



**HEXAGON**



**OEM7  
Commands and Logs  
Reference Manual**

---

# OEM7 Commands and Logs Reference Manual

**Publication Number:** OM-20000169

**Revision Level:** v22

**Revision Date:** November 2022

**Firmware Versions:**

- 7.08.14 / OM7MR0814RN0000: OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPar7, CPT7, CPT7700
- 7.08.01R1 / OM7MR0801RN0001: SMART7
- 7.08.01 / OM7MR0801RN0000: SMART2

## Proprietary Notice

ALIGN, GLIDE, Inertial Explorer, NovAtel, OCEANIX, OEM6, OEM7, PwrPak7, RELAY, RTK ASSIST, SPAN, STEADYLINE, TerraStar and Waypoint are trademarks of NovAtel, Inc., entities within the Hexagon Autonomy & Positioning division, their affiliated entities, and/or their licensors. All other trademarks are properties of their respective owners.

©2022 NovAtel Inc. All rights reserved. NovAtel is part of Hexagon. NovAtel makes no representation or warranty regarding the accuracy of the information in this publication. This document gives only a general description of the product(s) or service(s) offered by NovAtel and, except where expressly provided otherwise, shall not form part of any contract. Such information, the products and conditions of supply are subject to change without notice.

---

# Table of Contents

## Figures

## Tables

## Customer Support

## Foreword

## Chapter 1 Messages

1.1	ASCII	33
1.2	Abbreviated ASCII	35
1.3	Binary	37
1.3.1	Virtual Ports	42
1.4	Description of ASCII and Binary Logs with Short Headers	48
1.5	Message Responses	49
1.5.1	Abbreviated ASCII Response	49
1.5.2	ASCII Response	49
1.5.3	Binary Response	50
1.6	GLONASS Slot and Frequency Numbers	52
1.6.1	PRN Numbers	52
1.7	GPS Reference Time Status	53
1.8	Message Time Stamps	54
1.9	Decoding of the GPS Reference Week Number	55
1.10	32-Bit CRC	55

## Chapter 2 Commands

2.1	Command Formats	58
2.1.1	Optional Parameters	58
2.2	Command Settings	58
2.3	Command Defaults	59
2.4	ADJUST1PPS	60
2.5	ALIGNAUTOMATION	68
2.6	ANTENNAPOWER	70
2.7	ANTENNATYPE	72
2.8	APPROXPOSTIMEOUT	75
2.9	ASSIGN	76
2.10	ASSIGNALL	79
2.11	ASSIGNLBANDBEAM	82
2.12	AUTH	84
2.13	AUTOSURVEY	87
2.14	BASEANTENNAPCO	90
2.15	BASEANTENNAPCV	92

---

2.16	BASEANTENNATYPE	93
2.17	BDSECUTOFF	105
2.18	BESTVELTYPE	107
2.19	BLUETOOTHCONTROL	109
2.20	CANCONFIG	110
2.21	CCOMCONFIG	112
2.22	CLOCKADJUST	115
2.23	CLOCKCALIBRATE	117
2.24	CLOCKOFFSET	121
2.25	CNOUPDATE	123
2.26	COMCONTROL	124
2.27	CONFIGCODE	127
2.28	DATADECODESIGNAL	129
2.29	DATUM	133
2.30	DATUMTRANSFORMATION	135
2.31	DGPSTXID	138
2.32	DIFFCODEBIASCONTROL	140
2.33	DLLTIMECONST	141
2.34	DNSCONFIG	144
2.35	DOPPLERWINDOW	146
2.36	DUALANTENNAALIGN	148
2.37	DYNAMICS	149
2.38	ECHO	151
2.39	ECUTOFF	154
2.40	ELEVATIONCUTOFF	156
2.41	ETHCONFIG	159
2.42	EVENTINCONTROL	162
2.43	EVENTOUTCONTROL	165
2.44	EXTERNALCLOCK	168
2.45	FILEAUTOTRANSFER	171
2.46	FILECONFIG	173
2.47	FILEDELETE	175
2.48	FILEMEDIACONFIG	176
2.49	FILEROTATECONFIG	178
2.50	FILETRANSFER	180
2.51	FIX	182
2.52	FORCEGALE6CODE	185
2.53	FORCEGLOL2CODE	186
2.54	FORCEGPSL2CODE	188
2.55	FREQUENCYOUT	190
2.56	FRESET	193
2.57	GALECUTOFF	196
2.58	GENERATEALIGNCORRECTIONS	198
2.59	GENERATEDIFFCORRECTIONS	200

---

2.60	GENERATERTKCORRECTIONS .....	201
2.61	GEODETICDATUM .....	203
2.62	GGAQUALITY .....	205
2.63	GLIDEINITIALIZATIONPERIOD .....	207
2.64	GLOECUTOFF .....	208
2.65	HDTOUTTHRESHOLD .....	210
2.66	HEADINGOFFSET .....	211
2.67	ICOMCONFIG .....	213
2.68	INTERFACEMODE .....	216
2.68.1	SPAN Systems .....	216
2.69	IONOCONDITION .....	222
2.70	IPCONFIG .....	223
2.71	IPSERVICE .....	225
2.72	ITBANDPASSCONFIG .....	227
2.73	ITDETECTCONFIG .....	229
2.74	ITFRONTENDMODE .....	231
2.75	ITPROGFILTCONFIG .....	233
2.76	ITSPECTRALANALYSIS .....	236
2.77	ITWARNINGCONFIG .....	240
2.78	J1939CONFIG .....	241
2.79	LOCKOUT .....	244
2.80	LOCKOUTSATELLITE .....	245
2.81	LOCKOUTSYSTEM .....	246
2.82	LOG .....	247
2.82.1	Binary .....	248
2.82.2	ASCII .....	251
2.83	LOGIN .....	253
2.84	LOGOUT .....	255
2.85	LUA .....	256
2.86	MAGVAR .....	258
2.87	MARKCONTROL .....	261
2.88	MEDIAFORMAT .....	264
2.89	MODEL .....	266
2.90	MOVINGBASESTATION .....	268
2.91	NAVICECUTOFF .....	270
2.92	NMEABEIDOUTALKER .....	272
2.93	NMEAFORMAT .....	273
2.94	NMEATALKER .....	276
2.95	NMEAVERSION .....	278
2.96	NTRIPCONFIG .....	279
2.97	NTRIPSOURCETABLE .....	281
2.98	NVMRESTORE .....	282
2.99	NVMUSERDATA .....	283
2.100	OUTPUTDATUM .....	284

---

2.101	PDPFILTER .....	286
2.101.1	GLIDE Position Filter .....	286
2.102	PDPMODE .....	288
2.103	PGNCONFIG .....	290
2.104	\$PMDT .....	291
2.105	POSAVE .....	293
2.106	POSTIMEOUT .....	295
2.107	PPPBASICCONVERGEDCRITERIA .....	296
2.108	PPPCONVERGEDCRITERIA .....	298
2.109	PPPDYNAMICS .....	300
2.110	PPPDYNAMICSEED .....	301
2.111	PPPRESET .....	303
2.112	PPPSEED .....	304
2.113	PPPSOURCE .....	307
2.114	PPPTIMEOUT .....	308
2.115	PPSCONTROL .....	309
2.116	PPSCONTROL2 .....	312
2.117	PROFILE .....	315
2.118	PSRDIFFSOURCE .....	317
2.119	PSRDIFFSOURCETIMEOUT .....	319
2.120	PSRDIFFTIMEOUT .....	320
2.121	QZSSECUTOFF .....	321
2.122	RADARCONFIG .....	323
2.123	RAIMMODE .....	326
2.123.1	Detection strategy .....	326
2.123.2	Isolation strategy .....	326
2.124	REFERENCESTATIONTIMEOUT .....	328
2.125	RESET .....	329
2.126	RFINPUTGAIN .....	330
2.127	RTKANTENNA .....	332
2.128	RTKASSIST .....	334
2.129	RTKASSISTTIMEOUT .....	335
2.130	RTKDYNAMICS .....	337
2.131	RTKINTEGECRITERIA .....	339
2.132	RTKMATCHEDTIMEOUT .....	341
2.133	RTKNETWORK .....	342
2.134	RTKPORTMODE .....	344
2.135	RTKQUALITYLEVEL .....	346
2.136	RTKRESET .....	347
2.137	RTKSOURCE .....	348
2.138	RTKSOURCETIMEOUT .....	350
2.139	RTKSVENTRIES .....	351
2.140	RTKTIMEOUT .....	352
2.141	RTKTRACKINGCONTROL .....	353

---

2.142	SATEL4CONFIG	354
2.143	SATEL9CONFIG	357
2.144	SATEL9CONFIGL	359
2.145	SATEL9CONFIGN	361
2.146	SATELCONTROL	363
2.147	SATELDETECT	364
2.148	SATELSTARTUPDETECT	365
2.149	SAVECONFIG	366
2.150	SAVEETHERNETDATA	367
2.151	SBASCONTROL	369
2.152	SBASECUTOFF	372
2.153	SBASTIMEOUT	373
2.154	SELECTCHANCONFIG	374
2.155	SEND	380
2.156	SENDHEX	382
2.157	SERIALCONFIG	383
2.158	SERIALPROTOCOL	386
2.159	SETADMINPASSWORD	388
2.160	SETAPPROXPOS	389
2.161	SETAPPROXTIME	390
2.162	SETBASERECEIVERTYPE	392
2.163	SETBESTPOSCRITERIA	393
2.164	SETDIFFCODEBIASES	395
2.165	SETIONOTYPE	397
2.166	SETNAV	399
2.167	SETROVERID	402
2.168	SETTIMEBASE	403
2.169	SETTROPOMODEL	405
2.170	SETUTCLEAPSECONDS	406
2.171	SOFTLOADCOMMIT	407
2.172	SOFTLOADDATA	408
2.173	SOFTLOADFILE	409
2.174	SOFTLOADRESET	411
2.175	SOFTLOADSETUP	412
2.176	SOFTLOADSREC	414
2.177	STATUSCONFIG	415
2.178	STEADYLINE	417
2.179	STEADYLINEDIFFERENTIALTIMEOUT	419
2.180	SURVEYPOSITION	420
2.181	TECTONICSCOMPENSATIONSOURCE	423
2.182	TERRASTARAUTOCHANCONFIG	424
2.183	THISANTENNAPCO	426
2.184	THISANTENNAPCV	427
2.185	THISANTENNATYPE	428

2.186	TILTCOMPENSATIONCONTROL	429
2.187	TILTFILTER	430
2.188	TILTZERO	432
2.189	TRACKSIGNAL	434
2.190	TRACKSV	437
2.191	TUNNELESCAPE	439
2.192	UALCONTROL	441
2.193	UNASSIGN	443
2.194	UNASSIGNALL	445
2.195	UNDULATION	446
2.196	UNLOCKOUT	448
2.197	UNLOCKOUTALL	449
2.198	UNLOCKOUTSATTELLITE	450
2.199	UNLOCKOUTSYSTEM	451
2.200	UNLOG	452
2.200.1	Binary	452
2.200.2	ASCII	453
2.201	UNLOGALL	454
2.202	USBSTICKEJECT	455
2.203	USERDATUM	456
2.204	USEREXPDATUM	458
2.205	USERI2CBITRATE	461
2.206	USERI2CREAD	462
2.207	USERI2CWRITE	464
2.208	UTMZONE	467
2.209	WIFIALIGNAUTOMATION	469
2.210	WIFIAPCHANNEL	472
2.211	WIFIAPCONFIG	473
2.212	WIFIAPPASSKEY	474
2.213	WIFIAPSSID	475
2.214	WIFIMODE	476
2.215	WIFINETCONFIG	478

### Chapter 3 Logs

3.1	Log Types	481
3.1.1	Log Type Examples	481
3.2	ALIGNBSLNENU	483
3.3	ALIGNBSLNXYZ	485
3.4	ALIGNDOP	487
3.5	ALMANAC	488
3.6	AUTHCODES	491
3.7	AVEPOS	493
3.8	BDSALMANAC	495
3.9	BDSBCNAV1EPHEMERIS	497



---

3.10	BDSBCNAV1RAWMESSAGE	500
3.11	BDSBCNAV2EPHEMERIS	501
3.12	BDSBCNAV2RAWMESSAGE	504
3.13	BDSBCNAV3EPHEMERIS	505
3.14	BDSBCNAV3RAWMESSAGE	508
3.15	BDSCLOCK	509
3.16	BDSEPHEMERIS	511
3.17	BDSIONO	514
3.18	BDSRAWNAVSUBFRAME	515
3.19	BESTDATUMINFO	517
3.20	BESTGNSSDATUMINFO	519
3.21	BESTPOS	520
3.22	BESTSATS	528
3.23	BESTUTM	532
3.24	BESTVEL	535
3.25	BESTXYZ	538
3.26	BSLNXYZ	541
3.27	CHANCONFIGLIST	543
3.28	CLOCKMODEL	548
3.29	CLOCKSTEERING	550
3.30	DATUMTRANSFORMATIONS	553
3.31	DUALANTENNAHEADING	555
3.32	ETHSTATUS	557
3.33	FILELIST	558
3.34	FILESTATUS	560
3.35	FILESYSTEMCAPACITY	562
3.36	FILESYSTEMSTATUS	564
3.37	FILETRANSFERSTATUS	566
3.38	GALALMANAC	568
3.39	GALCLOCK	570
3.40	GALCNAVRAWPAGE	572
3.41	GALFNAVEPHEMERIS	573
3.42	GALFNAVRAWALMANAC	576
3.43	GALFNAVRAWEPHEMERIS	577
3.44	GALFNAVRAWPAGE	579
3.45	GALINAVEPHEMERIS	580
3.46	GALINAVRAWALMANAC	583
3.47	GALINAVRAWEPHEMERIS	584
3.48	GALINAVRAWWORD	585
3.49	GALIONO	587
3.50	GEODETICDATUMS	588
3.51	GLMLA	590
3.52	GLOALMANAC	593
3.53	GLOCLOCK	595

---

3.54	GLOEPHEMERIS	597
3.55	GLORAWALM	601
3.56	GLORAWEPHEM	603
3.57	GLORAWFRAME	605
3.58	GLORAWSTRING	607
3.59	GPALM	608
3.60	GPGGA	610
3.61	GPGGALONG	613
3.62	GPGLL	615
3.63	GPGRS	617
3.64	GPGSA	622
3.65	GPGST	624
3.66	GPGSV	626
3.67	GPHDT	629
3.68	GPHDTDUALANTENNA	630
3.69	GPRMB	631
3.70	GPRMC	634
3.71	GPSCNAVRAWMESSAGE	636
3.72	GPSEPHEM	639
3.73	GPVTG	643
3.74	GPZDA	645
3.75	HEADING2	646
3.76	HEADINGEXT	649
3.77	HEADINGEXT2	650
3.78	HEADINGRATE	651
3.79	HEADINGSATS	653
3.80	HWMONITOR	655
3.81	IONUTC	658
3.82	IPSTATS	660
3.83	IPSTATUS	661
3.84	ITBANDPASSBANK	663
3.85	ITDETECTSTATUS	665
3.86	ITFILTTABLE	667
3.87	ITPROGFILTBANK	671
3.88	ITPSDDETECT	673
3.88.1	PSD Samples	673
3.89	ITPSDFINAL	676
3.90	J1939STATUS	680
3.91	LBANDBEAMTABLE	682
3.92	LBANDTRACKSTAT	684
3.93	LOGLIST	687
3.93.1	Binary	688
3.93.2	ASCII	689
3.94	LUAFILELIST	690

---

3.95	LUAFILESYSTEMSTATUS	692
3.96	LUAOUTPUT	693
3.97	LUASTATUS	695
3.98	MARKPOS, MARK2POS, MARK3POS and MARK4POS	696
3.99	MARK1TIME, MARK2TIME, MARK3TIME and MARK4TIME	699
3.100	MASTERPOS	702
3.101	MATCHEDPOS	704
3.102	MATCHEDSATS	707
3.103	MATCHEDXYZ	709
3.104	MODELFEATURES	711
3.105	NAVICALMANAC	715
3.106	NAVICEPHEMERIS	717
3.107	NAVICIONO	720
3.108	NAVICRAWSUBFRAME	722
3.109	NAVICSYSCLOCK	723
3.110	NAVIGATE	725
3.111	NMEA Standard Logs	728
3.112	NOVATELXOBS	730
3.113	NOVATELXREF	731
3.114	OCEANIXINFO	732
3.115	OCEANIXSTATUS	735
3.116	PASSCOM, PASSAUX, PASSUSB, PASSETH1, PASSICOM, PASSNCOM	737
3.117	PASSTHROUGH	742
3.118	PDPDATUMINFO	743
3.119	PDPDOP	744
3.120	PDPDOP2	746
3.121	PDPPOS	748
3.122	PDPSATS	750
3.123	PDPVEL	752
3.124	PDPXYZ	753
3.125	PORTSTATS	755
3.126	PPPDATUMINFO	757
3.127	PPPPOS	758
3.128	PPPSATS	760
3.129	PPPSEEDAPPLICATIONSTATUS	762
3.130	PPPSEEDSTORESTATUS	764
3.131	PPPXYZ	765
3.132	PROFILEINFO	767
3.133	PSRDATUMINFO	769
3.134	PSRDOP	770
3.135	PSRDOP2	772
3.136	PSRPOS	774
3.137	PSRSATS	776
3.138	PSRVEL	778

