
```

Or

```
print,/par/waas/1/alp
```

Receiver Availability: This parameter is available in G3- and TG3-based receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

SBAS Parameters for Other Receivers

These TPS receivers have two independent channels that can be allocated to SBAS satellites. Either channel can track any one of the SBAS satellites.

Tracking selected SBAS satellite

`[/par/]/waas/<1|2>/sat`

Description: Tracks a SBAS satellite by assigning to it a PRN corresponding to one of the GPS satellites.

Access: read & write

Type: integer

Values: `[0...138]`

Default: `0`

Usage Guidelines: Fast acquisition/reacquisition for SBAS satellites is not very fast actually. Some parameters (such as ionospheric delays) are not stored in NV RAM so that TTFF (time to first fix) may be greater than that for GPS SVs.

Command Examples: Associate SBAS PRN 122 (Inmarsat AOR-W) with GPS PRN 32.
```
set,/par/waas/1/sat,12232
```
Or
```
set,waas/1/sat,12232
```

Associate SBAS PRN 134 (Inmarsat POR) with GPS PRN 12.
```
set,/par/waas/2/sat,13412
```
Or
```
set,waas/2/sat,13412
```

Stop tracking a SBAS satellite in the second channel.
```
set,/par/waas/2/sat,0
```
Or
```
set,waas/2/sat,0
```

Confirm which SBAS and GPS PRNs are used in the first channel.
```
print,/par/waas/1/sat
```
Or
```
print,waas/1/sat
```

Receiver Availability: This parameter is available in Hiper and Legacy families, Odyssey-RS, and other non G3-based boards and receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of additional information from SBAS satellite

`[/par/]/waas/raw`

Description: Outputs raw one-second and some other decoded SBAS data into serial port A.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Output raw data to port A.
`set,/par/waas/raw,on`
Or
`set,waas/raw,on`

Check the output.
`print,/par/waas/raw`
Or
`print,waas/raw`

Receiver Availability: This parameter is available in Hiper and Legacy families, Odyssey-RS, and other non G3-based boards and receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Use of true SBAS PRNs in SI message

`[/par/]/waas/numb`

Description: Outputs true PRNs in SI messages rather than associated with a GPS PRN.

Access: read & write

Type: boolean

Values: `off | on`

Default: `off`

Usage Guidelines: N/A

Command Examples: Enable outputting of true SBAS PRNs.
`set,/par/waas/numb,on`
Or
`set,waas/numb,on`

Check the current status.
`print,/par/waas/numb`
Or
`print,waas/numb`

Receiver Availability: This parameter is available in Hiper and Legacy families, Odyssey-RS, and other non G3-based boards and receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Maximum age of SBAS corrections

`[/par/]/waas/<1|2>/timeout`

Description: Specifies how long the receiver will wait for new SBAS code differential corrections before producing a stand-alone position.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – the receiver will wait for the new corrections during the amount of time specified by the `/par/pos/cd/maxage` parameter (up to 1200 seconds).
- `on` – the receiver will wait for the new corrections up to 12 seconds. If the age of the corrections exceeds 12 seconds, the receiver will compute a stand-alone position.

Default: `on`

Usage Guidelines: N/A

Command Examples: Set the parameter to the off value for the first channel.

```
set,/par/waas/1/timeout,off
```

Or

```
set,waas/1/timeout,off
```

Check the current status.

```
print,/par/waas/1/timeout
```

Or

```
print,waas/1/timeout
```

Receiver Availability: This parameter is available in Hiper and Legacy families, Odyssey-RS, and other non G3-based boards and receivers.

History: This parameter was introduced in firmware 2.3p4.

See Also: N/A

Common SBAS Parameters

This section describes SBAS parameters available in G3- and TG3-based receivers, and any other receivers.

Use of message type 0

`[/par/]/waas/<1|2>/instead0`

Description: Enables and disables the receiver to use any data from SBAS satellites if message type 0 is being broadcast.

Access: read & write

Type: integer

Values: `[0...99]`

- `0` – the receiver will not substitute message type 0 and therefore will not use any data from the SBAS satellite.
- `[1...99]` – the receiver will substitute the selected message type for message type 0 and therefore will interpret message type 0 as the message with the selected type.

Default: `0`

Usage Guidelines:

- SBAS specifications forbid to use any data from the satellites transmitting message type 0. This parameter allows you to override the limitation.
- Use non-default values with caution. Be sure to understand this parameter before making changes. Its usage may result in system damage, loss of data, or personal injury.

Command Examples: Interpret message type 0 as message type 2. Because message 0 contains same data as message 2, this command will allow you to use WAAS satellites.

```
set,/par/waas/1/instead0,2
```

Or

```
set,waas/1/instead0,2
```

Interpret message type 0 as message type 99. Currently you execute this command when you need to run an EGNOS satellite so that to make the receiver interpret message 0 as one of non-existing messages (e.g., message 99)

```
set,/par/waas/1/instead0,99
```

Or

```
set,waas/1/instead0,99
```

Confirm how message 0 is interpreted.

```
print,/par/waas/1/instead0
```

Or

```
print,waas/1/instead0
```

`[/par/]/waas/<1|2>/instead0` *(Continued)*

Receiver Availability:
- This parameter is available as `/par/waas/1/instead0` in G3- and TG3-based receivers.
- This parameter is available as `/par/waas/<1|2>/instead0` in all other receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Applying ionospheric corrections from SBAS satellite

`[/par/]/waas/<1|2>/ion`

Description: Enables and disables the receiver to apply ionospheric delay corrections broadcasted by a SBAS satellite to the GPS satellites.

Access: read & write

Type: integer

Values: `[0...2]`

- `0` – the receiver will not apply ionospheric corrections at all when computing a code differential position.
- `1` – the receiver will apply the available ionospheric corrections to the corresponding GPS satellites when computing the code differential position. If iono corrections are unavailable for some satellites, the receiver will apply the standard Klobucher iono model for these satellites.
- `2` – the receiver will apply the available ionospheric corrections to the corresponding GPS satellites and only these satellites will be used when computing the code differential position. If the number of GPS satellites with ionospheric corrections is not enough for computing the code differential position, the receiver will produce a standalone position.

Default: `2`

Usage Guidelines: Ionospheric corrections may not be available for some low-elevation GPS satellites in some boundary regions.

Command Examples: Disable the usage of iono corrections at all.

```
set,/par/waas/1/ion,0
```

Or

```
set,waas/1/ion,0
```

Query the current ionosphere correction mode.

```
print,/par/waas/1/ion
```

Or

```
print,waas/1/ion
```

Receiver Availability:

- This parameter is available as `/par/waas/1/ion` in G3- and TG3-based receivers.
- This parameter is available as `/par/waas/<1|2>/ion` in all other receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Method used for selecting maximum age of SBAS fast corrections

`[/par/]/waas/fastcor/mode`

Description: Specifies the method used for selecting the maximum age of SBAS fast corrections.

Access: read & write

Type: enumerated

Values: `auto | manual`

- `auto` – the receiver will automatically select the maximum age of fast corrections according to the broadcast data.
- `manual` – the receiver will apply a user-defined maximum age.

Default: `auto`

Usage Guidelines: If you select `manual`, specify the age using the `/par/waas/fastcor/maxage` parameter.

Command Examples: Set the mode for maximum age to manual.

```
set,/par/waas/fastcor/mode,manual
```

Or

```
set,waas/fastcor/mode,manual
```

Check the mode.

```
print,/par/waas/fastcor/mode
```

Or

```
print,waas/fastcor/mode
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

Maximum age of SBAS fast corrections in *manual* mode

`[/par/]/waas/fastcor/maxage`

Description: Specifies the maximum age of SBAS fast corrections in manual mode.

Access: read & write

Type: integer

Values: `[1...1200]`; expressed in seconds

Default: `18`

Usage Guidelines: This parameter is effective if `/par/waas/fastcor/mode` is set to `manual`.

Command Examples: Set the maximum age to 10 seconds.
`set,/par/waas/fastcor/maxage,10`
Or
`set,waas/fastcor/maxage,10`

Confirm the value of age.
`print,/par/waas/fastcor/maxage`
Or
`print,waas/fastcor/maxage,10`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.1.

See Also: N/A

OmniSTAR Parameters

OmniSTAR is a wide-area differential GPS service, using satellite broadcast techniques. Data from many widely-spaced Reference Stations are used in a proprietary multi-site solution to achieve sub-meter positioning over most land areas worldwide. This is done by mathematically weighting each reference station as a function of its relative position. The result is one set of RTCM-SC-104, Version 2 corrections, optimized for the user location. This makes OmniStar VBS highly suitable for mobile applications or dynamic positioning.

OmniSTAR is a member of the Fugro group of companies with offices throughout the world. Topcon has implemented the OmniStar Virtual Base Station (VBS) into the own firmware. For more information about OmniStar, visit http://www.omnistar.com/.



The parameters described in this section are applicable only to the MAP-R receivers.

VBS Version

`[/par/]/omni/ver`

Description:	Reports the VBS version.
Access:	read-only
Type:	float
Values:	N/A
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Confirm the VBS version. `print,/par/omni/ver` Or `print,omni/ver`
Receiver Availability:	This parameter is available only in the MAP-R receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

VBS status flags

`[/par/]/omni/stat`

Description: Reports the VBS status flags in hexadecimal format.

Access: read-only

Type: integer

Values: VBS_WARN_NEED_UPDATE = 0x8000 – Some data for the VBS need to be updated (the VBS update is performed automatically). Note however that during the updating procedure differential corrections are not available.
VBS_ERROR_NO_TIME = 0x0080 – No time is available from GPS engine.
VBS_ERROR_NO_POSITION = 0x0040 – No position is available from GPS engine.
VBS_ERROR_NO_ALMANAC = 0x0020 – VBS almanac data are unavailable. The VBS is still waiting for the data.
VBS_ERROR_NO_REMOTE_SITES = 0x0010 – Information about remote sites is not available. The VBS is still waiting for the data.
VBS_ERROR_LINK = 0x0008 – No subscription to the link with the current satellite.
VBS_ERROR_WET = 0x0004 – No subscription to work on the sea.
VBS_ERROR_REGION=0x0002 – No subscription to the current region.
VBS_ERROR_EXPIRATION=0x0001 – No subscription.

Default: N/A

Usage Guidelines: N/A

Command Examples: Print the status flag.

```
print,/par/omni/stat
```

Or

```
print,omni/stat
```

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Fugro ID

`[/par/]/omni/sub/num`

Description: Reports the Fugro ID number.

Access: read-only

Type: integer

Values: `[200000...299999]`

Default: N/A

Usage Guidelines: This ID is required by OmniSTAR for subscribing to their service.

Command Examples: Print the ID number.
```
print,/par/omni/sub/num
```
Or
```
print,omni/sub/num
```

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Subscription start time

`[/par/]/omni/sub/start`

Description: Reports the subscription start time.

Access: read-only

Type: integer

Values: N/A; seconds

Default: N/A

Usage Guidelines: N/A

Command Examples: Print the subscription start time.
```
print,/par/omni/sub/start
```
Or
```
print,omni/sub/start
```

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Subscription expiration time

`[/par/]/omni/sub/exp`

Description: Reports the subscription expiration time.

Access: read-only

Type: integer

Values: N/A; seconds

Default: N/A

Usage Guidelines: N/A

Command Examples: Print the subscription expiration time.
`print,/par/omni/sub/exp`
Or
`print,omni/sub/exp`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Subscription extension time

`[/par/]/omni/sub/ext`

Description: Reports the subscription extension time.

Access: read-only

Type: integer

Values: N/A; seconds

Default: N/A

Usage Guidelines: N/A

Command Examples: Print the subscription extension time.
`print,/par/omni/sub/ext`
Or
`print,omni/sub/ext`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Subscription time in hours

`[/par/]/omni/sub/hour`

Description:	Reports how much time is left for an OmniSTAR subscription that was purchased hourly.
Access:	read-only
Type:	integer
Values:	N/A; hours
Default:	N/A
Usage Guidelines:	The count of the remaining time starts as soon as VBS is turned on and stops when VBS is turned off.
Command Examples:	Print the subscription time. `print,/par/omni/sub/hour` Or `print,omni/sub/hour`
Receiver Availability:	This parameter is available only in the MAP-R receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Duplicate OmniSTAR output stream

`[/par/]/omni/dup`

Description: Specifies the name of output stream where the data decoded by OmniSTAR module (in RTCM format) are to be duplicated.

Access: read & write

Type: enumerated

Values: `port`

Default: `/dev/null`

Usage Guidelines: By default, no duplication is performed (i.e. the data is being sent to the position computation engine only).

Command Examples: Duplicate decoded OmniSTAR data to serial port D.

```
set,/par/omni/dup,/dev/ser/d
```

Or

```
set,omni/dup,/dev/ser/d
```

Confirm where the data are duplicated.

```
print,/par/omni/dup
```

Or

```
print,omni/dup
```

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Clearing the NVRAM of the OmniSTAR module

`[/par/]/omni/clear`

Description: Clears the data stored in the OmniSTAR's NVRAM (almanac, subscription, etc.).

Access: read & write

Type: list {alm,nvr,sit,sub} of boolean values

Values: `{y|n,y|n,y|n,y|n}`

Default: `{n,n,n,n}`

Usage Guidelines: After executing this parameter with the `set` command, the parameter's default values `n` are immediately restored.

Command Examples: Clear OmniSTAR's NVRAM.
`set,/par/omni/clear,{y,y,y,y}`
Or
`set,omni/clear,{y,y,y,y}`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Clearing OmniSTAR almanac in the OmniSTAR NVRAM

`[/par/]/omni/clear/alm`

Description: Clears the OmniSTAR almanac in the OmniSTAR NVRAM.

Access: read & write

Type: boolean

Values: `y|n`

Default: `n`

Usage Guidelines: After executing this parameter with the `set` command, the parameter's default value `n` is immediately restored.

Command Examples: Clear the OmniSTAR almanac.
`set,/par/omni/clear/alm,y`
Or
`set,omni/clear/alm,y`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Clearing OmniSTAR NVRAM in the receiver's NVRAM

`[/par/]/omni/clear/nvr`

Description: Clears the OmniSTAR NVRAM in the receiver's NVRAM.

Access: read & write

Type: boolean

Values: `y|n`

Default: `n`

Usage Guidelines: After executing this parameter with the `set` command, the parameter's default value `n` is immediately restored.

Command Examples: Clear the OmniSTAR NVRAM.
`set,/par/omni/clear/nvr,y`
Or
`set,omni/clear/nvr,y`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Clearing OmniSTAR sites in the OmniSTAR NVRAM

`[/par/]/omni/clear/sit`

Description: Clears the OmniSTAR sites in the OmniSTAR NVRAM.

Access: read & write

Type: boolean

Values: `y|n`

Default: `n`

Usage Guidelines: After executing this parameter with the `set` command, the parameter's default value `n` is immediately restored.

Command Examples: Clear the OmniSTAR sites.
`set,/par/omni/clear/sit,y`
Or
`set,omni/clear/sit,y`

Receiver Availability: This parameter is available only in the MAP-R receiver.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Clearing OmniSTAR subscription in the OmniSTAR NVRAM

`[/par/]/omni/clear/sit`

Description:	Clears the OmniSTAR subsription in the OmniSTAR NVRAM.
Access:	read & write
Type:	boolean
Values:	`y|n`
Default:	`n`
Usage Guidelines:	After executing this parameter with the `set` command, the parameter's default value `n` is immediately restored.
Command Examples:	Clear the OmniSTAR subscription. `set,/par/omni/clear/sub,y` Or `set,omni/clear/sub,y`
Receiver Availability:	This parameter is available only in the MAP-R receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

BINEX Parameters

This section describes the parameters used for enabling, disabling, and configuring various fields in the record 0x00.

Refer to the BINEX section of the UNAVCO Web site at http://binex.unavco.org/binex.html for a detailed description of the standard.

Contents of the 0x04 field for the 0x00 record

`[/par/]binex/site`

Description: Specifies the site name and its description.

Access: read & write

Type: string

Values: `[0...127]`

Default: ""

Usage Guidelines: With this parameter, you specify only the contents of the 0x04 field. Its output to the 0x00 record controls the parameter `/par/binex/00_00/04`.

Command Examples: Specify the site name.
`set,/par/binex/site,Tharsis`
Or
`set,binex/site,Tharsis`

Query the site name.
`print,/par/binex/site`
Or
`print,binex/site`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: Refer to the parameter `/par/binex/00_00/04` for how to activate the output of this field within the 0x00 record.

Contents of the 0x0f field for the 0x00 record

`[/par/]binex/data_id`

Description: Specifies a 4-character ASCII code used in the 0x0f field.

Access: read & write

Type: string

Values: `[0...4]`

Default: ""

Usage Guidelines:
- With this parameter, you specify only the contents of the 0x0f field. Its output to the 0x00 record controls the parameter `/par/binex/00_00/0f`.
- If you specify less than 4 characters, the receiver pads the string on the right to the required size with spaces.

Command Examples:
Specify the code.
`set,/par/binex/data_id,NB23`
Or
`set,binex/data_id,NB23`

Query the code.
`print,/par/binex/data_id`
Or
`print,binex/data_id`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: Refer to the parameter `/par/binex/00_00/0f` for how to activate the output of this field within the 0x00 record.

Enable output of the 0x04 field to the 0x00 record

`[/par/]binex/00_00/04`

Description: Enables and disables output of the contents of the 0x04 field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x04 is disabled for output
- `on` – 0x04 is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x04.

```
set,/par/binex/00_00/04,on
```

Or

```
set,binex/00_00/04,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/04
```

Or

```
print,binex/00_00/04
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x0f field to the 0x00 record

`[/par/]binex/00_00/0f`

Description: Enables and disables the output of the contents of the 0x0f field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x0f is disabled for output
- `on` – 0x0f is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x0f.

```
set,/par/binex/00_00/0f,on
```

Or

```
set,binex/00_00/0f,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/0f
```

Or

```
print,binex/00_00/0f
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x17 field to the 0x00 record

`[/par/]binex/00_00/17`

Description: Enables and disables the output of the contents of the 0x17 field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x17 is disabled for output
- `on` – 0x17 is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x17.

```
set,/par/binex/00_00/17,on
```

Or

```
set,binex/00_00/17,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/17
```

Or

```
print,binex/00_00/17
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x19 field to the 0x00 record

`[/par/]binex/00_00/19`

Description: Enables and disables the output of the contents of the 0x19 field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x19 is disabled for output
- `on` – 0x19 is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x19.

```
set,/par/binex/00_00/19,on
```

Or

```
set,binex/00_00/19,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/19
```

Or

```
print,binex/00_00/19
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x1a field to the 0x00 record

`[/par/]binex/00_00/1a`

Description: Enables and disables the output of the contents of the 0x1a field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x1a is disabled for output
- `on` – 0x1a is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x1a.

```
set,/par/binex/00_00/1a,on
```

Or

```
set,binex/00_00/1a,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/1a
```

Or

```
print,binex/00_00/1a
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x1b field to the 0x00 record

`[/par/]binex/00_00/1b`

Description: Enables and disables the output of the contents of the 0x1b field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x1b is disabled for output
- `on` – 0x1b is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x1b.

```
set,/par/binex/00_00/1b,on
```

Or

```
set,binex/00_00/1b,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/1b
```

Or

```
print,binex/00_00/1b
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x1d field to the 0x00 record

`[/par/]binex/00_00/1d`

Description: Enables and disables the output of the contents of the 0x1d field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x1d is disabled for output
- `on` – 0x1d is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x1d.

```
set,/par/binex/00_00/1d,on
```

Or

```
set,binex/00_00/1d,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/1d
```

Or

```
print,binex/00_00/1d
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of the 0x1f field to the 0x00 record

`[/par/]binex/00_00/1f`

Description: Enables and disables the output of the contents of the 0x1f field to the 0x00 record.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – 0x1f is disabled for output
- `on` – 0x1f is enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x1f.

```
set,/par/binex/00_00/1f,on
```

Or

```
set,binex/00_00/1f,on
```

Query to confirm the output is enabled.

```
print,/par/binex/00_00/1f
```

Or

```
print,binex/00_00/1f
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Enable output of all fields to the 0x00 record with a single parameter

`[/par/]binex/00_00/`

Description: Enables and disables the output of the contents of all fields to the 0x00 record.

Access: write-only

Type: boolean

Values: `off | on`

- `off` – all fields are disabled for output
- `on` – all fields are enabled for output

Default: `on`

Usage Guidelines: N/A

Command Examples: Enable output of 0x1f.

```
set,/par/binex/00_00/,on
```

Or

```
set,binex/00_00/,on
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Status of fields available for the 0x00 record

`[/par/]binex/00_00/`

Description: Verifies whether the fields are enabled for output to the 0x00 record or not.

Access: read-only

Type: list of boolean for fields 04, 0f, 17, 19, 1a, 1b, 1d, 1f

Values: `off | on`

- `off` – a field is disabled for output
- `on` – a field is enabled for output

Default: N/A

Usage Guidelines: N/A

Command Examples: Verify the status of fields.

```
print,/par/binex/00_00:on
```

Or

```
print,binex/00_00
```

Reply to the first command.

```
RE042/par/binex/
00_00={04=on,0f=on,17=on,19=on,1a=on,1b=on,1d=on
,1f=on}
```

It indicates that all fields are enabled for output to the record 0x00.

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

List of parameters used to control the 0x00 record

`[/par/]binex`

Description: Lists the parameters used to control the contents and output of the fields for the 0x00 record.

Access: read-only

Type: list of parameters

Values: `{site, data_id, 00_00}`

- for `site`, see
- for `data_id`, see
- for `00_00`, see

Default: N/A

Usage Guidelines: N/A

Command Examples: Query the parameters.

```
print,/par/binex:on
```

Or

```
print,binex
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 2.5p1.

See Also: N/A

Receiver Options

Receiver options enable or disable the various functions and features available for a TPS receiver. The options must be loaded onto the receiver to activate the desired functionality.

A receiver option can be purchased, leased, or purchased and leased at the same time.

A leased option is defined by the leased value and the expiration date.

A purchased option is defined only by the purchased value because such an option has no expiration date.

Because options can be both purchased and leased, the current value of the option displays the current effective value.

GRIL defines the following descriptors for each option:

- Option name – the name of the option represented as a 4-character code.
- Current value –indicates whether the option is in force at the present or not.
- Purchased value – indicates whether the option is purchased or not.
- Leased value – indicates whether the option is leased or not.
- Expiration date – the date the leased option will be disabled, if applicable.

Option values can be the following:

- -1 – the firmware or hardware version does not support this option.
- 0 – the receiver option is disabled.
- positive integer – the option is enabled with a specified value. Note that the current value will coincide with the larger of the purchased and leased values.

With the following parameters, you can retrieve information about the receiver options.

Current value for specified option

`[/par/]opts/<NAME>/cur`

Description: Reports the currently effective value for the specified option.

Access: read only

Type: integer

Values: `[-1...511]`

- `-1` – the firmware or hardware version does not support this option.
- `0` – the receiver option is disabled.
- `[1...511]` – the option is enabled with a specified value.

Default: N/A

`[/par/]opts/<NAME>/cur` *(Continued)*

Usage Guidelines: For a positive integer, the current value will coincide with the larger of the purchased and leased values.

Command Examples: Report the current value for the _GPS option.
`print,/par/opts/_GPS/cur`
Or
`print,opts/_GPS/cur`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to “Receiver Options” on page E-1 for a complete list of options available in a TPS receiver.

Purchased value for specified option

`[/par/]opts/<NAME>/purchased`

Description: Reports the purchased value for the specified option.

Access: read only

Type: integer

Values: `[0...511]`

Default: N/A

Usage Guidelines: The purchased option cannot be expired.

Command Examples: Report the purchased value for the DIST option.
`print,/par/opts/DIST/purchased`
Or
`print,opts/DIST/purchased`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to “Receiver Options” on page E-1 for a complete list of options available in a TPS receiver.

Leased value for specified option

`[/par/]opts/<NAME>/leased`

Description:	Reports the leased value for the specified option.
Access:	read only
Type:	integer
Values:	`[0...511]`
Default:	N/A
Usage Guidelines:	Always check the expiration date to ensure the option is in force at the present.
Command Examples:	Report the leased value for the DIST option. `print,/par/opts/DIST/leased` Or `print,opts/DIST/leased`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	Refer to “Receiver Options” on page E-1 for a complete list of options available in a TPS receiver.

Expiration date for specified option

`[/par/]opts/<NAME>/date`

Description: Reports the date the leased option will be disabled.

Access: read only

Type: string

Values: `ddmmyy`

- `dd` – day (01...31)
- `mm` – month (01...12)
- `yy` – year (00...99)

Default: N/A

Usage Guidelines: The expiration date is only available for a leased option. Zero is shown if the leased value is unavailable.

Command Examples: Report the expiration date for the DIST option.

```
print,/par/opts/DIST/date
```

Or

```
print,opts/DIST/date
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: Refer to “Receiver Options” on page E-1 for a complete list of options available in a TPS receiver.

Complete information about specified option

`[/par/]opts/<NAME>`

Description:	Reports the complete information about the specified option.
Access:	read only
Type:	list
Values:	`{cur,purchased,leased,date}`
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Report the complete information for the DIST option. `print,/par/opts/DIST` Or `print,opts/DIST`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	Refer to "Receiver Options" on page E-1 for a complete list of options available in a TPS receiver.

Complete information about all options

`[/par/]opts`

Description:	Reports the complete information about all options for the receiver.
Access:	read only
Type:	list
Values:	`{cur,purchased,leased,date}`
Default:	N/A
Usage Guidelines:	N/A
Command Examples:	Report the complete information about all options. `print,/par/opts` Or `print,opts`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in the first release of firmware.
See Also:	Refer to "Receiver Options" on page E-1 for a complete list of options available in a TPS receiver.

AutoSeed Functionality

The Auto-seed feature will provide users the ability to quickly setup and begin RTK operations without using an external interface to localize the base receiver. The user will simply set up over a mark and power up. The base receiver will automatically select a new position gathered by autonomous averaging and save it for later re-occupation. If the point has been previously occupied and the receiver position falls within proper tolerance, it will select a point from positions stored in memory.

Parameters for working with the current point

Name of the current point

`[/par/]ref/avg/list/curpnt/name`

Description: Specifies the name of the current point.

Access: read & write

Type: string [0...19]

Values: up to 19 alphanumeric characters

Default: `"UNUSED"`

Usage Guidelines: N/A

Command Examples: Set the name to point1.

```
set,/par/ref/avg/list/curpnt/name,point1
```

Or

```
set,ref/avg/list/curpnt/name,point1
```

Query the name of the current point.

```
print,/par/ref/avg/list/curpnt/name
```

Or

```
print,ref/avg/list/curpnt/name
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Identifier of the current point

`[/par/]ref/avg/list/curpnt/id`

Description: Specifies the identifier of the current point.

Access: read & write

Type: integer

Values: `[0...4095]`

Default: `0`

Usage Guidelines: N/A

Command Examples: Set the identifier to 3.
`set,/par/ref/avg/list/curpnt/id,3`
Or
`set,ref/avg/list/curpnt/id,3`

Query the identifier of the current point.
`print,/par/ref/avg/list/curpnt/id`
Or
`print,ref/avg/list/curpnt/id`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

L1PC coordinates of the current point

`[/par/]ref/avg/list/curpnt/pos/<xyz|geo>`

Description: Specifies the L1PC coordinates of the current point in WGS-84.

Access: read & write

Type: {datum,x,y,z}
{datum,lat,lon,alt}

Values: as defined in "Datum Parameters" on page 3-299

Default: `{W84,+6378137.0000,+0.0000,+0.0000}` for xyz
`{W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000}` for geo

Usage Guidelines: N/A

Command Examples: Set the Cartesian coordinates of the current point.

```
set,/par/ref/avg/list/curpnt/pos/
xyz,{W84,+2850683.2396,+2199377.4170,+5247241.5569}
```

Or

```
set,ref/avg/list/curpnt/curpnt/pos/
xyz,{W84,+2850683.2396,+2199377.4170,+5247241.5569}
```

Query the geodetic coordinates of the current point.

```
print,/par/ref/avg/list/curpnt/pos/geo
```

Or

```
print,ref/avg/list/curpnt/pos/geo
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

ARP coordinates of the current point

`[/par/]ref/avg/list/curpnt/arp/<xyz|geo>`

Description: Specifies the ARP coordinates of the current point in WGS-84.

Access: read & write

Type: {datum,x,y,z}
{datum,lat,lon,alt}

Values: as defined in "Datum Parameters" on page 3-299

Default: `{W84,+6378137.0000,+0.0000,+0.0000}` for xyz
`{W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000}` for geo

Usage Guidelines: N/A

Command Examples: Set the Cartesian coordinates of the current point.

```
set,/par/ref/avg/list/curpnt/arp/
xyz,{W84,+2850683.2396,+2199377.4170,+5247241.5569}
```

Or

```
set,ref/avg/list/curpnt/curpnt/arp/
xyz,{W84,+2850683.2396,+2199377.4170,+5247241.5569}
```

Query the geodetic coordinates of the current point.

```
print,/par/ref/avg/list/curpnt/arp/geo
```

Or

```
print,ref/avg/list/curpnt/arp/geo
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Automatic fill the fields of the current point

`[/par/]ref/avg/list/curpnt/fill`

Description: Fills the fields of the current point automatically.

Access: write-only

Type: enumerated

Values: `all | pos | name | id | 0...99`

- `all` – fill all fields.
- `pos` – fill the position field only.
- `name` – fill the name field only.
- `id` – fill the identifier field only.
- `0...99` – fill the fields using information from the point of the list.

Default: N/A

Usage Guidelines: Data and time will be written automatically upon completing a point-to-list addition.

Command Examples: Fill the position field.

```
set,/par/ref/avg/list/curpnt/fill,pos
```

Or

```
set,ref/avg/list/curpnt/fill,pos
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Parameters for working with the given point from the list

Name of the point

`[/par/]ref/avg/list/pnts/<N>/name`

Description:	Reports the name of the point.
Access:	read-only
Type:	string [0...19]
Values:	up to 19 alphanumeric characters
Default:	`"UNUSED"`
Usage Guidelines:	`N` ranges from 00 to 99.
Command Examples:	Query the name of point 01. `print,/par/ref/avg/list/pnts/01/name` Or `print,ref/avg/list/pnts/01/name`
Receiver Availability:	This parameter is available in all receivers.
History:	This parameter was introduced in firmware 3.2.
See Also:	N/A

L1PC coordinates of the point

`[/par/]ref/avg/list/pnts/<N>/pos/<xyz|geo>`

Description: Reports the L1PC coordinates of the point.

Access: read-only

Type: {datum,x,y,z}
{datum,lat,lon,alt}

Values: as defined in "Datum Parameters" on page 3-299

Default: `{W84,+6378137.0000,+0.0000,+0.0000}` for xyz
`{W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000}` for geo

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the geodetic coordinates of point 34.
```
print,/par/ref/avg/list/pnts/34/pos/geo
```
Or
```
print,ref/avg/list/pnts/34/pos/geo
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

ARP coordinates of the point

`[/par/]ref/avg/list/pnts/<N>/arp/<xyz|geo>`

Description: Reports the ARP coordinates of the point.

Access: read-only

Type: {datum,x,y,z}
{datum,lat,lon,alt}

Values: as defined in "Datum Parameters" on page 3-299

Default: `{W84,+6378137.0000,+0.0000,+0.0000}` for xyz
`{W84,N00d00m00.000000s,E000d00m00.000000s,+0.0000}` for geo

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the Cartesian coordinates of point 34.
```
print,/par/ref/avg/list/pnts/34/arp/xyz
```
Or
```
print,ref/avg/list/pnts/34/arp/xyz
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Date and time of the point

`[/par/]ref/avg/list/pnts/<N>/date`

Description: Reports the date and time of the point creation.

Access: read-only

Type: string

Values: date_spec time_spec

Default: `"88Y01M01D 00h00m00s"`

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the date&time of point 34.
```
print,/par/ref/avg/list/pnts/34/date
```
Or
```
print,ref/avg/list/pnts/34/date
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Index of the point

`[/par/]ref/avg/list/pnts/<N>/index`

Description: Reports the index of the point. It defines the point position in the list in accordance with the last sorting.

Access: read-only

Type: integer

Values: `[0...99]`

Default: `0`

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the index of point 34.
```
print,/par/ref/avg/list/pnts/34/index
```
Or
```
print,ref/avg/list/pnts/34/index
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

ID of the point

`[/par/]ref/avg/list/pnts/<N>/id`

Description: Reports identifier of the point. It is used as the reference station identifier.

Access: read-only

Type: integer

Values: `[0...4095]`

Default: `0`

Usage Guidelines:
- `N` ranges from 00 to 99.
- The range of values depends on the RTK format used.
  - –For RTCM 2.3, the receiver will use the values modulo 1024.
  - –For RTCM 3.0, the receiver will use the values modulo 4096.
  - –For CMR/CMR+, the receiver will use the values modulo 32.

Command Examples: Query ID of point 34.

```
print,/par/ref/avg/list/pnts/34/id
```

Or

```
print,ref/avg/list/pnts/34/id
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Point creation status

`[/par/]ref/avg/list/pnts/<N>/auto`

Description: Reports whether the point was created automatically.

Access: read-only

Type: boolean

Values: `n | y`

- `n` – the point was created automatically.
- `y` – the point was not created automatically.