---

3.139	PSRXYZ	780
3.140	QZSSALMANAC	783
3.141	QZSSCNAVRAWMESSAGE	785
3.142	QZSSEPHEMERIS	788
3.143	QZSSIONUTC	791
3.144	QZSSRAWALMANAC	793
3.145	QZSSRAWCNAVMESSAGE	795
3.146	QZSSRAWEPHEM	796
3.147	QZSSRAWSUBFRAME	798
3.148	RADARSTATUS	799
3.149	RAIMSTATUS	801
3.150	RANGE	804
3.151	RANGECMP	811
3.152	RANGECMP2	816
3.153	RANGECMP4	823
3.154	RANGEGPSL1	835
3.155	RAWALM	837
3.156	RAWCNAVFRAME	839
3.157	RAWEPHEM	840
3.158	RAWGPSSUBFRAME	842
3.159	RAWGPSWORD	843
3.160	RAWSBASFRAME	844
3.161	RAWSBASFRAME2	845
3.162	REFSTATION	847
3.163	REFSTATIONINFO	849
3.164	ROVERPOS	851
3.165	RTCMV3 Standard Logs	854
3.165.1	Legacy Observable Messages	854
3.165.2	MSM Observable Messages	855
3.165.3	Station and Antenna Messages	856
3.165.4	Ephemeris Messages	857
3.165.5	GLONASS Bias Information Message	858
3.166	RTKASSISTSTATUS	859
3.167	RTKDATUMINFO	861
3.168	RTKDOP	862
3.169	RTKDOP2	864
3.170	RTKPOS	866
3.171	RTKSATS	869
3.172	RTKVEL	871
3.173	RTKXYZ	873
3.174	RXCONFIG	876
3.175	RXSTATUS	878
3.176	RXSTATUSEVENT	891
3.177	SAFEMODESTATUS	894

---

3.178	SATEL4INFO	897
3.179	SATEL9INFO	899
3.180	SATELSTATUS	902
3.181	SATVIS2	904
3.182	SATXYZ2	907
3.183	SAVEDSURVEYPOSITIONS	910
3.184	SBAS0	912
3.185	SBAS1	913
3.186	SBAS2	914
3.187	SBAS3	917
3.188	SBAS4	919
3.189	SBAS5	921
3.190	SBAS6	923
3.191	SBAS7	926
3.192	SBAS9	929
3.193	SBAS10	931
3.194	SBAS12	933
3.195	SBAS17	935
3.196	SBAS18	937
3.197	SBAS24	939
3.198	SBAS25	942
3.199	SBAS26	945
3.200	SBAS27	947
3.201	SBASALMANAC	949
3.202	SOFTLOADSTATUS	951
3.203	SOURCETABLE	954
3.204	TECTONICSCOMPENSATION	957
3.205	TERRASTARINFO	959
3.206	TERRASTARSTATUS	962
3.207	TILTDATA	964
3.208	TIME	965
3.209	TIMESYNC	968
3.210	TRACKSTAT	969
3.211	TRANSFERPORTSTATUS	971
3.212	UPTIME	973
3.213	USERANTENNA	974
3.214	USERI2CRESPONSE	976
3.215	VALIDMODELS	979
3.216	VERIPOSINFO	981
3.217	VERIPOSSTATUS	983
3.218	VERSION	984
3.219	WIFIAPSETTINGS	989
3.220	WIFINETLIST	992
3.221	WIFISTATUS	994

---

## Chapter 4 SPAN Commands

4.1 ALIGNMENTMODE .....	999
4.2 CONNECTIMU .....	1001
4.3 DMICONFIG .....	1004
4.4 EXTERNALPVAS .....	1006
4.5 INPUTGIMBALANGLE .....	1011
4.6 INSALIGNCONFIG .....	1013
4.7 INSCALIBRATE .....	1016
4.8 INSCOMMAND .....	1019
4.9 INSSEED .....	1021
4.10 INSTHRESHOLDS .....	1023
4.11 INSZUPT .....	1025
4.12 RELINSAUTOMATION .....	1026
4.13 RELINSCONFIG .....	1028
4.14 SETALIGNMENTORIENTATION .....	1030
4.15 SETALIGNMENTVEL .....	1031
4.16 SETHEAVEWINDOW .....	1032
4.17 SETIMUEVENT .....	1033
4.18 SETIMUPORTPROTOCOL .....	1034
4.19 SETIMUSPECS .....	1035
4.20 SETINITAZIMUTH .....	1038
4.21 SETINSPROFILE .....	1039
4.22 SETINSROTATION .....	1041
4.23 SETINSTRANSFORMATION .....	1044
4.24 SETINSUPDATE .....	1048
4.25 SETMAXALIGNMENTTIME .....	1050
4.26 SETRELINSOUTPUTFRAME .....	1051
4.27 SETUPSENSOR .....	1053
4.28 TAGNEXTMARK .....	1055
4.29 TIMEDEVENTPULSE .....	1056

## Chapter 5 SPAN Logs

5.1 Logs with INS or GNSS Data .....	1059
5.2 BESTGNSSPOS .....	1060
5.3 BESTGNSSVEL .....	1062
5.4 CORRIMUDATA .....	1064
5.5 CORRIMUDATAS .....	1066
5.6 CORRIMUS .....	1068
5.7 DELAYEDHEAVE .....	1070
5.8 GIMBALLEDPVA .....	1071
5.9 HEAVE .....	1073
5.10 IMURATECORRIMUS .....	1074
5.11 IMURATEPVA .....	1076

---

5.12	IMURATEPVAS .....	1078
5.13	INSATT .....	1080
5.14	INSATTQS .....	1082
5.15	INSATTS .....	1084
5.16	INSATTX .....	1085
5.17	INSCALSTATUS .....	1090
5.18	INSCONFIG .....	1092
5.19	INSDATUMINFO .....	1096
5.20	INSPOS .....	1097
5.21	INSPOSS .....	1098
5.22	INSPOSX .....	1099
5.23	INSPVA .....	1101
5.24	INSPVACMP .....	1103
5.25	INSPVAS .....	1105
5.26	INSPVASDCMP .....	1107
5.27	INSPVAX .....	1109
5.28	INSSEEDSTATUS .....	1112
5.29	INSSPD .....	1114
5.30	INSSPDS .....	1115
5.31	INSSTDEV .....	1116
5.32	INSSTDEVS .....	1118
5.33	INSUPDATESTATUS .....	1120
5.34	INSVEL .....	1124
5.35	INSVELS .....	1125
5.36	INSVELUSER .....	1126
5.37	INSVELX .....	1129
5.38	MARK1PVA, MARK2PVA, MARK3PVA and MARK4PVA .....	1131
5.39	PASHR .....	1133
5.40	RAWDMI .....	1135
5.41	RAWIMU .....	1137
5.42	RAWIMUS .....	1161
5.43	RAWIMUSX .....	1166
5.44	RAWIMUX .....	1170
5.45	RELINSPVA .....	1174
5.46	SYNCHEAVE .....	1177
5.47	SYNCRELINSPVA .....	1178
5.48	TAGGEDMARK1PVA, TAGGEDMARK2PVA, TAGGEDMARK3PVA and TAGGEDMARK4PVA .....	1181
5.49	TSS1 .....	1183
5.50	VARIABLELEVERARM .....	1185

---

## Chapter 6 Responses

### APPENDIX A Example of Bit Parsing a RANGECMP4 Log

A.1 Reference Log Decoding .....	1195
A.1.1 Reference Header .....	1195
A.1.2 Reference Satellite and Signal Block: GPS .....	1195
A.1.3 Reference Measurement Block Header: GPS .....	1197
A.1.4 Reference Measurement Block: GPS .....	1197
A.1.5 Reference Primary Signal Measurement Block: GPS PRN 10 – L1CA .....	1198
A.1.6 Reference Secondary Signals Measurement Block: GPS PRN 10 – L2Y .....	1200
A.1.7 Reference Third Signals Measurement Block: GPS PRN 10 – L5Q .....	1202
A.1.8 Reference Satellite and Signal Block: GLONASS .....	1204
A.1.9 Reference Measurement Block Header: GLONASS PRN 38 .....	1206
A.1.10 Reference Primary Signal Measurement Block: GLONASS PRN 38 – L1CA .....	1206
A.2 Differential Log Decoding .....	1208
A.2.1 Differential Header .....	1209
A.2.2 Differential Satellite and Signal Block .....	1209
A.2.3 Differential Measurement Block Header .....	1210
A.2.4 Differential Measurement Block .....	1211
A.2.5 Differential Primary Signal Measurement Block GPS PRN 10 – L1CA .....	1212
A.2.6 Differential Secondary Signals Measurement Block GPS PRN 10 – L2Y .....	1214
A.2.7 Differential Third Signals Measurement Block GPS PRN 10 – L5Q .....	1216



# Figures

Figure 1: Byte Arrangements .....	33
Figure 2: 1PPS Alignment .....	60
Figure 3: ADJUST1PPS Connections .....	63
Figure 4: Pulse Width and 1PPS Coherency .....	191
Figure 5: HEADINGOFFSET Example .....	211
Figure 6: Illustration of Magnetic Variation and Correction .....	258
Figure 7: TTL Pulse Polarity .....	261
Figure 8: Moving Base Station 'Daisy Chain' Effect .....	269
Figure 9: Using the SEND Command .....	381
Figure 10: Illustration of SETNAV Parameters .....	399
Figure 11: Illustration of Undulation .....	446
Figure 12: The WGS84 ECEF Coordinate System .....	540
Figure 13: Navigation Parameters .....	725
Figure 14: Pass Through Log Data .....	740
Figure 15: Channel Tracking Example .....	806

# Tables

Table 1: Field Type .....	32
Table 2: ASCII Message Header Structure .....	34
Table 3: Abbreviated ASCII Message Header Structure .....	35
Table 4: Binary Message Header Structure .....	38
Table 5: Detailed Port Identifier .....	40
Table 6: Detailed Port Identifier – Virtual Ports .....	42
Table 7: Available Port Types .....	48
Table 8: Short ASCII Message Header Structure .....	49
Table 9: Short Binary Message Header Structure .....	49
Table 10: Binary Message Response Structure .....	50
Table 11: Binary Message Sequence .....	51
Table 12: PRN Numbers for Commands and Logs .....	52
Table 13: GPS Reference Time Status .....	53
Table 14: COM Port Signals Available for 1PPS .....	61
Table 15: ADJUST1PPS Mode .....	66
Table 16: ANTENNAPOWER Parameters .....	71
Table 17: User-Defined Antenna Type .....	74
Table 18: Channel State .....	78
Table 19: Satellite System .....	80
Table 20: L-Band Assignment Option .....	83
Table 21: AUTH Command State .....	86
Table 22: Frequency Type .....	91
Table 23: Antenna Type .....	94
Table 24: Radome Type .....	103
Table 25: CAN Port Speed .....	111
Table 26: CAN Protocol .....	114
Table 27: Control Options .....	125
Table 28: Port, Signal and Control Availability on OEM7 Receivers .....	126
Table 29: GNSS Signal Default and Configurability .....	129
Table 30: Signal Type .....	131
Table 31: Signal Type .....	142
Table 32: Satellite System .....	147
Table 33: User Dynamics .....	150
Table 34: Communications Port Identifiers .....	152
Table 35: Clock Type .....	170

---

Table 36: Pre-Defined Values for Oscillators .....	170
Table 37: Disk Full Action .....	179
Table 38: FIX Parameters .....	183
Table 39: Fix Types .....	184
Table 40: Signals Tracked – Channel Configuration and L2type Option .....	186
Table 41: GPS L2 Code Type .....	188
Table 42: Signals Tracked – Channel Configuration and L2type Option .....	189
Table 43: FRESET Target .....	194
Table 44: Datum Anchors .....	204
Table 45: ICOMCONFIG Protocol .....	215
Table 46: Serial Port Interface Modes .....	219
Table 47: IP Services .....	226
Table 48: RF Path Selection .....	230
Table 49: Frequency Bands .....	232
Table 50: Mode .....	232
Table 51: Programmable Filter ID .....	234
Table 52: Programmable Filter Mode .....	235
Table 53: Data Sources for PSD Samples .....	237
Table 54: Frequency Types .....	238
Table 55: FFT Sizes .....	239
Table 56: Binary Log Triggers .....	250
Table 57: ASCII Log Triggers .....	252
Table 58: NMEA Talkers .....	277
Table 59: Epoch Options .....	285
Table 60: Profile Option .....	316
Table 61: PSRDIFFSOURCE Type .....	318
Table 62: Response Modes .....	324
Table 63: RAIM Mode Types .....	327
Table 64: Network RTK Mode .....	342
Table 65: RTK Source Type .....	349
Table 66: Base Type .....	355
Table 67: Compatibility Mode .....	356
Table 68: Turn Radio On, Off or Factory Reset .....	363
Table 69: System Types .....	371
Table 70: COM Port Identifiers .....	385
Table 71: Parity .....	385

---

Table 72: Handshaking .....	385
Table 73: Ports Supporting RS-422 .....	387
Table 74: Selection Type .....	394
Table 75: System Used for Timing .....	404
Table 76: Available Set Up Commands .....	413
Table 77: STEADYLINE Mode .....	418
Table 78: Signal .....	435
Table 79: TRACKSV Command Condition .....	438
Table 80: User Accuracy Level Supplemental Position Types and NMEA Equivalents .....	441
Table 81: UTM Zone Commands .....	468
Table 82: Log Type Triggers .....	481
Table 83: Position Averaging Status .....	494
Table 84: Data Source .....	515
Table 85: Transformation Status .....	518
Table 86: Solution Status .....	523
Table 87: Position or Velocity Type .....	524
Table 88: GPS and GLONASS Signal-Used Mask .....	526
Table 89: Galileo and BeiDou Signal-Used Mask .....	526
Table 90: Extended Solution Status .....	527
Table 91: Supplemental Position Types and NMEA Equivalents .....	527
Table 92: Observation Statuses .....	529
Table 93: GPS Signal Mask .....	530
Table 94: GLONASS Signal Mask .....	531
Table 95: Galileo Signal Mask .....	531
Table 96: BeiDou Signal Mask .....	531
Table 97: QZSS Signal Mask .....	531
Table 98: NavIC Signal Mask .....	531
Table 99: Definitions .....	540
Table 100: CHANCONFIGLIST Signal Type .....	545
Table 101: Clock Model Status .....	549
Table 102: Clock Source .....	551
Table 103: Steering State .....	551
Table 104: File Type .....	559
Table 105: Mass Storage Device .....	561
Table 106: File Status .....	561
Table 107: Mass Storage Status .....	565

---

Table 108: File Transfer Status .....	567
Table 109: Signal Type .....	585
Table 110: Kp UTC Leap Second Descriptions .....	596
Table 111: GLONASS Ephemeris Flags Coding .....	600
Table 112: P1 Flag Range Values .....	600
Table 113: GPS Quality Indicators .....	612
Table 114: Position Precision of NMEA Logs .....	614
Table 115: Position Precision of NMEA Logs .....	616
Table 116: NMEA Positioning System Mode Indicator .....	616
Table 117: System and Signal IDs .....	619
Table 118: NMEA Positioning System Mode Indicator .....	633
Table 119: NMEA Positioning System Mode Indicator .....	635
Table 120: Signal Type .....	637
Table 121: URA Variance .....	642
Table 122: NMEA Positioning System Mode Indicator .....	644
Table 123: Solution Source .....	648
Table 124: Satellite System .....	654
Table 125: HWMONITOR Status Table .....	656
Table 126: DDC Filter Type .....	669
Table 127: ITFILTable Status Word .....	669
Table 128: Filter Switches .....	670
Table 129: Interference Detection Status Word .....	674
Table 130: RF Frequency Path .....	675
Table 131: Spectral Analysis Status Word .....	678
Table 132: Node Status .....	681
Table 133: L-Band Signal Tracking Status .....	685
Table 134: File System Status .....	692
Table 135: Lua Data Source .....	694
Table 136: Script Status .....	695
Table 137: Feature Status .....	712
Table 138: Feature Type .....	713
Table 139: GNSS Time Scales .....	724
Table 140: Navigation Data Type .....	727
Table 141: Oceanix Subscription Type .....	733
Table 142: Oceanix Subscription Details Mask .....	733
Table 143: Oceanix Region Restriction .....	734

---

Table 144: Decoder Data Synchronization State .....	735
Table 145: Region Restriction Status .....	736
Table 146: System Used for Timing .....	747
Table 147: Position Type .....	759
Table 148: PPP Seed Application Status .....	763
Table 149: Status Word .....	768
Table 150: System Used for Timing .....	773
Table 151: Signal Type .....	786
Table 152: Emulated Radar Status .....	800
Table 153: RAIM Mode Types .....	802
Table 154: Integrity Status .....	802
Table 155: Protection Level Status .....	803
Table 156: Channel Tracking Status .....	806
Table 157: Tracking State .....	808
Table 158: Correlator Type .....	808
Table 159: RINEX Mappings .....	809
Table 160: Range Record Format (RANGECMP only) .....	812
Table 161: StdDev-PSR Values .....	814
Table 162: Satellite Block of the Range Record Format (RANGECMP2 only) .....	817
Table 163: Signal Block of the Range Record Format (RANGECMP2 only) .....	818
Table 164: Std Dev PSR Scaling .....	819
Table 165: Std Dev ADR Scaling .....	820
Table 166: L1/E1/B1 Scaling .....	820
Table 167: Signal Type (only in RANGECMP2) .....	821
Table 168: Header .....	825
Table 169: Satellite and Signal Block .....	825
Table 170: Measurement Block Header .....	826
Table 171: Primary Reference Signal Measurement Block .....	827
Table 172: Secondary Reference Signals Measurement Block .....	828
Table 173: Primary Differential Signal Measurement Block .....	829
Table 174: Secondary Differential Signals Measurement Block .....	831
Table 175: Signal Bit Mask .....	832
Table 176: Lock Time .....	832
Table 177: ADR Std Dev .....	833
Table 178: Pseudorange Std Dev .....	834
Table 179: Base Station Status .....	848

---

Table 180: Base Station Type .....	848
Table 181: Legacy Observable Messages .....	855
Table 182: MSM Type Descriptions .....	855
Table 183: MSM Log Names .....	856
Table 184: MSM Message IDs .....	856
Table 185: Station and Antenna Messages .....	857
Table 186: Ephemeris Messages .....	857
Table 187: GLONASS Bias Information Message .....	858
Table 188: System Used for Timing .....	865
Table 189: Receiver Error .....	881
Table 190: Receiver Status .....	882
Table 191: Version Bits .....	884
Table 192: Auxiliary 1 Status .....	884
Table 193: Auxiliary 2 Status .....	886
Table 194: Auxiliary 3 Status .....	887
Table 195: Antenna Gain State .....	889
Table 196: Auxiliary 4 Status .....	889
Table 197: Status Word .....	892
Table 198: Event Type .....	893
Table 199: Safe Mode States .....	895
Table 200: Compatibility Mode .....	898
Table 201: ASCII and Binary Values .....	901
Table 202: Radio State .....	903
Table 203: Error Types .....	903
Table 204: Evaluation of UDREI .....	916
Table 205: SBAS Subsystem Types .....	950
Table 206: SoftLoad Status Type .....	951
Table 207: Tectonics Compensation Status .....	957
Table 208: TerraStar Subscription Type .....	960
Table 209: TerraStar Subscription Details Mask .....	960
Table 210: TerraStar Region Restriction .....	961
Table 211: Decoder Data Synchronization State .....	962
Table 212: TerraStar Local Area Status .....	963
Table 213: TerraStar Geogating Status .....	963
Table 214: USB Detection Type .....	971
Table 215: USB Mode .....	972

---

Table 216: User-Defined Antenna Type .....	975
Table 217: Error Code .....	977
Table 218: Operation Mode Code .....	978
Table 219: Veripos Operating Mode .....	981
Table 220: Veripos Subscription Details Mask .....	982
Table 221: Decoder Data Synchronization State .....	983
Table 222: Component Types .....	986
Table 223: Firmware and Boot Version Field Formats .....	987
Table 224: Wi-Fi Band .....	990
Table 225: Wi-Fi Security Protocol .....	990
Table 226: Wi-Fi Encryption Type .....	990
Table 227: Regulatory Region .....	991
Table 228: Wi-Fi Security Type .....	993
Table 229: Wi-Fi Status .....	995
Table 230: IMU Type .....	1002
Table 231: EXTERNALPVAS Updates Mask .....	1009
Table 232: EXTERNALPVAS Options Mask .....	1010
Table 233: COM Ports .....	1029
Table 234: Rotational Offset Types .....	1043
Table 235: Translation Offset Types .....	1046
Table 236: Translation Input Frame .....	1046
Table 237: Inertial Solution Status .....	1081
Table 238: Extended Solution Status .....	1086
Table 239: Alignment Indication .....	1088
Table 240: NVM Seed Indication .....	1089
Table 241: Offset Type .....	1091
Table 242: Source Status .....	1091
Table 243: Injection Status .....	1113
Table 244: Validity Status .....	1113
Table 245: DMI Update Status .....	1121
Table 246: ALIGN Update Values .....	1121
Table 247: INS Update Values .....	1122
Table 248: Extended Velocity Status .....	1128
Table 249: iIMU-FSAS IMU Status .....	1140
Table 250: HG1700 IMU Status .....	1141
Table 251: LN200 IMU Status .....	1143



Table 252: ISA-100C IMU Status .....	1144
Table 253: IMU-CPT IMU Status .....	1145
Table 254: KVH 1750 and KVH P-1750 IMU Status .....	1147
Table 255: HG1900 and HG1930 IMU Status .....	1148
Table 256: HG4930 AN01 IMU Status .....	1150
Table 257: HG4930 AN04 IMU Status .....	1151
Table 258: ADIS16488 and IMU-IGM-A1 IMU Status .....	1153
Table 259: STIM300 and IMU-IGM-S1 IMU Status .....	1154
Table 260: $\mu$ IMU IMU Status .....	1156
Table 261: G320N IMU Status .....	1157
Table 262: G370N IMU Status .....	1159
Table 263: Raw IMU Scale Factors .....	1164
Table 264: Response Messages .....	1186

# Customer Support

## NovAtel Knowledge Base

If you have a technical issue, visit the NovAtel Support page at [novatel.com/support](http://novatel.com/support). Through the *Support* page you can contact Customer Support, find papers and tutorials or download the latest firmware. To access the latest user documentation, visit [docs.novatel.com/OEM7](http://docs.novatel.com/OEM7).

## Before Contacting Customer Support

Before contacting NovAtel Customer Support about a software problem, perform the following steps:



If logging data over an RS-232 serial cable, ensure that the configured baud rate can support the data bandwidth (see **SERIALCONFIG** command). NovAtel recommends a minimum suggested baud rate of 230400 bps.

1. Log the data suggested in the appropriate *Troubleshooting Logs* section to a file on your computer for 15 minutes.
  - *General Troubleshooting Logs* on the next page
  - *Tracking and Interference Troubleshooting Logs* on the next page
  - *SPAN Troubleshooting Logs* on page 28
  - *RTK Troubleshooting Logs* on page 28
  - *PPP Troubleshooting Logs* on page 29
  - *ALIGN Troubleshooting Logs* on page 29

If using NovAtel Application Suite, log the Troubleshooting message set for 15 minutes.

2. Send the data file to NovAtel Customer Support: [support.novatel@hexagon.com](mailto:support.novatel@hexagon.com)
3. You can also issue a **FRESET** command to the receiver to clear any unknown settings.



The **FRESET** command will erase all user settings. You should know your configuration (by requesting the RXCONFIGA log) and be able to reconfigure the receiver before you send the **FRESET** command.

If you are having a hardware problem, send a list of the troubleshooting steps taken and the results.

## Contact Information

Log a support request with NovAtel Customer Support using one of the following methods:

### Log a Case and Search Knowledge:

Website: [novatel.com/support](http://novatel.com/support)

### Log a Case, Search Knowledge and View Your Case History: (login access required)

Web Portal: [shop.novatel.com/novatelstore/s/login/](http://shop.novatel.com/novatelstore/s/login/)

**E-mail:**

[support.novatel@hexagon.com](mailto:support.novatel@hexagon.com)

**Telephone:**

U.S. and Canada: 1-800-NOVATEL (1-800-668-2835)

International: +1-403-295-4900

## General Troubleshooting Logs

```
LOG RXSTATUSB ONCHANGED
LOG RAWEPHEMB ONNEW
LOG GLORAWEPHEMB ONNEW
LOG BESTPOSB ONTIME 1
LOG RANGE B ONTIME 1
LOG RXCONFIGA ONCE
LOG VERSIONA ONCE
LOG LOGLISTA ONCE
LOG PORTSTATSA ONTIME 10
LOG PROFILEINFOA ONCE
LOG HWMONITORA ONTIME 10
```

## Tracking and Interference Troubleshooting Logs

```
LOG VERSIONA ONCE
LOG RXCONFIGA ONCE
LOG CHANCONFIGLISTB ONCE
LOG PASSTHROUGH A ONNEW
LOG RXSTATUSB ONCHANGED
LOG CLOCKSTEERING B ONCHANGED
LOG RAWEPHEMB ONNEW
LOG GLORAWEPHEMB ONNEW
LOG GALINAVRAWEPHEMERIS B ONNEW
LOG BDSEPHEMERIS B ONNEW
LOG QZSSEPHEMERIS B ONNEW
LOG NAVICEPHEMERIS B ONNEW
LOG RAWALMB ONNEW
LOG GLORAWALMB ONNEW
LOG GALALMANACB ONNEW
LOG BDSALMANACB ONNEW
LOG QZSSALMANACB ONNEW
LOG NAVICALMANACB ONNEW
LOG IONUTCB ONNEW
LOG GLOCLOCKB ONNEW
LOG GALCLOCKB ONNEW
LOG BDSLOCKB ONNEW
LOG TRACKSTATB ONTIME 1
LOG RANGE B ONTIME 1
LOG BESTPOSB ONTIME 1
LOG SATVIS2B ONTIME 30
LOG ITDETECTSTATUSB ONCHANGED
```

For dual antenna receivers, add these logs:

```
LOG TRACKSTATB_1 ONTIME 1
LOG RANGE B_1 ONTIME 1
```

For interference issues add this log.

```
LOG ITPSDDETECTB ONNEW
```

For interference issues, when you have enough datalink bandwidth to handle large logs, add this log:

```
LOG ITPSDFINALB ONNEW
```

## SPAN Troubleshooting Logs

```
LOG RXSTATUSB ONCHANGED
LOG RAWEPHEMB ONNEW
LOG GLORAWEPHEMB ONNEW
LOG GALINAVRAWEPHEMERISB ONNEW
LOG BDSEPEHEMERISB ONNEW
LOG BESTPOSB ONTIME 1
LOG RANGE B ONTIME 1
LOG RXCONFIGA ONCE
LOG VERSIONA ONCE
LOG RAWIMUSXB ONNEW
LOG INSUPDATESTATUSB ONCHANGED
LOG INSPVAXB ONTIME 1
LOG INSCONFIGA ONCHANGED
LOG BESTGNSSPOSB ONTIME 1
```

## RTK Troubleshooting Logs

```
LOG RXSTATUSB ONCHANGED
LOG RAWEPHEMB ONNEW
LOG GLORAWEPHEMB ONNEW
LOG QZSSRAWEPHEMB ONNEW
LOG BDSRAWNAVSUBFRAMEB ONNEW
LOG GALFNAVRAWEPHEMERISB ONNEW
LOG GALINAVRAWEPHEMERISB ONNEW
LOG RANGE B ONTIME 1.0
LOG BESTPOSB ONTIME 1.0
LOG RXCONFIGB ONCE
LOG VERSIONB ONCE
LOG TRACKSTATB ONTIME 1.0
LOG RTKPOSB ONTIME 1.0
LOG MATCHEDPOSB ONNEW
LOG MATCHEDSATS B ONNEW
LOG RTKSATS B ONTIME 1.0
LOG PSRPOSB ONTIME 1.0
LOG RAWALMB ONNEW
LOG IONUTCB ONNEW
LOG GLORAWALMB ONNEW
LOG GLOCLOCKB ONNEW
LOG PASSTHROUGH B ONNEW
LOG CLOCKMODEL B ONTIME 1.0
LOG REFSTATIONB ONNEW
LOG RTKVELB ONTIME 1.0
```

## PPP Troubleshooting Logs

```
LOG RXSTATUSB ONCHANGED
LOG GPSEPHEMB ONNEW
LOG GLOEPHEMERISB ONNEW
LOG QZSEPHEMERISB ONNEW
LOG BDSEPHEMERISB ONNEW
LOG BDSBCNAV1EPHEMERISB ONNEW (firmware versions 7.08.03 and 7.08.10 and later)
LOG BDSBCNAV2EPHEMERISB ONNEW (firmware versions 7.08.03 and 7.08.10 and later)
LOG BDSBCNAV3EPHEMERISB ONNEW (firmware versions 7.08.03 and 7.08.10 and later)
LOG GALFNAVEPHEMERISB ONNEW
LOG GALINAVEPHEMERISB ONNEW
LOG RANGE B ONTIME 1.0
LOG BESTPOS B ONTIME 1.0
LOG RXCONFIG B ONCE
LOG VERSION B ONCE
LOG TRACKSTAT B ONTIME 10.0
LOG LBANDTRACKSTAT B ONTIME 1.0
LOG PPPPOS B ONTIME 1.0
LOG PPPSAT B ONTIME 1.0
LOG TERRASTARINFO B ONCHANGED
LOG TERRASTARSTATUS B ONCHANGED
LOG PSRPOS B ONTIME 1.0
LOG ALMANAC B ONNEW
LOG GLOALMANAC B ONNEW
LOG GALALMANAC B ONNEW
LOG BDSALMANAC B ONNEW
LOG QZSSALMANAC B ONNEW
LOG IONUTCB ONNEW
LOG GLOCLOCK B ONNEW
LOG LBANDBEAMTABLE B ONCHANGED
```

## ALIGN Troubleshooting Logs

```
LOG RXSTATUSB ONCHANGED
LOG RAWEPHEMB ONNEW
LOG GLORAWEPHEMB ONNEW
LOG GALINAVRAWEPHEMERISB ONNEW
LOG BDSEPHEMERISB ONNEW
LOG BESTPOS B ONTIME 1
LOG RANGE B ONTIME 1
LOG RXCONFIG A ONCE
LOG VERSION A ONCE
LOG LOGLISTA ONCE
LOG PORTSTATSA ONTIME 10
LOG ALIGNBSLNENUB ONNEW
LOG ALIGNBSLNXYZB ONNEW
LOG ALIGNDOPB ONNEW
LOG HEADING2 B ONNEW
LOG MASTERPOS B ONNEW
LOG ROVERPOS B ONNEW (This log can only be output if you have Y model ALIGN)
LOG HEADINGSATSA ONNEW (This log can only be output on the ALIGN rover)
LOG DUALANTENNAHEADING B ONTIME 1.0
LOG TRACKSTAT B_1 ONTIME 1 (Dual antenna receivers only)
LOG RANGE B_1 ONTIME 1 (dual antenna receivers only)
```

# Foreword

This manual describes each command and log the OEM7 receivers and SMART2 receiver are capable of accepting or generating. Sufficient detail is provided so you can understand the purpose, syntax and structure of each command or log. You will also be able to communicate with the receiver, enabling you to effectively use and write custom interfacing software for specific applications.

## Related Documents and Information

OEM7 and SMART2 products include the following:

- Support for all current and upcoming GNSS constellations
- Real-Time Kinematic (RTK) (not available on SMART2)
- L-Band capability including TerraStar licensed based corrections
- Satellite Based Augmentation System (SBAS) signal functionality
- Differential Global Positioning System (DGPS)
- National Marine Electronics Association (NMEA) standards, a protocol used by GNSS receivers to transmit data

For more information on these components, refer the Support page on our website at [novatel.com/support](http://novatel.com/support). For introductory information on GNSS technology, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

This manual does not address any of the receiver hardware attributes or installation information. Consult the product specific Installation and Operation User Manuals for information about these topics ([docs.novatel.com/OEM7](http://docs.novatel.com/OEM7)). Furthermore, should you encounter any functional, operational or interfacing difficulties with the receiver, refer to the NovAtel web site for warranty and support information.