Default: `n`

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the creation status of point 34.

```
print,/par/ref/avg/list/pnts/34/auto
```

Or

```
print,ref/avg/list/pnts/34/auto
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Point protection status

`[/par/]ref/avg/list/pnts/<N>/protect`

Description: Reports whether the point is protected from being overwritten.

Access: read-only

Type: boolean

Values: `n | y`

- `n` – the point is protected from being overwritten.
- `y` – the point is not protected from being overwritten.

Default: `n`

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Query the protection status of point 34.

```
print,/par/ref/avg/list/pnts/34/protect
```

Or

```
print,ref/avg/list/pnts/34/protect
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Distance between the current point and the given point

`[/par/]ref/avg/list/pnts/<N>/dist`

Description: Reports the distance between the current point and the given point.

Access: read-only

Type: float

Values: `[0.000...cur_value]`; expressed in meters

Default: `0.000`

Usage Guidelines:
- `N` ranges from 00 to 99.
- The coordinates of the current point are obtained from averaging.

Command Examples: Query the distance between the point 34 and the given point.

```
print,/par/ref/avg/list/pnts/34/dist
```

Or

```
print,ref/avg/list/pnts/34/dist
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Other parameters

Parameters of the points in the list

`[/par/]ref/avg/list/pnts/<N>`

Description: Reports the parameters of the points in the list.

Access: read-only

Type: list

Values: See the reply to the command.

Default: See the reply to the command.

Usage Guidelines: `N` can be one of the following: `used`, `auto`, `manual`, `all`.

Command Examples: Query the parameters of all points in the list.
`print,/par/ref/avg/list/pnts/all`
Or
`print,ref/avg/list/pnts/all`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Total numbers of the points with the given flag

`[/par/]ref/avg/list/pnts/count/<N>`

Description: Reports the total number of points with the given flag.

Access: read-only

Type: integer

Values: `[0...100]`

Default: `0`

Usage Guidelines: `N` can be one of the following: `used`, `auto`, `manual`, `free`.

Command Examples: Query the total number of auto points in the list.
`print,/par/ref/avg/list/pnts/count/auto`
Or
`print,ref/avg/list/pnts/count/auto`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Adding current point to the list

`[/par/]ref/avg/list/pnts/add`

Description: Adds the current point to the list.

Access: write-only

Type: integer

Values: `[-1...99`

- `-1` – index will be assigned automatically.
- `[0...99]` – index specified by the user.

Default: N/A

Usage Guidelines: N/A

Command Examples: Add the current point with index 22 to the list.
`set,/par/ref/avg/list/pnts/add,22`
Or
`set,ref/avg/list/pnts/add,22`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Deleting points from the list

`[/par/]ref/avg/list/pnts/del`

Description: Deletes the points from the list.

Access: write-only

Type: integer

Values: `[-1...99`

- `-1` – delete all points.
- `[0...99]` – delete a point with the specified index from the list.

Default: N/A

Usage Guidelines: N/A

Command Examples: Delete all points from the list.
`set,/par/ref/avg/list/pnts/del,-1`
Or
`set,ref/avg/list/pnts/del,-1`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Sorting points in the list

`[/par/]ref/avg/list/pnts/sort`

Description: Sorts the points in the list using the specified criterion.

Access: write-only

Type: enumerated

Values: `date | near | protect | name`

- `date` – sort the list by date.
- `near` – sort the list by proximity to the current point.
- `protect` – sort the list by the protection status.
- `name` – sort the list by name.

Default: N/A

Usage Guidelines: N/A

Command Examples: Sort the list by date.
`set,/par/ref/avg/list/pnts/sort,date`
Or
`set,ref/avg/list/pnts/sort,date`

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Making the point protected from being overwritten

`[/par/]ref/avg/list/pnts/<N>/option/protect`

Description: Sets the *protected* flag to the specified point in the list.

Access: write-only

Type: boolean

Values: `off | on`

- `off` – the point is not protected from being overwritten.
- `on` – the point is protected from being overwritten.

Default: N/A

Usage Guidelines: `N` ranges from 00 to 99.

Command Examples: Set the protected flag to point 34.

```
set,/par/ref/avg/list/pnts/34/option/protect,on
```

Or

```
set,ref/avg/list/pnts/34/option/protect,on
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in firmware 3.2.

See Also: N/A

Miscellaneous Parameters

Enter exception mode on errors

`[/par/]except`

Description: Specifies the receiver behaviour should an unrecoverable error occur.

Access: read & write

Type: boolean

Values: `off | on`

- `off` – if an unrecoverable error is detected, the receiver will enter exception mode.
- `on` – once an unrecoverable error is detected, the receiver will reset itself and then continue running. Here reset means a hardware reset similar to what the command `set,/par/reset,y` results in.

Default: `off`

Usage Guidelines: N/A

Command Examples: Do not enter to exception mode.

```
set,/par/except,on
```

Or

```
set,except,on
```

Query the parameter value.

```
print,/par/except
```

Or

```
print,except
```

Receiver Availability: This parameter is available in all receivers.

History: This parameter was introduced in the first release of firmware.

See Also: N/A

Receiver board temperature

`[/par/]dev/thermo/out`

Description:	Reports the receiver board's current temperature in degrees of Celsius
Access:	read-only
Type:	float
Values:	`cur_value`
Default:	`cur_value`
Usage Guidelines:	N/A
Command Examples:	Query the current temperature. `print,/par/dev/thermo/out` Or `print,dev/thermo/out`
Receiver Availability:	This parameter is unavailable in the GMS-2 receiver.
History:	This parameter was introduced in the first release of firmware.
See Also:	N/A

Configuring Messages

As described in "GRIL Concepts and Terms" on page 1-1, the receiver organizes all messages to groups and sets. This section describes the parameters and commands you can use to handle the messages in these categories.

You can apply the `list` and `print` commands to the message groups as follows:

```
print,/msg/jps
print,/msg/nmea
print,/msg/misc
print,/msg/jps/<NAME>
print,/msg/nmea/<NAME>
list,/msg/jps
list,/msg/nmea
list,/msg/jps/<NAME>
list,/msg/nmea/<NAME>
```

where `NAME` must be replaced with a specific message name.

Note that the `print` command enables the receiver to output the corresponding scheduling parameters for messages whereas the `list` command instructs the receiver to output only message names.

Example 1.

```
list,/msg/jps
RE03D{JP,MF,PM,EV,XA,XB,ZA,ZB,YA,YB,RT,RD,ST,LT,BP,TO,DO,OO,UO,GT,
RE040 NT,GO,NO,TT,PT,SI,NN,EL,AZ,SS,FC,RC,rc,PC,pc,CP,cp,DC,CC,cc,EC,
RE040 TC,R1,P1,1R,1P,r1,p1,1r,1p,D1,C1,c1,E1,F1,R2,P2,2R,2P,r2,p2,2r,
RE040 2p,D2,C2,c2,E2,F2,ID,PV,PO,PG,VE,VG,DP,SG,BI,SE,PS,GE,NE,GA,NA,
RE040 WE,WA,WO,GS,NS,rE,rM,rV,TM,MP,TR,MS,DL,TX,SP,SV,RP,RK,BL,AP,AB,
RE01F re,RM,RS,IO,JI,JE,NP,LH,EE,ET}
```

Example 2.

```
print,/msg/nmea:on
RE021/msg/nmea={GGA={1.00,0.00,0,0x0},
RE017 GLL={1.00,0.00,0,0x0},
RE017 GNS={1.00,0.00,0,0x0},
RE017 GRS={1.00,0.00,0,0x0},
RE017 GSA={1.00,0.00,0,0x0},
RE017 GST={1.00,0.00,0,0x0},
RE017 GSV={1.00,0.00,0,0x0},
RE017 HDT={1.00,0.00,0,0x0},
```

```
RE017 RMC={1.00,0.00,0,0x0},
RE017 ROT={1.00,0.00,0,0x0},
RE017 VTG={1.00,0.00,0,0x0},
RE017 ZDA={1.00,0.00,0,0x0},
RE019 P_ATT={1.00,0.00,0,0x0}}
```

Example 3.

```
print,/msg/jps/JP
RE011{1.00,0.00,0,0x0}
```

TPS receivers currently support the following three message sets:

- /msg/def – the default set of messages. It is used when starting data logging with the MINTER FN button or when running `em` and `out` commands with the second argument /msg/def/, e.g., `em,,/msg/def`.
- /msg/rtk/jps/min – the minimum adequate message set for an RTK system using slow modems. The set comprises the messages: RT, SI, rc, cp, 2r, 2p, BI, ET. For example, the command `em,/dev/ser/c,/msg/rtk/jps/min` allows your receiver to transmit necessary TPS RTK minimum correction set to port C.
- /msg/rtk/jps/max – the complete set of RTK messages. This set is preferable if the modem throughput is high enough. The set comprises the messages: RT, SI, rc, cp, DC, EC, 2r, 2p, D2, E2, BI, ET.

The contents of a message set can be found by using the commands `print` and `list` (in fact, this is similar to applying `print` and `list` to message groups). You will notice one small difference here. When querying a message set with `print` and `list`, you will be notified of not only the messages' individual names but also the names of the groups they are associated with, for example:

```
print,/par/out/cur/term
RE01C{nmea/GGA={1.00,0.00,0,0x0}}


list,/par/out/cur/term
RE00A{nmea/GGA}
```

Also, a query can be confined to messages from a particular group. In this case, the receiver will omit the group names reporting only the individual names, e.g.:

```
print,/par/out/cur/term/nmea
RE017{GGA={1.00,0.00,0,0x0}}
```

The commands `set`, `remove`, and `create` are currently applicable only to the message set `/msg/def`.

Examples.

`create,/msg/def/jps/GS` instructs the receiver, first, to add TPS message GS to the end of the default set of messages and, second, assign this message its default scheduling parameters.

`remove,/msg/def/jps/GA` instructs the receiver to remove TPS message GA from the default set of messages.

`set,/msg/def/jps/SI,{1,0,0,0}` instructs the receiver to change the scheduling parameters previously assigned to TPS message SI in the default set of messages.

Notes:

Messages

This chapter describes all of the messages supported by TPS receivers.

A message is the receiver output product that contains a set of data in a specific GRIL format. The receiver generates messages in response to the commands it receives. The messages are then conveyed to the specified destination for further manipulation. Destination could be a port or log file.

In This Chapter:

- "Message Specification" on page 4-3
- "Using Special Characters in Message Identifiers" on page 4-4
- "Time Scales Definition and Representation in Standard Messages" on page 4-4
- "Attention Users of Topcon Tools™ and Other TPS Post-processing Software" on page 4-5
- "Standard Binary Messages" on page 4-6
  - "Header Messages" on page 4-7
  - "Time Messages" on page 4-9
  - "Position and Velocity Messages" on page 4-21
  - "Satellite Data Messages" on page 4-36
  - "PPS and Event Marker Messages" on page 4-108
  - "Optimized Real-Time Application (ORTA) Messages" on page 4-111
  - "Miscellaneous Messages" on page 4-123
- "Standard Text Messages" on page 4-136
  - "General Messages" on page 4-138
  - "Command Response Messages" on page 4-175
- "Industry-approved Messages" on page 4-177
  - "NMEA sentences" on page 4-177
  - "RTCM 2.x Messages" on page 4-203
  - "RTCM 3.x Messages" on page 4-205

– “CMR Messages” on page 4-206
– “BINEX Records” on page 4-207

Message Specification

Each message is defined with its specifications as per the following categories:

- *Message Name* – provides the message ID as used in commands and the message title.
- *Description* – contains a concise explanation of what the message does.
- *Attributes* – contains properties of a message, including:
  - Message ID as it appears in a message stream.
  - The Length of Message Body (bytes) shows the decimal length of the message.

    For a fixed length message, *length* is a constant defining the message length. For a variable length message, *length* may be either an arithmetic expression depending on some other variables (e.g., nSats) or just the string *var*.

    The variable *nSats* denotes the actual number of satellites the given message involves.
  - Default Scheduling Specification.

    For messages included to the default message set, this attribute displays their default values of period, phase, count, and flags in this set. This specification is given for the receiver's default configuration. For more details about message sets, see "Message Sets" on page 1-16.

    For other messages, this attribute displays their default values of period, phase, count, and flags in a group the messages belong to. This specification is given for the receiver's default configuration. For more details about message groups, see "Message Groups" on page 1-16.

    This attribute will not show a user-defined scheduling specification.
- *Body* – provides the contents of a message.

  Some of the fields are shown in **`bold`** in the message body. This notation indicates that the field may contain a special value. For more details on the available special values, see "Special Values" on page 1-14.
- *Usage Guidelines* – provides information that ensures the proper use of a message. This might include detailed explanation of the message operation and result, prohibitions, and any special considerations.
- *Examples* – provides one or more examples demonstrating how the message in a stream.
- *History* – lists the firmware versions in which the message was introduced and modified, if relevant.
- *See Also* – directs you to other commands, parameters, or messages that might contain related information.

Some of the categories might have the N/A abbreviation standing for Not Applicable. It should be considered as a normal feature of the message and not a lack of functionality.

Using Special Characters in Message Identifiers

All of the predefined messages conform to the standard message format described in “Structure of Standard Messages” on page 1-14. You will notice that most of the messages have identifiers comprising only digits and/or English letters. In fact, the RT message with the ~~ identifier is the only message whose identifier uses the character "~". It makes sense as this message plays a very important part serving as an epoch delimiter. Thus there are special precautions in order to minimize the probability of losing this key message.

The idea of using highly distinctive identifiers for the messages that serve as delimiters is very clear. Should a message's checksum be wrong, just check its identifier. If neither of the identifier's characters coincides with "~", then it is very unlikely that this is a corrupted RT message. Therefore, you need not skip to the following RT message in this case.

On the other hand, if a message has the correct checksum yet one of the identifier characters is "~", then it would be safer to treat this message as a corrupted ~~ message. In this case – skip to the next RT message.

Similarly, the == identifier of the Event message, too, must be as distinctive as possible since the application software uses free-form events just as delimiters.

Time Scales Definition and Representation in Standard Messages

A TPS receiver may handle five time scales, specifically:

- T_g – GPS system time.
- T_u – UTC (USNO). Universal Coordinated Time supported by the U.S. Naval Observatory.
- T_n – GLONASS system time.
- T_s – UTC(SU). Universal Coordinated Time supported by the State Time and Frequency Service, Russia.
- T_r – Receiver time. This is the only time scale that is always available in your receiver (i.e., the other time scales listed above may or may not be currently available). In fact, TPS receiver always synchronizes its receiver time with one of the four global time scales: GPS time, UTC(USNO),

GLONASS time, or UTC(SU). The time scale thus selected is referred to as *receiver reference time* (T_{rr}).

NOTICE *Currently, TPS receivers can be synchronized to GPS system time only.*

Different time scales may have different time formats associated with them (e.g., GPS time uses such terms as *week number*, *time of week*, etc.). Note, however, that the receiver time representation will not depend on the selected receiver reference time.

Most of the binary messages do not contain reference time information inside. It would be excessive to use one and the same time tag with all of the many messages the receiver generates at the current epoch. When outputting receiver information available for the current epoch, you usually get various messages. Instead of supplying each of them with an individual time tag data field, the receiver uses a special message that carries receiver time information common for these messages. This message is called *Receiver Time*. In fact, Receiver Time is supposed to precede all of the other messages generated at the current epoch thus delimiting messages corresponding to different epochs. From a formal point of view, it is up to you to define the order of messages in the output stream. However, care should be taken to ensure that the order in which messages are written into the output stream does not break the epoch synchronization, which is very essential for post-processing the logged data with Topcon Tools™ and other TPS software packages.

There are some other messages having a time tag data field. One of them is called *Event*. Since an event may occur any time and the event time is absolutely independent of the epoch grid, this message type does require its own time tag.

Attention Users of Topcon Tools™ and Other TPS Post-processing Software

Some of the message types[5] are critical for Topcon Tools™ and other TPS post-processing software to be able to import and process tps files correctly.

Below is the list of the TPS messages that Topcon Tools™ will identify when parsing an imported tps `file:`

JP, MF, PM, EV, RD, RT, TO, UO, PO, VE, PV, DP, SI, NN, EL, RC, R1, R2, rc, r1, r2, 1R, 2R, 1r, 2r, CC, C1, C2, cc, c1, c2, PC, P1, P2, pc, p1, p2, CP, 1P, 2P, cp, 1p, 2p, DC, D2, TC, FC, F1, F2, EC, E1, E2, NE, GE, WE, GA, NA, IO, XA, XB, SS, SE, rM, rE, rV

In practice, however, most Topcon Tools™ users process raw data collected with TPS receivers that are configured to record only the default set of TPS messages, specifically:

5. And their place among other messages.

JP, MF, PM, EV, XA, XB, RT, RD, SI, NN, EL, FC, RC, DC, EC, TC, CP, 1R, 1P, 2R, 2P, E1, D2, E2, SS, SE, PV, ST, DP, TO, DO, UO, IO, GE, NE, GA, NA, WE, WA, WO

Great care should be exercised with respect to the *time messages* such as *RD* and *RT*. If a set of messages other than the default one is to be used, or if the adopted order of messages in the default set of messages is changed in some way, make sure that the updated set of messages maintains the *epoch synchronization*, i.e., the messages *RT* and *RD* precede the messages containing code, carrier phase, and other types of measurements collected at the current epoch. Should the user violate this condition, the software may not be able to correctly process corresponding raw data files.

Also note that Topcon Tools™ will not import a TPS file without the *MF* message. For more information about the *MF* message, refer to “MF – Messages format and identifier” on page 4-8.

Standard Binary Messages

This section describes all standard binary messages available in a TPS receiver.

Solution Types Definition used in Standard Binary Messages

Table 4-1 defines the solution types used in the `solType` field.

Table 4-1. Solution Types in Standard Binary Messages

Solution Type	Meaning
0	Invalid data
1	Standalone solution
2	Code differential (DGPS) solution
3	Phase differential (RTK) solution with float ambiguities
4	Phase differential (RTK) solution with fixed ambiguities
5	Entered value (i.e., not calculated by the receiver)

Header Messages

JP – Topcon Positioning Systems file identifier

Description

This message is intended as a file identifier for a whole variety of files created by using different TPS software. This message, which is always put at the beginning of the TPS file, serves two purposes.
First, it enables the processing program to easily identify the file type.
Second, this message usually contains some additional information about the origin of the corresponding file (e.g., what particular hardware was used to collect data this file contains).

Parameters

Message ID	JP
Length of the Message Body (bytes)	85
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x1c

Body

```
struct JPFileId {
  a1 id[5];           // File type identifier
  a1 description[80]; // Human-readable file description
};
```

Usage Guidelines

- The field `id` indicates the file type.
- The field `description` is padded to the required size with blanks if necessary.
- For TPS receivers, the JP message always contains the following information:
  - `id` = `RLOGF`,
  - `description` = `JPS <NAME> Receiver Log File` (blanks are omitted here), where `<NAME>` stands for the specific receiver name, e.g. `JPS GMS-2 Receiver Log File`.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

MF – Messages format and identifier

Description

This message provides information on the format of the messages and the byte order in which multi-byte binary types are stored inside the message bodies.

Parameters

Message ID	MF
Length of the Message Body (bytes)	9
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	60.0
count	0
flags	0x37

Body

```
struct {

  a1 id[2]='JP';  // JP identifier

  a1 majorVer[2]; // Format major version as decimal (e.g. 01)

  a1 minorVer[2]; // Format minor version as decimal (e.g. 00)

  a1 order;       // Bytes order:
                  //    0 - LSB first
                  //    1 - MSB first

  a1 crc[2];      // Checksum as hexadecimal

};
```

Usage Guidelines

- The size of this message is not subject to change. Therefore, the first 7 bytes of this message are always MF009JP.
- The message format's major version is updated if and only if some backward incompatible changes to the existing message format are made. Any other changes to the existing messages result in updating only the minor version.
- The data field `order` describes how multi-byte binary types are stored inside the message bodies. For message format version 1.0, order is always set to 0 (i.e., you cannot force the receiver to write binary data in the most significant bytes first order).
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

Time Messages

RD – Receiver date

Description

This message contains the date part of the full receiver time representation (T_r).

Parameters

Message ID	RD
Length of the Message Body (bytes)	6
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	120.0
count	0
flags	0x23

Body

```
struct RcvDate {

  u2 year;  // Current year [1...65534]

  u1 month; // Current month [1...12]

  u1 day;   // Current day [1...31]

  u1 base;  // Receiver reference time
            //       0 – GPS
            //       1 – UTC_USNO
            //       2 – GLONASS
            //       3 – UTC_SU
            // 4...254 – Reserved

  u1 cs;    // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

RT – Receiver time

Description

This message contains the time of day part of the full receiver time representation (T_r).

Parameters

Message ID	~~
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvTime {

  u4 tod; // T_r modulo 1 day (86400000 ms) [ms]

  u1 cs;  // Checksum

};
```

Usage Guidelines

- To enable/disable this message, use the message name `/msg/jps/RT` (not `/msg/jps/~~`).
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

ET – Epoch time

Description

This message serves as an end of epoch indicator and contains the time of the day stamp.

Parameters

Message ID	::
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct EpochTime {

  u4 tod; // T_r modulo 1 day (86400000 ms) [ms]

  u1 cs;  // Checksum

};
```

Usage Guidelines

To enable/disable this message, use the message name `/msg/jps/ET` (not `/msg/jps/::`).

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

UO – Universal Coordinated Time (UTC) parameters

Description

This message describes the relationship between UTC(USNO) and GPS time as specified by GPS subframe 4, page 18.

Parameters

Message ID	UO
Length of the Message Body (bytes)	24
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	3600.0
count	0
flags	0x23

Body

```
struct UTCtoGPSTimeOffset {
  f8 a0;    // Constant term of polynomial [s]
  f4 a1;    // First order term of polynomial [s/s]
  u4 tot;   // Reference time of week [s]
  u2 wnt;   // Reference week number []
  i1 dtls;  // Delta time due to leap seconds [s]
  u1 dn;    // Future reference day number [1...7] []
  u2 wnlsf; // Future reference week number []
  i1 dtlsf; // Future delta time due to leap seconds [s]
  u1 cs;    // Checksum
};
```

Usage Guidelines

- For how to convert GPS time into UTC(USNO), see ICD-GPS-200C, Revision IRN-200C-004 April 12, 2000.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Examples” on page 1-17 for more information about the default message set.

GT – GPS time

Description

This message contains the current GPS time of week and the week number.

Parameters

Message ID	GT
Length of the Message Body (bytes)	7
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct GPSTime {
  u4 tow; // Time of week [ms]
  u2 wn;  // GPS week number, modulo 1024 []
  u1 cs;  // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

NT – GLONASS time

Description

This message contains the current GLONASS time of day and the day number.

Parameters

Message ID	NT
Length of the Message Body (bytes)	7
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct GLOTime {
  u4 tod; // Time of day [ms]
  u2 dn;  // GLONASS day number, modulo 4 years starting from 1996 []
  u1 cs;  // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

WO – WAAS time

Description

This message describes the relationship between UTC and WAAS network time. This message has much in common with the UO message.

Parameters

Message ID	WO
Length of the Message Body (bytes)	31
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct WAASTime {

  f8 a0wnt; // Constant term of polynomial [s]

  f4 a1wnt; // First-order term of polynomial [s/s]

  u4 tot;   // Reference time of week [s]

  u2 wnt;   // Reference week number []

  i1 dtls;  // Delta time due to leap seconds [s]

  u1 dn;    // Future reference day number [1...7] []

  u2 wnlsf; // Future reference week number []

  i1 dtlsf; // Future delta time due to leap seconds [s]

  i1 utcsi; // UTC Standard Identifier:
            //      0 - UTC as operated by the Communications Research
            //          Laboratory (CRL), Tokyo, Japan
            //      1 - UTC as operated by the National Institute of
            //          Standards and Technology (NIST)
            //      2 - UTC as operated by the U. S. Naval Observatory (USNO)
            //      3 - UTC as operated by the International Bureau of
            //          Weights and Measures (BIPM)
            //  4...7 - Reserved

  u4 tow;   // Time of week [s]

  u2 wn;    // Week number []

  u1 cs;    // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Message Sets" on page 1-16 for more information about the default message set.

TO – Receiver reference time to receiver time offset

Description

This message contains the difference between receiver reference time (T_{rr}) and receiver time (T_r).

Parameters

Message ID	TO
Length of the Message Body (bytes)	9
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvTimeOffset {
  f8 val; // Trr - Tr [s]
  u1 cs;  // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Message Sets" on page 1-16 for more information about the default message set.

DO – Derivative of receiver time offset

Description

This message contains time derivative of the difference between receiver reference time (T_{rr}) and receiver time (T_r).

Parameters

Message ID	DO
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvTimeOffsetDot {

  f4 val; // Time Derivative of (Trr - Tr) [s/s]

  u1 cs;  // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

GO – GPS time to receiver time offset

Description

This message contains the difference between GPS system time (T_g) and receiver time (T_r).

Parameters

Message ID	GO
Length of the Message Body (bytes)	6
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvGPSimeOffset {
  i1 sec; // (Tg - Tr) [s]
  i4 ns;  // (Tg - Tr), modulo 1 second [ns]
  u1 cs;  // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

NO – GLONASS time to receiver time offset

Description

This message contains the difference between GLONASS system time (T_n) and receiver time (T_r).

Parameters

Message ID	NO
Length of the Message Body (bytes)	6
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvGLOTimeOffset {

  i1 sec; // (T_n - T_r) [s]

  i4 ns;  // (T_n - T_r), modulo 1 second [ns]

  u1 cs;  // Checksum

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

BP – Rough indicator of receiver synchronization to reference time

Description

This message contains an indication of the receiver clock's synchronization to the receiver reference time (T_{rr}).

Parameters

Message ID	BP
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvTimeAccuracy {

  f4 acc; // If this value >> 1 ms, it means that position is not computed
          // yet and therefore receiver clock may not be properly
          // synchronized to T_rr [s]

  u1 cs;  // Checksum

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

Position and Velocity Messages

ST – Solution time-tag

Description

This message specifies the receiver time of the current position solution.

Parameters

Message ID	ST
Length of the Message Body (bytes)	6
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SolutionTime {
  u4 time;    // Solution time. T_r modulo 1 day (86400000 ms) [ms]
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

- The time-tag may differ from the current receiver time if the receiver runs in RTK delay mode. In this case the time tag from this message is typically in the past with respect to the time tag of the current epoch.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

PO – Cartesian position

Description

This message contains the current position of the receiver's antenna in Cartesian coordinates and SEP[a] values.

Parameters

Message ID	PO
Length of the Message Body (bytes)	30
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Pos {
  f8 x, y, z; // Cartesian coordinates [m]
  f4 sigma;   // Position SEP [m]
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

This message always contains coordinates computed in WGS 84.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

a. SEP = Spherical Error Probable

VE – Cartesian velocity

Description

This message contains the current velocity vector of the receiver's antenna in Cartesian coordinates and SEP values.

Parameters

Message ID	VE
Length of the Message Body (bytes)	18
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Vel {
  f4 x, y, z; // Cartesian velocity vector [m/s]
  f4 sigma;   // Velocity SEP [m/s]
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

PV – Cartesian position and velocity

Description

This message contains the position and velocity of the receiver's antenna in Cartesian coordinates and SEP values.

Parameters

Message ID	PV
Length of the Message Body (bytes)	46
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PosVel {
  f8 x, y, z;     // Cartesian coordinates [m]
  f4 pSigma;      // Position SEP [m]
  f4 vx, vy, vz; // Cartesian velocities [m/s]
  f4 vSigma;      // Velocity SEP [m/s]
  u1 solType;     // Solution type
  u1 cs;          // Checksum
};
```

Usage Guidelines

- This message always contains coordinates computed in WGS 84.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

PG – Geodetic position

Description

This message contains the position of the receiver's antenna in geodetic coordinates and SEP values.

Parameters

Message ID	PG
Length of the Message Body (bytes)	30
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct GeoPos {
  f8 lat;     // Latitude [rad]
  f8 lon;     // Longitude [rad]
  f8 alt;     // Ellipsoidal height [m]
  f4 pSigma;  // Position SEP [m]
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

This message always contains coordinates computed in WGS 84.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

VG – NEU velocity

Description

This message contains the velocity of the receiver's antenna in the local horizon NEU coordinate system and SEP values.

Parameters

Message ID	VG
Length of the Message Body (bytes)	18
Default Scheduling Parameters	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct GeoVel {
  f4 lat;     // Northing velocity [m/s]
  f4 lon;     // Easting velocity [m/s]
  f4 alt;     // Height velocity [m/s]
  f4 pSigma;  // Velocity SEP [m/s]
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

SG – Position and velocity RMS errors

Description

This message contains RMS errors of the horizontal and vertical position and velocity of the receiver's antenna.

Parameters

Message ID	SG
Length of the Message Body (bytes)	18
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Rms {
  f4 hpos;     // Horizontal position RMS error [m]
  f4 vpos;     // Vertical position RMS error [m]
  f4 hvel;     // Horizontal velocity RMS error [m/s]
  f4 vvel;     // Vertical velocity RMS error [m/s]
  u1 solType; // Solution type
  u1 cs;       // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

SP – Position covariance matrix

Description

This message contains the entries of covariance matrix for position of the receiver's antenna.

Parameters

Message ID	SP
Length of the Message Body (bytes)	42
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PosCov {
  f4 xx;       // [m^2]
  f4 yy;       // [m^2]
  f4 zz;       // [m^2]
  f4 tt;       // [m^2]
  f4 xy;       // [m^2]
  f4 xz;       // [m^2]
  f4 xt;       // [m^2]
  f4 yz;       // [m^2]
  f4 yt;       // [m^2]
  f4 zt;       // [m^2]
  u1 solType; // Solution type
  u1 cs;       // Checksum
};
```

Usage Guidelines

The values of position covariance matrix are available only in RTK mode.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

SV – Velocity covariance matrix

Description

This message contains the entries of covariance matrix for velocity of the receiver's antenna.

Parameters

Message ID	SV
Length of the Message Body (bytes)	42
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct VelCov {

  f4 xx;       // [(m/s)^2]

  f4 yy;       // [(m/s)^2]

  f4 zz;       // [(m/s)^2]

  f4 tt;       // [(m/s)^2]

  f4 xy;       // [(m/s)^2]

  f4 xz;       // [(m/s)^2]

  f4 xt;       // [(m/s)^2]

  f4 yz;       // [(m/s)^2]

  f4 yt;       // [(m/s)^2]

  f4 zt;       // [(m/s)^2]

  u1 solType; // Solution type

  u1 cs;       // Checksum

};
```

Usage Guidelines

The values of velocity covariance matrix are available only in RTK mode.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

DP – Dilution of Precision (DOP)

Description

This message contains the dilution of precision parameters that comprise position and geometric dilutions of precision.

Parameters

Message ID	DP
Length of the Message Body (bytes)	14
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Dops {
  f4 hdop;    // Horizontal dilution of precision (HDOP) []
  f4 vdop;    // Vertical dilution of precision (VDOP) []
  f4 tdop;    // Time dilution of precision (TDOP) []
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

- Geometric dilution of precision GDOP = sqrt ($hdop^2 + vdop^2 + tdop^2$)
- Position dilution of precision PDOP = sqrt($hdop^2 + vdop^2$)
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

BL – Base line

Description

This message contains information on the baseline vector estimated by the RTK engine.

Parameters

Message ID	BL
Length of the Message Body (bytes)	34
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct BaseLine {
  f8 x, y, z; // Calculated baseline vector coordinates [m]
  f4 sigma;   // Baseline SEP [m]
  u1 solType; // Solution type
  i4 time;    // Receiver time of the baseline estimate [s]
  u1 cs;      // Checksum
};
```

Usage Guidelines

This message outputs in RTK mode only.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

BI – Base station information

Description	Parameters	
This message contains reference station information for RTK.	Message ID	BI
	Length of the Message Body (bytes)	28
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

```
struct BaseInfo {
  f8 x, y, z; // ECEF coordinates [m]
  u2 id;      // Reference station ID
  u1 solType; // Solution type
  u1 cs;      // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

ha – Heading and pitch

Description

This message contains information on the baseline vector estimated by the RTK engine.

Parameters

Message ID	ha
Length of the Message Body (bytes)	9
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct HeadP {

  f4 heading; // Heading of the baseline between the reference and rover
              // receivers [radians]

  f4 pitch;   // Pitch of the baseline between the reference and rover
              // receivers [radians]

  u1 cs;      // Checksum

};
```

Usage Guidelines

This message outputs in RTK mode only.

Examples

N/A

History

This message was introduced in firmware 2.6.

See Also

N/A

PS – Position statistics

Description

This message contains some statistics for RTK positioning.

Parameters

Message ID	PS
Length of the Message Body (bytes)	9
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PosStat {

  u1 solType;   // Solution type

  u1 gpsLocked; // Number of GPS SVs locked

  u1 gloLocked; // Number of GLONASS SVs locked

  u1 gpsAvail;  // Number of GPS SVs available for positioning

  u1 gloAvail;  // Number of GLONASS SVs available for positioning

  u1 gpsUsed;   // Number of GPS SVs used in positioning

  u1 gloUsed;   // Number of GLONASS SVs used in positioning

  u1 fixProg;   // Ambiguity fixing progress indicator controllable by
                // RTK engine [%]

  u1 cs;        // Checksum

};
```

Usage Guidelines

The `fixProg` field may vary between 0% and 100%.

- `0` – raw measurements are good enough and all available ambiguities are fixed. In practice, this field rarely takes values other than zero.
- `1...99` – not all of the ambiguities have been fixed. This is the case in the following situations:
  - – you have just launched the RTK engine and it is trying to get a first fixed solution.
  - – there is one or more problem satellites whose measurements prevent the engine from fixing all available ambiguities in batch.
  - – the receiver has just started tracking one or more new satellites. It will take the RTK engine some time to fix these new ambiguities.
- `100` – the RTK engine has just finished fixing all available ambiguities. This indicator will be dropped to zero immediately after it has reached 100%.