## Prerequisites

As this reference manual is focused on the OEM7 family commands and logging protocol, it is necessary to ensure the receiver has been properly installed and powered up according to the instructions outlined in the companion product specific user manuals ([docs.novatel.com/OEM7](http://docs.novatel.com/OEM7)).

## Commands and Logs Defaults and Structure

- The factory defaults for commands and logs are shown after the syntax but before the example in the command or log description.
- The letter H in the Binary Byte or Binary Offset columns of the commands and logs tables represents the header length for that command or log, see *Binary* on page 37.
- The number following 0x is a hexadecimal number.
- Default values shown in command tables indicate the assumed values when optional parameters have been omitted. Default values do not imply the factory default settings.
- Parameters surrounded by [ and ] are optional in a command or are required for only some instances of the command depending on the values of other parameters.
- Text displayed between < and > indicates the entry of a keystroke in the case of the command or an automatic entry in the case of carriage return <CR> and line feed <LF> in data output.
- In tables where no values are given they are assumed to be reserved for future use.

- Status words in ASCII logs are output as hexadecimal numbers and must be converted to binary format (and in some cases then also to decimal) to parse the fields because they are not fixed in 4-bits boundary. For an example of this type of conversion, see the RANGE log, *Table 156: Channel Tracking Status* on page 806.
- Conversions and their binary or decimal results are always read from right to left. For a complete list of hexadecimal, binary and decimal equivalents, refer to the [Unit Conversions](#) information available on our website at [novatel.com/support](http://novatel.com/support).
- ASCII log examples may be split over several lines for readability. In reality, only a single [CR][LF] pair is transmitted at the end of an ASCII log.

You can download the most up-to-date version of this manual from the OEM7 Documentation Portal ([docs.novatel.com/OEM7](http://docs.novatel.com/OEM7)).

## Conventions

The following conventions are used in this manual:



Information that supplements or clarifies text.



A caution that actions, operation or configuration may lead to incorrect or improper use of the hardware.



A warning that actions, operation or configuration may result in regulatory noncompliance, safety issues or equipment damage.

# Chapter 1 Messages

The receiver handles incoming and outgoing NovAtel data in three different message formats: Abbreviated ASCII, ASCII and Binary. This allows for a great deal of versatility in the way the OEM7 and SMART2 receivers can be used. All NovAtel commands and logs can be entered, transmitted, output or received in any of the three formats. The receiver also supports RTCMV3, NOVATELX and NMEA format messaging.

When entering an ASCII or abbreviated ASCII command to request an output log, the message type is indicated by the character appended to the end of the message name. 'A' indicates the message is ASCII and 'B' indicates binary. No character means the message is Abbreviated ASCII. When issuing binary commands, the output message type is dependent on the bit format in the message's binary header (refer to *Binary* on page 37).

Table 1: *Field Type* below below, describes the field types used in the description of messages.

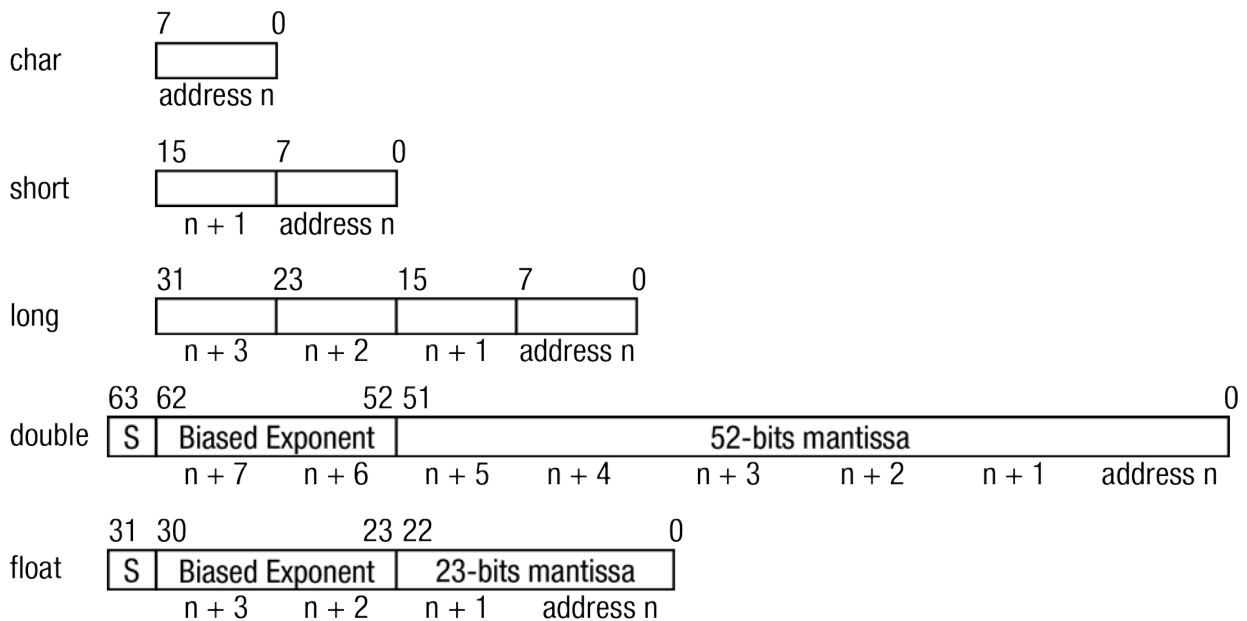
**Table 1: Field Type**

Type	Binary Size (bytes)	Description
Char	1	The <b>char</b> type is an 8-bit integer in the range -128 to +127. As a binary value, a two's complement format is used. This integer value may be the ASCII code corresponding to the specified character. In ASCII or Abbreviated ASCII this comes out as an actual character.
UChar	1	The <b>uchar</b> type is an 8-bit unsigned integer. Values are in the range from +0 to +255. In ASCII or Abbreviated ASCII this comes out as a number.
Short	2	The <b>short</b> type is 16-bit integer in the range -32768 to +32767. As a binary value, a two's complement format is used.
UShort	2	The same as short except it is not signed. Values are in the range from +0 to +65535.
Long	4	The <b>long</b> type is 32-bit integer in the range -2147483648 to +2147483647. As a binary value, a two's complement format is used.
ULong	4	The same as long except it is not signed. Values are in the range from +0 to +4294967295.
Double	8	The <b>double</b> type contains 64-bits: 1 for sign, 11 for the exponent and 52 for the mantissa. Its range is $\pm 1.7E308$ with at least 15 digits of precision. This is IEEE 754.
Float	4	The <b>float</b> type contains 32-bits: 1 for the sign, 8 for the exponent and 23 for the mantissa. Its range is $\pm 3.4E38$ with at least 7 digits of precision. This is IEEE 754.
Enum	4	A 4-byte enumerated type beginning at zero (an unsigned long). In binary, the enumerated value is output. In ASCII or Abbreviated ASCII, the enumeration label is spelled out.
GPSec	4	This type has two separate formats dependent on whether you requested a binary or an ASCII format output. For binary, the output is in milliseconds and is a <b>long</b> type. For ASCII, the output is in seconds and is a <b>float</b> type.



Type	Binary Size (bytes)	Description
Hex	n	<b>Hex</b> is a packed, fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs.
Hex Ulong	4	An unsigned, 32-bit integer in hexadecimal format. Values are in the range from +0 to +4294967295.
String	n	<b>String</b> is a variable length array of bytes that is null-terminated in the binary case and additional bytes of padding are added to maintain 4-byte alignment. The maximum byte length for each String field is shown in the row in the log or command tables.

**Figure 1: Byte Arrangements**



*Byte Arrangements* above shows the arrangement of bytes, within each field type, when used by IBM PC computers. All data sent to or from the OEM7 and SMART2 receivers is ordered least significant bit (LSB) first (little-endian). This is opposite to the most significant bit first (big-endian) ordering that is shown in *Byte Arrangements* above. Data is then stored in the receiver LSB first. For example, in char type data, the LSB is bit 0 and the most significant bit (MSB) is bit 7. See *Table 156: Channel Tracking Status* on page 806 for a more detailed example.

## 1.1 ASCII

ASCII messages are readable by both the user and a computer. The structures of all ASCII messages follow the general conventions as noted here:

1. The lead code identifier for each record is '#'.
2. Each log or command is of variable length depending on amount of data and formats.

3. All data fields are delimited by a comma ',' with two exceptions:
  - The first exception is the last header field which is followed by a ';' to denote the start of the data message.
  - The second exception is the last data field, which is followed by a '\*' to indicate end of message data.
4. Each log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters.  
 For example:  
 \*1234ABCD[CR][LF]. This value is a 32-bit CRC of all bytes in the log, excluding the '#' identifier and the asterisk preceding the eight CRC digits.  
 See *32-Bit CRC* on page 55 for the algorithm used to generate the CRC.
5. The receiver only accepts the following ASCII characters.
  - characters between space (ASCII value 32) and '~' (ASCII value 126) inclusive,
  - vertical tab (ASCII value 9)
  - line feed (ASCII value 10)
  - horizontal tab (ASCII value 11)
  - carriage return (ASCII value 13)
 Other values are discarded and can lead to unexpected results.
6. An ASCII string is one field and is surrounded by double quotation marks.  
 For example:  
 "ASCII string". If separators are surrounded by quotation marks then the string is still one field and the separator will be ignored (example, "xxx,xxx" is one field). Double quotation marks within a string are not allowed.
7. If the receiver detects an error parsing an input message, it returns an error response message. See *Responses* on page 1186 for a list of response messages from the receiver.

**Message Structure:**

header; data field..., data field..., data field... \*xxxxxxxx [CR][LF]

The ASCII message header structure is described in *Table 2: ASCII Message Header Structure* below.

**Table 2: ASCII Message Header Structure**

Field	Field Name	Field Type	Description	Ignored on Input
1	Sync	Char	Sync character. The ASCII message is always preceded by a single '#' symbol	N
2	Message	Char[]	The ASCII name of the log or command	N
3	Port	Char[]	The name of the port from which the log was generated. The string is made up of the port name followed by an _x where x is a number from 1 to 31 denoting the virtual address of the port. If no virtual address is indicated, it is assumed to be address 0	Y

Field	Field Name	Field Type	Description	Ignored on Input
4	Sequence #	Long	Used for multiple related logs. It is a number that counts down from N-1 to 0, where 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0	N
5	% Idle Time	Float	The minimum percentage of time the processor is idle, calculated once per second	Y
6	Time Status	Enum	The value indicates the quality of the GPS reference time (see <i>Table 13: GPS Reference Time Status</i> on page 53)	Y
7	Week	Ulong	GPS reference week number	Y
8	Seconds	GPSTime	Seconds from the beginning of the GPS reference week; accurate to the millisecond level	Y
9	Receiver Status	Ulong	An eight digit hexadecimal number representing the status of various hardware and software components of the receiver (see <i>Table 190: Receiver Status</i> on page 882)	Y
10	Reserved	Ulong	Reserved for internal use	Y
11	Receiver S/W Version	Ulong	A value (0 - 65535) representing the receiver software build number	Y
12	;	Char	The character indicates the end of the header	N

**Example Log:**

```
#RAWEPHEMA, COM1, 0, 55.5, SATTIME, 2072, 133140.000, 02000000, 58ba, 15761;32, 2072,
136800, 8b00602b57a606100004389101eefa4e0eed24e012f216600007608cd27,
8b00602b58282f02373454d33b986d01bd01a76ba710a2a10d008e21667f,
8b00602b58ae003384abe701001226ff6c6c1c9999f3c99fffa77c2f05c8*d3806ea3
```

## 1.2 Abbreviated ASCII

This message format is designed to make entering and viewing commands and logs simple. The data is represented as simple ASCII characters, separated by spaces or commas and arranged in an easy to understand format. There is no 32-bit CRC for error detection because it is meant for viewing by the user.

The Abbreviated ASCII message header structure is described in *Table 3: Abbreviated ASCII Message Header Structure* below.

**Table 3: Abbreviated ASCII Message Header Structure**

Field	Field Name	Field Type	Description	Ignored on Input
1	Sync	Char	Sync character. The Abbreviated ASCII message is always preceded by a single '<' symbol	N
2	Message	Char[]	The ASCII name of the log or command	N

Field	Field Name	Field Type	Description	Ignored on Input
3	Port	Char[]	The name of the port from which the log was generated. The string is made up of the port name followed by an _x where x is a number from 1 to 31 denoting the virtual address of the port. If no virtual address is indicated, it is assumed to be address 0	Y
4	Sequence #	Long	Used for multiple related logs. It is a number that counts down from N-1 to 0, where 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0	N
5	% Idle Time	Float	The minimum percentage of time the processor is idle, calculated once per second	Y
6	Time Status	Enum	The value indicates the quality of the GPS reference time (see <i>Table 13: GPS Reference Time Status</i> on page 53)	Y
7	Week	Ulong	GPS reference week number	Y
8	Seconds	GPSTime	Seconds from the beginning of the GPS reference week; accurate to the millisecond level	Y
9	Receiver Status	Ulong	An eight digit hexadecimal number representing the status of various hardware and software components of the receiver (see <i>Table 190: Receiver Status</i> on page 882)	Y
10	Reserved	Ulong	Reserved for internal use	Y
11	Receiver S/W Version	Ulong	A value (0 - 65535) representing the receiver software build number	Y
12	CR/LF	Char	A carriage return / line feed indicates the end of the header	N

**Example Command:**

```
log pppsats ontime 1
```

**Resultant Log:**

```
<PPPSATS USB1 0 51.5 FINESTEERING 2209 510815.000 12000020 ce3f 16809
< 45
< GPS 5 GOOD 00000003
< GPS 2 GOOD 00000003
< GPS 31 GOOD 00000003
< GPS 9 SUPPLEMENTARY 00000007
< GPS 18 GOOD 00000007
< GPS 6 SUPPLEMENTARY 00000007
< GPS 29 GOOD 00000003
< GPS 20 GOOD 00000003
< GPS 12 GOOD 00000003
< GPS 25 SUPPLEMENTARY 00000007
< QZSS 194 SUPPLEMENTARY 00000007
< SBAS 135 NOTUSED 00000000
< SBAS 138 NOTUSED 00000000
```

```
< SBAS 131 NOTUSED 00000000
< SBAS 133 NOTUSED 00000000
< GLONASS 19+3 GOOD 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 9-2 GOOD 00000003
< GLONASS 3+5 GOOD 00000003
< GLONASS 18-3 GOOD 00000003
< GLONASS 15 SUPPLEMENTARY 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 17+4 GOOD 00000003
< GALILEO 36 GOOD 0000000f
< GALILEO 2 GOOD 0000000f
< GALILEO 30 GOOD 0000000f
< GALILEO 34 GOOD 0000000f
< GALILEO 4 SUPPLEMENTARY 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 15 GOOD 0000000f
< GALILEO 9 GOOD 0000000f
< GALILEO 5 SUPPLEMENTARY 0000000f
< BEIDOU 25 GOOD 0000001d
< BEIDOU 34 GOOD 0000001d
< BEIDOU 23 GOOD 0000001d
< BEIDOU 22 GOOD 0000001d
< BEIDOU 44 GOOD 0000001d
< BEIDOU 43 GOOD 0000001d
< BEIDOU 42 GOOD 0000001d
< BEIDOU 28 SUPPLEMENTARY 0000001d
< BEIDOU 37 GOOD 0000001d
< BEIDOU 11 SUPPLEMENTARY 00000007
< BEIDOU 12 SUPPLEMENTARY 00000007
< BEIDOU 9 SUPPLEMENTARY 00000007
< BEIDOU 21 GOOD 0000001d
```

The array of 45 entries are offset from the left hand side and start with '<'.

## 1.3 Binary

Binary messages are strictly machine readable format. They are ideal for applications where the amount of data transmitted is fairly high. Due to the inherent compactness of binary as opposed to ASCII data, messages are much smaller. The smaller message size allows a larger amount of data to be transmitted and received by the receiver's communication ports. The structure of all binary messages follows the general conventions as noted here:

1. Basic format of:
  - *Header*: 3 Sync bytes plus 25-bytes of header information. The header length is variable as fields may be appended in the future. Always check the header length.
  - *Data*: variable
  - *CRC*: 4 bytes
2. The 3 Sync bytes will always be:



Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
5	Message ID	Ushort	This is the Message ID number of the log (see the command or log descriptions for the Message ID values of individual commands or logs)	2	4	N
6	Message Type	Char	Bits 0-4 = Measurement source <sup>1</sup> Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response bit (see <i>Message Responses</i> on page 49) 0 = Original Message 1 = Response Message	1	6	N
7	Port Address	Uchar	See <i>Table 5: Detailed Port Identifier</i> on the next page (decimal values >=32 may be used) (lower 8-bits only) <sup>2</sup>	1	7	N <sup>3</sup>
8	Message Length	Ushort	The length in bytes of the body of the message, not including the header nor the CRC	2	8	N
9	Sequence	Ushort	Used for multiple related logs. It is a number that counts down from N-1 to 0 where N is the number of related logs and 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0	2	10	N

<sup>1</sup>Bits 0-4 are used to indicate the measurement source. For dual antenna receivers, if bit 0 is set, the log is from the secondary antenna.

<sup>2</sup>The 8-bit size means you will only see 0xA0 to 0xBF when the top bits are dropped from a port value greater than 8-bits. For example, ASCII port USB1 will be seen as 0xA0 in the binary output.

<sup>3</sup>Recommended value is THISPORT (decimal 192).

Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
10	Idle Time	Uchar	Time the processor is idle, calculated once per second. Take the time (0 – 200) and divide by two to give the percentage of time (0 – 100%). 0% indicates the processor is fully occupied. Other values indicate the availability of the processor to take on tasks.	1	12	Y
11	Time Status	Enum	Indicates the quality of the GPS reference time (see <i>Table 13: GPS Reference Time Status</i> on page 53).	1 <sup>1</sup>	13	N <sup>2</sup>
12	Week	Ushort	GPS reference week number	2	14	N
13	ms	GPSec	Milliseconds from the beginning of the GPS reference week	4	16	N
14	Receiver Status	Ulong	32-bits representing the status of various hardware and software components of the receiver (see <i>Table 190: Receiver Status</i> on page 882)	4	20	Y
15	Reserved	Ushort	Reserved for internal use	2	24	Y
16	Receiver S/W Version	Ushort	A value (0 - 65535) representing the receiver software build number	2	26	Y

Table 5: Detailed Port Identifier

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
NO_PORTS	0	0	No ports specified
COM1	20	32	COM1
COM2	40	64	COM2
COM3	60	96	COM3
SPECIAL	a0	160	Unknown port
THISPORT	c0	192	Current COM port
FILE	e0	224	Port for logging to a file
USB1	5a0	1440	USB1

<sup>1</sup>This ENUM is not 4-bytes long but, as indicated in the table, is only 1-byte.

<sup>2</sup>Fields 12 and 13 (Week and ms) are ignored if Field 11 (Time Status) is invalid. In this case, the current receiver time is used. The recommended values for the three time fields are 0, 0, 0.



ASCII Port Name	Hex Port Value	Decimal Port Value	Description
USB2	6a0	1696	USB2
USB3	7a0	1952	USB3
AUX	8a0	2208	AUX port
COM4	ba0	2976	COM4
ETH1	ca0	3232	ETH1
IMU	da0	3488	IMU
ICOM1	fa0	4000	ICOM1
ICOM2	10a0	4256	ICOM2
ICOM3	11a0	4512	ICOM3
NCOM1	12a0	4768	NCOM1
NCOM2	13a0	5024	NCOM2
NCOM3	14a0	5280	NCOM3
ICOM4	15a0	5536	ICOM4
WCOM1	16a0	5792	WCOM1
COM5	17a0	6048	COM5
COM6	18a0	6304	COM6
BT1	19a0	6560	Bluetooth device
COM7	1aa0	6816	COM7
COM8	1ba0	7072	COM8
COM9	1ca0	7328	COM9
COM10	1da0	7584	COM10
CCOM1	1ea0	7840	CAN COM1
CCOM2	1fa0	8096	CAN COM2
CCOM3	20a0	8352	CAN COM3
CCOM4	21a0	8608	CAN COM4
CCOM5	22a0	8864	CAN COM5
CCOM6	23a0	9120	CAN COM6
ICOM5	26a0	9888	ICOM5
ICOM6	27a0	10144	ICOM6

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
ICOM7	28a0	10400	ICOM7
SCOM1	29a0	10656	SCOM1
SCOM2	2aa0	10912	SCOM2
SCOM3	2ba0	11168	SCOM3
SCOM4	2ca0	11424	SCOM4



The ports available vary based on the receiver.

### 1.3.1 Virtual Ports

Virtual ports can be used to create groups of logs which are all sent to the same physical port.

To remove logging to virtual ports, the virtual port must be specified or the \_ALL port must be used. For example:

```
LOG COM2_1 BESTPOS ONTIME 1
LOG COM2_2 TRACKSTAT ONTIME 10
```

Logs on the virtual ports will not be cleared by UNLOGALL COM2. To clear the logs COM2\_1, COM2\_2 or COM2\_ALL must be used.

**Table 6: Detailed Port Identifier – Virtual Ports**

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
COM1_ALL	1	1	All virtual ports for COM1
COM2_ALL	2	2	All virtual ports for COM2
COM3_ALL	3	3	All virtual ports for COM3
THISPORT_ALL	6	6	All virtual ports for the current port
FILE_ALL	7	7	All virtual ports for logging to file
ALL_PORTS	8	8	All virtual ports for all ports
USB1_ALL	d	13	All virtual ports for USB1
USB2_ALL	e	14	All virtual ports for USB2
USB3_ALL	f	15	All virtual ports for USB3
AUX_ALL	10	16	All virtual ports for the AUX
COM4_ALL	13	19	All virtual ports for COM4
ETH1_ALL	14	20	All virtual ports for ETH1
IMU_ALL	15	21	All virtual ports for IMU

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
ICOM1_ALL	17	23	All virtual ports for ICOM1
ICOM2_ALL	18	24	All virtual ports for ICOM2
ICOM3_ALL	19	25	All virtual ports for ICOM3
NCOM1_ALL	1a	26	All virtual ports for NCOM1
NCOM2_ALL	1b	27	All virtual ports for NCOM2
NCOM3_ALL	1c	28	All virtual ports for NCOM3
ICOM4_ALL	1d	29	All virtual ports for ICOM4
WCOM1_ALL	1e	30	All virtual ports for WCOM1
COM1_1	21	33	COM1, virtual port 1
...			
COM1_31	3f	63	COM1, virtual port 31
COM2_1	41	65	COM1, virtual port 1
...			
COM2_31	5f	95	COM2, virtual port 31
COM3_1	61	97	COM3, virtual port 1
...			
COM3_31	7f	127	COM3, virtual port 31
SPECIAL_1	a1	161	Unknown port, virtual port 1
...			
SPECIAL_31	bf	191	Unknown port, virtual port 31
THISPORT_1	c1	193	Current COM port, virtual port 1
...			
THISPORT_31	df	223	Current COM port, virtual port 31
FILE_1	e1	225	Virtual port 1 for logging to file
...			
FILE_31	ff	255	Virtual port 31 for logging to file
USB1_1	5a1	1441	USB1, virtual port 1
...			
USB1_31	5bf	1471	USB1, virtual port 31

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
USB2_1	6a1	1967	USB2, virtual port 1
...			
USB2_31	6bf	1727	USB2, virtual port 31
USB3_1	7a1	1953	USB3, virtual port 1
...			
USB3_31	7bf	1983	USB port 3, virtual port 31
AUX_1	8a1	2209	AUX port, virtual port 1
...			
AUX_31	8bf	2239	AUX port, virtual port 31
COM4_1	ba1	2977	COM4, virtual port 1
...			
COM4_31	bbf	3007	COM4, virtual port 31
ETH1_1	ca1	3233	ETH1, virtual port 1
...			
ETH1_31	cbf	3263	ETH1, virtual port 31
IMU_1	da1	3489	IMU, virtual port 1
...			
IMU_31	dbf	3519	IMU, virtual port 31
ICOM1_1	fa1	4001	ICOM1, virtual port 1
...			
ICOM1_31	fbf	4031	ICOM1, virtual port 31
ICOM2_1	10a1	4257	ICOM2, virtual port 1
...			
ICOM2_31	10bf	4287	ICOM2, virtual port 31
ICOM3_1	11a1	4513	ICOM3, virtual port 1
...			
ICOM3_31	11bf	4543	ICOM3, virtual port 31
NCOM1_1	12a1	4769	NCOM1, virtual port 1
...			

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
NCOM1_31	12bf	4799	NCOM1, virtual port 31
NCOM2_1	13a1	5025	NCOM2, virtual port 1
...			
NCOM2_31	13bf	5055	NCOM2, virtual port 31
NCOM3_1	14a1	5281	NCOM3, virtual port 1
...			
NCOM3_31	14bf	5311	NCOM3, virtual port 31
ICOM4_1	15a1	5537	ICOM4, virtual port 1
...			
ICOM4_31	15bf	5567	ICOM4, virtual port 31
WCOM1_1	16a1	5793	WCOM1, virtual port 1
...			
WCOM1_31	16bf	5823	WCOM1, virtual port 31
COM5_ALL	16c0	5824	All virtual ports for COM5
COM6_ALL	16c1	5825	All virtual ports for COM6
BT1_ALL	16c2	5826	All virtual ports for the Bluetooth device
COM7_ALL	16c3	5827	All virtual ports for COM7
COM8_ALL	16c4	5828	All virtual ports for COM8
COM9_ALL	16c5	5829	All virtual ports for COM9
COM10_ALL	16c6	5830	All virtual ports for COM10
CCOM1_ALL	16c7	5831	All virtual ports for CCOM1
CCOM2_ALL	16c8	5832	All virtual ports for CCOM2
CCOM3_ALL	16c9	5833	All virtual ports for CCOM3
CCOM4_ALL	16ca	5834	All virtual ports for CCOM4
CCOM5_ALL	16cb	5835	All virtual ports for CCOM5
CCOM6_ALL	16cc	5836	All virtual ports for CCOM6
ICOM5_ALL	16cf	5839	All virtual ports for ICOM5
ICOM6_ALL	16d0	5840	All virtual ports for ICOM6
ICOM7_ALL	16d1	5841	All virtual ports for ICOM7

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
SCOM1_ALL	16d2	5842	All virtual ports for SCOM1
SCOM2_ALL	16d3	5843	All virtual ports for SCOM2
SCOM3_ALL	16d4	5844	All virtual ports for SCOM3
SCOM4_ALL	16d5	5845	All virtual ports for SCOM4
COM5_1	17a1	6049	COM5, virtual port 1
...			
COM5_31	17bf	6079	COM5, virtual port 31
COM6_1	18a1	6305	COM6, virtual port 1
...			
COM6_31	18bf	6335	COM6, virtual port 31
BT1_1	19a1	6561	Bluetooth device, virtual port 1
...			
BT1_31	19bf	6591	Bluetooth device, virtual port 31
COM7_1	1aa1	6817	COM7, virtual port 1
...			
COM7_31	1abf	6847	COM7, virtual port 31
COM8_1	1ba1	7073	COM8, virtual port 1
...			
COM8_31	1bbf	7103	COM8, virtual port 31
COM9_1	1ca1	7329	COM9, virtual port 1
...			
COM9_31	1cbf	7359	COM9, virtual port 31
COM10_1	1da1	7585	COM10, virtual port 1
...			
COM10_31	1dbf	7615	COM10, virtual port 31
CCOM1_1	1ea1	7841	CAN COM1, virtual port 1
...			
CCOM1_31	1ebf	7871	CAN COM1, virtual port 31
CCOM2_1	1fa1	8097	CAN COM2, virtual port 1

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
...			
CCOM2_31	1fbf	8127	CAN COM2, virtual port 31
CCOM3_1	20a1	8353	CAN COM3, virtual port 1
...			
CCOM3_31	20bf	8383	CAN COM3, virtual port 31
CCOM4_1	21a1	8609	CAN COM4, virtual port 1
...			
CCOM4_31	21bf	8639	CAN COM4, virtual port 31
CCOM5_1	22a1	8865	CAN COM5, virtual port 1
...			
CCOM5_31	22bf	8895	CAN COM5, virtual port 31
CCOM6_1	23a1	9121	CAN COM6, virtual port 1
...			
CCOM6_31	23bf	9151	CAN COM6, virtual port 31
ICOM5_1	26a1	9889	ICOM5, virtual port 1
...			
ICOM5_31	26bf	9919	ICOM5, virtual port 31
ICOM6_1	27a1	10145	ICOM6, virtual port 1
...			
ICOM6_31	27bf	10175	ICOM6, virtual port 31
ICOM7_1	28a1	10401	ICOM7, virtual port 1
...			
ICOM7_31	28bf	10431	ICOM7, virtual port 31
SCOM1_1	29a1	10657	SCOM1, virtual port 1
...			
SCOM1-31	29bf	10687	SCOM1, virtual port 31
SCOM2_1	2aa1	10913	SCOM2, virtual port 1
...			
SCOM2_31	2abf	10943	SCOM2, virtual port 31

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
SCOM3_1	2ba1	11169	SCOM3, virtual port 1
...			
SCOM3_31	2bbf	11199	SCOM3, virtual port 31
SCOM4_1	2ca1	11425	SCOM4, virtual port 1
...			
SCOM4_31	2cbf	11455	SCOM4, virtual port 31



COM1\_ALL, COM2\_ALL, COM3\_ALL, COM4\_ALL, COM5\_ALL, THISPORT\_ALL, FILE\_ALL, ALL\_PORTS, USB1\_ALL, USB2\_ALL, USB3\_ALL, AUX\_ALL, ETH1\_ALL, ICOM1\_ALL, ICOM2\_ALL, ICOM3\_ALL, ICOM4\_ALL, ICOM5\_ALL, ICOM6\_ALL, ICOM7\_ALL, CCOM1\_ALL, CCOM2\_ALL, CCOM3\_ALL, CCOM4\_ALL, CCOM5\_ALL, CCOM6\_ALL, NCOM1\_ALL, NCOM2\_ALL, NCOM3\_ALL, SCOM1\_ALL, SCOM2\_ALL, SCOM3\_ALL, SCOM4\_ALL and WCOM1\_ALL are only valid for the **UNLOGALL** command.

Table 7: Available Port Types below provides examples of where each port type might be used.

**Table 7: Available Port Types**

Port Type	Description	Example of where it might be used
AUX	Auxiliary "serial" ports	An additional UART serial port available only on certain platforms
BTx	Bluetooth ports	These ports are used to connect over Bluetooth devices, when the receiver is equipped with a BT device
COMx	Serial Port	UART serial ports. Used when there is a physical RS-232 or RS-422 connection to the receiver
ICOMx	Internet ports	These ports are used when establishing TCP or UDP connections to the receiver over a network
NCOMx	NTRIP ports	These ports are used when establishing NTRIP connections to the receiver over a network
SCOMx	Script ports	Ports used by the Scripted User Interface (i.e. Lua)
USBx	USB "serial" ports	When the receiver is connected to an external host through USB, these ports are available
WCOMx	Web Server port	Ports used by Web Server applications, for receivers equipped with a web server

## 1.4 Description of ASCII and Binary Logs with Short Headers

One difference from the standard OEM7 logs is there are two possible headers for the ASCII and binary versions of the INS logs. Which header is used for a given log is described in the log definitions in the *SPAN Logs* on page 1058 chapter. The reason for the alternate short headers is that the normal OEM7 binary header



is quite long at 28 bytes. This is nearly as long as the data portion of many of the INS logs and creates excess storage and baud rate requirements.

These logs are set up in the same way as normal ASCII or binary logs except a normal ASCII or binary header is replaced with a short header (see *Table 8: Short ASCII Message Header Structure* below and *Table 9: Short Binary Message Header Structure* below).

**Table 8: Short ASCII Message Header Structure**

Field	Field Name	Field Type	Description
1	%	Char	% symbol
2	Message	Char	This is the name of the log
3	Week Number	Ushort	GNSS week number
4	Seconds	GPSec	Seconds from the beginning of the GNSS week (Same byte arrangement as a Float type)

**Table 9: Short Binary Message Header Structure**

Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset
1	Synch	Char	Hex 0xAA	1	0
2	Synch	Char	Hex 0x44	1	1
3	Synch	Char	Hex 0x13	1	2
4	Message Length	Uchar	Message length, not including header or CRC	1	3
5	Message ID	Ushort	Message ID number	2	4
6	Week Number	Ushort	GNSS week number	2	6
7	Milliseconds	GPSec	Milliseconds from the beginning of the GNSS week (Same byte arrangement as a Long type)	4	8

## 1.5 Message Responses

By default, if you input a message you get back a response. If desired, the **INTERFACEMODE** command (see page 216) can be used to disable response messages. The response will be in the exact format you entered the message (that is, binary input = binary response).