If the solution type is RTK fixed, the number of SVs with float ambiguities is
gpsAvail + gloAvail - (gpsUsed + gloUsed)

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

PT – Time of continuous position computation

Description

This message specifies the time interval over which continuous position computation has been possible.

Parameters

Message ID	PT
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PositionTime {
  u4 pt; // Position time [s]
  u1 cs; // Checksum
};
```

Usage Guidelines

If the receiver is unable to compute any position at the current epoch, the `pt` field resets to zero.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

Satellite Data Messages

This section focuses on messages containing satellite specific information. These kinds of messages include satellite measurements (code and carrier phase observables, elevations, azimuths, etc.) and navigation data (almanacs and ephemerides).

Different applications may utilize different sets of measurements. It is almost impossible to select a fixed set of combinations of satellite measurements that would be enough universal yet compact. Every individual measurement message provides some specific (homogeneous) data for the satellites tracked.

For any given epoch, data for particular satellite is put at the same position (index) in all the messages. To make it possible to handle data corresponding to satellites of different systems in a universal manner, each satellite is assigned its Universal Satellite Identifier (USI). The SI message establishes the mapping between Universal Satellite Identifier (USI) and the index for this particular satellite. The number of satellites an epoch has data for, `nSats`, should also be obtained from the SI message. This approach allows the minimization of the data transfer overheads (note that the gain increases as the number of satellites grows).

The following table describes USI allocation:

Table 4-2. Universal Satellite Identifier (USI) Allocation

USI Values	Assigned Satellites
0	Unused.
1...37	GPS SV PRNs.
38...70	USI - 45 (signed) = GLONASS SV FCNs.[a]
71...119	Reserved.
120...138	SBAS PRNs.
139...254	Reserved.
255	Unused.

a. USI = 70 (FCN = 25) designates a GLONASS SV whose FCN is unknown. Recommended for use when converting third-party GNSS measurement files into TPS format.

SI – Satellite indices

Description

This message contains an array of USIs indicating the currently tracked satellites whose elevation angles are equal or above the value specified by the parameter `/par/out/elm/<stream>`.

Parameters

Message ID	SI
Length of the Message Body (bytes)	nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x23

Body

```
struct SatIndex {

  u1 usi[nSats]; // USI array []

  u1 cs;         // Checksum

};
```

Usage Guidelines

- The SI message serves as a reference map for other measurement message types since it establishes the one-to-one correspondence between the satellite identifier and the array index allocated to this satellite. Bear in mind that this correspondence holds true until the next SI message is issued. Note that the new SI message may or may not differ from the previous one.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

- “Message Sets” on page 1-16 for more information about the default message set.
- The parameter `/par/out/elm/<stream>` on page 3-211.

NN – GLONASS satellite system numbers

Description

This message shows the orbit slot numbers the orbit slot number for every GLONASS satellite from the latest SI message.

Parameters

Message ID	NN
Length of the Message Body (bytes)	nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x23

Body

```
struct SatNumbers {
  u1 number[nSats]; // GLONASS SV orbit slot number []
  u1 cs;            // Checksum
};
```

Usage Guidelines

- In this message `nSats` designates the number of GLONASS satellites as specified in the corresponding SI message. Recall that GLONASS satellites have universal identifiers lying within the range [38…70].
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

EL – Satellite elevations

Description

This message contains elevations for all the satellites specified in the latest SI message.

Parameters

Message ID	EL
Length of the Message Body (bytes)	nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SatElevation {
  i1 elev[nSats]; // Elevation angle ranging between -90 and 90 [degrees]
  u1 cs;          // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

AZ – Satellite azimuths

Description

This message contains azimuths for all the satellites specified in the latest SI message.

Parameters

Message ID	AZ
Length of the Message Body (bytes)	nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SatAzimuth {
  u1 azim[nSats]; // Azimuth angle ranging between 0 and 180 [degrees*2]
  u1 cs;          // Checksum
};
```

Usage Guidelines

The notation `[degrees*2]` means that the corresponding values must be multiplied by two to restore actual azimuths in degrees.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

RC – Full L1 C/A pseudoranges

Description

Contains full L1 C/A pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	RC
Length of the Message Body (bytes)	8*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PR_L1CA {
  f8 prL1CA[nSats]; // L1 C/A pseudoranges [s]
  u1 cs;            // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

rc – Delta L1 C/A pseudoranges

Description

Contains delta L1 C/A pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	rc
Length of the Message Body (bytes)	4*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct DPR_L1CA {

  i4 dprL1CA[nSats]; // (L1 C/A pseudoranges [s] - 0.075 ) * 10^11

  u1 cs;             // Checksum

};
```

Usage Guidelines

Use the following formula to restore full C/A pseudoranges on L1 in seconds:

$$\textit{Full L1 C/A pseudoranges} = \textit{Delta L1 C/A pseudoranges} * 10^{-11} + 0.075$$

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

CC – L1 C/A pseudorange smoothing corrections

Description

Contains L1 C/A pseudorange smoothing corrections and corresponding smoothing intervals for all the satellites specified in the latest SI message.

Parameters

Message ID	CC
Length of the Message Body (bytes)	6*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct S_L1CA {
  Smooth smooth[nSats]; // L1 C/A pseudorange smoothing data
  u1 cs;                // Checksum
};
```

Usage Guidelines

The structure `Smooth` is defined as follows:

```
struct Smooth {
  f4 value;     // Smoothing correction [s]
  u2 interval;  // Smoothing interval [s]
};
```

To obtain smoothed C/A pseudoranges in seconds, just add the smoothing corrections to the corresponding raw pseudoranges:

Smoothed L1 C/A pseudoranges = Full L1 C/A pseudoranges from RC + smoothing corrections

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

cc – Short L1 C/A pseudorange smoothing corrections

Description

Contains short L1 C/A pseudorange smoothing corrections for all the satellites specified in the latest SI message.

Parameters

Message ID	cc
Length of the Message Body (bytes)	2*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SS_L1CA {

  i2 SSmooth[nSats]; // Smoothing correction [s*10^-11]

  u1 cs;             // Checksum

};
```

Usage Guidelines

Use the following formula to compute smoothed C/A pseudoranges in seconds:

*Smoothed L1 C/A pseudoranges = Full L1 C/A pseudoranges from RC + smoothing corrections * 10^{-11}*

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

R1 – Full L1 P pseudoranges

Description

Contains full L1 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	R1
Length of the Message Body (bytes)	8*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PR_L1P {
  f8 prL1P[nSats]; // L1 P pseudoranges [s]
  u1 cs;           // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

r1 – Delta L1 P pseudoranges

Description

Contains delta L1 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	r1
Length of the Message Body (bytes)	4*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct DPR_L1P {

  i4 dprL1P[nSats]; // (L1 P pseudoranges [s] - 0.075) * 10^11

  u1 cs;            // Checksum

};
```

Usage Guidelines

Use the following formula to restore full L1 P pseudoranges in seconds:

$$Full\ L1\ P\ pseudoranges = Delta\ L1\ P\ pseudoranges * 10^{-11} + 0.075$$

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

1R – Relative L1 P pseudoranges

Description

Contains relative L1 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	1R
Length of the Message Body (bytes)	4*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RPR_L1P {

  f4 rprL1P[nSats]; // Full L1 P pseudorange from R1 - Full L1 C/A pseudorange
                    // from RC, [s]

  u1 cs;            // Checksum

};
```

Usage Guidelines

- Use the following formula to restore full L1 P pseudorange in seconds:

  Full L1 P pseudorange = Full L1 C/A pseudorange from RC + Relative L1 P pseudorange from 1R

- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

1r – Relative delta L1 P pseudoranges

Description

This message contains relative delta P/L1 pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	1r
Length of the Message Body (bytes)	2*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RDPR_L1P {
  i2 rdprL1P[nSats]; // (Full L1 P pseudorange from R1 [s] -
                     // - Full L1 C/A pseudorange from RC [s] - 2^-7) * 10^11
  u1 cs;             // Checksum
};
```

Usage Guidelines

- Use the following formula to restore full L1 P pseudorange in seconds:
  *Full L1 P pseudorange = Full L1 C/A pseudorange from RC + Relative delta L1 P pseudorange from 1r * * 10^{-11} + 2 * 10^{-7}*
- This message is the most compact L1 P pseudorange format.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

C1 – L1 P pseudorange smoothing corrections

Description

Contains L1 P pseudorange smoothing corrections and corresponding smoothing intervals for all the satellites specified in the latest SI message.

Parameters

Message ID	C1
Length of the Message Body (bytes)	6*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct S_L1P {
  Smooth smooth[nSats]; // L1 P pseudorange smoothing data
  u1 cs;                // Checksum
};
```

Usage Guidelines

The structure `Smooth` is defined as follows:

```
struct Smooth {
  f4 value;     // Smoothing correction [s]
  u2 interval;  // Smoothing interval [s]
};
```

To obtain smoothed L1 P pseudoranges in seconds, just add the smoothing corrections to the corresponding raw pseudoranges:

Smoothed L1 P pseudoranges = Full L1 P pseudoranges from R1 + smoothing corrections

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

c1 – Short L1 P pseudorange smoothing corrections

Description

Contains short L1 P smoothing corrections for all the satellites specified in the latest SI message.

Parameters

Message ID	c1
Length of the Message Body (bytes)	2*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SS_L1P {

  i2 SSmooth[nSats]; // Smoothing correction [s*10^-11]

  u1 cs;             // Checksum

};
```

Usage Guidelines

Use the following formula to compute smoothed L1 P pseudoranges in seconds:

*Smoothed L1 P pseudoranges = Full L1 P pseudoranges from R1 + smoothing corrections * 10^{-11}*

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

R2 – Full L2 P pseudoranges

Description

Contains full L2 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	R2
Length of the Message Body (bytes)	8*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PR_L2P {
  f8 prL2P[nSats]; // L2 P pseudoranges [s]
  u1 cs;           // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

r2 – Delta L2 P pseudoranges

Description

Contains delta L2 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	r2
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct DPR_L2P {

  i4 dprL2P[nSats]; // (L2 P pseudoranges [s] - 0.075) * 10^11

  u1 cs;            // Checksum

};
```

Usage Guidelines

Use the following formula to restore full L2 P pseudoranges in seconds:

$$Full\ L2\ P\ pseudoranges = Delta\ L2\ P\ pseudoranges * 10^{-11} + 0.075$$

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

2R – Relative L2 P pseudoranges

Description

Contains relative L2 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	2R
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RPR_L2P {

  f4 rprL2P[nSats]; // Full L2 P pseudorange from R2 - Full L1 C/A
                    // pseudorange // from RC, [s]

  u1 cs;            // Checksum

};
```

Usage Guidelines

- Use the following formula to restore full L2 P pseudorange in seconds:

  Full L2 P pseudorange = Full L1 C/A pseudorange from RC + Relative L2 P pseudorange from 2R

- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

2r – Relative delta L2 P pseudoranges

Description

This message is based on the most compact P/L2 pseudorange format.
This message contains relative delta L2 P pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	2r
Length of the Message Body (bytes)	2*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RDPR_L2P {
  i2 rdprL2P[nSats]; // (Full L2 P pseudorange from R2 [s] -
                     // - Full L1 C/A pseudorange from RC [s] - 2^-7) *
                     // * 10^11
  u1 cs;             // Checksum
};
```

Usage Guidelines

- Use the following formula to restore full L2 P pseudorange in seconds:
  *Full L2 P pseudorange = Full L1 C/A pseudorange from RC + Relative delta L2 P pseudorange from 2r * * 10^{-11} + 2 * 10^{-7}*
- This message is the most compact L2 P pseudorange format.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

C2 – L2 P pseudorange smoothing corrections

Description

Contains L2 P pseudorange smoothing corrections and corresponding smoothing intervals for all the satellites specified in the latest SI message.

Parameters

Message ID	C2
Length of the Message Body (bytes)	6*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct S_L2P {
 Smooth smooth[nSats]; // L2 P smoothing correction data
  u1 cs;               // Checksum
};
```

Usage Guidelines

The structure `Smooth` is defined as follows:

```
struct Smooth {
  f4 value;     // Smoothing correction [s]
  u2 interval;  // Smoothing interval [s]
};
```

To obtain smoothed L2 P pseudoranges in seconds, just add the smoothing corrections to the corresponding raw pseudoranges:

Smoothed L2 P pseudoranges = Full L2 P pseudoranges from R2 + smoothing corrections

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

c2 – Short L2 P pseudorange smoothing corrections

Description

Contains short L2 P smoothing corrections for all the satellites specified in the latest SI message.

Parameters

Message ID	c2
Length of the Message Body (bytes)	2*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SS_L2P {

  i2 SS_L2P[nSats]; // Smooth correction [s*10^-11]

  u1 cs;            // Checksum

};
```

Usage Guidelines

Use the following formula to compute smoothed L2 P pseudoranges in seconds:

*Smoothed L2 P pseudoranges = Full L2 P pseudoranges from R2 + smoothing corrections * 10^{-11}*

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

R3 – Full GPS L2C and GLONASS L2 C/A pseudoranges

Description

Contains full GPS L2C and GLONASS L2 C/A pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	R3
Length of the Message Body (bytes)	8*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PR_L2C {
  f8 prL2C[nSats]; // GPS L2C and GLONASS L2 C/A pseudoranges [s]
  u1 cs;           // Checksum
};
```

Usage Guidelines

This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

3R – Relative GPS L2C / GLONASS L2 C/A pseudoranges

Description

Contains relative GPS L2C / GLONASS L2 C/A pseudoranges for all the satellites specified in the latest SI message.

Parameters

Message ID	3R
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RPR_L2C {

  f4 rprL2C[nSats]; // Full GPS L2C/GLONASS L2 C/A pseudorange from R3 -
                    // - Full L1 C/A pseudorange from RC, [s]

  u1 cs;            // Checksum

};
```

Usage Guidelines

- Use the following formula to restore full L2 P pseudorange in seconds:

  Full L2C pseudorange = Full L1 C/A pseudorange from RC + Relative L2C pseudorange from 3R

- This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

PC – Full L1 C/A carrier phases

Description

Contains the full C/A carrier phases for all the satellites specified in the latest SI message.

Parameters

Message ID	PC
Length of the Message Body (bytes)	8*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct CP_L1CA {
  f8 cpL1CA[nSats]; // L1 C/A carrier phasea [cycles]
  u1 cs;            // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

pc – Short L1 C/A carrier phases

Description

Contains short L1 C/A carrier phases for all the satellites specified in the latest SI message.

Parameters

Message ID	pc
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SCP_L1CA {

  u4 scpL1CA[nSats]; // Short L1 C/A carrier phases [cycles/1024]

  u1 cs;             // Checksum

};
```

Usage Guidelines

- Use the following formula to compute full L1 C/A carrier phases in cycles:

  Full L1 C/A carrier phase = Short L1 C/A carrier phase / 1024

- Short carrier phases from this message will have discontinuities due to periodic rollovers used to represent this kind of carrier phases. This is due to the less memory-consuming carrier phase representation. Such rollovers do not occur very frequently (approximately once every 15 minutes or more). If you want to remove this kind of discontinuity from your carrier phases, you can use the following simple correction technique in the C programming language:

```
static unsigned long prev = value_from_pc;
static long rollOvers=0;
unsigned long curr = value_from_pc;
if ( curr < (1<<30) && prev > (1<<30)*3 ) rollOvers++;
if ( curr > (1<<30)*3 && prev < (1<<30) ) rollOvers--;
prev=curr;
double phase=((1<<16)*double(rollOvers)*(1<<16) + curr) / 1024.; // cycles
```

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

CP – L1 C/A carrier phases computed relative to pseudoranges from the RC message

Description

Contains the differences between the full L1 C/A carrier phases and the matching pseudoranges from the RC message for all the satellites specified in the latest SI message.

Parameters

Message ID	CP
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L1CA_RC {

  f4 rcpL1CARC[nSats]; // Full L1 C/A carrier phase from PC / L1 frequency -
                       // - full L1 C/A pseudorange from RC, [s]

  u1 cs;               // Checksum

};
```

Usage Guidelines

Use the following formula to compute full L1 C/A carrier phases in cycles:

*Full L1 C/A carrier phase = (L1 C/A carrier phase from CP + L1 C/A pseudorange from RC) * L1_freq*,

where L1_freq is the nominal L1 carrier frequency in Hz for the corresponding satellite

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

cp – L1 C/A carrier phases computed relative to pseudoranges from the rc message

Description

Contains the differences between the full L1 C/A carrier phases and the matching pseudoranges from the rc message for all the satellites specified in the latest SI message.

Parameters

Message ID	cp
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L1CA_rc {

  i4 rcpL1CArc[nSats]; // Full L1 C/A carrier phase from PC / L1 frequency -
                       // - delta L1 C/A pseudorange from rc, [s*2^-40]

  u1 cs;               // Checksum

};
```

Usage Guidelines

Use the following formula to compute full L1 C/A carrier phases in cycles:

*Full L1 C/A carrier phase = (L1 C/A carrier phase from cp * 2^{-40} + delta L1 C/A pseudorange from rc) * L1_freq*,
where L1_freq is the nominal L1 carrier frequency in Hz for the corresponding satellite

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

P1 – Full L1 P carrier phases

Description	Parameters	
Contains the L1 P carrier phases for all the satellites specified in the latest SI message.	Message ID	P1
	Length of the Message Body (bytes)	8*nSats + 1
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

```
struct CP_L1P {
  f8 cpL1P[nSats]; // Full L1 P carrier phases [cycles]
  u1 cs;           // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

p1 – Short L1 P carrier phases

Description

Contains short L1 P carrier phases for all the satellites specified in the latest SI message.

Parameters

Message ID	p1
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SCP_L1P {
  u4 scpL1P[nSats]; // Short L1 P carrier phases [cycles/1024]
  u1 cs;            // Checksum
};
```

Usage Guidelines

- Use the following formula to compute full L1 P carrier phases in cycles:

  Full L1 P carrier phase = Short L1 P carrier phase / 1024

- Short carrier phases from this message will have discontinuities due to periodic rollovers used to represent this kind of carrier phases. This is due to the less memory-consuming carrier phase representation. Such rollovers do not occur very frequently (approximately once every 15 minutes or more). If you want to remove this kind of discontinuity from your carrier phases, you can use the following simple correction technique in the C programming language:

```
static unsigned long prev = value_from_pc;
static long rollOvers=0;
unsigned long curr = value_from_pc;
if ( curr < (1<<30) && prev > (1<<30)*3 ) rollOvers++;
if ( curr > (1<<30)*3 && prev < (1<<30) ) rollOvers--;
prev=curr;
double phase=((1<<16)*double(rollOvers)*(1<<16) + curr) / 1024.; // cycles
```

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

1P – L1 P carrier phases computed relative to pseudoranges from the RC message

Description

Contains the differences between the full L1 P carrier phases and the corresponding pseudoranges from the RC message for all the satellites specified in the latest SI message.

Parameters

Message ID	1P
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L1P_RC {

  f4 rcpL1PRC[nSats]; // Full L1 P carrier phase from P1 / L1 frequency -
                      // - full L1 C/A pseudorange from RC, [s]

  u1 cs;              // Checksum

};
```

Usage Guidelines

- Use the following formula to compute full L1 P carrier phases in cycles:

  *Full L1 P carrier phase = (L1 P carrier phase from 1P + L1 C/A pseudorange from RC) * L1_freq*,
  where L1_freq is the nominal L1 carrier frequency in Hz for the corresponding satellite

- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

1p – L1 P carrier phases computed relative to pseudoranges from the rc message

Description

Contains the differences between the full L1 P carrier phases and the corresponding pseudoranges from the rc message for all the satellites specified in the latest SI message.

Parameters

Message ID	1p
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L1P_rc {
  i4 rcpL1Prc[nSats]; // Full L1 P carrier phase from P1 / L1 frequency -
                      // - delta L1 C/A pseudorange from rc, [s*2^-40]
  u1 cs;              // Checksum
};
```

Usage Guidelines

Use the following formula to compute full L1 P carrier phases in cycles:

*Full L1 P carrier phase = (L1 P carrier phase from 1p * 2^{-40} + delta L1 C/A pseudorange from rc) * L1_freq*,
where L1_freq is the nominal L1 carrier frequency in Hz for the corresponding satellite

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

P2 – Full L2 P carrier phases

Description	Parameters	
Contains the full L2 P carrier phases for all the satellites specified in the latest SI message.	Message ID	P2
	Length of the Message Body (bytes)	8*nSats + 1
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

```
struct CP_L2P {
  f8 cpL2P[nSats]; // Full L2 P carrier phases [cycles]
  u1 cs;           // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

p2 – Short L2 P carrier phases

Description

Contais short L2 P carrier phases for all the satellites specified in the latest SI message.

Parameters

Message ID	p2
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SPC_L2P {

  u4 scpL2P[nSats]; // Short L2 P carrier phases [cycles/1024]

  u1 cs;            // Checksum

};
```

Usage Guidelines

- Use the following formula to compute full L2 P carrier phases in cycles:

  Full L2 P carrier phase = Short L2 P carrier phase / 1024

- Short carrier phases from this message will have discontinuities due to periodic rollovers used to represent this kind of carrier phases. This is due to the less memory-consuming carrier phase representation. Such rollovers do not occur very frequently (approximately once every 15 minutes or more). If you want to remove this kind of discontinuity from your carrier phases, you can use the following simple correction technique in the C programming language:

```
static unsigned long prev = value_from_pc;
static long rollOvers=0;
unsigned long curr = value_from_pc;
if ( curr < (1<<30) && prev > (1<<30)*3 ) rollOvers++;
if ( curr > (1<<30)*3 && prev < (1<<30) ) rollOvers--;
prev=curr;
double phase=((1<<16)*double(rollOvers)*(1<<16) + curr) / 1024.; // cycles
```

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

2P – L2 P carrier phases computed relative to pseudoranges from the RC message

Description

Contains the differences between the full L2 P carrier phases and the corresponding pseudoranges from the RC message for all the satellites specified in the latest SI message.

Parameters

Message ID	2P
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L2P_RC {

  f4 rcpL2PRC[nSats]; // Full L2 P carrier phase from P2 / L2 frequency -
                      // - full L1 C/A pseudorange from RC, [s]

  u1 cs;              // Checksum

};
```

Usage Guidelines

- Use the following formula to compute full L2 P carrier phases in cycles:

  *Full L2 P carrier phase = (L2 P carrier phase from 2P + L1 C/A pseudorange from RC) * L2_freq*,
  where L2_freq is the nominal L2 carrier frequency in Hz for the corresponding satellite

- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Examples” on page 1-17 for more information about the default message set.

2p – L2 P carrier phases computed relative to pseudoranges from the rc message

Description

Contains the differences between the full L2 P carrier phases and the corresponding pseudoranges from the rc message for all the satellites specified in the latest SI message.

Parameters

Message ID	2p
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L2P_rc {
  i4 rcpL2Prc[nSats]; // Full L2 P carrier phase from P2 / L2 frequency –
                      // - delta L1 C/A pseudorange from rc, [s*2^-40]
  u1 cs;              // Checksum
};
```

Usage Guidelines

Use the following formula to compute full L2 P carrier phases in cycles:

*Full L2 P carrier phase = (L2 P carrier phase from 2p * 2^{-40} + delta L1 C/A pseudorange from rc) * L2_freq*,
where L2_freq is the nominal L2 carrier frequency in Hz for the corresponding satellite

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

P3 – Full GPS L2C and GLONASS L2 C/A carrier phases

Description	Parameters	
Contains the full GPS L2C and GLONASS L2 C/A carrier phases for all the satellites specified in the latest SI message.	Message ID	P3
	Length of the Message Body (bytes)	8*nSats + 1
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

```
struct CP_L2C {
  f8 cpL2C[nSats]; // Full GPS L2C and GLONASS L2 C/A carrier phases [cycles]
  u1 cs;           // Checksum
};
```

Usage Guidelines

This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

3P – GPS L2C and GLONASS L2 C/A carrier phases computed relative to pseudoranges from the RC message

Description

Contains the differences between the full GPS L2C and GLONASS L2 C/A carrier phases and the corresponding pseudoranges from the RC message for all the satellites specified in the latest SI message.

Parameters

Message ID	3P
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RCP_L2C_RC {

  f4 rcpL2CRC[nSats]; // Full GPS L2C and GLONASS L2 C/A carrier phasea -
                      // - full L1 C/A pseudorange from RC, [s]

  u1 cs;              // Checksum

};
```

Usage Guidelines

- Use the following formula to compute full L2 C carrier phases in cycles:

  *Full L2 C carrier phase = (L2 C carrier phase from 3P + L1 C/A pseudorange from RC) * L2_freq*,
  where L2_freq is the nominal L2 carrier frequency in Hz for the corresponding satellite

- This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

DC – L1 C/A Doppler

Description

Contains L1 C/A doppler estimates for all the satellites specified in the latest SI message.

Parameters

Message ID	DC
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct D_L1CA {

  i4 dL1CA[nSats]; // L1 C/A Doppler [Hz*10^-4]

  u1 cs;           // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

D1 – L1 P Doppler

Description

Contains L1 P doppler estimates for all the satellites specified in the latest SI message.

Parameters

Message ID	D1
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct D_L1P {

  i4 dL1P[nSats]; // L1 P Doppler [Hz*10^-4]

  u1 cs;          // Checksum

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

D2 – L2 P Doppler

Description

Contains L2 P doppler estimates for all the satellites specified in the latest SI message.

Parameters

Message ID	D2
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct D_L2P {

  i4 dL2P[nSats]; // L2 P Doppler [Hz*10^-4]

  u1 cs;          // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

D3 – GPS L2C and GLONASS L2 C/A Doppler

Description

Contains GPS L2C and GLONASS L2 C/A doppler estimates for all the satellites specified in the latest SI message.

Parameters

Message ID	D3
Length of the Message Body (bytes)	4*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct D_L2C {

  i4 dL2C[nSats]; // Doppler [Hz*10^-4]

  u1 cs;          // Checksum

};
```

Usage Guidelines

This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

EC – L1 C/A signal-to-noise ratios (SNR)

Description

Contains SNRs in the L1 C/A channel for all the satellites specified in the latest SI message.

Parameters

Message ID	EC
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L1CA {
  u1 snrL1CA[nSats]; // SNRs in L1 C/A [dB*Hz]
  u1 cs;             // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

CE – L1 C/A signal-to-noise ratios x 4

Description

This message contains SNRs in the L1 C/A channel multiplied by 4 for all the satellites specified in the latest SI message.

Parameters

Message ID	CE
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L1CAx4 {
  u1 snrL1CAx4[nSats]; // SNRs in L1 C/A [dB*Hz/4]
  u1 cs;               // Checksum
};
```

Usage Guidelines

This message is primarily intended for UNAVCO.

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

N/A

E1 – L1 P signal-to-noise ratios

Description

Contains SNRs in the L1 P channel for all the satellites specified in the latest SI message.

Parameters

Message ID	E1
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L1P {
  u1 snrL1P[nSats]; // SNRs in L1 P [dB*Hz]
  u1 cs;            // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

1E – L1 P signal-to-noise ratios x 4

Description

This message contains SNRs in the L1 P channel multiplied by 4 for all the satellites specified in the latest SI message.

Parameters

Message ID	1E
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L1Px4 {
  u1 snrL1Px4[nSats]; // SNRs in L1 P [dB*Hz/4]
  u1 cs;              // Checksum
};
```

Usage Guidelines

This message is primarily intended for UNAVCO.

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

N/A

E2 – L2 P signal-to-noise ratios

Description

Contains SNRs in the L2 P channel for all the satellites specified in the latest SI message.

Parameters

Message ID	E2
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L2P {
  u1 snrL2P[nSats]; // SNRs in L2 P [dB*Hz]
  u1 cs;            // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

2E – L2 P signal-to-noise ratios x 4

Description

This message contains SNRs in the L2 P channel multiplied by 4 for all the satellites specified in the latest SI message.

Parameters

Message ID	2E
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L2Px4 {
  u1 snrL2Px4[nSats]; // SNRs in L2 P [dB*Hz/4]
  u1 cs;              // Checksum
};
```

Usage Guidelines

This message is primarily intended for UNAVCO.

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

N/A

E3 – GPS L2C and GLONASS L2 C/A signal-to-noise ratios

Description

Contains SNRs in the GPS L2C and GLONASS L2 C/A channel for all the satellites specified in the latest SI message.

Parameters

Message ID	E3
Length of the Message Body (bytes)	nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SNR_L2C {
  u1 snrL2C[nSats]; // SNRs in L2C [dB*Hz]
  u1 cs;            // Checksum
};
```

Usage Guidelines

This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

TC – Time since last loss-of-lock on particular L1 C/A channel

Description

Reports time elapsed since the last loss-of-lock in the L1 C/A channel for every satellite specified in the latest SI message.

Parameters

Message ID	TC
Length of the Message Body (bytes)	nSats*2+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct TrackingTimeL1CA {

  u2 tt[nSats]; // Tracking time [s]

  u1 cs;        // Checksum

};
```

Usage Guidelines

- Each satellite is allocated its own TC-time counter. Count-up begins with zero and stops when the counter reaches the maximum value the `u2` data type allows.
- The TC-time counters are not subject to rollovers.
- Given a satellite, TC-time count starts as soon as the L1 C/A signal is locked on. Should a loss-of-lock occur when tracking the L1 C/A signal, the TC-time counter resets to zero.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

TT – Time since last loss-of-lock on all L1 C/A channels

Description

This message indicates the time elapsed since the last complete loss-of-lock on all L1 C/A channels.

Parameters

Message ID	TT
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct TrackingTime {

  u4 tt; // Tracking time [s]

  u1 cs; // Checksum

};
```

Usage Guidelines

- Count-up begins with zero and stops when the counter reaches the maximum value the `u4` data type allows.
- The TT-time counter is not subject to rollovers.
- TT-time count starts as soon as the first L1 C/A signal is locked on. Should a loss-of-lock occur on the last L1 C/A signal, the TT-time counter resets to zero.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

FC – L1 C/A signal lock loop flags

Description

Contains the L1 C/A signal lock loop flags for all the satellites specified in the latest SI message.

Parameters

Message ID	FC
Length of the Message Body (bytes)	nSats*2 + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct FlagsL1CA {

  u2 flags[nSats]; // L1 C/A lock loop flags [bit-field]

  u1 cs;           // Checksum

};
```

Usage Guidelines

- See Table 4-3 on page 87 for flags definition.
- The simplest approach to L1 C/A data validation is keeping track of only bit #6. As long as this bit remains set for a particular satellite, all of this satellite's C/A measurements are considered good.
  The receiver normally utilizes very narrow DLL bandwidths, thus, quite a long settling-down time (tens of seconds). In fact, it is not worth waiting until the pseudorange noise error reaches its steady-state level. Note that bit #6 is set as soon as the measured pseudoranges become accurate enough[a] (i.e., irrespective of whether the formal settling-down period is over or not). On the other hand, for code differential applications, pseudorange accuracy is of critical importance.
  If bit #4 is set, this indicates that the corresponding pseudoranges are generated after the loop having reached the steady state and therefore are considered the least noisy.
  For applications based on various carrier phase differencing techniques, bi t#5 is of special interest. If set, this means that a loss-of-lock occurred in the PLL between the previous and the current epochs, which is very likely to have resulted in a carrier phase cycle slip. Bit #5 is updated at the same rate at which raw measurements are generated in the receiver. If FC message is output rarer than this, then TC message is used as a PLL loss-of-lock indicator.
  In fact, it is not infrequent that raw data are used even if bit #6 is not set. In such cases, however, all responsibility for providing valid results rests with the user.
  Bits ##0...3 are used for internal purposes.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Message Sets" on page 1-16 for more information about the default message set.

a. say, at a level of unmodeled pseudorange errors

Table 4-3. Flags Definition

Bit #	Hex Mask	Description
0	0x0001	PLL is in phase lock
1	0x0002	Satellite signal strength is sufficient
2	0x0004	Satellite uses guiding data from the common loop
3	0x0008	Satellite data are used in the common loop, which in turn controls this satellite's loops
4	0x0010	DLL is in steady state phase lock
5	0x0020	Loss-of-lock occurred in PLL between the previous and the current epochs
6	0x0040	Integral indicator: all L1 C/A data are good
7	0x0080	Not used
8	0x0100	Preamble detected
9...15		Reserved

F1 – L1 P signal lock loop flags

Description

This message contains the L1 P signal lock loop flags for all the satellites specified in the latest SI message.

Parameters

Message ID	F1
Length of the Message Body (bytes)	nSats*2 + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct FlagsL1P {
  u2 flagsL1P[nSats]; // L1 P lock loop flags [bit-field]
  u1 cs;              // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

Message FC on page 4-86 for flags definition.

F2 – L2 P signal lock loop flags

Description

This message contains the L2 P signal lock loop flags for all the satellites specified in the latest SI message.

Parameters

Message ID	F2
Length of the Message Body (bytes)	nSats*2 + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct FlagsL2P {
  u2 flagsL2P[nSats]; // L2 P lock loop flags [bit-field]
  u1 cs;              // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

Message FC on page 4-86 for flags definition.

F3 – GPS L2C and GLONASS L2 C/A signal lock loop flags

Description

Contains the GPS L2C and GLONASS L2 C/A signal lock loop flags for all the satellites specified in the latest SI message.

Parameters

Message ID	F3
Length of the Message Body (bytes)	2*nSats + 1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct FlagsL2C {
  u2 flagsL2C[nSats]; // L2C lock loop flags [bit-field]
  u1 cs;              // Checksum
};
```

Usage Guidelines

This message is only for G3-based boards and receivers.