### 1.5.1 Abbreviated ASCII Response

The response is just the leading '<' followed by the response string, for example: <OK.

### 1.5.2 ASCII Response

The response is the full header with the message name being identical except ending in an 'R' (for response). The body of the message consists of a string of characters for the response. For example:

```
#FRESETR, COM1, 0, 73.0, UNKNOWN, 0, 0.000, 00000000, 06e5, 0; OK*55c70910
```

The prefix "ERROR:" is added to the response message (see *Table 264: Response Messages* on page 1186) except when the command is accepted. When the command is accepted, the message response is "OK".

### 1.5.3 Binary Response

The response is similar to an ASCII response except that it follows the binary protocols, see *Table 10: Binary Message Response Structure* below.

*Table 11: Binary Message Sequence* on the next page is an example of the sequence for requesting and then receiving BESTPOSB. The example is in hex format. When you enter a hex command, you may need to add a 'x' or '0x' before each hex pair, depending on your code. For example:

**0xAA0x440x120x1C0x010x000x02** and so on.

**Table 10: Binary Message Response Structure**

	Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset
B I N A R Y	1	Sync	Char	Hexadecimal 0xAA	1	0
	2	Sync	Char	Hexadecimal 0x44	1	1
	3	Sync	Char	Hexadecimal 0x12	1	2
	4	Header Length	Uchar	Length of the header	1	3
	5	Message ID	Ushort	Message ID number	2	4
	6	Message Type	Char	Response Bit 1 = Response Message	1	6
	7	Port Address	Uchar	See <i>Table 5: Detailed Port Identifier</i> on page 40	1	7
H E A D E R	8	Message Length	Ushort	The length in bytes of the body of the message (not including the CRC)	2	8
	9	Sequence	Ushort	Normally 0	2	10
	10	Idle Time	Uchar	Idle time	1	12
	11	Time Status	Enum	<i>Table 13: GPS Reference Time Status</i> on page 53	1 <sup>1</sup>	13
	12	Week	Ushort	GPS reference week number	2	14
	13	ms	GPSTime	Milliseconds into GPS reference week	4	16
	14	Receiver Status	Ulong	<i>Table 190: Receiver Status</i> on page 882	4	20
	15	Reserved	Ushort	Reserved	2	24

<sup>1</sup>This ENUM is not 4-bytes long but as indicated in the table is only 1 byte.

	Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset
	16	Receiver S/W Version	Ushort	Receiver software build number	2	26
I D	17	Response ID	Enum	The enumeration value corresponding to the response message ( <i>Table 264: Response Messages</i> on page 1186)	4	28
H E X	18	Response	Hex	String containing the ASCII response in hex coding to match the ID above (for example, 0x4F4B = OK)  The prefix "ERROR:" is added to the response message (see <i>Table 264: Response Messages</i> on page 1186) except when the command is accepted. When the command is accepted, the message response is "OK".	variable	32

Table 11: Binary Message Sequence

Direction	Sequence	Data
To Receiver	LOG Command Header	AA44121C 01000240 20000000 1D1D0000 29160000 00004C00 55525A80
	LOG Parameters	20000000 2A000000 02000000 00000000 0000F03F 00000000 00000000
	Checksum	2304B3F1
From Receiver	LOG Response Header	AA44121C 01008220 06000000 FFB4EE04 605A0513 00004C00 FFFF5A80
	LOG Response Data	01000000 4F4B
	Checksum	DA8688EC
From Receiver	BESTPOSB Header	AA44121C 2A000220 48000000 90B49305 B0ABB912 00000000 4561BC0A
	BESTPOSB Data	00000000 10000000 1B0450B3 F28E4940 16FA6BBE 7C825CC0 0060769F 449F9040 A62A82C1 3D000000 125ACB3F CD9E983F DB664040 00303030 00000000 00000000 0B0B0000 00060003
	Checksum	42DC4C48

## 1.6 GLONASS Slot and Frequency Numbers

When a GLONASS PRN in a log is in the range 38 to 61, then that PRN represents a GLONASS Slot Number where the Slot Number shown is the actual GLONASS Slot Number plus 37.

Similarly, the GLONASS Frequency shown in logs is the actual GLONASS Frequency plus 7.

For example:

```
<RANGE COM1 0 82.0 FINESTEERING 1729 155076.000 02004000 5103 11465
46
31 0 24514687.250 0.064 -128825561.494675 0.010 3877.473 45.0 563.310
18109c04
...
46 5 24097664.754 0.213 -128680178.570435 0.014 -3740.543 40.6 10098.600
08119e44
...
8 0 39844800.076 0.043 -160438471.200694 0.013 -392.547 42.5 12038.660
00349c84
```

when 31 is a GPS satellite, 8 is a BeiDou satellite and 46 is a GLONASS satellite. Its actual GLONASS Slot Number is 9 and its frequency is -2.

Refer to *PRN Numbers* below for more information about GLONASS PRN numbers. Also, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss) for more information.

### 1.6.1 PRN Numbers

The PRN and SVID ranges for the logs and commands that use them are shown in the following table.

**Table 12: PRN Numbers for Commands and Logs**

Command/Log	GPS PRN	SBAS PRN	SBAS QZSS L1S PRN	GLONASS Slot	Galileo SVID	QZSS PRN	BDS PRN	NavIC PRN
ASSIGN	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
ASSIGNALL	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
LOCKOUTSATELLITE	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
SBASCONTROL	-	120-158	183-192	-	-	-	-	-
TRACKSV	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
UNLOCKOUTSATELLITE	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
RANGE	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
RANGECMP	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14
RANGECMP2	1-32	120-158	183-192	1-24	1-36	193-202	1-63	1-14
RANGECMP4	1-32	120-158	183-192	1-24	1-36	193-202	1-63	1-14
RANGEGPSL1	1-32	-	-	-	-	-	-	-
SATVIS2	1-32	120-158	183-192	1-24	1-36	193-202	1-63	1-14
TRACKSTAT	1-32	120-158	183-192	38-61	1-36	193-202	1-63	1-14

## 1.7 GPS Reference Time Status

All reported receiver times are subject to a qualifying time status. The status indicates how well a time is known (see *Table 13: GPS Reference Time Status* below).

**Table 13: GPS Reference Time Status**

GPS Reference Time Status (Decimal)	GPS Reference Time Status (ASCII)	Description
20	UNKNOWN	Time validity is unknown
60	APPROXIMATE	Time is set approximately
80	COARSEADJUSTING	Time is approaching coarse precision
100	COARSE	This time is valid to coarse precision
120	COARSESTEERING	Time is coarse set and is being steered
130	FREEWHEELING	Position is lost and the range bias cannot be calculated
140	FINEADJUSTING	Time is adjusting to fine precision
160	FINE	Time has fine precision
170	FINEBACKUPSTEERING	Time is fine set and is being steered by the backup system
180	FINESTEERING	Time is fine set and is being steered
200	SATTIME	Time from satellite. Only used in logs containing satellite data such as ephemeris and almanac

There are several distinct states the receiver goes through.

When the **CLOCKADJUST** command (see page 115) is enabled:

- UNKNOWN (initial state)
- COARSESTEERING (initial coarse time set)
- FINESTEERING (normal operating state)
- FINEBACKUPSTEERING (when the backup system is used for a time)
- FREEWHEELING (when range bias becomes unknown)

When the **CLOCKADJUST** command (see page 115) is disabled:

- UNKNOWN (initial state)
- COARSE (initial coarse time set)
- FINE (normal operating state)

On startup and before any satellites are tracked, the receiver can not possibly know the current time. As such, the receiver time starts counting at GPS reference week 0 and second 0.0. The time status flag is set to UNKNOWN.

If time is input to the receiver using the **SETAPPROXTIME** command (see page 390), the time status will be APPROXIMATE.

After time information is decoded from the navigation data and validated, the receiver time is set with an accuracy of approximately  $\pm 10$  milliseconds. This state is qualified by the COARSE or COARSESTEERING time status flag depending on the state of the CLOCKADJUST switch (for more information, refer to the **CLOCKADJUST** command on page 115).

Once a position is known and range biases are being calculated, the internal clock model will begin modeling the range biases also known as the receiver clock offset.

Modeling will continue until the model is a good estimation of the actual receiver clock behavior. At this time, the receiver time will again be adjusted, this time to an accuracy of  $\pm 1$  microsecond. This state is qualified by the FINE time status flag.

The final logical time status flag depends on whether **CLOCKADJUST** is enabled or not. If CLOCKADJUST is disabled, the time status flag will never improve on FINE. The time will only be adjusted again to within  $\pm 1$  microsecond if the range bias gets larger than  $\pm 250$  milliseconds. If CLOCKADJUST is enabled, the time status flag is set to FINESTEERING and the receiver time is continuously updated (steered) to minimize the receiver range bias.

If a solution cannot be computed with the primary satellite system, it will attempt to use a backup system (if available). When the backup system is used and time is computed, the time status is set to FINEBACKUPSTEERING. If the position is lost and the range bias cannot be calculated, the time status is degraded to FREEWHEELING.



See also *Message Time Stamps* below and the **SETTIMEBASE** command on page 403.

## 1.8 Message Time Stamps

All NovAtel format messages generated by the OEM7 and SMART2 receivers have a GPS reference time stamp in their header. GPS reference time is referenced to UTC with zero point defined as midnight on the night of January 5, 1980. The time stamp consists of the number of weeks since that zero point and the number of seconds since the last week number change (0 to 604,799). GPS reference time differs from UTC time since leap seconds are occasionally inserted into UTC and GPS reference time is continuous. In addition, a small error (less than 1 microsecond) can exist in synchronization between UTC and GPS reference time. The TIME log reports both GNSS and UTC time and the offset between the two.

The data in synchronous logs (for example, RANGE, BESTPOS, TIME) are based on a periodic measurement of satellite pseudoranges. The time stamp on these logs is the receiver estimate of GPS reference time at the time of the measurement.

Other log types (asynchronous and polled) are triggered by an external event and the time in the header may not be synchronized to the current GPS reference time. Logs that contain satellite broadcast data (for example, ALMANAC, GPSEPHEN) have the transmit time of their last subframe in the header. In the header of differential time matched logs (for example, MATCHEDPOS) is the time of the matched reference and local observation that they are based on. Logs triggered by a mark event (for example, MARKPOS, MARK1TIME) have the estimated GPS reference time of the mark event in their header. In the header of polled logs (for example, LOGLIST, PORTSTATS, VERSION) is the approximate GPS reference time when their data was generated. However, when asynchronous logs are triggered ONTIME, the time stamp will represent the time the log was generated and not the time given in the data.

For more information about log types, see *Log Types* on page 481.

## 1.9 Decoding of the GPS Reference Week Number

The GPS reference week number provided in the raw satellite data is the 10 least significant bits (or 8 least significant bits in the case of the almanac data) of the full week number. When the receiver processes the satellite data, the week number is decoded as defined in the log description. In all log headers, the full week number is given starting from week 0 or January 6, 1980.

## 1.10 32-Bit CRC

The ASCII and Binary OEM7 family and SMART2 message formats all contain a 32-bit CRC for data verification. This allows the user to ensure the data received (or transmitted) is valid with a high level of certainty.

The C functions below may be implemented to generate the CRC of a block of data.

```
#define CRC32_POLYNOMIAL 0xEDB88320L
/* -----
Calculate a CRC value to be used by CRC calculation functions.
----- */
unsigned long CRC32Value(int i) {
    int j;
    unsigned long ulCRC;
    ulCRC = i;
    for ( j = 8 ; j > 0; j-- ) {
        if ( ulCRC & 1 )
            ulCRC = ( ulCRC >> 1 ) ^ CRC32_POLYNOMIAL;
        else
            ulCRC >>= 1;
    }
    return ulCRC;
}

/* -----
Calculates the CRC-32 of a block of data all at once
ulCount - Number of bytes in the data block
ucBuffer - Data block
----- */
unsigned long CalculateBlockCRC32( unsigned long ulCount, unsigned char
*ucBuffer ) {
    unsigned long ulTemp1;
    unsigned long ulTemp2;
    unsigned long ulCRC = 0;
    while ( ulCount-- != 0 ) {
        ulTemp1 = ( ulCRC >> 8 ) & 0x00FFFFFFL;
        ulTemp2 = CRC32Value( ((int) ulCRC ^ *ucBuffer++ ) & 0xFF );
        ulCRC = ulTemp1 ^ ulTemp2;
    }
    return( ulCRC );
}
```



The NMEA checksum is an XOR of all the bytes (including delimiters such as ',' but excluding the \* and \$) in the message output. It is therefore an 8-bit and not a 32-bit checksum.

Not all logs may be available. Every effort is made to ensure examples are correct, however, a checksum may be created for promptness in publication. In this case it will appear as '9999'.

**Example:**

BESTPOSB and BESTPOSA from an OEM7 or SMART2 receiver.

**Binary Log Message:**

```
0xAA, 0x44, 0x12, 0x1C, 0x2A, 0x00, 0x02, 0x20, 0x48, 0x00, 0x00, 0x00,
0x90, 0xB4, 0x93, 0x05, 0xB0, 0xAB, 0xB9, 0x12, 0x00, 0x00, 0x00, 0x00,
0x45, 0x61, 0xBC, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00,
0x1B, 0x04, 0x50, 0xB3, 0xF2, 0x8E, 0x49, 0x40, 0x16, 0xFA, 0x6B, 0xBE,
0x7C, 0x82, 0x5C, 0xC0, 0x00, 0x60, 0x76, 0x9F, 0x44, 0x9F, 0x90, 0x40,
0xA6, 0x2A, 0x82, 0xC1, 0x3D, 0x00, 0x00, 0x00, 0x12, 0x5A, 0xCB, 0x3F,
0xCD, 0x9E, 0x98, 0x3F, 0xDB, 0x66, 0x40, 0x40, 0x00, 0x30, 0x30, 0x30,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x0B, 0x00, 0x00,
0x00, 0x06, 0x00, 0x03, 0x42, 0xdc, 0x4c, 0x48
```

Below is a demonstration of how to generate the CRC from both ASCII and BINARY messages using the function described above.



When you pass the data into the code that follows, exclude the checksum shown in ***bold italics*** above. It is ***42dc4c48***.

**Binary Checksum Calculation:**

```
#include <stdio.h>
#include <string.h>
#include <inttypes.h>

void main() {
    // Expect checksum 0x42, 0xDC, 0x4C, 0x48 (42dc4c48)
    unsigned char buffer[] = {0xAA, 0x44, 0x12, 0x1C, 0x2A, 0x00, 0x02, 0x20,
0x48, 0x00,
                                0x00, 0x00, 0x90, 0xB4, 0x93, 0x05, 0xB0, 0xAB,
0xB9, 0x12,
                                0x00, 0x00, 0x00, 0x00, 0x45, 0x61, 0xBC, 0x0A,
0x00, 0x00,
                                0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x1B, 0x04,
0x50, 0xB3,
                                0xF2, 0x8E, 0x49, 0x40, 0x16, 0xFA, 0x6B, 0xBE,
0x7C, 0x82,
                                0x5C, 0xC0, 0x00, 0x60, 0x76, 0x9F, 0x44, 0x9F,
0x90, 0x40,
                                0xA6, 0x2A, 0x82, 0xC1, 0x3D, 0x00, 0x00, 0x00,
0x12, 0x5A,
                                0xCB, 0x3F, 0xCD, 0x9E, 0x98, 0x3F, 0xDB, 0x66,
0x40, 0x40,
                                0x00, 0x30, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
                                0x00, 0x00, 0x0B, 0x0B, 0x00, 0x00, 0x00, 0x06,
0x00, 0x03};
```



```
//Note that the CRC on the binary data will be little-endian ordered.
unsigned long CRCle = CalculateBlockCRC32(sizeof(buffer), buffer);

//big-endian users (such as Atmel AVR32 users) may swap endianness as
follows
unsigned long CRCbe = __builtin_bswap32(CRCle);

printf("\n\n%s %lx \n", "Computed binary checksum (little-endian): ",
CRCle);
printf("%s %" PRIx32 "\n", "Computed binary checksum (big-endian): ",
CRCbe);

}
```

Note that the above checksum function (`CalculateBlockCRC32`) must also be included to execute this code.