Examples

N/A

History

This message was introduced in firmware 3.2.

See Also

Message FC on page 4-86 for flags definition.

SS – Satellite navigation status

Description

Reports navigation status for all the satellites specified in the latest SI message. In addition, this message indicates which receiver positioning mode is on.

Parameters

Message ID	SS
Length of the Message Body (bytes)	nSats+2
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x22

Body

```
struct NavStatus {
  u1 ns[nSats]; // Navigation status
  u1 solType;   // Solution type
  u1 cs;        // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

- “Message Sets” on page 1-16 for more information about the default message set.
- “Satellite Navigation Status Codes” on page D-1 for available navigation status codes.

ID – Ionospheric delays

Description

This message contains estimated ionospheric delays as computed by using the L1 minus L2 frequency combination.

Parameters

Message ID	ID
Length of the Message Body (bytes)	4*nSats+1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct IonoDelay {
  f4 delay[nSats]; // Ionospheric delay [s]
  u1 cs;           // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

GA – GPS almanac

Description

This message contains the current GPS almanac data.

Parameters

Message ID	GA
Length of the Message Body (bytes)	47
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	1800.0
count	0
flags	0x22

Body

```
struct GPSAlm {

  u1 sv;       // SV PRN number within the range [1...37]

  i2 wna;      // Almanac reference week []

  i4 toa;      // Almanac reference time of week [s]

  u1 healthA; // Health summary (from almanac), [bit-field]
              //   0...4 - code for health of SV signal components
              //   5...7 - navigation data health indicators

  u1 healthS; // Satellite health (page 25 of subframe 5) []

  u1 config;  // Satellite configuration (page 25 of subframe 4), [bit-field]
              //   0...2 - satellite configuration
              //       3 - anti-spoofing flag
              //   4...7 - reserved

  //======= Clock data =======

  f4 af1;      // Polynomial coefficient [s/s]

  f4 af0;      // Polynomial coefficient [s]

  //===== Ephemeris data =====

  //----- Keplerian orbital parameters -----

  f4 rootA;    // Square root of the semi-major axis [m^1/2]

  f4 ecc;      // Eccentricity []

  f4 m0;       // Mean anomaly at reference time [semi-circles]

  f4 omega0;  // Longitude of ascending node of orbit plane at the start of
              // week 'wna' [semi-circles]

  f4 argPer;  // Argument of perigee [semi-circles]
```

```
  //----- Corrections to orbital parameters -----
  f4 deli;     // Correction to inclination angle [semi-circles]
  f4 omegaDot; // Rate of right ascension [semi-circle/s]
  u1 cs;       // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

GE – GPS ephemeris

Description	Parameters	
This message contains the current GPS ephemeris data.	Message ID	GE
	Length of the Message Body (bytes)	123
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	900.0
	count	0
	flags	0x22

Body

```
struct GPSEphemeris {

  u1 sv;        // SV PRN number within the range [1...37]

  u4 tow;       // Time of week [s]

  u1 flags;     // Flags (see GPS ICD for details) [bit-field]:
                //      0 - curve fit interval
                //      1 - data flag for L2 P-code
                //   2, 3 - code on L2 channel
                //      4 - anti-spoof (A-S) flag (from HOW)
                //      6 - ephemeris was retrieved from the non-volatile
                //          memory
                //      7 - reserved

  //===== Clock data (Subframe 1) =====

  i2 iodc;      // Issue of data, clock []

  i4 toc;       // Clock data reference time [s]

  i1 ura;       // User range accuracy index []

  u1 healthS;   // Satellite health []

  i2 wn;        // Week number []

  f4 tgd;       // Estimated group delay differential [s]

  f4 af2;       // Polynomial coefficient [s/s^2]

  f4 af1;       // Polynomial coefficient [s/s]

  f4 af0;       // Polynomial coefficient [s]

  //===== Ephemeris data (Subframes 2 and 3) =====

  i4 toe;       // Ephemeris reference time [s]

  i2 iode;      // Issue of data, ephemeris []
```

```
  //----- Keplerian orbital parameters -----

  f8 rootA;    // Square root of the semi-major axis [m^1/2]

  f8 ecc;      // Eccentricity []

  f8 m0;       // Mean anomaly at reference time (wn,toe) [semi-circles]

  f8 omega0;   // Longitude of ascending node of orbit plane at the start of
               // week 'wn' [semi-circles]

  f8 inc0;     // Inclination angle at reference time [semi-circles]

  f8 argPer;   // Argument of perigee [semi-circles]

  //----- Corrections to orbital parameters -----

  f4 deln;     // Mean motion difference from computed value [semi-circle/s]

  f4 omegaDot; // Rate of right ascension [semi-circle/s]

  f4 incDot;   // Rate of inclination angle [semi-circle/s]

  f4 crc;      // Amplitude of the cosine harmonic correction term to the
               // orbit radius [m]

  f4 crs;      // Amplitude of the sine harmonic correction term to the
               // orbit radius [m]

  f4 cuc;      // Amplitude of the cosine harmonic correction term to the
               // argument of latitude [rad]

  f4 cus;      // Amplitude of the sine harmonic correction term to the
               // argument of latitude [rad]

  f4 cic;      // Amplitude of the cosine harmonic correction term to the
               // angle of inclination [rad]

  f4 cis;      // Amplitude of the sine harmonic correction term to the
               // angle of inclination [rad]

  u1 cs;       // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

NA – GLONASS almanac

Description	Parameters	
This message contains the current GLONASS almanac data.	Message ID	NA
	Length of the Message Body (bytes)	46
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	1800.0
	count	0
	flags	0x22

Body

```
struct GloAlmanac {

  u1 sv;     // Satellite orbit slot number within [1...24] []

  i1 frqNum; // Satellite frequency channel number [-7...24] []

  i2 dna;    // Day number within 4-year period starting with the leap year []

  f4 tlam;   // Time of the first ascending node passage on day 'dna' [s]

  u1 health; // Satellite health as specified by 'Cn' [bit-field]
             //      0 - '1' indicates healthy SV, '0' - unhealthy
             //  1...7 - reserved

  //===== Clock data =====

  f4 tauN;   // Coarse time correction to SV clock with respect to GLONASS
             // system time [s]

  f8 tauSys; // Correction to GLONASS system time with respect to UTC(SU) [s]

  //===== Ephemeris data =====

  f4 ecc;    // Eccentricity at reference time 'tlam' []

  f4 lambda; // Longitude of ascending node at reference
             // time 'tlam' [semi-circles]

  f4 argPer; // Argument of perigee at reference time 'tlam' [semi-circles]

  f4 delT;   // Correction to mean Draconic period at reference
             // time 'tlam' [s/period]

  f4 delTdt; // Rate of change of Draconic period [s/period^2]

  f4 deli;   // Correction to inclination at reference
             // time 'tlam'[semi-circles]

  u1 cs;     // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

NE – GLONASS ephemeris

Description

This message contains the current GLONASS ephemeris data.

Parameters

Message ID	NE
Length of the Message Body (bytes)	80
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	900.0
count	0
flags	0x22

Body

```
struct GLOEphemeris {

  u1 sv;     // Satellite orbit slot number [1...24] []

  i1 frqNum; // Satellite frequency channel number [-7...24] []

  i2 dne;    // Day number within 4-year period []
             // (Ephemeris reference time 'tb' is specified for day 'dne')

  i4 tk;     // Frame start time within current day [s]

  i4 tb;     // Ephemeris reference time [s]

  u1 health; // Satellite health. [bit-field]
             //      0 - MSB taken from Bn word which indicates satellite
             //          health:
             //          1 - indicates 'satellite is unhealthy'
             //          0 - indicates 'satellite is healthy'
             //      1 - If set, this flag indicates that the time-frequency
             //          params 'tau' and 'gamma' may be wrong. (Note that
             //          TPS receiver performs several 'internal' data
             //          consistency checks allowing detection of problem
             //          broadcast parameters).
             //      2 - If set, this flag indicates that initial
             //          conditions 'r[3]' and 'v[3]' may be wrong.
             //      3 - SV health (Cn word) status from almanac:
             //          0 - indicates 'satellite is unhealthy'
             //          1 - indicates 'satellite is healthy'
             //      4 - If set, this flag indicates that SV health status
             //          from almanac is available
             //  5...7 - reserved
```

```
  //===== Ephemeris data ======

  u1 age;    // Age of operational information (En) [days]

  u1 flags;  // Flags (for details on the flags, see GLONASS ICD) [bit-field]
             //   0...1 - p1 word
             //         2 - p2 word
             //         3 - p3 word
             //   4...5 - 2 LSB taken from Bn word
             //         6 - '1' means this ephemeris was retrieved from
             //             non-volatile memory
             //         7 - reserved

  f8 r[3];   // Satellite PE-90 coordinates [km]

  f4 v[3];   // Satellite PE-90 velocities [km/s]

  f4 w[3];   // Satellite PE-90 accelerations due to Luni-Solar
             // gravitational perturbations [km/s^2]

   //===== Clock data ======

  f8 tauSys; // Time correction to GLONASS time scale (vs. UTC(SU))
             // tauSys = TUTC(SU) - TGLN [s]

  f4 tau;    // Correction to satellite clock (vs. GLONASS time)
             // tau = TGLN - TSV [s]

  f4 gamma;  // Rate of satellite clock offset [s/s]

  u1 flags2; // [bit-field]:
             //   0...1 - GLONASS-M P
             //         2 - GLONASS-M P4
             //   3...4 - GLONASS-M M
             //         5 - GLONASS-M ln (fifth string)
             //   6...7 - reserved

  f4 tgps;   // Tau GPS

  f4 dtaun;  // Delta Tau N

  i2 Na;     // Day number to which tauS is referenced

  i2 Nt;     // Nt

  u1 Ft;     // Ft

  u1 cs;     // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Message Sets" on page 1-16 for more information about the default message set.

WA – WAAS almanac

Description

This message contains an almanac set of data.

Parameters

Message ID	WA
Length of the Message Body (bytes)	39
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct WAASAlm {

  u1 WAAS_PRN;       // WAAS SV PRN number within [120...138]

  u1 GPS_PRN;        // GPS SV PRN associated with this WAAS SV

  u1 id;             // Data ID

  u1 healthS;        // Satellite health [bit-field]
                     // bit 0:
                     //   0 - Ranging ON
                     //   1 - Ranging is OFF
                     // bit 1:
                     //   0 - Corrections ON
                     //   1 - Corrections OFF
                     // bit 2:
                     //   0 - Broadcast Integrity ON
                     //   1 - Broadcast Integrity OFF
                     // bit 3: reserved
                     // bits 4...7 are all set to zero

  u4 tod;            // Time of the day [s]

  i4 xg, yg, zg;     // ECEF coordinates [m]

  f4 vxg, vyg, vzg; // ECEF velocity [m/s]

  u4 tow;            // Time of GPS week, this almanac was received at

  u2                 // GPS week, this almanac was received at

  u1 cs;             // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

WE – WAAS ephemeris

Description

This message contains a set of WAAS satellite orbit parameters used to predict satellite location at any time.

Parameters

Message ID	WE
Length of the Message Body (bytes)	71
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct WAASEhemeris {

  u1 WAAS_PRN;          // WAAS SV PRN number within [120...138]

  u1 GPS_PRN;           // GPS SV PRN associated with WAAS SV

  u1 iod;               // Issue of data

  u1 acc;               // WAAS SV accuracy (For details, see section
                        // 20.3.3.3.1.3 of ICD-GPS-200C,Revision IRN-200C-004
                        // April 12, 2000)

  u4 tod;               // Reference time (seconds of the day)[s]

  f8 xg, yg, zg;        // ECEF coordinates [m]

  f4 vxg, vyg, vzg;     // ECEF velocity [m/s]

  f4 vvxg, vvyg, vvzg; // ECEF acceleration [m/s^2]

  f4 agf0;              // WAAS SV clock offset factor "ao" [s]

  f4 agf1;              // WAAS SV clock offset factor "a1" [s/s]

  u4 tow;               // Time of GPS week, this ephemeris was received at

  u2 wn;                // GPS week, this ephemeris was received at

  u1 cs;                // Checksum

};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Message Sets" on page 1-16 for more information about the default message set.

GD – GPS raw navigation data

Description

This message contains GPS raw navigation data.

Parameters

Message ID	GD
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x2

Body

```
struct GPSNavData {

  u1 recSize;         // Size of satellite data record (currently 42)

  u1 SatData dat[N]; // Satellite data, where N can be derived from
                     // the following expression:
                     // N = ([Message Length] - 2) / recSize

  u1 cs;             // Checksum


struct SatData {

  u1 prn;            // Satellite PRN

  u1 cnt;            // Counter which is updated upon receiving a new
                     // sub-frame for the given satellite

  i4 data[10];       // GPS sub-frame contents (every 4-bytes word contains
                     // 30 LSB of GPS navigation data)

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

LD – GLONASS raw navigation data

Description

This message contains GLONASS raw navigation data.

Parameters

Message ID	LD
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x2

Body

```
struct GlnNavData {

  u1 recSize;         // Size of satellite data record (currently 18)

  u1 SatData dat[N]; // Satellite data, where N can be derived from
                      // the following expression:
                      // N = ([Message Length] - 2) / recSize

  u1 cs;              // Checksum


struct SatData {

  i1 fcn;             // Satellite FCN

  u1 cnt;             // Counter which is updated upon receiving a new
                      // string of GLONASS sub-frame for the given FCN

  i4 data[4];         // GLONASS sub-frame contents (every 4-bytes word
                      // contains 25 LSB of GLONASS sub-frame)

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

N/A

PPS and Event Marker Messages

ZA, ZB – PPS offset

Description

Contains the offset between the scheduled PPS time and the actual pulse edge's arrival time.

Parameters

Message ID	ZA, ZB
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PPSOffset {

  f4 offs; // PPS offset [ns]

  u1 cs;   // Checksum

};
```

Usage Guidelines

- Due to a hardware limitation, PPS signals are discrete with a resolution of 25 ns or 5 ns depending on the board. You may use this message to compensate for this discrete error in TPS receivers
- When the pulse edge is advanced relative to the scheduled time, the offset is positive.
- When the pulse edge is delayed relative to the scheduled time, the offset is negative.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

YA, YB – Time offset at PPS generation time

Description

This message contains the offset between PPS reference time (T_{ppsr}) and receiver time (T_r) at the moment of PPS generation.

Parameters

Message ID	YA, YB
Length of the Message Body (bytes)	10
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvTimeOffsAtPPS {

  f8 offs;   // [T_ppsr - T_r][s]

  u1 tScale; // PPS reference time scale
             //       0 – GPS system time
             //       1 – UTC (USNO). Universal Coordinated Time
             //           supported by the U.S. Naval Observatory.
             //       2 – GLONASS system time
             //       3 – UTC (SU). Universal Coordinated Time
             //           supported by the State Time and Frequency Service,
             //           Russia
             // 4...254 – Reserved

  u1 cs;     // Checksum

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

XA, XB – External event

Description

Contains the event time tag in corresponding time scale modulo one day.

Parameters

Message ID	XA, XB
Length of the Message Body (bytes)	10
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x30

Body

```
struct ExtEvent {

  i4 ms;      // ms part of event time tag

  i4 ns;      // ns part of event time tag

  u1 tScale; // PPS reference time scale
             //       0 – GPS system time
             //       1 – UTC (USNO. Universal Coordinated Time
             //           supported by the U.S. Naval Observatory.
             //       2 – GLONASS system time
             //       3 – UTC (SU. Universal Coordinated Time
             //           supported by the State Time and Frequency Service,
             //           Russia
             // 4...254 – Reserved

  u1 cs;      // Checksum

};
```

Usage Guidelines

- This message belongs to the default set of messages.
- To make the receiver generate these messages, take the following steps:
  - set the parameter `/par/dev/event/a/in` or `/par/dev/event/b/in` to `on` and
  - issue an appropriate `em` or `out` command.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

Optimized Real-Time Application (ORTA) Messages

ORTA messages contain different yet logically related observables, position and time information as one data set. For example, the message rM is comprised of all of the code and carrier phase measurements available in the receiver for the given epoch.

These messages are primarily designed for applications where you require high data integrity together with minimizing the amount of data transmitted over a communication link.

You can control the structure of ORTA messages by means of parameters from “Governing Optimized Real Time Application (ORTA) Messages” on page 3-333.

The following tables will explain the relationships between the ORTA messages rE, rM, rV and the basic TPS messages.

The rM message can be used in place of the following basic messages:

Pseudorange measurements	RC, rc, R1, r1, 1R, 1r, R2, r2, 2R, 2r
Carrier phase measurements	PC, pc, CP, cp, P1, p1, 1P, 1p, P2, p2, 2P, 2p
Doppler	DC, D1, D2
Signal lock loop flags	FC, F1, F2, F3
Signal-to-noise ratios	EC, E1, E2, E3
Satellite navigation status	SS
Satellite Indices	SI
GLONASS satellite system numbers	NN
Receiver date and time	RD, RT
Time since last loss-of-lock	TC
Receiver reference time to receiver time offset	TO

The rV message will replace the following basic messages:

Position and velocity	PO, VE, PV
Solution time tag	ST

The rE message will replace the following basic messages:

Receiver date and time	RD, RT

You can receive ORTA messages in an arbitrary order of precedence. Before decoding a message its checksum must be computed. Recall that the checksum is computed for all the bytes starting from the header of the message up to the checksum positions exclusive. For how to compute CRC-16, refer to “16-Bit Checksum” on page A-2.

rE – Reference epoch

Description

Contains date and time for the given epoch.

Parameters

Message ID	rE
Length of the Message Body (bytes)	10
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RefEpoch {

  u2 sample;  // Sample number []

  u2 scale;   // Time scale ID, leap second status and week/day part of
              // epoch representation [bit-field]
              // 15...13: Time scale ID:
              //          0 – GPS, 1 – GLONASS, 2 – UTC
              // 12...11: Leap second status:
              //           0 – no leap second epoch
              //           1 – positive leap second
              //           2 – negative leap second
              //           3 – leap second status is unknown
              //           (this flag shows whether a leap second occurred
              //           at the current epoch)
              //  10...0: Week/day representation:
              //          (a) if time scale ID is GPS:
              //                week number [0-1023],
              //                1024 indicates unknown week number
              //          (b) if time scale ID is GLONASS:
              //                day number within 4-year period [1-1461],
              //                0 indicates unknown day number
              //          (c) if time scale ID is UTC:
              //                day number within the year [1-366],
              //                0 indicates unknown day number;

  u4 reftime; // Seconds part of epoch representation [ms:
              // (a) if time scale ID is GPS, "reftime" contains seconds
              //     of GPS week;
              // (b) if time scale ID is GLONASS, "reftime" contains seconds
              //     of GLONASS day;
              // (c) if time scale ID is UTC, "reftime" contains seconds
              //     of UTC day;

  u2 crs16;   // 16-bit CRC

};
```

Usage Guidelines

The field `sample` serves two main purposes:

- it allows you to preserve data integrity since messages referenced to a specific epoch will all have the same sample number.
- it allows you to keep track of the number of lost messages issued through a given port since the sample number is incremented when the next epoch starts.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Governing Optimized Real Time Application (ORTA) Messages” on page 3-333.

rM – Raw Measurements

Description

Contains all of observables available in the receiver for the given epoch.

Parameters

Message ID	rM
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct SlotRec {
  // Note: The zeroth element of the array Slot[i],i=0,...,M-1, unlike the
  // other elements, does not contain corrections to the reference pseudorange
  // from the Header structure.
  // To provide the user with additional information, the flag "svst" is used
  // for "delrange" in the zeroth slot.

union {

  i2 svst;     // SV status [bit-field]:
               // 15...11: GLONASS slot number [0...24], 0 – unknown
               //          For GPS SV the bit-fields are undefined
               //  10...6: Channel number [0...31]
               //          31 indicates that channel number is not available
               //   5...0: SV navigation status (see [SS] message)

  i2 delrange; // Delta pseudorange [0.02 meters]:
               // [full pseudorange for given slot]-[refrange]
  };

  u4 word1;    // Packed data 1 [bit-field]:
               // 31...12: [carrier phase] - [refrange]
               //          -2^19/+(2^19 -1)[0.0005 meters]
               //  11...9: slot ID:
               //          0 – C/A L1, 1 – P1, 2 – P2, 3 – C/A L2,
               //          4 – L5, 5,6,7 – reserved
               //       8: Reserved
               //       7: Signal lock loop flags are available
               //       6: Lock time is available
               //   5...0: Signal-to-noise ratio [dB*Hz]

  u2 flags;    // Signal lock loop flags (see FC message). Available for
               // versions 0 and 1 only.
```

```
  u2 lock;      // Packed data 2. Available for versions 0 and 1 only,
                // [bit-field]:
                // 15...12: Fractional part of signal-to-noise ratio [0.1 dB*Hz]
                // 11...10: Reserved
                //   9...0: Lock time [0.1 second] Tracking time since last
                //          loss of lock. Varies between 0 and 102.3 seconds.
                //          Gets stuck at 102.3 s after the actual tracking
                //          time exceeds this value (until another loss of
                //          lock occurs).

  u4 word2;     // Packed data 3. Available for version 0 only, [bit-field]:
                // 31...7: Doppler, -2^24/+(2^24 -1) [0.001 Hz]
                //  6...0: Reserved
};

struct Header {

  u4 refrange; // Reference pseudorange [0.02 meters]

  u1 svid;      // SV ID (see SI message)

  u1 num;       // Number of slot records (M) [bit-field]:
                // 7...3: Reserved
                // 2...0: Number of slot records minus one (M)

};

struct SvData {

struct Header  Head;    // Header

struct SlotRec Slot[M]; // Slot record

};

struct RawMeas {

  u2 sample;  // Sample number []

  u2 scale;   // Time scale ID, leap second status and week/day part of
              // epoch representation [bit-field]
              // 15...13: time scale ID:
              //          0 - GPS, 1 - GLONASS, 2 - UTC
              // 12...11: leap second status:
              //           0 - no leap second epoch
              //           1 - positive leap second
              //           2 - negative leap second
              //           3 - leap second status is unknown
              //           (this flag shows whether a leap second occurred
              //           at the current epoch)
              //  10...0: week/day representation:
              //          (a) if time scale ID is GPS:
              //                week number [0-1023],
              //                1024 indicates unknown week number
```

```
              //          (b) if time scale ID is GLONASS:
              //                day number within 4-year period [1-1461],
              //                0 indicates unknown day number
              //          (c) if time scale ID is UTC:
              //                day number within the year [1-366],
              //                0 indicates unknown day number;

  u4 reftime; // Seconds part of epoch representation [ms]:
              // (a) if time scale ID is GPS, "reftime" contains seconds
              //     of GPS week;
              // (b) if time scale ID is GLONASS, "reftime" contains seconds
              //     of GLONASS day;
              // (c) if time scale ID is UTC, "reftime" contains seconds
              //     of UTC day;

  i2 clock;   // Clock offset. Contains zeros for version 2, [bit-field]:
              // 15...2: Clock offset -2^13/+(2^13-1) [125 nanoseconds]
              //  1...0: Clock offset ID:
              //            0 - Clock offset is unavailable
              //            1 - [GPS - Receiver time]
              //            2 - [GLONASS - Receiver time]
              //            3 - Reserved

  u2 flags;   // Flags [bit-field]:
              // 15...13: Message version [0...7]
              //  12...8: Total number of "Svd" records (N)
              //   7...5: This value plus 6 makes the length of the structure
              //          "Header" in bytes
              //   4...0: This value plus 10 (for message versions 0 and 1)
              //          or 6 (for message versions 2...7) makes the length
              //          of the structure "SlotRec" in bytes

struct SvData Svd[N]; // SV data

  u2 crs16;   // 16-bit CRC

};
```

Usage Guidelines

The field `sample` serves two main purposes:

- it allows you to preserve data integrity since messages referenced to a specific epoch will all have the same sample number.
- it allows you to keep track of the number of lost messages issued through a given port since the sample number is incremented when the next epoch starts.

When handling this message, the following rules must be observed:

- You should retrieve from the message its version number and the lengths of the structures `Header` and `SlotRec`. These parameters are necessary to maintain compatibility with older software products in case the message structure is modified in the future. At present, there are three versions, 0, 1, and 2.
  - Version 0: the rM message is output as is (i.e, in its original form).
  - Version 1: mainly intended for RTK applications. In this version, the field `u4 word2` (i.e. Doppler) is removed from the structure `SlotRec` altogether, which results in a more compact data set as compared against version 0.

  – Version 2: provides further data reduction by excluding the fields `u2 flags`, `u2 lock` and `u4 word2` from the structure `SlotRec`. Also, the `i2 clock` field contains zeroes in this version.

- Next, you should retrieve the `Total number of "Svd" records` flag. Although it is possible to decode the message by using the message length from the message's header, taking into account the `Total number of "Svd" records` flag simplifies the decoding.
- The field `refrange` from the structure `SvData` serves as a reference for all the other code and carrier phase measurements available for the given satellite. In other words, all the measurements other than the reference pseudo-range, are represented as deltas referenced to a common reference value. Such an approach allows the reduction of message length. The first parameter in the structure `SlotRec` should be handled depending on whether the structure's slot number is zero or not.
- The field `num` from the structure `Header` shows the total number of slot records (see the structure `SlotRec`). For example, if only L1 C/A measurements are enabled in the message, the field `num` will be zero.

Examples

N/A

History

- This message was introduced in the first release of firmware.
- Fractional part of signal-to-noise ratio (bits 15...12 from the field `u2 lock`) was implemented in firmware 2.4.
- Version 2 was added in firmware 2.4.

See Also

“Governing Optimized Real Time Application (ORTA) Messages” on page 3-333.

rV – Position and velocity

Description

Contains the position and velocity of the receiver's antenna in the selected coordinate system.

Parameters

Message ID	rV
Length of the Message Body (bytes)	42
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct PVVector {

  u2 sample; // Sample number []

  u2 delta;  // [bit-field]:
             // 15...5: Difference between the raw measurement time tag
             //         (available from either rM or rE message) and
             //         the position time tag [-1024...1023],
             //         [5 milliseconds]
             //    4-0: reserved

  u4 word1;  // 32 MSB of Position X-component

  u4 word2;  // [bit-field]:
             // 31...24: 8 LSB of Position ECEF X-component [10^-4 meters]
             //          or Latitude [10^-11 rad]
             //          or Grid (Local) X-component [10^-4 meters]
             //      23: "1" indicates that Position is valid
             // 22...21: 0 – Position is given in ECEF system
             //          1 – Position is given in geodetic coordinates
             //              (latitude, longitude, height above ellipsoid)
             //          2 – Position is given in grid (or local)
             //              coordinates
             //          3 – Reserved
             // 20...16: Number of GPS Sats used in computation
             //      15: "1" indicates that Velocity is valid
             // 14...13: Reserved
             //  12...8: Number of GLONASS Sats used in computation
             //   7...4: Position computation mode
             //          (see "NP – Navigation position" on page 4-155)
             //   3...0: Velocity computation mode
             //          (see "NP – Navigation position" on page 4-155))

  u4 word3;  // 32 MSB of Position Y-component
```

```
  u4 word4;  // [bit-field]:
             // 31...24: 8 LSB of Position Y-component [10^-4 meters]
             //          or Latitude [10^-11 rad]
             //          or Grid (Local) X-component [10^-4 meters]
             // 23...15: PDOP * 10 []
             //  14...0: RMS velocity error [10^-3 meters]

  u4 word5;  // 32 MSB of Position Z-component

  u4 word6;  // [bit-field]:
             // 31...24: 8 LSB of Position Z-component [10^-4 meters]
             //          or Height above ellipsoid [10^-4 meters]
             // 23...20: Reserved
             //  19...0: RMS Position error [10^-3 meters]

  u4 word7;  // [bit-field]:
             // 31...4: velocity X-component [10^-4 m/s]
             //         or East component (if type 1 or 2 is selected
             //         in bits 22...21 of the u4 word2 field)
             //  3...2: Reserved
             //  1...0: MSB of TPS datum number (refer to See Also below)

  u4 word8;  // [bit-field]:
             // 31...4: velocity Y-component [10^-4 m/s]
             //         or North component (if type 1 or 2 is selected
             //         in bits 22...21 of the u4 word2 field)
             //  3...0: bits 7...4 of datum ID

  u4 word9;  // [bit-field]:
             // 31...4: velocity Z-component [10^-4 m/s]
             //         or Up component (if type 1 or 2 is selected
             //         in bits 22...21 of the u4 word2 field)
             //  3...0: LSB of TPS datum number

  u2 crs16;  // 16-bit CRC
};
```

Usage Guidelines

The field `sample` serves two main purposes:

- it allows you to preserve data integrity since messages referenced to a specific epoch will all have the same sample number.
- it allows you to keep track of the number of lost messages issued through a given port since the sample number is incremented when the next epoch starts.

Examples

N/A

History

- This message was introduced in the first release of firmware.
- Bits 22...21 of the `u4 word2` field were implemented in firmware 2.4.

See Also

- “Governing Optimized Real Time Application (ORTA) Messages” on page 3-333.
- “Reference Ellipsoids and Local Datums” on page H-1 for TPS datum numbers.

rT – Receiver's clock offsets

Description

Contains the clock parameters.

Parameters

Message ID	rT
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct ClkOffs {

  u4 word1; // [bit-field]:
            //     31: Reserved
            //     30: If set, improved timing mode is turned on.
            // 29...0: Clock offset [10^-4 meters]; bit combination
            //         0x20000000 means that clock offset is not
            //         available or exceeds +/- 536870911

  u4 word2; // [bit-field]:
            // 31...29: Reserved
            // 28...26: Navigation system (0 – GPS, 1 – GLN)
            //  25...0: Derivative of clock offset [10^-4 m/s]
            //          bit combination 0x2000000 means that clock
            //          offset is not available or exceeds +/- 33554431
};

struct ClockOffsets {

  u2 sample;        // Sample number []

  u2 reserved;      // Reserved for future extensions

  u1 recSize;       // Size of data block that corresponds to the given
                    // satellite system in bytes (currently 8 bytes)

  ClkOffs  Offs[N]; // Clock offsets, where N can be derived from
                    // the following expression:
                    // N = ([Message Length] - 7) / recSize)

  u2 crs16;         // 16-bit CRC

};
```

Usage Guidelines

The field `sample` serves two main purposes:

- it allows you to preserve data integrity since messages referenced to a specific epoch will all have the same sample number.
- it allows you to keep track of the number of lost messages issued through a given port since the sample number is incremented when the next epoch starts.

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

"Governing Optimized Real Time Application (ORTA) Messages" on page 3-333.

Miscellaneous Messages

IO – Ionospheric parameters

Description

This message contains ionospheric correction parameters from GPS subframe 4, page 18.

Parameters

Message ID	IO
Length of the Message Body (bytes)	39
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	3600.0
count	0
flags	0x23

Body

```
struct IonoParams{

  u4 tot;    // Time of week [s]

  u2 wn;     // Week number (taken from the first subframe)

// The coefficients of a cubic equation representing the amplitude of the
// vertical delay
  f4 alpha0; // [s]

  f4 alpha1; // [s/semicircles]

  f4 alpha2; // [s/semicircles^2]

  f4 alpha3; // [s/semicircles^3]

// The coefficients of a cubic equation representing the period of the model
  f4 beta0;  // [s]

  f4 beta1;  // [s/semicircles]

  f4 beta2;  // [s/semicircles^2]

  f4 beta3;  // [s/semicircles^3]

  u1 cs;     // Checksum

};
```

Usage Guidelines

- This ionospheric model mainly used by single frequency GPS receivers.
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

- ICD-GPS-200C, Revision IRN-200C-004 April 12, 2000 for more information about this ionosphere model.
- “Message Sets” on page 1-16 for more information about the default message set.

LT – Message output latency

Description

This message contains the difference between the actual output time of the first of the messages sent to the chosen output stream at the given epoch, and this epoch's time-tag.

Parameters

Message ID	LT
Length of the Message Body (bytes)	2
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Latency {

  u1 lt; // output latency [ms]

  u1 cs; // Checksum

};
```

Usage Guidelines

The latency can differ for different streams depending on the number of messages enabled for output to the streams. For example, the more messages are output to port A, the bigger the latency of port B; this is because the receiver begins outputting messages to port B only after it has finished outputting messages to port A.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

>> – Wrapping data

Description

This message is used to wrap up arbitrary data.

Parameters

Message ID	>>
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	N/A

Body

```
struct WrapData {

  u1 sid;         // Source identifier

  u1 data[size]; // Data received from the source. The size of the wrapped
                 // data (in bytes) is equal to the message length from
                 // the header less 3 (size=L-3).

  a1 cs[2];      // Checksum formatted as hexadecimal

};
```

Usage Guidelines

This message serves two purposes:

- Wrapping data from an input stream which has been set to the *wrapped echo* mode. The message is then generated whenever data occur on the stream. In this case, the `sid` field can take one of the following values:
  - `a...d`: serial ports A...D, `/dev/ser/a`,...,`/dev/ser/d`
  - `A...E`: TCP ports A...E, `/dev/tcp/a`,...,`/dev/tcp/d`
  - `U`: USB port, `/dev/usb/a`
- Wrapping any receiver's message into >> message before output. In this case, the `sid` field takes a value equal to 1 - count, where *count* is the parameter from the message scheduling specification.

You cannot enable or disable this message for output through the `em` and `dm` commands. This message is generated automatically according to the internal algorithm.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

"Defining Message Generation" on page 1-18 for details on the *count* parameter.

LH – Logging history

Description

This message contains history of logging of satellites data into a particular stream.

Parameters

Message ID	LH
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x2

Body

```
struct LoggingHistory  {

  u1 svsCount;                    // Number of SVs in this history

  u1 targetStream;                // The stream the history is gathered for
                                  // (see >> message for details)

  u2 issue;                       // The issue of the history. It is
                                  // incremented every time the history
                                  // is changed.

  u2 bitsCount;                   // Number of bits in this history. This
                                  // history contains "bitsCount" bits for
                                  // every SV specified in the "uids" field.

  u4 lastBitTime;                 // Time since the last history shift [ms]

  u1 uids[svsCount];              // Array of SVs UIDs (UIDs are the same as
                                  // in SI message)

  u1 zeroFill[fillCount];         // Padding to align the next field on 4
                                  // bytes boundary.
                                  // FillCount = (4 - (svsCount % 4)) % 4

  u4 hist[elemsCount][svsCount]; // History bits. For every SV the bits
                                  // are packed into array of "u4" values.
                                  // Most significant bit of the first
                                  // element of the array represents most
                                  // recent bit of the history.
                                  // Least significant bit of the last
                                  // element of the array represents the
                                  // oldest bit of the history.
                                  // The number of "u4" elements in the array
                                  // is just enough to hold "bitsCount" bits.
                                  // "svsCount" bit arrays are put into the
                                  // message in the order specified by the
                                  // "uids"field.
                                  // elemsCount = bitsCount + 31) / 32
```

```
  u1 cs;                          // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“History Logger Parameters” on page 3-183 for a description of parameters governing logging history.

EV – Event data

Description

This message contains free-form event or the receiver's internal event.

Parameters

Message ID	==
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x30

Body

```
struct Event {

  u4 time;    // Receiver time of event issue modulo one day [ms]

  u1 id;      // Event identifier:
              // 0 - free form event; The event was generated by the
              //     event command.
              // 1 - internal event; The event was generated automatically
              //     whenever a receiver option is expired or whenever the
              //     receiver makes the RTK position unavailable due to the
              //     limitation imposed by the DIST option.
              //     In the first case, the data field

  u1 data[]; // Event contents:
              // If u1 id is set to 0, this field contains data specified
              // by the event command.
              // If u1 id is set to 1, this field has the
              // format OPT,<OPTNAME>, where OPTNAME is the name of the
              // option expired or OPT,DIST,RANGE.

  u1 cs;      // Checksum

};
```

Usage Guidelines

- To enable or disable this message, use the message name `/msg/jps/EV` (not `/msg/jps/==`).
- This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

- "event" on page 2-16 for more information about the `event` command.
- "Message Sets" on page 1-16 for more information about the default message set.

OO – Oscillator offset

Description

This message contains the difference between the nominal oscillator frequency set at the factory, and the actual oscillator frequency at a given instant (normalized to nominal frequency).

Parameters

Message ID	OO
Length of the Message Body (bytes)	5
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct RcvOscOffs {

  f4 val; // Difference between nominal and actual frequencies
          // (normalized to nominal frequency) [s/s]

  u1 cs;  // Checksum

};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

SE – Security

Description

This message is intended for TPS internal use.

Parameters

Message ID	SE
Length of the Message Body (bytes)	6
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Security {
  u1 data[5]; // TPS proprietary data
  u1 cs;      // Checksum
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“Message Sets” on page 1-16 for more information about the default message set.

SM – Security for rM message

Description

This message is intended for use with the rM message being recorded to a log file for post-processing.

Parameters

Message ID	SM
Length of the Message Body (bytes)	8
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct Security {
  u1 data[6]; // TPS proprietary data
  u2 crc16;   // 16-bit CRC
};
```

Usage Guidelines

This message belongs to the default set of messages.

Examples

N/A

History

This message was introduced in firmware 2.4.

See Also

“Optimized Real-Time Application (ORTA) Messages” on page 4-111.

EE – Epoch end

Description

This message itself carries no information. It is intended just as an *end-of-epoch* marker used for real-time applications.

Parameters

Message ID	ll
Length of the Message Body (bytes)	1
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

```
struct EpochEnd {
  u1 cs; // Checksum
};
```

Usage Guidelines

- To enable/disable this message, use the message name `/msg/jps/EE` (not `/msg/jps/||`).
- For greater integrity, it is recommended to use the message ET instead of EE.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

“ET – Epoch time” on page 4-11.

ND – mmGPS data

Description	Parameters	
Contains data derived from GPS and mmGPS engines.	Message ID	ND
	Length of the Message Body (bytes)	50
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

```
struct Ndata {
  f8 NLat;      // GPS lattitude
  f8 NLon;      // GPS longitude
  f8 NAltGPS;   // GPS height
  f8 NAlt;      // mmGPS height
  f4 NSigma;    // mmGPS height RMS
  u1 solType;   // Solution type
  u1 trBatInf;  // Status information obtained from the sensor
  u1 rcvBatInf; // Status information obtained from the sensor
  u1 errInf;    // Status information obtained from the sensor
  f8 angle;     // Nbeam angle used in calculations
  u1 id;        // Channel ID explicitly set or chosen by the rover
  u1 cs;        // Checksum
};
```

Usage Guidelines

N/A

Examples

N/A

History

This message was introduced in firmware 2.5.

See Also

N/A

Standard Text Messages

This section begins with the format notation used to describe the ASCII message formats presented below. A TPS ASCII message's format specifies its structure, parameter types and the number of significant digits for each parameter representation.

The following data type specificators are used to distinguish between different data:

- D – designates decimal integer parameter type.
- X – designates hexadecimal integer parameter type.
- F – designates float or double parameter type.
- C – designates character parameter type.
- S – designates string type (note that strings may have arbitrary lengths).
- E – designates exponential format.

Given a numeric parameter, digits preceding the data type specificator designate the number of digits in this parameter format. There may be two, one or no digits specified before the data type specificator. For float and double parameters, the first digit defines the length of the integer part of the parameter representation (integer part descriptor) whereas the second digit defines the length of the fractional part (fractional part descriptor). If there are two digits used in the parameter format notation, these are separated by a dot "." (hex 2E).

A parameter's fractional part descriptor can be variable, i.e. its length is programmable with an appropriate receiver command. In this case the fractional part descriptor shows the parameter range, which is put inside square brackets (e.g., %0.[1-4]F).

Integer part descriptor may be omitted, which means that the integer part of the output value may be as long as necessary. Note that the delimiting dot before the corresponding fractional part descriptor will still exist.

If an integer parameter has no integer part descriptor before its data type specificator (either X or D), the receiver will output all the significant digits. Leading zeroes will be added if the actual length of the integer part is shorter than specified by the descriptor.

If a format notation includes the plus symbol "+", your receiver will output signed values (as usual, characters "+" (hex 2B) and "-" (hex 2D) are used to denote positive and negative values, respectively).

If a parameter format is bracketed by round braces, it means that this format applies to a batch of "homogeneous" parameters (note that the number of messages in such a batch may vary).

When a TPS text message is output, its individual parameters are delimited from each other with comma (hex 2C). Parameter format notation always starts with symbol "%" (hex 25).

A back slash followed by two hexadecimal digits designates that the corresponding ASCII character will be put into the message in this specific place. In addition, the reader will notice that various non-reserved symbols (lower and upper case English letters, arithmetic operation signs, curly braces, etc.) are used together with the above described parameter formats.

Table 4-4 illustrates the format notation as defined above.

Table 4-4. Data Format in Standard Text Messages

Format Specified	Output Data Representation
%5.2F	00027.89
%5.2F * %+4.2F	67283.67 * +5678.22
Value: %+5.2F	Value: +00000.35
%+2.0F	+07
%0.4F	0.1234
%+7.5F	-0000039.67432
5-2=%D	5-2=3
%D	2467
%+3D	-052
%+3D	+964
%2X	0A
{%2D,%+.4F}	{04,-45.8027}
%C%C%C	XYZ
%S	This is a string
\4A	J
\50	P
(%2.1F)	45.8,98.0,04.7,88.3
%.3E	1.234E-04
%.3E	-5.789E+02

General Messages

DL – Data link status

Description	Attributes	
This message displays the status of the data links associated with the corresponding.	Message ID	DL
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`DLINK`	Title of the message.
2	`%1D`	Total number of data links (0…5). The rest of the message is available only if this number is non-zero. Otherwise the total number of data links value is immediately followed by the checksum.
3	`({%C,%C,%S,%3D,%4D,%4D,%.2F},`, where	Group of parameters associated with the given data link. The total number of such groups is determined by the previous field.
	`%C`	Data link identifier: • A...D – serial ports • E...I – TCP ports • L – Bluetooth • U – USB port
	`%C`	Decoder identifier: • C – CMR decoder • J – TPS decoder • R – RTCM 2.x decoder • T – RTCM 3.0 decoder.
	`%S`	Reference station identifier.
	`%3D`	Time, in seconds, elapsed since receiving last message (maximum value = 999). Estimated with an accuracy of 1 sec.
	`%4D`	Number of received messages (between 0001 and 9999). If no message received, this data field contains zero.

3	%4D	Number of corrupt messages (between 0001 and 9999). If no corrupt messages detected, this data field is set to zero.
	%.2F	Data link quality in percent (0-100).
4	@%2X	Checksum

Usage Guidelines

N/A

Example

DL02C,DLINK,1,{D,R,0036,000,1396,0001,100.00},@D3

History

- This message was introduced in the first release of firmware.
- Decoder identifier T was added in firmware 2.4.

See Also

N/A

GS – GPS SV status

Description	Attributes	
This message describes the status of GPS satellites.	Message ID	GS
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`GPSVST`	Title of the message.
2	`%2D`	Total number of locked GPS satellites.
3	`({%2D,%2D,%3D,{%2D,%2D,%2D,%2D},%2D}`, where	The total number of such 8-parameter sections will coincide with the number of locked SVs.
	`%2D`	GPS PRN number.
	`%2D`	Elevation in degrees.
	`%3D`	Azimuth in degrees.
	`%2D`	L1 C/A signal-to-noise ratio [dB*Hz].
	`%2D`	L1 P signal-to-noise ratio [dB*Hz].
	`%2D`	L2 P signal-to-noise ratio [dB*Hz].
	`%2D`	L2C signal-to-noise ratio [dB*Hz].
	`%2D`	Channel navigation status (see “Satellite Navigation Status Codes” on page D-1).
4	`@%2X`	Checksum

Usage Guidelines

The parameter `%2D` for L2C signal-to-noise ratio appears only when the receiver tracks these signals.

Example

GS118,GPSVST,10,{27,10,210,{39,19,19},00},{13,49,260,{49,40,40},00},{16,24,100,{42,27,27},00},{07,15,205,{42,24,24,40},00},{04,37,287,{49,35,35},00},{23,89,150,{57,49,50},00},{25,35,201,{49,35,35},00},{31,19,047,{41,27,27,00},00},{20,50,148,{51,40,41},00},{02,00,000,{45,25,25},06},@1C

History

- This message was introduced in the first release of firmware.
- The parameter `%2D` for L2C signal-to-noise ratio was added in firmware 3.1

See Also
N/A

NS – GLONASS SV status

Description

This message describes the status of GLONASS satellites.

Attributes

Message ID	NS
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`GLSVST`	Title of the message.
2	`%2D`	Total number of locked GLONASS satellites.
3	`({%2D,%2D,%2D,%3D,{%2D,%2D,%2D,%2D},%2D})`, where	The status of one locked GLONASS satellite. The total number of such 9-parameter sections will coincide with the number of locked SVs.
	`%2D`	GLONASS SV Orbit Slot Number. If for a GLONASS SV its orbit slot number is reported as zero, this means that this parameter hasn't yet been detected.
	`%2D`	GLONASS SV Frequency Channel Number.
	`%2D`	Elevation in degrees.
	`%3D`	Azimuth in degrees.
	`%2D`	L1 C/A signal-to-noise ratio [dB*Hz].
	`%2D`	L1 P signal-to-noise ratio [dB*Hz].
	`%2D`	L2 P signal-to-noise ratio [dB*Hz].
	`%2D`	L2 C/A signal-to-noise ratio [dB*Hz].
	`%2D`	Channel navigation status (see “Satellite Navigation Status Codes” on page D-1).
4	`@%2X`	Checksum

Usage Guidelines

The parameter `%2D` for L2 C/A signal-to-noise ratio appears only when the receiver tracks these signals.

Example

NS0A8,GLSVST,05,{14,04,06,208,{40,39,36,36},00},{11,00,11,035,{44,43,33,33},00},{19,03,10,303,{42,41,36},00},{20,02,17,354,{43,43,40,40},00},{04,06,29,312,{47,46,42},00},@5B

History

- This message was introduced in the first release of firmware.
- The parameter `%2D` for L2 C/A signal-to-noise ratio was added in firmware 3.1

See Also

N/A

RS – Reference station status

Description	Attributes	
This message contains parameters related to the reference station status.	Message ID	RS
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`REFST`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest digits designate seconds)
4	`{%C,%C,%C,%C}`, where	Checking reference station location. • V – means that the difference between the current receiver coordinates and the user-defined reference coordinates does not exceed the specified limit • N – means that the difference between the current receiver coordinates and the user-defined reference coordinates is greater than the specified limit See the parameter `/par/ref/limit` on page 3-353.
	`%C`	The APC reference station coordinates used for referencing GPS data (see the parameters `/par/ref/pos/gps/...`).
	`%C`	The APC reference station coordinates used for referencing GLONASS data (see the parameters `/par/ref/pos/glo/...`).
	`%C`	The ARP reference station coordinates used for referencing GPS data (see the parameters `/par/ref/arp/gps/...`).
	`%C`	The ARP reference station coordinates used for referencing GLONASS data (see the parameters `/par/ref/arp/glo/...`).
5	`@%2X`	Checksum

Usage Guidelines

N/A

Example

RS020,REFST,V,113004.00,{N,N,N,N},@FC

History

This message was introduced in the first release of firmware.

See Also

N/A

JI – Jamming Suppressor information

Description	Attributes	
This message contains information about the detected in-band interference.	Message ID	JI
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`JAMINF`	Title of the message.
2	`({%C,%S})`, where	Receiver can detect interference signals within the following five bands: GPS L1, GPS L2, GLONASS L1, GLONASS L2, and L5. Thus five consecutive pairs, each in the format `char,string`, where the character value specifies the number of the detected jamming signals and the string value indicates the aggregate interference strength for the corresponding band.
	`%C`	Number of detected jamming signals: • _ – Number of jamming signals is unknown • 0 – No jamming signal detected • 1 – One jamming signal detected • 2 – Two jamming signals detected • N – More than two jamming signals detected
	`%S`	Aggregate in-band interference strength: • ____ – No jamming detected • Weak – Slight • Aver – Moderate • High – Strong • Hard – Severe
3	`@%2X`	Checksum

Usage Guidelines

- Jamming Suppressor is available for JPS_3, JPS_4_x, Euro_x, E_GGD, and EGGDT_x receiver board versions, where “x” stands for the board revision number.
- Jamming Suppressor is available for EG3_x boards and NET-G3 receivers starting with firmware 3.3.
- The option _AJM must be enabled. The EG3_x boards and NET-G3 receivers may also require hardware modification.

Example

Two interference signals in the GPS L1 band and one interference signal in the GPS L2 band:
JE062,JAMEXT,GPS_L1,{1574000.3,-106,20},{1578311.8,-147, 5},GPS_L2,{1227999.8,-140, 5},@1E

History

This message was introduced in the first release of firmware.

See Also

N/A

JE – Extended Jamming Suppressor information

JE – Extended Jamming Suppressor information

Description	Attributes	
This message contains additional quantitative information on detected in-band interference.	Message ID	JE
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`JAMEXT`	Title of the message.
2	(`%S({%7.1F,%+3D,%2D,})`), where	String value contains the band descriptor. Band descriptor is followed by a sequence of *jamming info triplets* describing all individual jamming signals detected in the corresponding band. Each particular jamming info triplet describes a separate interference signal. If Jamming Suppressor has detected no jamming signals, the message body will contain the string `none`. If you enable JE message when anti-jamming is off (i.e., when the parameter `/par/ajm/mode` is set to `off`), the message body will contain the string `off`.
	`%S`	The band descriptor (GPS_L1, GPS_L2, GLN_L1 or GLN_L2).
	`%7.1F`	Estimated jamming frequency measured in kHz.
	`%+3D`	Estimated jamming power measured in dBW. Due to the jamming power estimator's latency, this field will contain the string `estm` (not a specific number) if the message is output during the estimator's settling-down period.
	`%2D`	Selected rejection bandwidth measured in kHz (at a -40 dB cut-off level). This parameter can take one of the following discrete values: 5 kHz, 10 kHz, 20 kHz, 40 kHz or 80 kHz.
3	`@%2X`	Checksum

Usage Guidelines

- The JE message will provide information only on the two bands enabled by the parameter `/par/ajm/mode` (see "Anti-jamming" on page 3-242). Recall that the following pairs of bands are supported:
  - GPS/L1+GPS/L2
  - GLONASS/L1+GLONASS/L2
  - GPS/L1+ GLONASS/L1
  - GPS/L2+GLONASS/L2
- Jamming Suppressor is available for JPS_3, JPS_4_x, Euro_x, E_GGD, and EGGDT_x receiver board versions, where "x" stands for the board revision number.
- Jamming Suppressor is available for EG3_x boards and NET-G3 receivers starting with firmware 3.3.
- The option _AJM must be enabled. The EG3_x boards and NET-G3 receivers may also require hardware modification.

Example

Two interference signals in the GPS L1 band and one interference signal in the GPS L2 band:
JE062,JAMEXT,GPS_L1,{1574000.3,-106,20},{1578311.8,-147, 5},GPS_L2,{1227999.8,-140, 5},@1E

History

This message was introduced in the first release of firmware.

See Also

N/A

MS – RTCM 2.x status message

Description

This message describes the status of RTCM 2.x rover station.

Attributes

Message ID	MS
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	RTSMST	Title of the message.
2	%3D	Time, in seconds, elapsed since last message was received (maximum value = 999). Estimated with an accuracy of 1 second.
3	%4D	Number of received messages (between 0001…9999). If no message has been received, this data field contains zero.
4	%4D	Number of corrupt messages (between 0001…9999). If there are no corrupt messages detected, this data field is set to zero.
5	%.2F	Data link quality in percent (0-100).
6	@%2X	Checksum

Usage Guidelines

This message is considered obsolete. Use DL instead.

Example

MS020,RTCMST,999,0000,0000,100.00,@1F

History

This message was introduced in the first release of firmware.

See Also

“DL – Data link status” on page 4-138.

TX – Text message

Description

This message allows the user to view text information derived on the rover end from messages RTCM Type 16, RTCM Type 36 and CMR Type 2.

Attributes

Message ID	TX
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x2

Body

Field	Format	Description
1	`TEXT`	Title of the message.
2	`%1D`	Total number of the messages (0…2). The rest of the message is available only if this number is non-zero. Otherwise the *total number of the messages* value is immediately followed by the checksum.
3	`({%02D/%02D,%02D:%02D:%02D, "%S",%C,%C,%S})`, where	Group of parameters associated with the given text message (note that the total number of such groups is determined by the previous parameter).
	`%02D/%02D`	Date of receiving of the message (MM/DD – month/day).
	`%02D:%02D:%02D`	UTC time of receiving of the message (HH:MM:SS).
	`"%S"`	String value of up to 90 characters.
	`%C`	Decoder identifier: • A – shows that information has been decoded from RTCM messages 23 and 24 • R – shows that information has been decoded from RTCM messages 16 and 36 • T – shows that information has been decoded from RTCM 3.0 • C – CMR decoder • J – TPS decoder
	`%C`	Data link identifier (see “DL – Data link status” on page 4-138).
	`%S`	Reference station identifier.
4	`@%2X`	Checksum

Usage Guidelines

N/A

Example

TX00B,TEXT,0,@BE

History

- This message was introduced in the first release of firmware.
- Two decoder identifiers, A and T, were added in firmware 2.4.

See Also

N/A

RM – Results of RAIM processing

Description	Attributes	
This message contains RAIM output data.	Message ID	RM
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`RAIM`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	`%1D`	RAIM indication: • 0 – no anomalous measurements have been detected (position is valid) • 1 – RAIM is not able to detect anomalous measurements (for example, due to poor geometry or limited number of visible satellites) - position may be badly affected by anomalous measurements • 2 – RAIM has detected and excluded anomalous measurements (position is valid) – Generally speaking, if more than one bad measurement is detected, there is no guarantee that the RAIM has excluded all bad measurements with the specified probability. However, in most cases, RAIM runs properly even if more than one bad measurement has been detected. – If the RAIM indicator is other than 2, this data field is followed by the checksum. • 3 – RAIM is turned off (position may be badly affected by poor measurements).
5	`%D`	Total number of excluded bad measurements.

6	(%C%2D)	IDs of the satellites with bad measurements. The total number of such satellites is determined by the previous parameter. • Navigation system identifier: – G designates GPS satellites – R or F both designate GLONASS satellites. The F flag is used for R until the receiver has determined the satellite's slot number • Satellite system number (or, for GLONASS, satellite frequency channel number): – GPS SV PRN (follows after the G flag) – GLONASS SV slot number (follows after the R flag) or GLONASS SV frequency channel number (follows after the F flag)
7	@%2X	Checksum

Usage Guidelines

N/A

Example

RM017,RAIM,V,102046.00,0,@41

History

This message was introduced in the first release of firmware.

See Also

N/A

NP – Navigation position

Description	Attributes	
This message includes the receiver's navigational and positioning parameters.	Message ID	NP
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`NAVPOS`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	`%1D`	Position computation indicator. If the indicator equals 0, position is valid and the message will contain more meaningful information (see the following data fields). If it is non-zero, position is not valid and this data field will be followed by the checksum. • 0 – Position is valid • 1 – Too many iterations have been made • 2 – Singular matrix • 3 – Not enough data for position computation • 4 – Either or both altitude and speed exceed specified threshold values 5 – PDOP exceeds specified threshold value. See the parameter `/par/pos/pdop` on page 3-278.

5	%C%C	Position (the first symbol) and velocity (the second symbol) computation mode: • A – Autonomous mode • D – Code differential mode • C – RTK positioning with codes • F – RTK positioning with float integers • R – RTK positioning with fixed integers • P – Entered position (i.e., not computed by the receiver)
6	{%2D,%2D}	Number of GPS and GLONASS satellites used in position computation.
7	%S	Reference geodetic datum identifier.
8	%C%2Do%2D'%2.6F"	Latitude: hemisphere (N – northern, S – southern), degrees, minutes and seconds.
9	%C%3Do%2D'%2.6F"	Longitude: hemisphere (E – eastern, W – western), degrees, minutes and seconds.
10	%+5.4F	Altitude above ellipsoid [meters].
11	%C	Geoidal separation indicator: • V – means geoidal separation is valid; • N – means geoidal separation is not valid.
12	%%+.4F	Geoidal separation: the difference between the earth ellipsoid and geoid (mean-sea-level) defined by the reference datum [meters].
13	%.2F	Horizontal dilution of precision (HDOP).
14	%.2F	Vertical dilution of precision (VDOP).
15	%.3F	Horizontal position RMS error [meters].
16	%.3F	Vertical position RMS error [meters].
17	%.4F	Horizontal velocity [km/h].
18	%+.4F	Vertical velocity [km/h].
19	%3.3F	True heading [degrees].
20	%C	Magnetic heading indicator: • V – means magnetic heading is valid; • N – means magnetic heading is not valid.
21	%3.3F	Magnetic heading [degrees].
22	%.3F	Horizontal velocity RMS error [meters/second].
23	%.3F	Vertical velocity RMS error [meters/second].
24	%.2F	Data link quality in percent (0-100).

25	%3D	Time [in seconds] elapsed since last RTCM, CMR or TPS message was received (maximum value = 999). Estimated with an accuracy of 1 second.
26	@%2X	Checksum

Usage Guidelines

N/A

Example

NP0B3,NAVPOS,V,102209.00,0,AA,{10,04},W84,N55o43'20.990715",E037o39'04.090081",+00163.8049,V,+014.4309,0.72,1.01,1.647,2.294,0.0891,+0.0464,304.680,V,296.572,0.016,0.023,100.00,999,@74

History

This message was introduced in the first release of firmware.

See Also

N/A

MP – Position in given map projection or local coordinates

Description

The message describes the receiver position in the specified map projection or local coordinate system.

Attributes

Message ID	MP
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`MAPRJ`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest digits designate seconds).
4	`%1D`	Position computation indicator. If the indicator equals 0, position is valid and the message will contain more meaningful information (see the following data fields). If it is non-zero, position is not valid and this data field will be followed by the checksum. For specific values, see “NP – Navigation position” on page 4-155.
5	`%C`	Grid (or local) coordinates indicator: • V – means that Grid (or local) coordinates are valid • N – means that Grid (or local) coordinates are not valid.
6	`%S`	Grid system ID: • UTM – Universal Transversal Mercator • TM – User-defined Transversal Mercator projection • STER – User-defined Stereographic projection • LOC – Local coordinates.
7	`%2D`	Zone of the grid system (00, if not available).
8	`%+.4F`	Northern component of grid coordinates or X component of local coordinates [meters].
9	`%+.4F`	Eastern component of grid coordinates or Y component of local coordinates [meters].

10	%+5.4F	Altitude above ellipsoid or local horizon (for local coordinates) [meters].
11	@%2X	Checksum

Usage Guidelines

N/A

Example

MP048,MAPRJ,V,102537.00,0,V,UTM,37,+6176019.2041,+415275.3340,+00163.9156,@0A

History

This message was introduced in the first release of firmware.

See Also

N/A

TR – Time residuals

Description

This message is intended for various time transfer applications. It contains information allowing the user to match an external clock to a specific GNSS satellite's time scale.

Attributes

Message ID	TR
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`TIMERES`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest digits designate seconds).
4	`%2D`	Total number of satellites used in position computation.
5	`({%C%2D,%.2F,%D})`, where	Group of parameters associated with the given satellite (note that the total number of such groups is determined by the previous parameter).
	`%C`	Navigation system identifier: • G – designates GPS satellites, • R or F – both designate GLONASS satellites. The F flag is used for R until the receiver has determined the satellite's slot number.
	`%2D`	Satellite system number (or, for GLONASS, satellite frequency channel number): • GPS SV PRN (follows after the G flag), • GLONASS SV slot number (follows after the R flag) or GLONASS SV frequency channel number (follows after the F flag).

5	%.2F	Time residual in nanoseconds for the given satellite. This satellite-specific time correction is defined as V1 - V2, where • V1 – the system time[a] to the receiver time offset as estimated using this particular satellite (i.e. the difference between the satellite's own clock[b] and the receiver clock); • V2 – the system time to the receiver time offset as estimated using all of the satellites belonging to this navigation system. Note that it is V2 that governs PPS signals generated in the receiver.
	%D	Issue of data for satellite ephemeris: • For GPS satellites, this parameter includes IODE (Issue Of Data, Ephemeris). • For GLONASS satellites, this parameter includes Ephemeris reference time t_b (7 LSB) and the indicator (MSB), which is set to 1 if the ephemeris has been updated without a change in t_b. The interpretation of this parameter for GLONASS satellites is subject to change in the future.
6	@%2X	Checksum

Usage Guidelines

N/A

Example

TR0D0,TIMERES,V,103814.00,12,{G21,+0.74,50},{R20,-2.74,55},{G15,-5.14,142},{G03,-3.23,62},
{G26,+1.34,81},{G22,+2.66,2},{G08,+4.49,40},{R04,+15.87,55},{G19,+1.86,84},{R19,-11.20,55},
{G18,-0.21,1},{R06,-3.97,55},@25

History

This message was introduced in the first release of firmware.

See Also

N/A

a. *System time* means GPS and GLONASS system time for GPS and GLONASS satellites, respectively
b. Assuming that the satellite clock has already been corrected to the corresponding system time, either GPS or GLONASS time, by applying the broadcast frequency-time corrections. The reader will notice that these "satellite-specific time residuals" first of all describe such effects as SA, multipath and atmospheric delay, which are all satellite-specific.

TM – Clock offsets and their time derivatives

Description

The message contains clock offsets (the difference between receiver time scale and GNSS system times) and their derivatives.

Attributes

Message ID	TM
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`TIMING`	Title of the message.
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	`%1D`	Position computation indicator. If it equals 0, this message contains more meaningful information (see the following data fields). If this indicator is non-zero, this data field is followed by the checksum. Position is valid only if the position computation indicator is equal to 0. For how to interpret other values, see “NP – Navigation position” on page 4-155.
5	`{%2D,%2D}`	Number of GPS and GLONASS satellites used in position computation.
6	`{{%C`	GPS time parameters indicator: • V – means that GPS time parameters are valid • N – means that GPS time parameters are not valid.
7	`%+.4F`	Receiver time scale - GPS system time [meters].
8	`%+.4F}`	Derivative of (Receiver time scale - GPS system time) [m/s].
9	`{%C`	GLONASS time parameters indicator: • V – means that GLONASS time parameters are valid • N – means that GLONASS time parameters are not valid.
10	`%+.4F`	Receiver time scale - GLONASS system time [meters].

11	%+.4F}}	Derivative of [Receiver time scale - GLONASS system time) [m/s].
12	%.2F	Time dilution of the precision (TDOP).
13	%.3F	Clock offsets RMS error [meters].
14	%.3F	RMS of the derivatives of clock offsets [m/s].
15	%1D	The Improved Timing mode indicator. If it equals 0, this means the Improved Timing mode is turned off. If this parameter is 1 the mode is turned on.
16	@%2X	Checksum

Usage Guidelines

N/A

Example

TM06C,TIMING,V,110600.00,0,A,{08,04},{{V,-79840.4939,-577.8016},{V,-80239.2567,-577.8182}},1.17,2.640, 0.026,0,@3C

History

This message was introduced in the first release of firmware.

See Also

N/A

RP – Reference station parameters

Description

The message contains the reference station parameters such as the station's coordinates, antenna offsets, station ID, etc. These parameters are used in RTK on the rover side.
These parameters are available via RTCM Messages Types 3, 22, 23, 24 or 32, RTCM v3.0, CMR Message Type 1 or TPS Message BI.

Attributes

Message ID	RP
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x2

Body

Field	Format	Description
1	`REFPAR`	Title of the message.
2	`%1D`	Total number of groups containing the reference station parameters. In the current version of the message, this field can be set to 0 or 1. • 0 means that the reference station parameters are not valid. • 1 means that this message contains valid information.
3	`({%S`	Reference geodetic datum identifier.
4	`%C%2Dd%2Dm%2.6Fs`	Latitude: hemisphere (N – northern, S – southern), degrees, minutes and seconds.
5	`%C%3Dd%2Dm%2.6Fs`	Longitude: hemisphere (E – eastern, W – western), degrees, minutes and seconds.
6	`%.4F`	Altitude above ellipsoid [meters].
7	`%C`	Antenna height indicator: • V – means antenna height is valid; • N – means antenna height is not valid.
8	`%.4F`	Antenna height [meters].
9	`%C`	Antenna East/North offsets indicator: • V – means East/North offsets are valid; • N – means East/North offsets are not valid.
10	`%.4F`	Antenna North offset [meters].
11	`%.4F`	Antenna East offset [meters].
12	`%C`	Decoder identifier (see “DL – Data link status” on page 4-138).
13	`%C`	Data link identifier (see “DL – Data link status” on page 4-138).
14	`%S})`	Reference station identifier.

15	@%2X	Checksum

Usage Guidelines

N/A

Example

RP00D,REFPAR,0,@16

History

This message was introduced in the first release of firmware.

See Also

N/A

RK – RTK parameters

Description

The message contains some RTK-related parameters, such as the difference between the base receiver time and the rover receiver time, estimated probability of fixing ambiguities correctly, etc.

Attributes

Message ID	RK
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	RTKPAR	Title of the message.
2	%C	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid.
3	%6.2F	UTC time of the position fix (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	%D	Fixing ambiguity progress, in percentage [].
5	%.1F	Estimated probability of fixing ambiguities correctly [].
6	%.2F	χ^2 for the position fix [].
7	%+.4F	Time shift between the rover and base receiver times [multiplied by the speed of light and presented in meters].
8	%+.4F	Derivative of time offset between the rover and the base [m/s].
9	%.2F	Root-mean-squared single differenced ionosphere error as estimated by the RTK engine [meters].
10	@%2X	Checksum

Usage Guidelines

N/A

Example

RK019,RTKPAR,V,114532.00,N,@19

History

This message was introduced in the first release of firmware.

See Also
N/A

AP – Position covariance matrix

Description	Attributes	
This message is a text version of the message SP.	Message ID	AP
	Length of the Message Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`POSCOV`	Title of the message.
2	`%C`	UTC time indicator: • V – means UTC time is valid. • N – means UTC time is not valid.
3	`%6.2F`	UTC time of the position fix (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	`%C`	Position covariance matrix indicator: • V – means that Position covariance matrix is available. • N – means that Position covariance matrix is not available. If this indicator is N, the indicator is followed by the checksum.
5	`%C`	Position computation mode (see "NP – Navigation position" on page 4-155).
6	`%.3E`	Cov(1,1)
7	`%.3E`	Cov(2,2)
8	`%.3E`	Cov(3,3)
9	`%.3E`	Cov(4,4)
10	`%.3E`	Cov(1,2)
11	`%.3E`	Cov(1,3)
12	`%.3E`	Cov(1,4)
13	`%.3E`	Cov(2,3)
14	`%.3E`	Cov(2,4)
15	`%.3E`	Cov(3,4)

16	@%2X	Checksum

Usage Guidelines

The values of position covariance matrix are available only in RTK mode.

Example

AP019,POSCOV,V,115640.00,N,@C2

History

This message was introduced in the first release of firmware.

See Also

N/A

AB – Baseline length

Description

This message contains the length and coordinates of the reference station-rover baseline.

Attributes

Message ID	AB
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	BASLI	Title of the message.
2	%C	UTC time indicator: • V – means UTC time is valid • N – means UTC time is not valid.
3	%6.2F	UTC time of the position fix (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	%C	Baseline length indicator: • V – means that baseline length is available; • N – means that baseline length is not available. If this indicator is N, the indicator is followed by the checksum.
5	%C	Position computation mode (see “NP – Navigation position” on page 4-155).
6	%.4F	X(ref) - X(rover) [meters]
7	%.4F	Y(ref) - Y(rover) [meters]
8	%.4F	Z(ref) - Z(rover) [meters]
9	%.4F	Position SEP [meters].
10	@%2X	Checksum

Usage Guidelines

N/A

Example

AB019,BASLIN,V,161135.00,N,@F3

History

This message was introduced in the first release of firmware.

See Also

N/A

WS – SBAS status

Description

This message contains the status of locked SBAS satellites.

Attributes

Message ID	WS
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`WSSVST`	Title of the message.
2	`%2D`	Total number of locked SBAS satellites.
3	`({%2D,%2D,%3D,{%2D},%2D})`, where	Data section comprises the parameters described below. The total number of such 5-parameter sections will coincide with the number of locked SBAS SVs.
4	`%2D`	SBAS PRN number.
5	`%2D`	Elevation [degrees].
6	`%3D`	Azimuth [degrees].
7	`%2D`	L1 C/A signal-to-noise ratio [dB*Hz].
8	`%2D`	Channel navigation status (see “Satellite Navigation Status Codes” on page D-1).
9	`@%2X`	Checksum

Usage Guidelines

This message is available for G3-based boards only.

Example

WS038,WSSVST,02,{120,00,000,{35},29},{126,00,000,{39},29},@E6

History

This message was introduced in firmware 3.1.

See Also

N/A

XD – Data from NMEA XDR sentence

Description

This message contains data taken from the NMEA XDR sentence.

Attributes

Message ID	XD
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x2

Body

Field	Format	Description
1	`%02D`	Total number of groups that contain the contents of XDR messages.
2	`({%C,%02D,%02D%02D%02D,%02D%02D%02D,{%C%C,%S}}`), where	Data section comprises the parameters described below.
3	`%C`	Source ID from which the message was received.
4	`%02D`	Message counter for this port.
5	`%02D%02D%02D`	Date of data reception (day, month, year).
6	`%02D%02D%02D`	Time of data reception (hour, minute, second).
7	`%C%C`	Message source identifier (e.g., WI or YX).
8	`%S`	Message data as text.
9	`@%2X`	Checksum

Usage Guidelines

This message is available for all receivers.

Example

N/A

History

This message was introduced in firmware 3.2.

See Also

N/A

PM – Receiver parameters

Description

This message contains the current values of most of the receiver parameters.

Parameters

Message ID	PM
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	0.0
phase (sec)	0.0
count	0
flags	0x37

Body