### ASCII Log Message:

```
#BESTPOSA,COM1,0,78.0,FINESTEERING,1427,325298.000,00000000,6145,2748;
SOL_COMPUTED,SINGLE,51.11678928753,-114.03886216575,1064.3470,-16.2708,
WGS84,2.3434,1.3043,4.7300,"",0.000,0.000,7,7,0,0,0,06,0,03*9c9a92bb
```

The checksum for this log is given above, it is `9c9a92bb`.

### ASCII:

```
#include <stdio.h>
#include <string.h>

void main() {
    //Remember to escape " characters as \"
    char *msgBlock =
    "BESTPOSA,COM1,0,78.0,FINESTEERING,1427,325298.000,00000000,\
    6145,2748;SOL_COMPUTED,SINGLE,51.11678928753,-114.03886216575,\
    1064.3470,-\
    16.2708,WGS84,2.3434,1.3043,4.7300,\"\",0.000,0.000,7,7,0,0,0,06,0,03";

    unsigned long CRC = CalculateBlockCRC32(strlen(msgBlock), (unsigned
char*)msgBlock);

    printf("\n\n%s %s\n", "Demonstrating CRC computed for the block:",
msgBlock);
    printf("\n\n%s %lu\n", "CRC32 in Decimal is: ", CRC);
    printf("%s %lx\n", "CRC32 in Hex is: ", CRC);
}
```

Note that the above checksum function (`CalculateBlockCRC32`) must also be included to execute this code.

# Chapter 2 Commands

Commands are used to control and configure OEM7 receivers.

## 2.1 Command Formats

The receiver accepts commands in 3 formats as described in *Messages* on page 32:

- Abbreviated ASCII
- ASCII
- Binary

Abbreviated ASCII is the easiest to use for your input. The other two formats include a CRC for error checking and are intended for use when interfacing with other electronic equipment.

The following are examples of the same command in each format:

### Abbreviated ASCII Example:

```
LOG COM1 BESTPOSB ONTIME 1[CR]
```

### ASCII Example:

```
#LOGA,THISPORT,0,0,UNKNOWN,0,0.0,0,0,0;COM1,BESTPOSB,ONTIME,1.000000,0.000000,N  
OHOLD*ec9ce601[CR]
```

### Binary Example:

```
AA44121C 010000C0 20000000 00FF0000 00000000 00000000 00000000 20000000  
2A000000 02000000 00000000 0000F03F 00000000 00000000 00000000 34D32DC1
```

### 2.1.1 Optional Parameters

Many commands have nested optional parameters where an optional parameter requires the optional parameter before it to be present. This is noted in the Abbreviated ASCII Syntax as:

```
Command [OPT_1 [OPT_2 [OPT_3]]]
```

In this syntax example, OPT\_1 and OPT\_2 must be provided if you want to provide a value for OPT\_3. These leading two options are required even if you want to use the defaults for OPT\_1 and OPT\_2.

## 2.2 Command Settings

There are several ways to determine the current command settings of the receiver:

1. Request an **RXCONFIG** log (see page 876). This log provides a listing of all commands issued to the receiver and their parameter settings. It also provides the most complete information.
2. For some specific commands, logs are available to indicate all their parameter settings. The **LOGLIST** log (see page 687) shows all active logs in the receiver beginning with the **LOG** command (see page 247).
3. Request a log of the specific command of interest to show the parameters last entered for that command. The format of the log produced is exactly the same as the format of the specific command with updated header information.



Requesting a log for specific command is useful for most commands. For commands repeated with different parameters (for example, **SERIALCONFIG** and **LOG**), only the most recent set of parameters used is shown. To view all sets of parameters, try method 1 or 2 above.

#### Abbreviated ASCII Example:

```
log fix
<FIX COM1 0 49.5 FINESTEERING 2072 405099.446 02000008 37d7 15761
<      NONE -10000.000000000000 -10000.000000000000 -10000.0000
```

## 2.3 Command Defaults

When the receiver is first powered up or after a **FRESET** command (see page 193), all commands revert to their factory default settings. When you use a command without specifying its optional parameters, it may have a different command default than the factory default. The **SAVECONFIG** command (see page 366) can be used to save these defaults. Use the **RXCONFIG** log (see page 876) to reference any default command and log settings.

Factory default settings for individual commands are stated in the following commands, organized alphabetically by command name.



**FRESET STANDARD** causes all previously stored user configurations saved to non-volatile memory to be erased (including Saved Config, Saved Almanac, Saved Ephemeris and L-Band-related data, excluding subscription information).



Ensure that all windows, other than the Console window, are closed in NovAtel's Connect user interface application before you issue the **SAVECONFIG** command (see page 366).

## 2.4 ADJUST1PPS

### *Adjusts the receiver clock*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to adjust the receiver clock or as part of the procedure to transfer time between receivers. The number of pulses per second (PPS) is always set to 1 Hz with this command. It is typically used when the receiver is not adjusting its own clock and is using an external reference frequency.

To disable the automatic clock adjustment, refer to the **CLOCKADJUST** command on page 115. To configure the receiver to use an external reference oscillator, see the **EXTERNALCLOCK** command on page 168.

The **ADJUST1PPS** command can be used to:

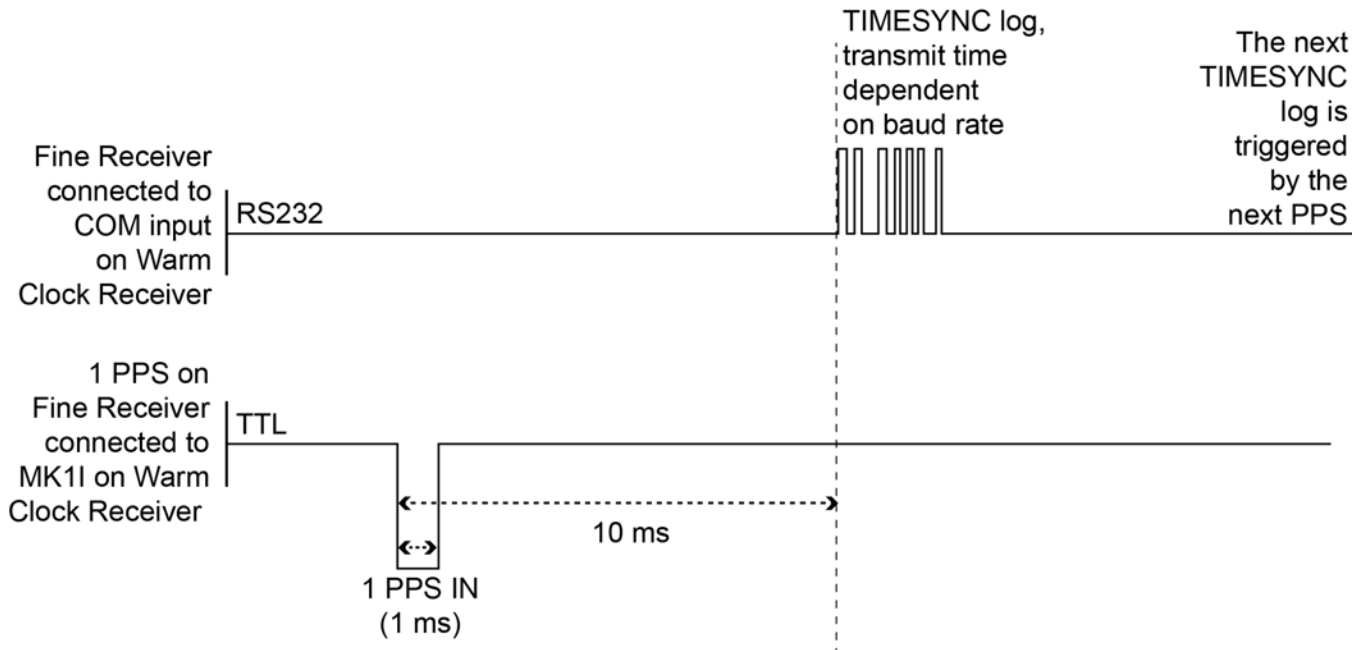
- Manually shift the phase of the clock
- Adjust the phase of the clock so the output 1PPS signal matches an external signal
- Set the receiver clock close to that of another GNSS receiver
- Set the receiver clock exactly in phase of another GNSS receiver

**i**

1. The resolution of the clock synchronization is 20 ns.
2. To adjust the 1PPS output, when the receiver's internal clock is being used and the **CLOCKADJUST** command is enabled, use the **CLOCKOFFSET** command on page 121.
3. If the 1PPS rate is adjusted, the new rate does not start until the next second begins.

*Figure 2: 1PPS Alignment* below shows the 1PPS alignment between a Fine and a Warm Clock receiver. See also the **TIMESYNC** log on page 968 and Transferring Time Between Receivers in the [OEM7 Installation and Operation User Manual](#).

**Figure 2: 1PPS Alignment**



The 1PPS is obtained from different receivers in different ways.



In *Figure 2: 1PPS Alignment* on the previous page, the examples are for the transfer of time. If you need position, you must be tracking satellites and your receiver must have a valid almanac.

Alternatively, the 1PPS signal can be set up to be output on a COM port using the **COMCONTROL** command (see page 124). The accuracy of the 1PPS is less using this method, but may be more convenient in some circumstances.

**Table 14: COM Port Signals Available for 1PPS**

OEM719	OEM729	OEM7600	OEM7700	OEM7720	PwrPak7	CPT7
COM1 Tx	COM1 Tx	COM1 Tx	COM1 Tx	COM1 Tx	COM1 Tx	COM1 Tx
COM2 Tx	COM2 Tx	COM2 Tx	COM2 Tx	COM2 Tx	COM2 Tx	COM2 Tx
	COM2 RTS	COM2 RTS	COM2 RTS	COM2 RTS		
	COM3 Tx	COM3 Tx	COM3 Tx	COM3 Tx	COM3 Tx	
		COM4 Tx	COM4 Tx	COM4 Tx		
		COM5 Tx	COM5 Tx	COM5 Tx		

To find out the time of the last 1PPS output signal, use the TIMESYNCA/B output message (see the **TIMESYNC** log on page 968) which can be output serially on any available COM port, for example:

```
LOG COM1 TIMESYNCA ONTIME 1
```

**Message ID: 429**

**Abbreviated ASCII Syntax:**

```
ADJUST1PPS mode [period] [offset]
```

**Factory Default:**

```
ADJUST1PPS OFF
```

**ASCII Example:**

```
ADJUST1PPS MARK CONTINUOUS 250
```



Use the **ADJUST1PPS** command to synchronize two OEM7 cards in a primary/secondary relationship to a common external clock.

At the Primary Receiver:

```
LOG COM2 TIMESYNCA ONTIME 1
INTERFACEMODE COM2 NOVATEL NOVATEL OFF
CLOCKADJUST DISABLE
EXTERNALCLOCK OCXO 10MHZ (choose rubidium, cesium or user instead and choose 5MHz instead if necessary)
```

At the Secondary Receiver:

```
INTERFACEMODE COM2 NOVATEL NOVATEL OFF
CLOCKADJUST DISABLE
ADJUST1PPS MARK (or markwithtime or time depending on your connection (see Figure 3: ADJUST1PPS Connections on the next page)
EXTERNALCLOCK OCXO 10MHZ (you can choose rubidium, cesium or user instead and choose 5MHz instead if necessary)
```

Connections:

Null modem cable connected from Primary COM2 to Secondary COM2

OCXO signal sent through a splitter to feed both the Primary and Secondary Receiver external clock inputs

Primary 1PPS connected to Secondary MKI

Connect everything before applying power. If power is applied and the OEM7 receivers have acquired satellites before the OCXO and/or 1PPS = MKI is set up, the times reported by the TIMESYNC logs still diverge. Note that after the clock model was stabilized at state 0, the time difference between the Primary and Secondary Receivers reported by the TIMESYNC log was less than 10 ns.



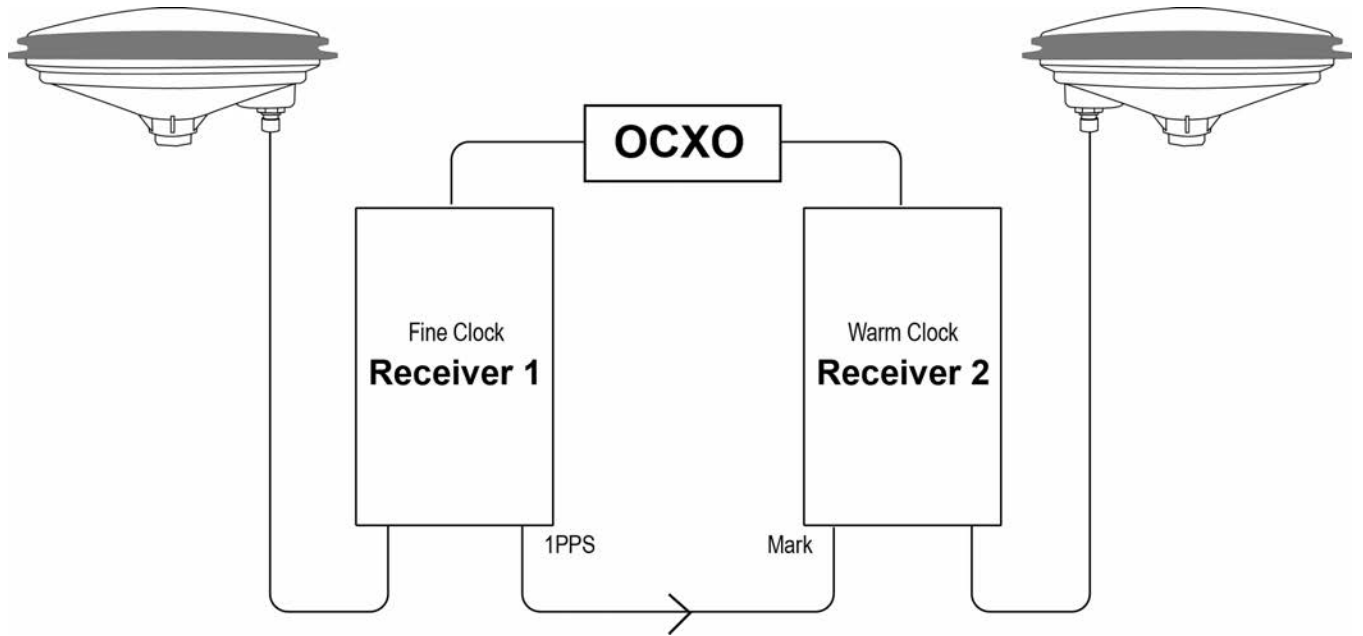
When connecting two receivers to transfer time, disable responses on the COM port used to connect the receivers by issuing the following command on both receivers:

```
INTERFACEMODE COM2 NOVATEL NOVATEL OFF
```

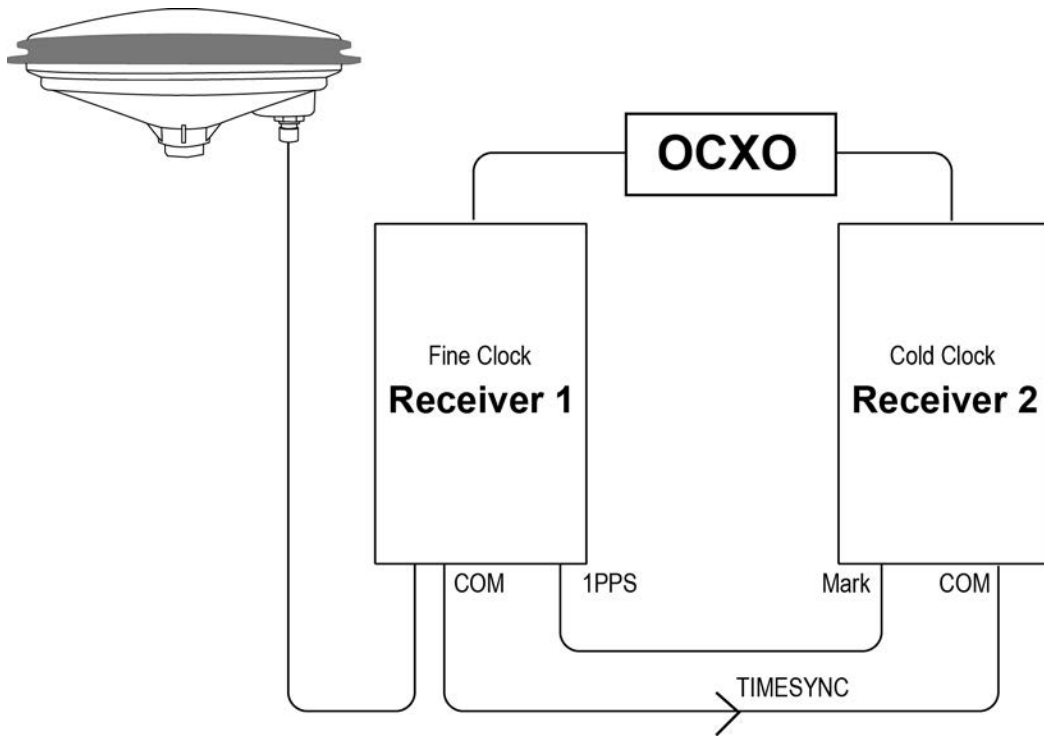


The following examples are for the transfer of time. If you need position, you must be tracking satellites and your receiver must have a valid almanac.