```
struct RcvParams {

  a1 params[];     // Parameters values

  a1 del[2] = ",@" // Checksum delimiter

  a1 cs[2];        // Checksum formatted as hexadecimal

};
```

Usage Guidelines

- By default, this message will be put at the beginning of a log file. Also, a PM message will be recorded into the log file every time one of the receiver parameters is changed.
- At present, the starting PM messages and the PM messages generated at the time of updating some of the receiver parameters have the following format:

  `PMXXXNAME=VALUE,@CC`, where
  `XXX` denotes the message length,
  `NAME` denotes the parameter name,
  `VALUE` denotes the parameter value,
  `CC` is the hexadecimal checksum.
- The other PM messages have a slightly different format, specifically:

  `PMXXX {VALUES},@CC`, where
  `XXX` denotes the message length,
  `VALUES` denote the values of the parameters for the given group,
  `CC` is the hexadecimal checksum.

Examples

N/A

History

This message was introduced in the first release of firmware.

See Also

N/A

Command Response Messages

The commands you send to the receiver may generate reply messages. The contents of a reply message will depend on what particular command has invoked this reply message.
See “Commands” on page 2-1 for more information about TPS receiver commands.

If a command has been sent successfully (i.e., returns no error), the receiver responds with RE message provided the command identifier (%%) has been used with the command.

If a command is unsuccessful for some reason (invalid command name, invalid parameter in the command, disabled options, etc.), the receiver always responds with ER message. The contents of ER message specify what is wrong with the issued command.

RE – Receiver ACK reply

Description

This message indicates the successful execution of a command.

Parameters

Message ID	RE
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	N/A
phase (sec)	N/A
count	N/A
flags	N/A

Body

```
struct RE {
  a1 reply[]; // ACK reply
};
```

Usage Guidelines

The receiver always generates the message automatically. You cannot modify its schedule specification.

Examples

RE002%%

History

This message was introduced in the first release of firmware.

See Also

N/A

ER – Receiver NAK reply

Description

This message indicates that an error occurred during the handling of a command.

Parameters

Message ID	ER
Length of the Message Body (bytes)	var
Default Scheduling Specification	
period (sec)	N/A
phase (sec)	N/A
count	N/A
flags	N/A

Body

```
struct ER {

  a1 reply[]; // NAK reply with error description

};
```

Usage Guidelines

- The receiver always generates the message automatically. You cannot modify its schedule specification.
- Below is a list of NAK replies that you may see while issuing the commands to the receiver:
  - `wrong command`
  - `1st parameter missed|wrong 1st parameter| 2nd parameter missed|wrong 2nd parameter`
  - `value missed|value out of range`
  - `wrong file name|file open error|file system not mounted|file read error| file already exists|can't create file|can't remove file|broken file| incompatible file version|file write error`
  - `data transfer open|data transfer error`
  - `wrong option name|bad option date|bad option value|option disabled`
  - `terminated by user|wrong receiver id|access denied|operation in progress| no file system device|dup loop|not supported|device is not accessible| wrong hardware version|incompatible file version|can't lock.`

Examples

ER01A%%{2,,wrong 1st parameter}

History

This message was introduced in the first release of firmware.

See Also

N/A

Industry-approved Messages

This section describes NMEA sentences, RTCM ver 2.x and 3 messages, CMR messages, and BINEX records.

NMEA sentences

General NMEA Format. Each NMEA sentence has the following format:

$AACCC,c-c*hh<CR><LF>

$:	Start of the sentence
AACCC:	Address field
	The first 2 characters are the identifier of transmitting terminal.
	The last 3 characters are the identifier of the message type.
c-c:	Data block
*:	Checksum separator
hh:	Checksum
<CR><LF>:	End of sentence (carriage return, line feed)

For each NMEA sentence, a NULL field will be used when one or more parameters are not reliable or not available. This field is described with 2 commas (,,) or a comma and an asterisk (,*) depending on the sentence position.

TOPCON receivers support the following transmission identifiers:

- GP – Global Positioning System (GPS)
- GL – GLONASS
- GN – Global Navigation Satellite System (GNSS)

In general, the transmission identifier notifies the receiving terminal whether the position information included in the sentence is generated by GPS alone, GLONASS alone, or the combination of GPS with GLONASS. Among actual NMEA sentences, however, there are some sentences that do not show the satellite used for positioning.

Fields. Latitude and Longitude

The latitude is shown as DDMM.MMMM and the longitude as DDDMM.MMMM.

DD and DDD represent a degree and MM.MMMM is an arcminute with a decimal point.

Direction

The characters representing the north latitude, south latitude, east longitude, and west longitude are N, S, E, and W, respectively

Time

Time is expressed as UTC in the format of HHMMSS.

HH denotes hours between 00 and 23, MM denotes minutes between 00 and 59, and SS denotes seconds between 00 and 59.

Supported Sentences. Table 4-5 lists NMEA sentences the TPS receivers support.

Table 4-5. Supported NMEA Sentences

Sentences	Description
GGA	Data on time, position, and positioning
GLL	Position and positioning mode
GMP	Map projection fix data
GNS	Data on time, position and positioning of GPS+GLONASS (GNSS)
GRS	Residual error of distance for each satellite
GSA	Operation mode, satellite used, and DOP of GNSS receiver
GST	Statistics of position errors
GSV	Number of satellites, satellite number, elevation angle, azimuthal angle, and SNR
HDT	Heading
RMC	Positioning time, date, position, course, and speed
ROT	Rate of turn
VTG	Course and speed
ZDA	UTC, day, month, year, and local time zone
P_ATT	Attitude parameters

GGA – Global Positioning System fix data

Description	Attributes	
This sentence outputs data on time, position and positioning.	Sentence ID	GGA
	Length of the Sentence Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`$GPGGA`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position fix (first two digits designate hours, the next two designate minutes and the remaining digits designate seconds).
3	`%4.[1-7]F`	Latitude in selected datum (first two digits designate degrees and the remaining digits designate minutes of arc).
4	`%C`	Latitude hemisphere: N – northern, S – southern.
5	`%5.[1-7]F`	Longitude in selected datum (first three digits designate degrees and the remaining digits designate minutes of arc).
6	`%C`	Longitude hemisphere: E – eastern, W – western.
7	`%1D`	GPS quality indicator: • 0 – Fix not available or invalid • 1 – GPS SPS Mode (single point mode), fix valid • 2 – Differential GPS SPS Mode, fix valid • 3 – GPS PPS Mode (single point mode), fix valid • 4 – RTK Fix solution • 5 – RTK Float solution • 6 – Estimated (dead reckoning) mode • 7 – Manual input mode • 8 – Simulator mode
8	`%2D`	Number of satellites used for position computation.
9	`%.2F`	Horizontal dilution of precision (HDOP).
10	`%+5.[1-4]F`	Altitude above geoid in selected datum [meters].
11	`%C`	Symbol M (denote that altitude is in meters).

12	%+.[1-4]F	Geoidal separation: the difference between the earth ellipsoid and geoid defined by the reference datum [meters].
13	%C	Symbol M (denotes that geoidal separation is in meters).
14	%.1F	Age of differential GPS data [seconds].
15	%4D	Differential reference station ID (an integer between 0000 and 1023).
16	*%2X\0D\0A	Checksum

Usage Guidelines

- The transmission identifier of the GGA sentence is always set to GP, regardless of whether positioning is calculated with GPS only, GLONASS only, or a combination of GPS and GLONASS.
- When the receiver uses GPS+GLONASS data for RTK and DGPS positioning, the age of differential GPS data and the differential reference station ID will be shown regarding the GPS data.
  In contrast, when the receiver uses GLONASS data only, the age of differential GPS data and the differential reference station ID will be shown regarding the GLONASS data.
- In general, when operating a GPS/GLONASS receiver, use the GNS sentence rather than the GGA sentence. The GGA sentence is used mainly for GPS only receivers.

Example

$GPGGA,095257.00,5543.3503351,N,03739.0685196,E,1,16,0.76,148.9262,M,14.4309,M,,*51

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GLL – Geographic position – latitude/longitude

Description	Attributes	
This sentence outputs data on the current latitude/longitude and positioning state.	Sentence ID	GLL
	Length of the Sentence Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`$GNGLL`	TALKER Identifier
2	`%4.[1-7]F`	Latitude in selected datum (first two digits designate degrees and the remaining digits designate minutes of arc).
3	`%C`	Latitude hemisphere: N – northern, S – southern.
4	`%5.[1-7]F`	Longitude in selected datum (first three digits designate degrees and the remaining digits designate minutes of arc).
5	`%C`	Longitude hemisphere: E – eastern, W – western.
6	`%6.[0-2]F`	UTC time of position (first two digits designate hours, the next two digits designate minutes and the remaining digits designate seconds).
7	`%C`	Status field: • V – Invalid for all values of positioning system mode indicator • A – Autonomous • D – Differential • P – Precise • R – RTK with fixed integers • F – RTK with floating integers

8	%C	Positioning system mode indicator: • A – Autonomous. Satellite system used in non-differential mode in position fix • D – Differential. Satellite system used in differential mode in position fix • E – Estimated (dead reckoning) mode • M – Manual input mode • S – Simulator mode • N – Data not valid
9	*%2X\0D\0A	Checksum

Usage Guidelines

N/A

Example

$GNGLL,5543.3509666,N,03739.0684602,E,114734.00,A,A*7D

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GMP – GNSS map projection fix data

Description	Attributes	
This sentence contains fix data for single or combined navigation systems in grid (or local coordinates) expressed in the given map projection.	Sentence ID	GMP
	Length of the Sentence Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`$GNGMP`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position fix (first two digits designate hours, the next two digits designate minutes and the rest digits designate seconds).
3	`%S`	Map projection identification.
4	`%S`	Map zone.
5	`%.[1-4]F`	X (Northern) component of grid (or local) coordinates in meters.
6	`%.[1-4]F`	Y (Northern) component of grid (or local) coordinates in meters.
7	`%C%C`	Mode indicator. For details, see “GNS – GNSS fix data” on page 4-185.
8	`%2D`	Total number of satellites used for position computation.
9	`%.2F`	Horizontal dilution of precision (HDOP).
10	`%+5.[1-4]F`	Altitude above geoid in the selected datum [meters].
11	`%+.[1-4]F`	Geoidal separation: the difference between the earth ellipsoid and geoid defined by the reference datum [meters].
12	`%.1F`	Age of differential data [seconds]. For details, see “GNS – GNSS fix data” on page 4-185.
13	`%4D`	Differential reference station ID (this is an integer between 0000 and 1023). For details, see “GNS – GNSS fix data” on page 4-185.
14	`*%2X\0D\0A`	Checksum

Usage Guidelines

N/A

Example

$GNGMP,134433.00,UTM,M37,6176017.8574,415276.4262,AA,15,0.86,151.8513,14.4309,,*60

History

This sentence was introduced in firmware 2.4.

See Also

N/A

GNS – GNSS fix data

Description

This sentence outputs data on time, position, and positioning of GPS+GLONASS (GNSS).

Attributes

Sentence ID	GNS
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNGNS`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position fix (first two digits designate hours, the next two designate minutes and the remaining digits designate seconds).
3	`%4.[1-7]F`	Latitude in selected datum (first two digits designate degrees and the remaining digits designate minutes of arc).
4	`%C,`	Latitude hemisphere: N – northern, S – southern.
5	`%5.[1-7]F`	Longitude in selected datum (first three digits designate degrees and the remaining digits designate minutes of arc).
6	`%C`	Longitude hemisphere: E – eastern, W – western.
7	`%C%C`	Mode indicator: variable length valid character field type with the first two characters currently defined. The first character indicates the use of GPS satellites, the second character indicates the use of GLONASS satellites. The indicators are: • N – No fix. Satellite system not used in position fix, or fix not valid. • A – Autonomous. Satellite system used in non-differential mode in position fix. • D – Differential. Satellite system used in differential mode in position fix. • P – Precise. Satellite system used in precision mode. Precision mode is defined as: no deliberate degradation (such as Selective Availability) and higher resolution code (P-code) is used to compute position fix.

7		• R – Real time kinematic. Satellite system used in RTK mode with fixed integers. • F – Float RTK. Satellite system used in real time kinematic mode with floating integers. • E – Estimated (dead reckoning) mode. • M – Manual input mode. • S – Simulator mode.
8	%2D	Total number of satellites used for position computation.
9	%.2F	Horizontal dilution of precision (HDOP).
10	%+5.[1-4]F	Altitude above geoid in selected datum [meters].
11	%+.[1-4]F	Geoidal separation: the difference between the earth ellipsoid and geoid defined by the reference datum [meters].
12	%.1F	Age of differential GPS data [seconds].
13	%4D	Differential reference station ID (an integer between 0000 and 1023).
14	*%2X\0D\0A	Checksum

Usage Guidelines

- When the receiver performs RTK or DGPS positioning using GPS alone or GLONASS alone, one GNS sentence will be output for the positioning result. When the receiver performs RTK or DGPS using GPS+GLONASS, three serial GNS sentences will be output for each positioning result.
- The first one of three serial GNS sentences outputs the majority of information and plays the most important role. The other two sentences output the respective information on GPS and GLONASS (in particular, the number of satellites, the age of differential data, and reference station ID).

Example

$GNGNS,122310.20,3722.425671,N,12258.856215,W,DD,14,0.9,1005.543,6.5,,*74<CR><LF>
$GPGNS,122310.20,,,,,,7,,,,5.2,23*4D<CR><LF>
$GLGNS,122310.20,,,,,,7,,,,3.0,23*55<CR><LF>

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GRS – GNSS range residuals

Description

This sentence contains range residuals and is used to support Receiver Autonomous Integrity Monitoring (RAIM).

Attributes

Sentence ID	GRS
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNGRS`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position fix (first two digits designate hours, the next two designate minutes and the remaining digits designate seconds).
3	`%1D`	Mode: • 0 – residuals were used to calculate the position given in the matching GGA or GNS sentence; • 1 – residuals were recomputed after the GGA or GNS position was computed. Currently, the receiver uses only the first mode (Mode = 0).
4	`(%+.3F)` or `(%+.0F)`	A sequence of range residuals (in meters). Sequence length depends on the number of satellites used in the position solution. Order must match order of satellite ID numbers in GSA. When GRS is used GSA and GSV are generally required. If the range residual exceeds 99.9 meters, then the decimal part is discarded, resulting in an integer (-103.7 becomes -103). The maximum value for this field is 999.
5	`*%2X\0D\0A`	Checksum

Usage Guidelines

- When either GPS or GLONASS is used, the transmission identifier will be set to GP or GL, respectively.
- If GPS and GLONASS are used together, the receiver will create 2 GRS sentences. The first sentence indicates the residual error of distance for GPS and the next sentence indicates the residual error of distance for GLONASS. The transmission identifier of each sentence uses GN, which represents the residual error of distance for GNSS.

Example

$GNGRS,131444.00,0,0.296,0.972,2.302,-0.799,1.914,0.154,-0.104,-0.775,-0.384,,,*7F
$GNGRS,131444.00,0,3.643,3.496,-5.486,-1.588,-3.163,,,,,,,*53

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GSA – GNSS DOP and active satellites

Description

This sentence outputs the operating mode of the GNSS receiver, the satellites used in the position solution reported by the GGA or GNS sentence, and DOP values.

Attributes

Sentence ID	GSA
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNGSA`	TALKER Identifier
2	`%C`	Switching Mode: • M – Manual, forced to operate in 2D or 3D mode • A – Automatic, allowed to automatically switch 2D/3D
3	`%D`	Positioning Mode: • 1 – Fix not available • 2 – 2D • 3 – 3D
4	`(%2D)`	A sequence of satellite ID numbers. Sequence length is variable (depends on the amount of satellites used in solution). 1...32: GPS PRN numbers – 1...32: NMEA satellite ID numbers 1...24: GLONASS slot numbers – 65...68: NMEA satellite ID numbers
5	`%.2F`	Position dilution of precision (PDOP).
6	`%.2F`	Horizontal dilution of precision (HDOP).
7	`%.2F`	Vertical dilution of precision (VDOP).
8	`*%2X\0D\0A`	Checksum

Usage Guidelines

- When either GPS or GLONASS is used, the transmission identifier will be set to GP or GL, respectively.
- If GPS and GLONASS are used together, the receiver will create 2 GSA sentences. The first and second sentences relate to the GPS and GLONASS satellites, respectively. Both the sentences, however, will have the same TALKER ID, GN, and the same DOP values (the latter are actually computed for the combined constellation). The TALKER ID GN indicates that this pair of sentences relate to the one and the same GNSS solution.

Example

$GNGSA,M,3,01,18,19,11,03,09,14,28,22,,,,1.36,0.74,1.14*16
$GNGSA,M,3,74,87,65,72,,,,,,,,,1.36,0.74,1.14*19

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GST – GNSS pseudorange error statistics

Description

This sentence outputs the statistics of position errors used to support RAIM.

Attributes

Sentence ID	GST
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNGST`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position (first two digits designate hours, the next two designate minutes and the remaining digits designate seconds).
3	`%.3F`	Estimated standard deviation of the range input's error. "SV Range input", which is used in the navigation process, includes this satellite's pseudo-range and the corresponding DGNSS correction [meters].
4	`%.3F`	Semi-major axis of error ellipse [meters].
5	`%.3F`	Semi-minor axis of error ellipse [meters].
6	`%.3F`	Orientation of semi-major axis of error ellipse [degrees from true north].
7	`%.3F`	RMS latitude error [meters].
8	`%.3F`	RMS longitude error [meters].
9	`%.3F`	RMS altitude error [meters].
10	`*%2X\0D\0A`	Checksum

Usage Guidelines

When the solution computed by the receiver is not RTK fixed or RTK float, fields 3, 4 and 5 are null fields.

Example

$GNGST,131125.00,2.944,,,,1.208,1.121,2.439*6D

History

This sentence was introduced in the first release of firmware.

See Also

N/A

GSV – GNSS satellites in view

Description

This sentence outputs the number of satellites, satellite number, elevation angle, azimuthal angle, and SNR.

Attributes

Sentence ID	GSV
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GPGSV`	TALKER Identifier
2	`%1D`	Total number of messages, 1 to 3.
3	`%1D`	Sentence number, 1 to 3.
4	`%2D`	Total number of satellites in view.
5	`(%2D,%2D,%3D,%2D)`	Satellite ID number (see GSA for ID numbers), elevation in degrees, azimuth in degrees and C/A signal-to-noise ratio (SNR) in dB*Hz.
6	`*%2X\0D\0A`	Checksum

Usage Guidelines

- The number of sets consisting of satellite number, elevation angle, azimuthal angle, and SNR is variable (up to four sets for each sentence).
- If the number of satellites exceeds four, multiple sentences will be output. The first field indicates the total number of sentences and the second field indicates the order of this sentence. GPS and GLONASS sentences are separately output. The identifier of the GPS sentence is GP, and of the GLONASS is GL.
- The number of satellites may exceed 12; however, the NMEA standard specifies that only three sentences are permitted for one epoch and that data on maximum 12 satellites can be output. If the number of satellites exceeds 12, some satellites may not be included in the epochs for which the GSV sentence has been given.

Example

One epoch with the GSV sentences:
$GPGSV,3,1,10…<CR><LF>
$GPGSV,3,2,10…<CR><LF>
$GPGSV,3,3,10…<CR><LF>
$GLGSV,2,1,7…<CR><LF>
$GLGSV,2,2,7…<CR><LF>

History

This sentence was introduced in the first release of firmware.

See Also

N/A

HDT – Heading, True

Description

This sentence outputs the direction.

Attributes

Sentence ID	HDT
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNHDT`	TALKER Identifier
2	`%.3F`	True Heading in degrees.
3	`%C`	Symbol T indicates true heading.
4	`*%2X\0D\0A`	Checksum

Usage Guidelines

N/A

Example

$GNHDT,273.928,T*2E

History

This sentence was introduced in the first release of firmware.

See Also

N/A

RMC – Recommended minimum specific GNSS data

Description

This sentence outputs time, date, position, course, and speed data provided by a GNSS navigation receiver.

Attributes

Sentence ID	RMC
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNRMC`	TALKER Identifier
2	`%6.[0-2]F`	UTC time of position fix (first two digits designate hours, the next two designate minutes and the remaining digits designate seconds).
3	`%C`	Position state: • A – Data valid • V – Navigation receiver warning
4	`%4.[1-7]F`	Latitude in selected datum (first two digits designate degrees and the remaining digits designate minutes of arc).
5	`%C`	Latitude hemisphere: N – northern, S – southern.
6	`%5.[1-7]F`	Longitude in selected datum (first three digits designate degrees and the remaining digits designate minutes of arc).
7	`%C`	Longitude hemisphere: E – eastern, W – western.
8	`%.[1-4]F`	Speed over ground (horizontal speed) [knots].
9	`%3.[1-3]F`	Course over ground (true course) [degrees].
10	`%S`	Date (day/month/year).
11	`%3.[1-3]F`	Magnetic variation [degrees].
12	`%C`	Magnetic variation direction: E – eastern, W – western.

13	%C	Positioning mode • A – Autonomous. Satellite system used in non-differential mode in position fix. • D – Differential. Satellite system used in differential mode in position fix. • E – Estimated (dead reckoning) mode. • M – Manual input mode. • S – Simulator mode. • N – Data not valid.
14	*%2X\0D\0A	Checksum

Usage Guidelines

N/A

Example

$GNRMC,084937.00,A,5543.3499436,N,03739.0689535,E,0.1487,174.939,110208,8.103,W,A*08

History

This sentence was introduced in the first release of firmware.

See Also

N/A

ROT – Rate of turn

Description

This sentence outputs rate of turn.

Attributes

Sentence ID	ROT
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNROT`	TALKER Identifier
2	`%.3F`	Rate of turn, degrees/minute, "-" = bow turns to port else bow turns to starboard.
3	`%C`	• A – data valid • V – data invalid
4	`*%2X\0D\0A`	Checksum

Usage Guidelines

This message is available if heading mode is turned on (i.e. `/par/pos/pd/hd/mode` is set to `on`).

Example

$GNROT,,V*16

History

This sentence was introduced in the first release of firmware.

See Also

N/A

VTG – Course over ground and ground speed

Description

This sentence outputs the travelling direction and velocity.

Attributes

Sentence ID	VTG
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNVTG`	TALKER Identifier
2	`%3.[1-3]F`	True course [degrees].
3	`%C`	Symbol T indicates True course.
4	`%3.[1-3]F`	Magnetic course [degrees].
5	`%C`	Symbol M indicates Magnetic course.
6	`%.[1-4]F`	Horizontal speed [knots].
7	`%C`	Symbol N indicates that horizontal speed is given in knots.
8	`%.[1-4]F`	Horizontal speed [km/h].
9	`%C`	Symbol K indicates that horizontal speed is given in km/h.
10	`%C`	Positioning system mode indicator • A – Autonomous. Satellite system used in non-differential mode in position fix. • D – Differential. Satellite system used in differential mode in position fix. • E – Estimated (dead reckoning) mode. • M – Manual input mode. • S – Simulator mode. • N – Data not valid.
11	`*%2X\0D\0A`	Checksum

Usage Guidelines

N/A

Example

$GNVTG,079.161,T,071.052,M,0.0354,N,0.0656,K,A*33

History

This sentence was introduced in the first release of firmware.

See Also

N/A

ZDA – Time & Date

Description

This sentence outputs UTC, day, month, year, and local time zone.

Attributes

Sentence ID	ZDA
Length of the Sentence Body (bytes)	var
Default Scheduling Specification	
period (sec)	1.0
phase (sec)	0.0
count	0
flags	0x0

Body

Field	Format	Description
1	`$GNZDA`	TALKER Identifier
2	`%6.[0-2]F`	UTC time (first two digits designate hours, next two digits designate minutes and the remaining digits designate seconds).
3	`%2D`	Day (varies between 01…31).
4	`%2D`	Month (varies between 01…12).
5	`%4D`	Year.
6	`%2D`	Local zone hours (varies from -13 to +13).
7	`%2D`	Local zone minutes (varies from 00 to 59).
8	`*%2X\0D\0A`	Checksum

Usage Guidelines

- The local time zone hour and minute refers to the difference from UTC.
- To specify values of local zone hours and local zone minutes, use parameter “Local time zone (offset)” on page 3-291.

Example

$GNZDA,095644.00,11,02,2008,00,00*7A

History

This sentence was introduced in the first release of firmware.

See Also

N/A

ATT – Attitude parameters

Description	Attributes	
This sentence outputs attitude parameters.	Sentence ID	P_ATT
	Length of the Sentence Body (bytes)	var
	Default Scheduling Specification	
	period (sec)	1.0
	phase (sec)	0.0
	count	0
	flags	0x0

Body

Field	Format	Description
1	`$PTPSR`	TALKER Identifier
2	`%C`	UTC time indicator: • V – means that UTC time is valid • N – means that UTC time is not valid
3	`%6.2F`	UTC time of position (the first two digits designate hours, the next two digits designate minutes and the rest of the digits designate seconds).
4	`%.3F`	True Heading in degrees.
5	`%.3F`	Pitch in degrees.
6	`%C`	Base position type: • N – not available • A – autonomous • D – code differential • F – RTK float • R – RTK fixed This field can be useful provided TPS messages are used for broadcasting RTK data.
7	`*%2X\0D\0A`	Checksum

Usage Guidelines

- This is a TPS proprietary NMEA sentence having the following format: `$PTPSR,MsgID,c--c*hh<CR><LF>`
- To enable ATT sentence, use the command as shown in the following example: `em,,nmea/P_ATT`

Example

$PTPSR,ATT,V,140415.00,,,*69

History

This sentence was introduced in the first release of firmware.

See Also

N/A

RTCM 2.x Messages

RTCM (Radio Technical Commission For Maritime Services) SC-104 (Special Committee 104) has developed a standard for differential GNSS (Global Navigation Satellite Systems) service. For details, see RTCM recommended standards for differential GNSS (Global Navigation Satellite System) service, version 2.3, August 20, 2001. (RTCM PAPER 136-2001/SC104-STD).

The standard formulates recommendations in the following areas:

1. Data message and format – The message elements that make up the corrections, the status messages, the station parameters and ancillary data are defined in some details.
2. User interface – A standard interface is defined which enables a receiver to be used in concert with a variety of different data links.

The formats both GPS/GLONASS RTK data and GPS/GLONASS differential corrections are defined as well as the formats of the messages, which includes, e.g., reference station parameters, special message etc.

This standard is used, for example, for broadcasting the differential corrections by the radio-beacon based differential services located near coastal waters all over the world.

Every RTCM message consists of a variable number of 30-bit words (the reader will notice some resemblance to the structure of the GPS Navigation Message). The first two words in any RTCM message serve as the message header. The header comprises the following data fields: preamble, message type, reference station ID, modified Z-count, sequence number, length of frame and station health. The contents of the other words will depend on the type of the message.

List of RTCM 2.x Messages Supported by TPS Receiver

Table 4-6 on page 204 contains a list of RTCM messages currently supported by TPS receivers.

The column Type indicates the message type number as specified in the RTCM standard.

The second column shows the message's TPS-specific identifier (or simply ID). When you enable a particular RTCM message with an appropriate command, you specify the message ID, not the message type.

The third column, Title, describes the message contents.

The fourth column, Period, specifies the default periods of the corresponding RTCM messages.

The last column explains how to calculate an RTCM message's length in 30-bit words. Note that according to the RTCM standard it takes the receiver five bytes to transmit a 30-bit word. This is because the two MSB in each of these five bytes are reserved (more precisely, set to some pre-defined values). Thus the following formula to compute the actual length of an RTCM message [in bits]:

[Length in bits] = [Length in 30-bit words] * 5 * 8

In this table, N designates the total number of satellites and S designates the length (in characters) of the user-specified text.

Table 4-6. List of Supported RTCM 2.x Messages

Type	ID	Title	Period [s]	Length in 30-bit words
1	1	Differential GPS corrections	1	2+N*2-N/3
3	3	GPS reference station parameters	10	2+4
6	6	GPS null frame	30	2
9	9	GPS partial correction set	1	2+N*2-N/3
15	15	Ionospheric delay message	60	2+N*2-N/2
16	16	GPS special message	30	2+1+(S-1)/3
18	18	RTK uncorrected carrier phases	1	3+N*2 for L1 C/A or L1 P data 2*(3+N*2) for L1+L2 data
19	19	RTK uncorrected pseudoranges	1	3+N*2 for L1 C/A or L1 P data 2*(3+N*2) for L1+L2 data
20	20	RTK carrier phase corrections	1	3+N*2 for L1 C/A or L1 P data 2*(3+N*2) for L1+L2 data
21	21	RTK/High accuracy pseudorange corrections	1	3+N*2 for L1 C/A or L1 P data 2*(3+N*2) for L1+L2 data
22	22	Extended reference station parameters	10	(2+1) to (2+3)
23	23	Antenna Type Definition Record	10	2+1+[N_{ant}+N_{ser}], where • N_{ant} – number of characters for antenna descriptor • N_{ser} – number of characters for serial number
24	24	Reference Station Antenna Reference Point Parameter	10	• 5 (if the antenna height is not included) • 6 (if the antenna height is included)
31	31	Differential GLONASS corrections	1	2+N*2-N/3
32	32	GLONASS reference station parameters	10	2+4
34	34	GLONASS partial correction set	1	2+N*2-N/3
34	65	GLONASS null frame (34th message with N=0)	30	2
36	36	GLONASS special message	30	2+1+(S-1)/3

RTCM 3.x Messages

The TPS receivers support the most recent versions 3.0 and 3.1 of binary messages developed by the Radio Technical Commission for Maritime Services Special Committee 104 for differential applications. For details, see RTCM recommended standards for differential GNSS (Global Navigation Satellite System) service, version 3.0, February, 10, 2004. (RTCM PAPER 30-2004/ SC104-STD).

The following RTCM 3.0 and 3.1 messages are supported:

- 1003 – GPS Basic RTK, L1 & L2
- 1004 – GPS Extended RTK, L1 & L2
- 1005 – Stationary Antenna Reference Point without Height Information
- 1006 – Stationary Antenna Reference Point with Height Information
- 1007 – Antenna Descriptor
- 1008 – Antenna Descriptor & Serial Number
- 1011 – GLONASS Basic RTK, L1 & L2
- 1012 – GLONASS Extended RTK, L1 & L2
- 1019 – GPS ephemeris (in accordance with current RTCM documents)
- 1020 – GLONASS ephemeris (in accordance with current RTCM documents)
- 4087G – Fugro proprietary message containing GPS ephemerides
- 4087N – Fugro proprietary message containing GLONASS ephemerides
- 4091t – Proprietary text message. This message provides a possibility to transmit a text message from the reference receiver to the rover receiver. The third-party receivers cannot use this message.

New message group, `/msg/rtcm3`, has been introduced to comprise RTCM 3.0 and 3.1 related messages.

To enable the output of message 1004 on serial port C with the period 1 second, issue the following command: `em,/dev/ser/c,/msg/rtcm3/1004:1.0`

To enable the output of messages 1019 and 1020 on serial port D with the period 1 second, issue the following command: `em,/dev/ser/d,/msg/rtcm3/{1019,1020}`

CMR Messages

The Compact Measurement Record (CMR) format was developed by Trimble Navigation Limited and now is approved for public use. The format is suitable for communication links that have a minimum of 2400 baud throughput (assuming that only GPS data is used). It provides significant advantages over RTCM 2.x messages (the latter are at least twice as long as compared against their CMR counterparts). It should be noted however that the original version of the CMR format does not allow for GLONASS. Therefore the original format definition needs to be expanded to include GLONASS.

List of CMR Messages Supported by TPS Receivers

Table 4-7 lists CMR messages currently supported by TPS receivers. The column Type indicates the message type as specified in the CMR standard except the message type var(3). TPS defined this type to overcome some limitations existing in the CMR protocol.

The second column shows what TPS specific identifiers (IDs) are assigned to different CMR messages. When enabling a CMR message with an appropriate command, the user specifies its ID, not type.

The third column, Title, describes what kind of data each CMR message contains.

The fourth column, Period, defines default periods for CMR messages.

The last column explains how to calculate the length of a CMR message (in bytes).

In this table, N designates the total number of satellites; S is the length (in characters) of Long Station ID; FREQ takes 1 and 2 for single- and dual-frequency measurements, respectively; Var indicates that the type is subject to change.

Table 4-7. List of CMR Messages Supported by TPS Receivers

Type	ID	Title	Period [s]	Length in bytes
0	0	GPS observables	1	6 + N * (8 + (FREQ-1) * 7)
1	1	Reference station coordinates	10	31
2	2	Reference station description	10	31+S
var (3)	10	GLONASS observables (TPS proprietary message)	1	6 + N * (8 + (FREQ-1) * 7)
+	9	Scrolling Station Information.[a]	1	16

a. This message has been developed for reducing the peak throughput of the CMR format. It is used for CMR message types 1 and 2. Data from this message type is transmitted by "frames". Each frame is 16 bytes in size (7-byte message body plus 9 auxiliary bytes), and is transmitted together with CMR Message Types 0 and 10, which allows considerable reduction of the data link peak load.

BINEX Records

BINEX, for “BINary EXchange”, is an operational binary format standard for GPS/GLONASS/SBAS research purposes. It has been designed to allow encapsulation of all (or most) of the information currently allowed for in RINEX OBS, GPS RINEX NAV, GLONASS RINEX NAV, RINEX MET, IONEX, SP3, SINEX, and so on, plus other GNSS-related data and metadata as encountered, including next-generation GNSS.

This group comprises the following BINEX records and subrecords:

- 00_00 – contains information about the site, monument, marker, reference point, and equipment setup for the collection of GPS, GLONASS, and other GNSS type data and other possible site-related information like meteorological, geophysical, equipment, and so on.
- 01_01 – contains GNSS navigation information for a specific satellite.
- 7E_00 – contains ancillary site data, e.g., site meteorological data (pressure, temperature, humidity), local geophysical data (tilt, strain), and so on.
- 7D_00 – contains state variables, e.g., internal temperature, power, and so on.
- 7F_02 – contains information on GNSS time tag, channel, constellation, observables.
- 7F_03 – contains observation data.
- 7F_04 – contains epochs where the receiver should be tracking, but is unable to track any SVs.

For a detailed information on BINEX specifications, refer to UNAVCO’s Web site at http://binex.unavco.org/.

For parameters used to modify these records, refer to “BINEX Parameters” on page 3-536.

Computing Checksum

This appendix contains algorithms used in TPS receivers for calculating 8-bit and 16-bit checksums.

For commands, the checksum is computed starting with the command's first non-blank byte and ending with the (@) inclusive.

For messages, the checksum is computed starting with the first byte of the message identifier and ending with the byte immediately preceding the checksum field inclusive.

8-Bit Checksum

The 8-bit checksum is computed according to the following algorithm:

```
       typedef unsigned char u1;
       u1 cs(u1 const* buf, int count) {
         enum {
           bits   = 8,
           lShift = 2,
           rShift = bits - lShift
         };
       #define ROT_LEFT(val) ((val << lSshift) | (val >> rShift))
         u1 res = 0;
         while(count--)
           res = ROT_LEFT(res) ^ *buf++;
         return ROT_LEFT(res);
}
```

16-Bit Checksum

The 16-bit checksum is computed according to the following algorithm:

```
typedef unsigned short Crc16;
enum {
  WIDTH = 16, // Width of poly (number of most significant bits).
  POLY = 0x1021, // Poly. Bit #16 is set and hidden.
  BYTE_BITS = 8, // Number of bits in byte.
  TABLE_SIZE = 1 << BYTE_BITS,  // Size of table.
  MSB_MASK = 1 << (WIDTH - 1)   // Mask for high order bit in word.
};

// Table (generated by 'crc16init()'.
static Crc16 table[TABLE_SIZE];

// Initializes the table. Should be called once before the first call to
// 'crc16()'
void crc16init(void)
{
  Crc16 i;
  for(i = 0; i < TABLE_SIZE; ++i) {
    Crc16 val = i << (WIDTH - BYTE_BITS);
    int j;
    for(j = 0; j < BYTE_BITS; ++j)
      val = (val << 1) ^ ((val & MSB_MASK) ? POLY : 0);
    table[i] = val;
  }
}

// Calculates CRC16 of 'cnt' bytes from 'src' and returns result.
// Starting value of CRC16 is supplied by caller in 'crc'
Crc16 crc16(Crc16 crc, void const* src, int cnt)
{
  unsigned char const* s = (unsigned char const*)src;
  while(cnt--)
    crc = (crc << BYTE_BITS) ^
      table[(crc >> (WIDTH - BYTE_BITS)) ^ *s++];
  }
  return crc;
}
```

Set the starting value of CRC16 to zero when you use the crc16() function to compute the checksum of the receiver message.

TPS Data Transfer Protocol

This appendix describes the TPS Data Transfer Protocol (TDTP). The terms *transmitter* and *receiver* are used to name the communicating parties in transfer protocol descriptions. We will call the GNSS receiver the *TPS receiver* in this section to distinguish it from the receiving end of the protocol called receiver.

All multi-byte fields are sent in least significant byte (LSB) first order.

Protocol Description

Data transfer protocol is primarily designed for downloading measurement files from TPS receivers to a host computer. The process of downloading/uploading should be initiated by command(s) sent to the TPS receiver that specifies which file should be transmitted. After a transfer is initiated, parties should use the protocol described here.

The protocol is a fixed-size block protocol with checksum and the ability to re-send a block multiple times if a transmission error occurs.

The format of a single block could be represented by the following C structure:

```
       struct Block {
         u1 type;       // Block type:
                        //  ORDINAL (SOB, 0x02)
                        //  LAST (EOT, 0x04)
                        //  ABORT (#, 0x23)
         // The following fields do not exist for block type 'ABORT'.
         union {
           u2 number;   // Block number for block of type
                        // ORDINAL (0 - based).
           i2 count;    // Number of bytes of data in the
                        // block of type LAST. Value -1
                        // indicates transfer error. In this
                        // case 'data' field of the block
                        // contains a zero-terminated string
                        // describing the error type.
         };
         u1 data[size]; // Data. In the block of type LAST
                        // only the first 'count' bytes are
                        // valid. The rest of the bytes are filled
                        // with zeroes.
         u2 cs;         // Checksum of bytes starting from
                        // the 'type' field up to, but not
                        // including, 'cs' field. It is calculated
                        // through all blocks starting from block 0.
         u1 eob;        // End Of Block marker (EOB, 0x03).
};
```

The transmitter sends blocks of this format in response to the receiver's request. Requests are single byte values. There are three types of requests:

- Request to re-send the latest block (NACK, negative acknowledge, 0x15).
- Request to send next block (ACK, positive acknowledge, 0x06).
- One of the characters in the range [!-/] (ASCII codes [0x21-0x2F]) to abort transfer. The particular value sent denotes which error occurred. The transmitter should ignore the rest of the values.

After the transfer protocol is initiated, the receiver should send NACK request to instruct the transmitter to send the first block (block number zero (0)). If an error occurs in the received block, the receiver could send NACK to instruct the transmitter to re-send the block. After the block is successfully received (including test for data integrity), the receiver sends ACK to instruct the transmitter to send the next block. If ACK or NACK was sent but no data is received from the transmitter, ACK or NACK can be resent. In the latter case there is the possibility of losing synchronization between the transmitter and receiver (i.e. the receiver thinks the transmitter received only one ACK but the transmitter actually received two ACKs). To deal with this situation the block number is implemented. The receiver can detect if synchronization is lost by checking that the block number stored in the received block matches the required block number. The first block sent will have a block number of zero (0) and each following block will be incremented by one (1).

The protocol can be terminated by the following events:

- Transmitter sends a block of type LAST with a positive "count" field. This is a normal end of transfer. In this case the last "count" bytes of data are in the "data" field of the block. The rest of the bytes of the "data" field are filled with zeroes.
- Transmitter sends a block of type LAST with a "count" field equal to "-1" (minus one). This means that an error on the transmitter end has occurred. The "data" field of the block contains a zero-terminated ASCII string describing the error in this case.
- Transmitter sends a block of type ABORT (i.e. sends "#" instead of SOB or EOT). This means that the receiver should immediately terminate its operation.
- Receiver sends a code in the range [0x21-0x2F] instead of ACK or NACK. This means that an error on the receiver end has occurred. In this case the particular value sent denotes the error code.

Computing Checksum

The CRC16 checksum is calculated from the bytes of blocks starting from the field "type" up to but not including the field "cs". The initial value of "cs" is set to zero (0) at the beginning of the transfer session. Each additional block uses the initial value for "cs" obtained from the previous successfully sent block.

Assuming the received block is stored into the buffer named "block", the control sum checking could be done as follows:

```
unsigned char block[1 + 2 + 512 + 2 + 1];
Crc16 crc = 0;
...
/* NOTE: crc isn't reset to 0 at each block. */
crc = crc16(crc, block, 1 + 2 + 512);
if(crc == crcReceived)
  /* Checksum is correct */
else
  /* Checksum is wrong */
```

To achieve this result, the transmitter calculates the "cs" field as follows:

```
unsigned char block[1 + 2 + 512 + 2 + 1];
Crc16 crc = 0;
...
/* NOTE: crc isn't reset to 0 at each block. */
crc = crc16(crc, block, 1 + 2 + 512);
/* Resulting 'crc' is output by transmitter LSB
first. */
```

Notes:

Free-form Events

The free-form events are intended to forward arbitrary text information to any output stream of the receiver without interpreting this information in the receiver. They are used with the `event` command for sending of predefined free-form events to a log file. As a result, a post-processing software will treat the contents of the file taking into account the recorded free-form events.

Preface

The receiver uses the following format for free-form events:

<DATA_NAME>=<value>

where <DATA_NAME> designates one of the following names:

_ANT, _ANH, _DYM, _SIT, _CAN, _SAV, _DSC, _FEA.

<DATA_NAME> determines the physical type of the corresponding event. By convention, all free-form events created by TPS software must start with underscore. To avoid format conflicts, third-party program developers who create their own TPS receiver interfacing software should not use the leading underscore in their own ("non-TPS") <DATA_NAME>s unless these are syntactically and semantically identical to the TPS counterparts.

Currently supported TPS Free-Form Events

_ANT Event (ANTENNA TYPE)

This event indicates TPS antenna type used. The value of an _ANT event is a non-empty string comprising up to 20 ASCII characters. Although this event can contain an arbitrary string, it is highly recommended to use only antenna type names approved and standardized by NGS. For officially adopted antenna type identifiers, visit the GPS Antenna Calibration page on the NGS web site at http://www.ngs.noaa.gov/ANTCAL/.

Examples:

```
_ANT=TOP_CR3_GGD
_ANT=TOP_HIPER_GD
```

_ANH Event (ANTENNA HEIGHT)

The value of an _ANH event has the format <height_value>[s], where height_value is an ASCII representation of the measured antenna height in meters.

Antenna height is expressed as a float in the usual (not scientific) notation so that the integer part is delimited from the fractional one with a dot. The optional s character indicates, if present, that the corresponding value relates to the measured slant. If no optional character is used, this indicates the measured vertical.

Examples:

```
_ANH=2.000
_ANH=1.543s
```

_DYM Event (DYNAMICS)

The value of a _DYM event can be either STATIC or DYNAMIC.

STATIC indicates that the antenna phase center (APC) is fixed at the moment this event is issued (and that the APC will probably remain static for some time after issuing this event, say, until another _DYM event appears).

On the contrary, DYNAMIC indicates that the APC will probably not be static after the event is issued.

Examples:

```
_DYM=STATIC
_DYM=DYNAMIC
```

_SIT Event (SITE)

The value of a _SIT event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use. In the future, however, we plan to relax the above mentioned limitations on _SIT <value>.

By TPS convention, _SIT<value> serves as an occupation (or site) identifier. Note that this kind of information plays a very important part in post-processing applications.

Example.

```
_SIT=P1-34_aBcD
```

_CAN Event (CANCEL)

The value of a _CAN event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use.

A CANCEL event is issued if the user wants the post-processing software to ignore the earlier entered occupation name.

Post-processing software is supposed to take a particular CANCEL event into account if the following two conditions exist:

- the CANCEL event is preceded by at least one SITE event in the log-file;
- the CANCEL event and the latest of the SITE events preceding it have the same value. If it is not the case, the CANCEL event is considered "false" and is discarded.

A CANCEL event, if identified as valid, will instruct the post-processing software to "cancel" the given occupation, i.e., all the data collected over the interval between the corresponding SITE and CANCEL events will not be attributed to any of the measured occupations.

_SAV Event (SAVE)

The value of a _SAV event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use.

Post-processing software is supposed to take a particular SAVE event into account if the following two conditions exist:

- the SAVE event is preceded by at least one SITE event in the log-file;
- the SAVE event and the latest of the SITE events preceding it have the same value. If it is not the case, the SAVE event is considered "false" and is discarded.

SAVE events are intended to notify the post-processing software that the corresponding occupation is finished and that all of the data recorded at this occupation are considered valid.

By TPS convention, a SAVE event, if considered "true", indicates that, as a rule, data recorded after issuing this SAVE event do not belong to the occupation whose name is specified in the field _SAV <value>.

Example.

```
_SAV=P1-34_aBcD
```

_DSC Event (DESCRIPTION)

This event type enables the user to associate a free format comment with the current occupation. Issuing a _DSC event while not within a particular site scope still can be quite useful. Interpreting "out of scope" description events is a software-specific procedure.

A DESCRIPTION event's value can be an arbitrary sequence of 8-bit bytes excluding

- <CR> (0x0D),
- <LF> (0x0A),
- <double quote> " (0x22).

Note that double backslash (i.e. \\) is used to represent backslash itself.

In DESCRIPTION events, it is assumed that all of the bytes whose values are less than 128 as well as the two-byte sequences \r, \n, and \' will represent ASCII characters. However, no assumptions are made regarding other bytes that can be found in this type of event, i.e., interpreting such bytes is software-specific.

_FEA Event (FEATURE)

This event type allows the user to specify a "spatial feature" for an object related to the GIS project being done.

This object can have more than one attribute associated with it so that the field operator will have to do work similar to filling in a questionnaire form.

The general format of _FEA event is as follows:

```
_FEA=EntityName:FieldName|FieldType|Units=ValueAsText
```

where `EntityName=AuthorityID.EntityNameProperly`, `FieldType` and `Units` are optional (by default, "string" and "meter").

There exist four field types: s, i, f, d (string, integer, float, date and time, respectively). The symbol "|"(0X7C) serves as a delimiter.

Also, if a feature event contains a numeric value, it may be necessary for the operator to specify the unit of measurement (meters, seconds, etc.). The letter designating the unit of measurement is put before the equality sign ("="), which is followed by the corresponding numeric value (see the example below).

Example.

Supposing we want to define a few features for a light post while performing our GIS survey. Assume that the following information needs to be entered into the receiver log file:

POST (name of the object that we enter the features for)

Height: 4.2 meters

Material: aluminum

Condition: poor

Age: 99 years

In this example, the four feature events will look as follows:

```
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Height|f|m=4.2
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Material|s=aluminum
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Condition|s=poor
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Age|i|year=99
```

Additional Comments on _CAN and _SAV

For post-processing software, it would be incorrect to consider the free-form events CANCEL and SAVE as the only possible end-of-occupation indicators. In some scenarios, the end of an occupation may coincide with the end of the file, or with the beginning of another occupation (_SIT event), or with some other events. If the situation is especially involved, the post-processing software may need to prompt the user for additional information in order to correctly recognize the occupation boundaries, etc.

Notes:

Satellite Navigation Status Codes

This appendix explains navigation codes available in the binary messages SS and rM and text messages GS and NS.

Codes 0-3 and 45-62 will show, for the given satellite, which raw data measurements have been used for position computation. The rest of the codes will indicate why this satellite has been excluded from position computation.

The notation *Reserved* indicates that the flag might be used in future versions of firmware and you can safely ignore it for the current firmware.

Table D-1 lists the codes available for a TPS receiver.

Table D-1. Satellite Navigation Status Structure

Code	Description
00	L1 C/A code used for absolute position computation
01	L1 P code used for absolute position computation
02	L2 P code used for absolute position computation
03	Ionosphere-free combination used for absolute position computation
04	Measurements unavailable
05	Ephemeris unavailable
06	Unhealthy SV (as follows from operational (=ephemeris) SV health)
07	Time-Frequency parameters from the ephemeris data set may be wrong[a]
08	Initial conditions (position and velocity vectors) from the ephemeris data set may be wrong[a]
09	Almanac SV health indicator unavailable for this satellite[a]

Table D-1. Satellite Navigation Status Structure (Continued)

Code	Description
10	Unhealthy SV [as follows from the almanac SV health indicator][a]
11	Alert flag (from the word “HOW”) is set[b]
12	URA indicates the absence of accuracy prediction for this SV[b]
13	User excluded this SV from position computation
14	User excluded this SV with this frequency channel number from position computation[a]
15	This SV is excluded from solution because its system number is unknown[a]
16	This SV has an elevation lower than the specified mask angle
17	Unable to compute position with the given ephemeris data (for example, too many iterations have been produced for solving Kepler's equation)
18	Ephemeris data is too old
19	This SV does not belong to the constellation the user has selected
20	Differential data from reference station unavailable for given satellite (applicable only when receiver runs in DGPS)
21	Reserved
22	RAIM has detected wrong measurements
23	SNR below specified minimum level
24, 25	Reserved
26	DLL not settled
27	Ionospheric corrections are not received from reference station
28	Coarse code outlier (hundreds of meters or more) has been detected
29	Ephemeris index unavailable (no ephemeris information; similar to code 05)
30	This SV is not used in RTK processing (similar to code 20 but is used specifically for RTK)
31	This SV is excluded from RTK processing because a wrong combination of measurements has been detected.
32	This SV is excluded from OmniSTAR HP position computation
33-50	Reserved
51	L1 C/A slot is used in RTK processing
52	L1 P slot is used in RTK processing
53	L2 P slot is used in RTK processing

Table D-1. Satellite Navigation Status Structure (Continued)

Code	Description
54	L1 P & L2 P measurements are used in RTK processing
55	L1 C/A & L2 P measurements are used in RTK processing
56	This SV is used for OmniSTAR HP position computation
57-62	Reserved
63	Satellite navigation status is undefined

a. GLONASS only
b. GPS only

Notes:

Receiver Options

Receiver options enable or disable the various functions and features available for a TPS receiver. The options must be loaded onto the receiver to activate the desired functionality.

Topcon Positioning Systems issues Option Authorization Files (OAF) to activate the specific options that customers purchase. An OAF allows customers to customize and configure the receiver according to particular needs, thus only purchasing those options needed. The utility software PC-CDU is used to load, or transfer, an OAF to the receiver.

Options Available for a TPS Receiver

Table E-1 lists all options available for a TPS receiver.

Table E-1. Receiver Options

Option Name	Description	Values
_GPS	Enables/disables the use of GPS satellites.	1 – enabled 0 – disabled
_GLO	Enables/disables the use of GLONASS satellites.	0 – enabled 1 – disabled -1 – not supported
_L1_	Enables/disables the receiver to track the L1 carrier frequency and use the C/A-code measurements for position computation.	1 – enabled 0 – disabled
_L2_	Enables/disables the receiver to track the L2 carrier frequency and use the P-code measurements on L1 and L2 for position computation.	1 – enabled 0 – disabled -1 – not supported
CIND	Enables/disables the Cinderella option. This option sets all receiver options to their maximum allowable values for 24 hours every other Tuesday at GPS midnight.	1 – enabled 0 – disabled

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
_POS	Displays the maximum allowable position update rate for autonomous and code differential positioning. Measured in Hertz.	0, 1, 2, 5, 10, 20 0 (zero) means the option is disabled
_RAW	Displays the maximum allowable raw measurement update rate. Measured in Hertz.	0, 1, 2, 5, 10, 20 0 (zero) means the option is disabled
_MEM	Displays the maximum amount of storage space for raw data files. Measured in Mbytes.	-1 – not supported 0...128 – enabled for specified memory space 129...511 – enabled for specified memory space N according with the following equation: N = 128 + (_MEM - 128) * 16
COOP	Enables/disables the use of the Co-Op tracking feature.	1 – enabled 0 – disabled -1 – not supported
_PPS	Enables/disables the receiver to output 1 PPS signal.	0 – no 1 PPS output 1 – single 1 PPS output (PPS A) 2 – dual 1 PPS output
EVNT	Enables/disables the receiver to accept event marker inputs.	0 – no event marker input 1 – single event marker input (Event A) 2 – dual event marker input
_AJM	Enables/disables the receiver to use Jamming Suppressor.	0 – disabled 1 – enabled outside USA 2 – enabled for USA only -1 – not supported
_MPR	Enables/disables the use of a special signal processing technique that significantly reduces the effects of multipath.	1 – enabled 0 – disabled
_FRI	If enabled, allows the receiver to operate with an external high-stability frequency source.	1 – enabled 0 – disabled
_FRO	Enables the receiver to generate a very stable 20 MHz frequency.	1 – enabled 0 – disabled
RS_A	Enables/disables communications through port A and displays the maximum available baud rate at which the receiver communicates with an external device. Measured in Kbaud.	0, 9, 19, 38, 56, 115, 230, 460 0 (zero) means the option is disabled The option displays 115 if no value is loaded.

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
RS_B	Enables/disables communications through port B and displays the maximum available baud rate at which the receiver communicates with an external device. Measured in Kbaud.	0, 9, 19, 38, 56, 115, 230, 460 0 (zero) means the option is disabled
RS_C	Enables/disables communications through port C and displays the maximum available baud rate at which the receiver communicates with an external device. Measured in Kbaud.	0, 9, 19, 38, 56, 115, 230, 460. 0 (zero) means the option is disabled
RS_D	Enables/disables communications through port D and displays the maximum available baud rate at which the receiver communicates with an external device. Measured in Kbaud.	0, 9, 19, 38, 56, 115, 230, 460. 0 (zero) means the option is disabled
RAIM	Enables/disables Receiver Autonomous Integrity Monitoring.	1 – enabled 0 – disabled -1 – not supported
_DTM	Enables/disables the receiver to output position information in a datum other than WGS 84 or PE-90.	1 – enabled 0 – disabled
MAGN	Enables/disables the receiver to output the magnetic declination.	1 – enabled 0 – disabled
_GEO	Enables/disables the receiver to use an embedded low-accuracy geoid model with RMS of about 3 m.	1 – enabled 0 – disabled
_WPT	Not used. Retained for future use.	always disabled
WAAS	Enables/disables the receiver to use the SBAS service.	1 – enabled 0 – disabled
OMNI	Enables/disables the receiver to use the OmniSTAR VBS satellite differential service. This option applies to the Map-R and Map-RT receivers. This option requires subscription to the OmniSTAR VBS service.	1 – enabled 0 – disabled -1 – not supported
CDIF	Enables/disables the receiver to run in DGPS (Code Differential) mode.	1 – enabled 0 – disabled
PDIF	Enables/disables the receiver to run in RTK mode at a specified position update rate. Values are given in Hertz.	0 – disabled 1 – enabled @ 1 Hz 2 – enabled @ 2 Hz 5 – enabled @ 5 Hz 10 – enabled @ 10 Hz 20 – enabled @ 20 Hz -1 – RTK not supported

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
RTMO	Allows the receiver to output RTCM 2.x messages. This is a bit-field option represented as decimal.	0 – RTCM 2.x output disabled 1 (bit #0 is set) – DGPS RTCM 2.x output enabled 2 (bit #1 is set) – RTK RTCM 2.x output enabled 3 (bits #1 and #2 are set) – DGPS&RTK RTCM 2.x output enabled
RTMI	Allows the receiver to accept RTCM 2.x messages via a certain number of ports. The value represents the number of ports from which the receiver is able to simultaneously accept RTCM 2.x messages.	0 – RTCM 2.x input disabled 1, 2, 3, 4, 5 – RTCM 2.x input enabled on the indicated number of ports
CMRO	Allows the receiver to output CMR messages. This is a bit-field option represented as decimal.	0 – CMR output disabled 1 (bit #0 is set) – CMR output enabled
CMRI	Allows the receiver to accept CMR messages. The value represents the number of ports from which the receiver is able to simultaneously accept CMR messages.	0 – CMR input disabled 1 – CMR input enabled on a port -1 – CMR input not supported
_LIM	Enables/disables the receiver to work beyond COCOM restriction.	0 – always disabled
_CPH	Enables/disables true carrier phase output. • If the option is enabled, true carrier phase is output. • If the option is disabled, integral Doppler is output for true carrier phase. In this case the PDIF option will not be fully available because only float solutions can be obtained when RTK using integral Doppler for true carrier phase.	1 – enabled 0 – disabled
ETHR	Enables/disables the receiver to communicate over Ethernet. This option applies only to the receivers with Ethernet port.	1 – enabled 2 – disabled -1 – not supported
_USB	Enables/disables communications through the USB port.	0 – disabled 1 – enabled
OCTO	Enables/disables the receiver to act as an attitude determination unit. Retained for future use.	0 – disabled 1 – enabled for heading and pitch computation 2 – enabled for heading, pitch, and roll computation
AUTH	Enable/disables the receiver to be authorized to work with a number of programs. This is a bit-field option represented as decimal.	0 – not authorized 1 (bit #0 is set) – authorized for Pinnacle 2 (bit #1 is set) – authorized for PC-CDU MS A combination of these values is also available.

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
JPSO	Allows the receiver to output TPS messages. This is a bit-field option represented as decimal.	0 – TPS output disabled 1 (bit #0 is set) – TPS output enabled
JPSI	Allows the receiver to accept TPS messages via a certain number of ports. The value represents the number of ports from which the receiver is able to simultaneously accept TPS messages.	0 – TPS input disabled 1, 2, 3, 4, 5 – TPS input enabled on the indicated number of ports
_TCP	Enables/disables the receiver to establish a certain number (5 max.) of simultaneous Telnet-like connections.	0 – disabled 1, 2, 3, 4, 5 – number of enabled Telnet connections
_FTP	Enables/disables the receiver to establish an FTP connection (1 maximum).	0 – disabled 1 – enabled -1 – not supported
DIST	Determines the maximum allowed distance in hundreds of meters between the reference and rover stations. If the distance between the reference and rover stations exceeds the specified limit, the receiver will not provide the RTK position.	0 – disabled 511 – enabled without restriction 1...510 – enabled for the specified distance (for example, 40 means 4 km)
CORI	Enables/disables the corresponding serial port to be set to any differential data input mode. This is a bit-field option represented as decimal.	0 – differential data inputs disabled on all serial ports 1 (bit #0 is set) – serial port A enabled for differential data input 2 (bit #1 is set) – serial port B enabled for differential data input 4 (bit #2 is set) – serial port C enabled for differential data input 8 (bit #3 is set) – serial port D enabled for differential data input A combination of these values is also available.
LAT1	Specifies the latitude of the upper left corner of the rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to x90, where x = 0 stands for North hemisphere (positive numbers), x = 1 stands for South hemisphere (negative numbers)

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
LON1	Specifies the longitude of the upper left corner of the rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to 360
LAT2	Specifies the latitude of the lower right corner of the rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to x90, where x = 0 stands for North hemisphere (positive numbers), x = 1 stands for South hemisphere (negative numbers)
LON2	Specifies the longitude of the lower right corner of the rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to 360
L_CS	Checksum of the LAT1, LON1, LAT2, LON2 options. This checksum is computed according to the following algorithm: `L_CS = LAT1 ^ LON1 ^ LAT2 ^ LON2` `if (L_CS == 0) L_CS = 1` If the checksum of the loaded option is not equal to the checksum computed in the receiver, its current value is set to zero and the receiver will not compute the position and output raw data measurements within the specified rectangle.	from 0 to 511 0 – indicates that the receiver is disabled to compute its position and output raw data measurements
LAT3	Specifies the latitude of the upper left corner of the second rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to x90, where x = 0 stands for North hemisphere (positive numbers), x = 1 stands for South hemisphere (negative numbers)
LON3	Specifies the longitude of the upper left corner of the second rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to 360
LAT4	Specifies the latitude of the lower right corner of the second rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to x90, where x = 0 stands for North hemisphere (positive numbers), x = 1 stands for South hemisphere (negative numbers)

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
LON4	Specifies the longitude of the lower right corner of the rectangle area within which the receiver can produce the position information and output measurement data. Measured in degrees.	from 0 to 360
LCS2	Checksum of the LAT3, LON3, LAT4, LON4 options. This checksum is computed according to the following algorithm: `LCS2 = LAT3 ^ LON3 ^ LAT4 ^ LON4` `if (LCS2 == 0) LCS2 = 1` If the checksum of the loaded option is not equal to the checksum computed in the receiver, its current value is set to zero and the receiver will not compute the position and output raw data measurements within the specified rectangle.	from 0 to 511 0 – indicates that the receiver is disabled to compute its position and output raw data measurements
RM3I	Allows the receiver to accept RTCM 3.0 messages. The value represents the number of ports from which the receiver is able to simultaneously accept RTCM 3.0 messages.	0 – RTCM 3.0 input disabled 1 – RTCM 3.0 input enabled on a port.
RM3O	Allows the receiver to output RTCM 3.0 messages. This is a bit-field option represented as decimal.	0 –RTCM 3.0 output disabled 1 (bit #0 is set) – DGPS RTCM 3.0 output enabled 2 (bit #1 is set) – RTK RTCM 3.0 output enabled 3 (bits #1 and #2 are set) – DGPS&RTK RTCM 3.0 output enabled
SEED	Enables/disables the reference receiver to operate in AutoSeed mode.	0 – disabled 1 – enabled
_IMU	Enables/disables the receiver to use an embedded Inertial Measurement Unit (IMU) for determining attitude parameters and outputting smoothed trajectories.	0 – disabled 1 – enabled
OMSV	Enables/disables the receiver to use various levels of OmniSTAR service. This is a bit-field option represented as decimal. This option requires subscription to the enabled OmniSTAR service.	0 – disabled 1 (bit #0 is set) – OmniSTAR VBS enabled 2 (bit #1 is set) – OmniSTAR XP enabled 4 (bit #2 is set) – OmniSTAR HP enabled A combination of these values is also available.

Table E-1. Receiver Options (Continued)

Option Name	Description	Values
TPBN	Enables/disables TruPath specific features. This is a bit-field option represented as decimal.	0 – disabled -1 – not supported 1 (bit #0 is set) – Job management enabled 2 (bit #1 is set) – Nudge enabled 4 (bit #2 is set) – Coverage management enabled 8 (bit #3 is set) – Implement management enabled A combination of these values is also available.
TPPM	Enables/disables wayline modes available for the TruPath. This is a bit-field option represented as decimal.	0 – disabled -1 – not supported 1 (bit #0 is set) – Lines enabled 2 (bit #1 is set) – Contours (curves) enabled 4 (bit #2 is set) – Center pivot enabled A combination of these values is also available.
TPVC	Enables/disables the TruPath to control a vehicle depending on the steering arrangement. This is a bit-field option represented as decimal.	0 – disabled -1 – not supported 1 (bit #0 is set) – Front steering enabled 2 (bit #1 is set) – Rear steering enabled 4 (bit #2 is set) – Track steering enabled 8 (bit #3 is set) – Articulated steering enabled 16 (bit #4 is set) – Four-wheel steering enabled A combination of these values is also available.
_L2C	Enables/disables the receiver to track GPS L2C and GLONASS C/A L2 signals and use the corresponding code and carrier phase observations for position computation.	0 – disabled 1 – enabled -1 – not supported

Obsolete Options

Table E-2 lists obsolete options available for a TPS receiver. They are no longer used in the receiver and remain for compatibility with some software products that check for their availability. The current values of obsolete options CDDB, CDDR, RTKB, and RTKR are set according to the current values of their alternative new options provided that these obsolete options are not loaded. This approach ensures that a third-party software will work properly should it check the status of these obsolete options when only corresponding new options are loaded.

Table E-2. Obsolete Options

CDDB	Enables/disables the receiver to be configured as a DGPS Base.	0 – disabled 1 – enabled
CDDR	Enables/disables the receiver to be configured as a DGPS Rover.	0 – disabled 1 – enabled
RTKB	Enables/disables the receiver to be configured as an RTK Base.	0 – disabled 1 – enabled
RTKR	Enables/disables the receiver to be configured as an RTK Rover with the desired RTK position update rate. Measured in Hertz.	0, 1, 2, 5, 10, 20 0 (zero) means the option is disabled
INFR	Enables/disables communications through an infrared port.	0 – disabled 1 – enabled -1 – not supported
_PAR	Enables/disables communications through a parallel port.	0 – disabled 1 – enabled -1 – not supported
_FRH	Allows use of the internal Spread Spectrum modem in frequency hopping mode.	0 – disabled 1 – enabled -1 – not supported
_DSS	Allow use of the internal Spread Spectrum modem in spread spectrum mode.	0 – disabled 1 – enabled -1 – not supported
_CDU	Enables/disables the receiver to be connected to a CDU-1 device.	0 – disabled 1 – enabled -1 – not supported
_BLT	Enables/disables the receiver to communicate via Bluetooth.	0 – disabled 1 – enabled -1 – not supported

Notes:

Obsolete Parameters and Messages

This appendix describes the obsolete parameters and messages. They are still maintained to ensure compatibility between the firmware and existing programs. None of them, however, are guaranteed to remain functional in the future versions of firmware. It is strongly recommended to replace them with the alternative new parameters or messages (if available) or disable their functionality altogether.

Obsolete Parameters

Table F-1 lists obsolete parameters and their replacements starting with firmware version 2.4. For obsolete parameters in older firmware versions, contact TPS Customer Support.

Table F-1. Obsolete Parameters

Obsolete Parameters	Replacement Parameters/Recommended Actions
Firmware 2.4	
/par/rover/base/ant/got/offs	/par/rover/base/ant/got/m_offs/val
Firmware 2.5	
/par/ref/nbeam//<id>/geo/fix	/par/ref/nbeam/<id>/geo/fix
/par/ref/nbeam//<id>/xyz/fix	/par/ref/nbeam/<id>/xyz/fix
/par/ref/nbeam//<id>/geo/slf	/par/ref/nbeam/<id>/geo/slf

Table F-1. Obsolete Parameters (Continued)

Obsolete Parameters	Replacement Parameters/Recommended Actions
/par/ref/nbeam//<id>/xyz/slf	/par/ref/nbeam/<id>/xyz/slf
/par/ref/nbeam/geo/fix	/par/ref/nbeam//geo/fix
/par/ref/nbeam/xyz/fix	/par/ref/nbeam//xyz/fix
/par/ref/nbeam/geo/slf	/par/ref/nbeam//geo/slf
/par/ref/nbeam/xyz/slf	/par/ref/nbeam//xyz/slf
Firmware 3.0	
/par/pos/filt/mode	Disable functionality
/par/pos/filt/type	Disable functionality
/par/pos/filt/num	Disable functionality
/par/pos/filt/maxdel	Disable functionality
/par/pos/filt/reset	Disable functionality
/par/pos/filt/weight	Disable functionality
Firmware 3.1p2	
/par/ppp/xt	/par/ppp/modem
Firmware 3.2	
/par/lock/pcode	/par/lock/gps/pcode /par/lock/glo/pcode
/par/ant/inp	/par/ant/rcv/inp
/par/ant/curinp	/par/ant/rcv/curinp

Obsolete Messages

Currently, there is only one message in this topic.

Starting with firmware 3.2, binary message EN is obsolete. Use binary message E3 as a replacement.

Maintenance Parameters

Some of the parameters that exist in GRIL are considered as service or troubleshooting parameters. These parameters are primarily intended for TPS factory or support personnel only. Change their values only under the guidance of a TPS technician.

Table G-1. Service and Troubleshooting Parameters

Name	Parameter
Verification of block recording	/par/dev/blk/a/verify
File system version formatted as hexadecimal	/par/dev/blk/a/ver
Processor's clock rate	/par/cpu/frq
Processor load	/par/load
Flash memory access time	/par/rcv/flash/waitstates
ASIC frequency	/par/asic/curfrq
Calculation of ambiguity fixing statistics using full firmware resets	/par/pos/pd/hist/mode
A priori (precise) coordinates of the test baseline	/par/pos/pd/hist/pr/x /par/pos/pd/hist/pr/y /par/pos/pd/hist/pr/z
Probability of fixing ambiguities	/par/pos/pd/hist/out
Total number of fixed solutions	/par/pos/pd/hist/num
Percentage of wrong fixed solutions	/par/pos/pd/hist/bad
Resetting current statistics for ambiguity fixing	/par/hist/reset

Table G-1. Service and Troubleshooting Parameters (Continued)

Name	Parameter
Collecting ambiguity fixing time statistics using RTK engine resets only	/par/rtk/dbi/rest
Decode CMR messages in accordance with firmware 2.2	/par/cmr/base/idxset /par/cmr/rover/idxset
Boot loader version	/par/rcv/ver/boot
Number of bad blocks in receiver memory	/par/dev/blk/a/bad_blocks
List of statistics for the file system buffer	/par/stat/fsb
Size of the file system buffer	/par/stat/fsb/sz
Maximum number of bytes in the file system buffer since last receiver start-up	/par/stat/fsb/max
Current number of bytes in the file system buffer	/par/stat/fsb/cnt
TCP/IP network statistics	/par/net/stat
P-code adaptation parameter	/par/raw/ploops/adapt
Half-integer ambiguity fixing on L1 and/or L2	/par/pos/pd/si
Use of SBAS range rate corrections	/par/waas/1/zrate
Applying tropospheric model when working with SBAS	/par/waas/1/trop

Reference Ellipsoids and Local Datums

Reference Ellipsoids and Local Datums

Information on reference ellipsoids and local datums presented in this Appendix was borrowed from [3]

Ellipsoid ID	Semi-major axis a (meters)	Reciprocal of flattening 1/f	Description
AA	6377563.396	299.3249646	Airy 1830
AN	6378160.0	298.25	Australian National
BR	6377397.155	299.1528128	Bessel 1841 (Ethiopia Indonesia Japan and Korea)
BN	6377483.865	299.1528128	Bessel 1841 (Namibia)
CC	6378206.4	294.9786982	Clarke 1866
CD	6378249.145	293.465	Clarke 1880 (As accepted by DMA)
EB	6377298.556	300.8017	Everest (Brunei and E.Malaysia (Sabah and Sarawak))
EA	6377276.345	300.8017	Everest (India 1830)
EC	6377301.243	300.8017	Everest (India 1956)
EF	6377309.613	300.8017	Everest (Pakistan)

Ellipsoid ID	Semi-major axis a (meters)	Reciprocal of flattening 1/f	Description
EE	6377304.063	300.8017	Everest (W.Malaisia and Singapore 1948)
ED	6377295.664	300.8017	Everest (W.Malaisia 1969)
RF	6378137.0	298.257222101	Geodetic Reference System 1980
HE	6378200.0	298.3	Helmert 1906
HO	6378270.0	297.0	Hough 1960
ID	6378160.0	298.247	Indonesian 1974
IN	6378388.0	297.0	International 1924
KA	6378245.0	298.3	Krassovsky 1940
AM	6377340.189	299.3249646	Modified Airy
FA	6378155.0	298.3	Modified Fischer 1960
SA	6378160.0	298.25	South American 1969
WD	6378135.0	298.26	WGS 1972
WE	6378137.0	298.257223563	WGS 1984
PE	6378136.0	298.257839303	PE-90

List of DATUMS Supported by TPS Receivers

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
0	WGS 1984	W84	WGS84	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	PZ 1990	P90	PZ-90	0.0	0.0	1.0	0.0	0.0	0.0	-0.2062648
2	WGS 1972	W72	WGS72	0.0	0.0	4.5	0.227	0.0	0.0	-0.554
3	ADINDAN	ADI-M (Ethiopia and Sudan)	Clarke1880	-166	-15	204	Zeros			
4	ADINDAN	ADI-E (Burkina Faso)	Clarke1880	-118	-14	218				
5	ADINDAN	ADI-F (Cameroon)	Clarke1880	-134	-2	210				
6	ADINDAN	ADI-A (Ethiopia)	Clarke1880	-165	-11	206				
7	ADINDAN	ADI-C (Mali)	Clarke1880	-123	-20	220				
8	ADINDAN	ADI-D (Senegal)	Clarke1880	-128	-18	224				
9	ADINDAN	ADI-B (Sudan)	Clarke1880	-161	-14	205				
10	AFGOOYE	AFG (Somalia)	Krassovsky1940	-43	-163	45				
11	ARC 1950	ARF-M (Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia and Zimbabwe)	Clarke1880	-143	-90	-294				
12	ARC 1950	ARF-A (Botswana)	Clarke1880	-138	-105	-289				
13	ARC 1950	ARF-H (Burundi)	Clarke1880	-153	-5	-292				
14	ARC 1950	ARF-B (Lesotho)	Clarke1880	-125	-108	-295				
15	ARC 1950	ARF-C (Malawi)	Clarke1880	-161	-73	-317				
16	ARC 1950	ARF-D (Swaziland)	Clarke1880	-134	-105	-295				
17	ARC 1950	ARF-E (Zaire)	Clarke1880	-169	-19	-278				
18	ARC 1950	ARF-F (Zambia)	Clarke1880	-147	-74	-283				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
19	ARC 1950	ARF-G (Zimbabwe)	Clarke1880	-142	-96	-293	Zeros			
20	ARC 1960	ARS-M (Kenya and Tanzania)	Clarke1880	-160	-6	-302				
21	ARC 1960	ARS-A (Kenya)	Clarke1880	-157	-2	-299				
22	ARC 1960	ARS-B (Tanzania)	Clarke1880	-175	-23	-303				
23	AYABELLE LIGHTHOUSE	PHA (Djibouti)	Clarke1880	-79	-129	145				
24	BISSAU	BID (Guinea-Bissau)	International1924	-173	253	27				
25	CAPE	CAP (South Africa)	Clarke1880	-136	-108	-292				
26	CARTHAGE	CGE (Tunisia)	Clarke1880	-263	6	431				
27	DABOLA	DAL (Guinea)	Clarke1880	-83	37	124				
28	EUROPEAN 1950	EUR_F (Egypt)	International1924	-130	-117	-151				
29	EUROPEAN 1950	EUR_T (Tunisia)	International1924	-112	-77	-145				
30	LEIGON	LEH (Ghana)	Clarke1880	-130	29	364				
31	LIBERIA 1964	LIB (Liberia)	Clarke1880	-90	40	88				
32	MASSAWA	MAS (Eritrea (Ethiopia))	Bessel1841	639	405	60				
33	MERCHICH	MER (Morocco)	Clarke1880	31	146	47				
34	MINNA	MIN-A (Cameroon)	Clarke1880	-81	-84	115				
35	MINNA	MIN-B (Nigeria)	Clarke1880	-92	-93	122				
36	M'PORALOKO	MPO (Gabon)	Clarke1880	-74	-130	42				
37	NORTH SAHARA 1959	NSD (Algeria)	Clarke1880	-186	-93	310				
38	OLD EGYPTIAN 1907	OEG (Egypt)	Helmert1906	-130	110	-13				
39	POINT 58	PTB (Burkina Faso and Niger)	Clarke1880	-106	-129	165				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
40	POINTE NOIRE 1948	PTN (Congo)	Clarke1880	-148	51	-291	Zeros			
41	SCHWARZECK	SCK (Namibia)	Bessel1841NAM	616	97	251				
42	SIERRA LEONE 1960	SRL (Sierra Leone)	Clarke1880	-88	4	101				
43	VOIROL 1960	VOR (Algeria)	Clarke1880	-123	-206	219				
44	AIN EL ABD 1970	AIN-A (Bahrain Island)	International1924	-150	-250	-1				
45	AIN EL ABD 1970	AIN-B (Saudi Arabia)	International1924	-143	-236	7				
46	DJAKARTA (BATAVIA)	BAT (Sumatra (Indonesia))	Bessel1841	-377	681	-50				
47	EUROPEAN 1950	EUR-H (Iran)	International1924	-117	-132	-164				
48	HONG KONG 1963	HKD (Hong Kong)	International1924	-156	-271	-189				
49	HU-TZU-SHAN	HTN (Taiwan)	International1924	-637	-549	-203				
50	INDIAN	IND-B (Bangladesh)	Everest Ind1830	282	726	254				
51	INDIAN	IND-I (India and Nepal)	Everest Ind1956	295	736	257				
52	INDIAN 1954	INF-A (Thailand)	Everest Ind1830	217	823	299				
53	INDIAN 1960	ING-A (Vietnam (near 16 degree N))	Everest Ind1830	198	881	317				
54	INDIAN 1960	ING-B (Con Sol Island (Vietnam))	Everest Ind1830	182	915	344				
55	INDIAN 1975	INH-A (Thailand)	Everest Ind1830	210	814	289				
56	INDONESIAN 1974	IDN (Indonesia)	Indonesian1974	-24	-15	5				
57	KANDAWALA	KAN (Sri Lanka)	Everest Ind1830	-97	-787	86				
58	KERTAU 1948	KEA (West Malaysia and Singapore)	Everest WMS1948	-11	851	5				
59	NAHRWAN	NAH-A (Masirah Island (Oman))	Clarke1880	-247	-148	369				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
60	NAHRWAN	NAH-B (United Arab Emirates)	Clarke1880	-249	-156	381	Zeros			
61	NAHRWAN	NAH-C (Saudi Arabia)	Clarke1880	-243	-192	477				
62	OMAN	FAH (Oman)	Clarke1880	-346	-1	224				
63	QATAR NATIONAL	QAT (Qatar)	International1924	-128	-283	22				
64	SOUTH ASIA	SOA (Singapore)	Modified Fisher	7	-10	-26				
65	TIMBALAI 1948	TIL (Brunei and East Malaysia (Sarawak and Sabah))	Everest BEM	-679	669	-48				
66	TOKYO	TOY-M (Japan, Okinawa and South Korea)	Bessel1841	-148	507	685				
67	TOKYO	TOY-A (Japan)	Bessel1841	-146	508	682				
68	TOKYO	TOY-B (South Korea)	Bessel1841	-147	506	687				
69	TOKYO	TOY-C (Okinawa)	Bessel1841	-146	508	682				
70	AUSTRALIAN GEODETIC 1966	AUA (Australia and Tasmania)	Australian	-133	-48	148				
71	AUSTRALIAN GEODETIC 1984	AUG (Australia and Tasmania)	Australian	-134	-48	149				
72	COORDINATE SYSTEM 1937 OF ESTONIA	EST (Estonia)	Bessel1841	374	150	588				
73	EUROPEAN 1950	EUR-M (Mean Solution (Austria, Belgium, Denmark, Finland, France, FRG, Gibraltar, Greece, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden and Switzerland))	International1924	-87	-98	-121				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
74	EUROPEAN 1950	EUR-A (Western Europe (Limited to Austria, Denmark, France, FRG, Netherlands, and Switzerland))	International1924	-87	-96	-120	Zeros			
75	EUROPEAN 1950	EUR-E (Cyprus)	International1924	-104	-101	-140				
76	EUROPEAN 1950	EUR-G (England, Channel Islands, Scotland and Shetland islands)	International1924	-86	-96	-120				
77	EUROPEAN 1950	EUR-K (England, Ireland, Scotland and Shetland islands)	International1924	-86	-96	-120				
78	EUROPEAN 1950	EUR-B (Greece)	International1924	-84	-95	-130				
79	EUROPEAN 1950	EUR-I (Italy (Sardinia))	International1924	-97	-103	-120				
80	EUROPEAN 1950	EUR-J (Italy (Sicily))	International1924	-97	-88	-135				
81	EUROPEAN 1950	EUR-L (Malta)	International1924	-107	-88	-149				
82	EUROPEAN 1950	EUR-C (Norway and Finland)	International1924	-87	-95	-120				
83	EUROPEAN 1950	EUR-D (Portugal and Spain)	International1924	-84	-107	-120				
84	EUROPEAN 1979	EUS (Mean Solution (Austria, Finland, Netherlands, Norway, Spain, Sweden and Switzerland))	International1924	-86	-98	-119				
85	HJORSEY 1955	HJO (Iceland)	International1924	-73	46	-86				
86	IRELAND 1965	IRL (Ireland)	Modified AIRY	506	-122	611				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
87	ORDNANCE SURVEY OF GREAT BRITAIN 1936	OGB-M (Mean solution (England, Isle of Man, Scotland, Shetland Islands and Wales))	AIRY1830	375	-111	431	Zeros			
88	ORDNANCE SURVEY OF GREAT BRITAIN 1936	OGB-A (England)	AIRY1830	371	-112	434				
89	ORDNANCE SURVEY OF GREAT BRITAIN 1936	OGB-B (England, Isle of Man, and Wales)	AIRY1830	371	-111	434				
90	ORDNANCE SURVEY OF GREAT BRITAIN 1936	OGB-C (Scotland and Shetland Islands)	AIRY1830	384	-111	425				
91	ORDNANCE SURVEY OF GREAT BRITAIN 1936	OGB-D (Wales)	AIRY1830	370	-108	434				
92	ROME 1940	MOD (Sardinia)	International1924	-225	-65	9				
93	S-42 (PULKOVO 1942)	SPK-A (Hungary)	Krassovsky1940	28	-121	-77				
94	S-42 (PULKOVO 1942)	SPK-B (Poland)	Krassovsky1940	23	-124	-82				
95	S-42 (PULKOVO 1942)	SPK-C (Czechoslovakia)	Krassovsky1940	26	-121	-78				
96	S-42 (PULKOVO 1942)	SPK-D (Latvia)	Krassovsky1940	24	-124	-82				
97	S-42 (PULKOVO 1942)	SPK-E (Kazakhstan)	Krassovsky1940	15	-130	-84				
98	S-42 (PULKOVO 1942)	SPK-F (Albania)	Krassovsky1940	24	-130	-92				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
99	S-42 (PULKOVO 1942)	SPK-G (Romania)	Krassovsky1940	28	-121	-77	Zeros			
100	S-42 (PULKOVO 1942)	SPK-0 (former Soviet Union (not standard designation))	Krassovsky1940	26.3	-132.6	-76.3	-0.12	-0.22	-0.4	-0.9
101	S-JYSK	CCD (Czechoslovakia (prior to 1 Jan 1993))	Bessel1841	589	76	480	Zeros			
102	CAPE CANAVERAL	CAC (Mean solution (Florida and Bahamas))	Clarke1866	-2	151	181				
103	NORTH AMERICAN 1927	NAS-C (Mean solution (CONUS))	Clarke1866	-8	160	176				
104	NORTH AMERICAN 1927	NAS-B (Western united states: Arizona, Arkansas, California, Colorado, Idaho, Iowa, Kansas, Montana, Nebraska, Nevada, New Mexico, North Dakota, Oklahoma, Oregon, South Dakota, Texas, Utah, Washington and Wyoming)	Clarke1866	-8	159	175				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
105	NORTH AMERICAN 1927	NAS-A (Eastern united states: Alabama, Connecticut, Delaware, District of Columbia, Florida, Georgia, Illinois, Indiana, Kentucky, Louisiana, Maine, Maryland, Massachusetts, Michigan, Minnesota, Mississippi, Missouri, New Hampshire, New Jersey, New York, North Carolina, Ohio, Pennsylvania, Rhode Island, South Carolina, Tennessee, Vermont, Virginia, West Virginia and Wisconsin)	Clarke1866	-9	161	179	Zeros			
106	NORTH AMERICAN 1927	NAS-D (Alaska (excluding Aleutian Islands))	Clarke1866	-5	135	172				
107	NORTH AMERICAN 1927	NAS-V (Aleutian Islands (East of 180W))	Clarke1866	-2	152	149				
108	NORTH AMERICAN 1927	NAS-W (Aleutian Islands (West of 180W))	Clarke1866	2	204	105				
109	NORTH AMERICAN 1927	NAS-Q (Bahamas (excluding San Salvador Island))	Clarke1866	-4	154	178				
110	NORTH AMERICAN 1927	NAS-R (San Salvador Island)	Clarke1866	1	140	165				
111	NORTH AMERICAN 1927	NAS-E (Canada: Mean solution (including Newfoundland))	Clarke1866	-10	158	187				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
112	NORTH AMERICAN 1927	NAS-F (Canada: Alberta and British Columbia)	Clarke1866	-7	162	188	Zeros			
113	NORTH AMERICAN 1927	NAS-G (Eastern Canada: Newfoundland, New Brunswick, Nova Scotia and Quebec)	Clarke1866	-22	160	190				
114	NORTH AMERICAN 1927	NAS-H (Canada: Manitoba and Ontario)	Clarke1866	-9	157	184				
115	NORTH AMERICAN 1927	NAS-I (Canada: Northwest territories and Saskatchewan)	Clarke1866	4	159	188				
116	NORTH AMERICAN 1927	NAS-J (Canada: Yukon)	Clarke1866	-7	139	181				
117	NORTH AMERICAN 1927	NAS-O (Canal zone)	Clarke1866	0	125	201				
118	NORTH AMERICAN 1927	NAS-P (Caribbean: Antigua Island, Barbados, Barbuda, Caicos Islands, Cuba, Dominican Republic, Grand Cayman, Jamaica and Turks Islands)	Clarke1866	-3	142	183				
119	NORTH AMERICAN 1927	NAS-N (Central America: Belize, Costa Rica, El Salvador, Guatemala, Honduras and Nicaragua)	Clarke1866	0	125	194				
120	NORTH AMERICAN 1927	NAS-T (Cuba)	Clarke1866	-9	152	178				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
121	NORTH AMERICAN 1927	NAS-U (Greenland (Hayes Peninsula))	Clarke1866	11	114	195	Zeros			
122	NORTH AMERICAN 1927	NAS-L (Mexico)	Clarke1866	-12	130	190				
123	NORTH AMERICAN 1983	NAR-A (Alaska (excluding Aleutian Islands))	GRS1980	0	0	0				
124	NORTH AMERICAN 1983	NAR-E (Aleutian Islands)	GRS1980	-2	0	4				
125	NORTH AMERICAN 1983	NAR-B (Canada)	GRS1980	0	0	0				
126	NORTH AMERICAN 1983	NAR-C (CONUS)	GRS1980	0	0	0				
127	NORTH AMERICAN 1983	NAR-H (Hawaii)	GRS1980	1	1	-1				
128	NORTH AMERICAN 1983	NAR-D (Mexico and Central America)	GRS1980	0	0	0				
129	BOGOTA OBSERVATORY	BOO (Colombia)	International1924	307	304	-318				
130	CAMPO INCHAUSPE 1969	CAI (Argentina)	International1924	-148	136	90				
131	CHUA ASTRO	CHU (Paraguay)	International1924	-134	229	-29				
132	CORREGO ALEGRE	COA (Brazil)	International1924	-206	172	-6				
133	PROVISIONAL SOUTH AMERICAN 1956	PRP-M (Mean solution (Bolivia, Chile, Colombia, Ecuador, Guyana, Peru and Venezuela))	International1924	-288	175	-376				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
134	PROVISIONAL SOUTH AMERICAN 1956	PRP-A (Bolivia)	International1924	-270	188	-388	Zeros			
135	PROVISIONAL SOUTH AMERICAN 1956	PRP-B (Northern Chile (near 19 S))	International1924	-270	183	-390				
136	PROVISIONAL SOUTH AMERICAN 1956	PRP-C (Southern Chile (near 43 S))	International1924	-305	243	-442				
137	PROVISIONAL SOUTH AMERICAN 1956	PRP-D (Colombia)	International1924	-282	169	-371				
138	PROVISIONAL SOUTH AMERICAN 1956	PRP-E (Ecuador)	International1924	-278	171	-367				
139	PROVISIONAL SOUTH AMERICAN 1956	PRP-F (Guyana)	International1924	-298	159	-369				
140	PROVISIONAL SOUTH AMERICAN 1956	PRP-G (Peru)	International1924	-279	175	-379				
141	PROVISIONAL SOUTH AMERICAN 1956	PRP-H (Venezuela)	International1924	-295	173	-371				
142	PROVISIONAL SOUTH CHILEAN 1963 (also known as HITO XVIII 1963)	HIT (Southern Chile (near 53 S))	International1924	16	196	93				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
143	SOUTH AMERICAN 1969	SAN-M (Mean solution (Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Trinidad and Tobago, and Venezuela))	South American1969	-57	1	-41	Zeros			
144	SOUTH AMERICAN 1969	SAN-A (Argentina)	South American1969	-62	-1	-37				
145	SOUTH AMERICAN 1969	SAN-B (Bolivia)	South American1969	-61	2	-48				
146	SOUTH AMERICAN 1969	SAN-C (Brazil)	South American1969	-60	-2	-41				
147	SOUTH AMERICAN 1969	SAN-D (Chile)	South American1969	-75	-1	-44				
148	SOUTH AMERICAN 1969	SAN-E (Colombia)	South American1969	-44	6	-36				
149	SOUTH AMERICAN 1969	SAN-F (Ecuador (excluding Galapagos Islands))	South American1969	-48	3	-44				
150	SOUTH AMERICAN 1969	SAN-J (Baltra, Galapagos Islands)	South American1969	-47	26	-42				
151	SOUTH AMERICAN 1969	SAN-G (Guyana)	South American1969	-53	3	-47				
152	SOUTH AMERICAN 1969	SAN-H (Paraguay)	South American1969	-61	2	-33				
153	SOUTH AMERICAN 1969	SAN-I (Peru)	South American1969	-58	0	-44				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
154	SOUTH AMERICAN 1969	SAN-K (Trinidad and Tobago)	South American1969	-45	12	-33	Zeros			
155	SOUTH AMERICAN 1969	SAN-L (Venezuela)	South American1969	-45	8	-33				
156	ZANDERIJ	ZAN (Suriname)	International1924	-265	120	-358				
157	ANTIGUA ISLAND ASTRO 1943	AIA (Antigua, Leeward Islands)	Clarke1880	-270	13	62				
158	ASCENSION ISLAND 1958	ASC (Ascension Island)	International1924	-205	107	53				
159	ASTRO DOS 71/4	SHB (St. Helena Island)	International1924	-320	550	-494				
160	BERMUDA 1957	BER (Bermuda Islands)	Clarke1866	-73	213	296				
161	DECEPTION ISLAND	DID (Deception Island, Antarctica)	Clarke1880	260	12	-147				
162	FORT THOMAS 1955	FOT (Nevis, St. Kitts, Leeward Islands)	Clarke1880	-7	215	225				
163	GRACIOSA BASE SW 1948	GRA (Faial, Graciosa, Pico, Sao Jorge and Terceira Islands (Azores))	International1924	-104	167	-38				
164	ISTS 061 ASTRO 1968	ISG (South Georgia Islands)	International1924	-794	119	-298				
165	L.C.5 ASTRO 1961	LCF (Cayman Brac Island)	Clarke1866	42	124	147				
166	MONTSERRAT ISLAND ASTRO 1958	ASM (Montserrat, Leeward Islands)	Clarke1880	174	359	365				
167	NAPARIMA, BWI	NAP (Trinidad and Tobago)	International1924	-10	375	165				
168	OBSERVATORIO METEOROLOGICO 1939	FLO (Corvo and Flores Islands (Azores))	International1924	-425	-169	81				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
169	PICO DE LAS NIEVES	PLN (Canary Islands)	International1924	-307	-92	127	Zeros			
170	PORTO SANTO 1936	POS (Porto Santo and Madeira Islands)	International1924	-499	-249	314				
171	PUERTO RICO	PUR (Puerto Rico and Virgin Islands)	Clarke1866	11	72	-101				
172	QORNOQ	QUO (South Greenland)	International1924	164	138	-189				
173	SAO BRAZ	SAO (Sao Miguel, Santa Maria Islands (Azores))	International1924	-203	141	53				
174	SAPPER HILL 1943	SAP (East Falkland Island)	International1924	-355	21	72				
175	SELVAGEM GRANDE 1938	SGM (Salvage Islands)	International1924	-289	-124	60				
176	TRISTAN ASTRO 1968	TDC (Tristan da Cunha)	International1924	-632	438	-609				
177	ANNA 1 ASTRO 1965	ANO (Cocos Islands)	Australian	-491	-22	435				
178	GAN 1970	GAA (Republic of Maldives)	International1924	-133	-321	50				
179	ISTS 073 ASTRO 1969	IST (Diego Garcia)	International1924	208	-435	-229				
180	KERGUELEN ISLAND 1949	KEG (Kerguelen Island)	International1924	145	-187	103				
181	MAHE 1971	MIK (Mahe Island)	Clarke1880	41	-220	-134				
182	REUNION	REU (Mascarene Islands)	International1924	94	-948	-1262				
183	AMERICAN SAMOA 1962	AMA (American Samoa Islands)	Clarke1866	-115	118	426				
184	ASTRO BEACON "E" 1945	ATF (Iwo Jima)	International1924	145	75	-272				
185	ASTRO TERN ISLAND (FRIG) 1961	TRN (Tern Island)	International1924	114	-116	-333				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
186	ASTRONOMIC AL STATION 1952	ASQ (Marcus Island)	International1924	124	-234	-25	Zeros			
187	BELLEVUE (IGN)	IBE (Efate and Erromango Islands)	International1924	-127	-769	472				
188	CANTO ASTRO 1966	CAO (Phoenix Islands)	International1924	298	-304	-375				
189	CHATHAM ISLAND ASTRO 1971	CHI (Chatham Island (New Zealand))	International1924	175	-38	113				
190	DOS 1968	GIZ (Gizo Island (New Georgia Islands))	International1924	230	-199	-752				
191	EASTER ISLAND 1967	EAS (Easter Island)	International1924	211	147	111				
192	GEODETIC DATUM 1949	GEO (New Zealand)	International1924	84	-22	209				
193	GUAM 1963	GUA (Guam)	Clarke1866	-100	-248	259				
194	GUX 1 ASTRO	DOB (Guadalcanal 1924 Island)	International1924	252	-209	-751				
195	JOHNSTON ISLAND 1961	JOH (Johnston Island)	International1924	189	-79	-202				
196	KUSAIE ASTRO 1951	KUS (Caroline Islands, Fed. States of Micronesia)	International1924	647	1777	-1124				
197	LUZON	LUZ-A (Philippines (Excluding Mindanao Island))	Clarke1866	-133	-77	-51				
198	LUZON	LUZ-B (Mindanao Island)	Clarke1866	-133	-79	-72				
199	MIDWAY ASTRO 1961	MID (Midway Islands)	International1924	912	-58	1227				
200	OLD HAWAIIAN	OHA-M (Mean solution Hawaii)	Clarke1866	61	-285	-181				
201	OLD HAWAIIAN	OHA-A (Hawaii)	Clarke1866	89	-279	-183				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
202	OLD HAWAIIAN	OHA-B (Kauai)	Clarke1866	45	-290	-172				
203	OLD HAWAIIAN	OHA-C (Maui)	Clarke1866	65	-290	-190				
204	OLD HAWAIIAN	OHA-D (Oahu)	Clarke1866	58	-283	-182				
205	PITCAIRN ASTRO 1967	PIT (Pitcairn Island)	International1924	185	165	42				
206	SANTO (DOS) 1965	SAE (Espirito Santo Island)	International1924	170	42	84				
207	VITI LEVU 1916	MVS (Viti Levu Island (Fiji Islands))	Clarke1880	51	391	-36				
208	WAKE-ENIWETOK 1960	ENW (Marshall Islands)	Hough1960	102	52	-38				
209	WAKE ISLAND ASTRO 1952	WAK (Wake Atoll)	International1924	276	-57	149	Zeros			
210	BUKIT RIMPAH	BUR (Bangka and Belitung Islands (Indonesia))	Bessel1841	-384	664	-48				
211	CAMP AREA ASTRO	CAZ (Camp McMurdo Area, Antarctica)	International1924	-104	-129	239				
212	EUROPEAN 1950	EUR-S (Iraq, Israel, Jordan, Kuwait, Lebanon, Saudi Arabia and Syria)	International1924	-103	-106	-141				
213	GUNUNG SEGARA	GSE (Kalimantan (Indonesia))	Bessel1841	-403	684	41				
214	HERAT NORTH	HEN (Afghanistan)	International1924	-333	-222	114				
215	HERMANNSKOGEL	HER (Yugoslavia (Prior to 1990) Slovenia, Croatia, Bosnia and Herzegovina, Serbia)	Bessel1841	682	-203	480				
216	INDIAN	IND-P (Pakistan)	Everest PAK	283	682	231				

Relationships between local datums and WGS84

##	Geodetic DATUM	DATUM ID	Reference Ellipsoid	Translation Vector (meters)			Scale (ppm) and Rotation Vector (seconds of arc)			
				DX	DY	DZ	Scale	Ex	Ey	Ez
217	PULKOVO 1942	PUK (Russia)	Krassovsky1940	28	-130	-95	Zeros			
218	TANANARIVE OBSERVATORY 1925	TAN (Madagaskar)	International1924	-189	-242	-91				
219	VOIROL 1874	VOI (Tunisia/ Algeria)	Clarke1880	-73	-247	227				
220	YACARE	YAC (Uruguay)	Clarke1866	-155	171	37				

Notes:

A	RELEASE, NO CHANGE	3315
REV	**Description**	**ECO**

Topcon Positioning Systems, Inc.
7400 National Drive, Livermore, CA 94550
800·443·4567 www.topconpositioning.com

ISO 9001:2000
FM 68448

GRIL Reference Manual
P/N: 27-040004-01 Rev A 6/08
©2008 Topcon Corporation All rights reserved. No unauthorized duplication.