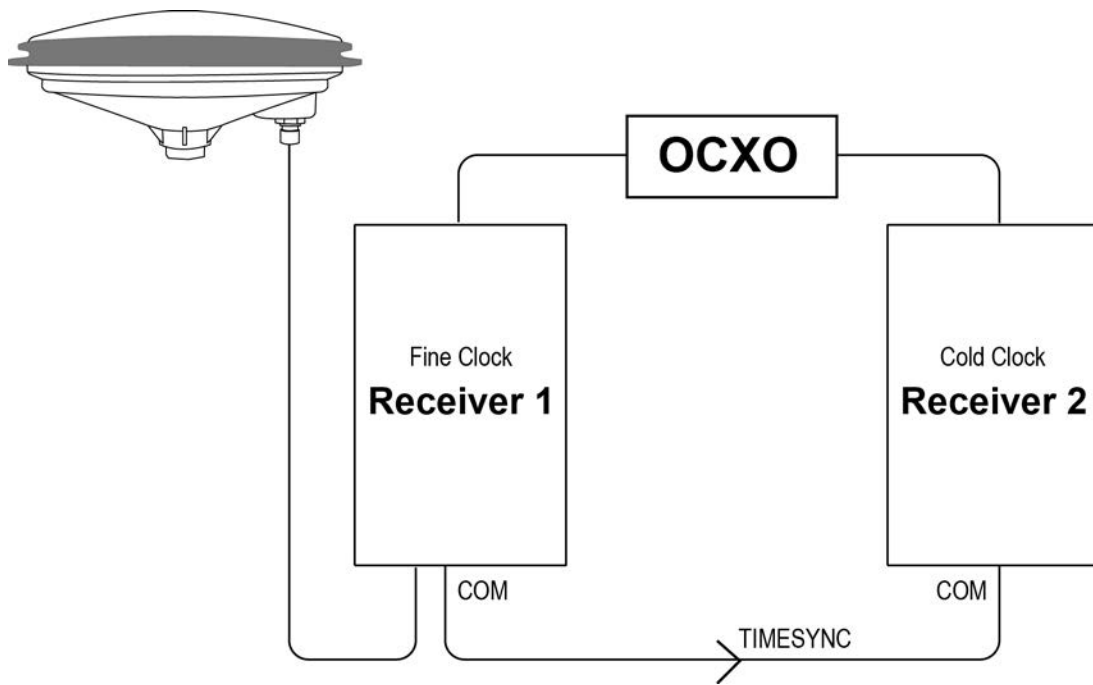
**Figure 3: ADJUST1PPS Connections**



`adjust1pps mark` (if Receiver 2 is not in coarsetime, the input is ignored)



`adjust1pps markwithtime` (will get to finetime)



`adjust1pps time` (will only get to coarsetime)

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ADJUST1PPS header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	mode	See Table 15: ADJUST1PPS Mode on page 66		Sets the ADJUST1PPS mode.	Enum	4	H



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	period	ONCE	0	The time is synchronized only once (default). The <b>ADJUST1PPS</b> command must be reissued if another synchronization is required	Enum	4	H+4
		CONTINUOUS	1	The time is continuously monitored and the receiver clock is corrected if an offset of more than 50 ns is detected			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	offset	-2147483648 to +2147483647 (ns)		Allows the operator to shift the Secondary clock in 20 ns increments. In MANUAL mode, this command applies an immediate shift of this offset in ns to the receiver clock. In MARK and MARKWITHTIME mode, this offset shifts the receiver clock with respect to the time of arrival of the MK11 event. If this offset is zero, the Secondary aligns its 1PPS to that of the signal received in its MK11 port. For example, if this value was set to 50, then the Secondary would set its 1PPS 50 ns ahead of the input signal and if this value was set to -100 then the would set its clock to 100 ns behind the input signal. Typically, this offset is used to correct for cable delay of the 1PPS signal (default=0)	Long	4	H+8

Table 15: ADJUST1PPS Mode

ASCII Value	Binary Value	Description
OFF	0	Disables ADJUST1PPS
MANUAL	1	Immediately shifts the receivers time by the offset field in ns. The period field has no effect in this mode. This command does not affect the clock state.

ASCII Value	Binary Value	Description
MARK	2	<p>Shifts the receiver time to align its 1PPS with the signal received in the MK11 port adjusted by the offset field in ns. The effective shift range is <math>\pm 0.5</math> s.</p> <p>Only the MK11 input can be used to synchronize the 1PPS signal. Synchronization cannot be done using the MK21 input offered on some receivers.</p>
MARKWITHTIME	3	<p>Shifts the receiver time to align its 1PPS with the signal received in the MK11 port adjusted by the offset field in ns, and sets the receiver TOW and week number, to that embedded in a received <b>TIMESYNC</b> log (see page 968). Also sets the receiver Time Status to that embedded in the <b>TIMESYNC</b> log (see page 968), which must have arrived between 800 and 1000 ms prior to the MK11 event (presumably the 1PPS from the Primary), or it is rejected as an invalid message.</p> <p>See <i>Figure 2: 1PPS Alignment</i> on page 60 and <i>TIMESYNC</i> on page 968. Also refer to Transferring Time Between Receivers in the <a href="#">OEM7 Installation and Operation User Manual</a>.</p>
TIME	4	<p>If the receiver clock is not at least COARSEADJUSTED, this command enables the receiver to COARSE adjust its time upon receiving a valid <b>TIMESYNC</b> log (see page 968) in any of the ports. The clock state embedded in the <b>TIMESYNC</b> log (see page 968) must be at least FINE or FINESTEERING before it is considered. The receiver does not use the MK11 event in this mode.</p>

## 2.5 ALIGNAUTOMATION

### *Configures ALIGN plug-and-play feature*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command configures the ALIGN plug and play feature. Use this command to enable/disable the plug and play feature, to set the rover COM port to which master is connected, to set the baud rate for communication, to set the intended operation rate using this command and to enable/disable sending the HEADINGEXTB/HEADINGEXT2B back to the Master receiver.

On issuing this command at the ALIGN Rover, the Rover will automatically sync with the Master and configure it to send corrections at the specified baud rate and specified data rate.



This command should only be issued at ALIGN Rover.

**Message ID: 1323**

#### Abbreviated ASCII Syntax:

```
ALIGNAUTOMATION option [comport] [baudrate] [datarate] [headingextboption]
[interfacemode]
```

#### Factory Default:

```
ALIGNAUTOMATION disable
```

#### Example:

```
ALIGNAUTOMATION enable com2 230400 10 ON
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ALIGNAUTOMATION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	ENABLE	1	Enable or disable the plug-and-play feature	Enum	4	H
		DISABLE	0				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	comport	COM1, COM2 or COM3		Rover COM port to which master is connected ( <i>Table 70: COM Port Identifiers</i> on page 385) (default=COM2)	Enum	4	H+4
4	baudrate	9600, 19200, 38400, 57600, 115200, 230400 or 460800		Intended baud rate for data transmission (default=230400)	Ulong	4	H+8
5	datarate	1, 2, 4, 5, 10 or 20		Rate (in Hz) at which heading output is required (default=10 Hz)	Ulong	4	H+12
6	headingextb option	OFF	0	Enable or disable sending HEADINGEXTB/HEADINGEXT2B back to the Master (default=ON)	Enum	4	H+16
		ON	1				
7	interfacemode	See <i>Table 46: Serial Port Interface Modes</i> on page 219		Serial port interface mode (default=None)	Enum	4	H+20

## 2.6 ANTENNAPOWER

### *Controls power to the antenna*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This command enables or disables the supply of electrical power from the internal power source of the receiver to the Low Noise Amplifier (LNA) of an active antenna. Refer to the [OEM7 Installation and Operation User Manual](#) for further information about supplying power to the antenna.

There are several bits in the receiver status that pertain to the antenna (see *Table 190: Receiver Status* on page 882). These bits indicate whether the antenna is powered and whether it is open circuited or short circuited.

**Message ID: 98**

#### Abbreviated ASCII Syntax:

```
ANTENNAPOWER switch
```

#### Factory Default:

```
ANTENNAPOWER ON
```

#### ASCII Examples:

```
ANTENNAPOWER ON
```

```
ANTENNAPOWER OFF
```



If a short circuit or other problem causes an overload of the current supplied to the antenna, the receiver hardware shuts down the power supplied to the antenna. To restore power, power cycle the receiver. The Receiver Status word, available in the **RXSTATUS** log (see page 878), provides more information about the cause of the problem.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ANTENNAPOWER header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	switch	See <i>Table 16: ANTENNAPOWER Parameters</i> on the next page		Antenna power setting	Enum	4	H

Table 16: ANTENNAPOWER Parameters

Binary	ASCII	Description
0	OFF	Disables antenna power. On dual antenna receivers, disables antenna power for both antennas.
1	ON	Enables antenna power. On dual antenna receivers, enables antenna power for both antennas.
3	PRIMARY_ON_SECONDARY_OFF	Enables primary antenna power and disables secondary antenna power. <b>Note:</b> Dual antenna receivers only
4	PRIMARY_OFF_SECONDARY_ON	Disables primary antenna power and enables secondary antenna power. <b>Note:</b> Dual antenna receivers only



The OEM7 dual antenna receivers are: OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7.

## 2.7 ANTENNATYPE

### Store the user defined antenna type

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use the **ANTENNATYPE** command to control the storage of up to five user-defined antennas. Values are entered and interpreted according to the conventions in the IGS ANTEX format.

The current list of user-defined antennas stored by the receiver can be viewed using the **USERANTENNA** log on page 974. The **THISANTENNATYPE** command (see page 428) and **BASEANTENNATYPE** command (see page 93) control the use of the antennas in RTK.

**Message ID: 2281**

#### Abbreviated ASCII Syntax:

```
ANTENNATYPE Action AntennaType [AntennaName] [NumberOfFrequencies]
[Frequency] [NorthOffset] [EastOffset] [UpOffset] [PCVArray] ...
```



The “Number of frequencies” determines the number of frequencies for which Phase Center Offsets (PCOs) and Phase Center Variations (PCVs) are stored. For simplicity, the syntax above shows a single frequency. The ASCII Example 1 below shows two frequencies being added. PCOs and PCVs can be defined for up to 24 frequencies.

#### ASCII Example 1:

```
ANTENNATYPE ADD USER_ANTENNA_1 NOVCUSTOM 2 GPSL1 0.09 0.0 51.74 0 -0.03
-0.11 -0.20 0.23 -0.17 -0.04 0.14 0.26 0.25 0.07 -0.24 -0.54 -0.67 -0.49
-0.02 0.55 0.84 0.47 GPSL2 -1.54 1.66 52.00 0.00 0.0 0.0 -0.03 -0.13 -0.28
-0.48 -0.61 -0.56 -0.28 0.23 0.84 1.29 1.31 0.77 -0.16 -0.95 -0.81 0.97
```

#### ASCII Example 2:

```
ANTENNATYPE REMOVE USER_ANTENNA_1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ANTENNATYPE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	Action	ADD	0	Add a user antenna	Enum	4	H
		REMOVE	1	Delete a user antenna			
3	AntennaType	<i>Table 17: User-Defined Antenna Type on the next page</i>		User defined antenna type	Enum	4	H+4
4	AntennaName			Name of the user defined antenna	Char[16]	16	H+8
5	NumberOf Frequencies			Number of frequencies for which corrections are stored	Ulong	8	H+24
6	Frequency	<i>See Table 22: Frequency Type on page 91</i>		The frequency for which the phase center corrections are valid.	Enum	4	H+32
7	NorthOffset			North phase center offset (millimetres) Valid range: -326.66 to 326.66	Float	4	H+36
8	EastOffset			East phase center offset (millimetres) Valid range: -326.66 to 326.66	Float	4	H+40
9	UpOffset			Up phase center offset (millimetres) Valid range: -326.66 to 326.66	Float	4	H+44

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
10	PCVArray			19 element array of Phase Center Variations in 5-degree elevation increments from 90 degrees elevation to 0 degrees (millimetres)  Valid range: -326.66 to 326.66	Float[19]	76	H+48
11	Next Frequency = H + 32 + (Number of frequencies x 92)						

Table 17: User-Defined Antenna Type

Binary	ASCII
1001	USER_ANTENNA_1
1002	USER_ANTENNA_2
1003	USER_ANTENNA_3
1004	USER_ANTENNA_4
1005	USER_ANTENNA_5

## 2.8 APPROXPOSTIMEOUT

### *Timeout for approximate positions in the receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the approximate position timeout of the receiver. Approximate positions are used upon startup of the receiver to populate the satellite visibility lists (see **SATVIS2** log on page 904) that aid in initial acquisition of signals. By default, an approximate position times out after 150 seconds and the receiver reverts to a cold search. This command is typically used in a high dynamic application where an initial position becomes inaccurate.

**Message ID: 1513**

#### **Abbreviated ASCII Syntax:**

```
APPROXPOSTIMEOUT timeout
```

#### **Factory Default:**

```
APPROXPOSTIMEOUT 150
```

#### **ASCII Example:**

```
APPROXPOSTIMEOUT 20
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	APPROXPOSTIMEOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Timeout	0 to 86400		Timeout in seconds. Default = 150	Ulong	4	H

## 2.9 ASSIGN

### *Assigns a channel to a PRN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



The **ASSIGN** command should only be used by advanced users.



1. Assigning SV channel sets the forced assignment bit in the channel tracking status field which is reported in the RANGE and TRACKSTAT logs.
2. Assigning a PRN to a SV channel does not remove the PRN from the search space of the automatic searcher; only the SV channel is removed (that is, the searcher may search and lock onto the same PRN on another channel). See *Table 12: PRN Numbers for Commands and Logs* on page 52 for the PRN available for the **ASSIGN** command.
3. GLONASS SVs cannot be assigned if there is no information on GLONASS frequencies and matching slot numbers.
4. OEM7 cards have 4 channels available for SBAS. They automatically use the healthy GEO satellites with the highest elevations. Use the **ASSIGN** command to enter a GEO PRN manually.
5. The **ASSIGN** and **UNASSIGN** commands are not accepted for L-Band channels. The **ASSIGNLBANDBEAM** command (see page 82) should be used for L-Band channels.
6. Manually assigned satellites are not reported in the **RANGECMP4** log (see page 823).

This command may be used to aid in the initial acquisition of a satellite by manually overriding the automatic satellite/channel assignment and reacquisition processes. The command specifies that the indicated tracking channel search for:

- a specified satellite
- at a specified Doppler frequency
- within a specified Doppler window

The instruction remains in effect for the specified SV channel and PRN, even if the assigned satellite is below the elevation cutoff. If the satellite Doppler offset of the assigned SV channel exceeds that specified by the window parameter of the **ASSIGN** command, the satellite may never be acquired or reacquired. If a channel has been manually assigned, and the channel is changed to AUTO tracking, then the channel is idled immediately and returns to automatic mode.

To cancel the effects of **ASSIGN**, issue one of the following:

- The **ASSIGN** command with the state set to AUTO
- The **UNASSIGN** command (see page 443)
- The **UNASSIGNALL** command (see page 445)

These immediately return SV channel control to the automatic search engine

**Message ID: 27**

**Abbreviated ASCII Syntax:**

```
ASSIGN channel [state] [prn [Doppler [Doppler window]]]
```

**ASCII Example 1:**

```
ASSIGN 0 ACTIVE 29 0 2000
```

In example 1, the first SV channel is searching for satellite PRN 29 in a range from -2000 Hz to 2000 Hz until the satellite signal is detected.

**ASCII Example 2:**

```
ASSIGN 11 28 -250 0
```

SV channel 11 is searching for satellite PRN 28 at an offset of -250 Hz only.

**ASCII Example 3:**

```
ASSIGN 11 IDLE
```

SV channel 11 is idled and does not attempt to search for satellites.



For dual antenna receivers:

- On the primary antenna, the SV channel is from 0 to N-1, where N is the number of channels in the primary antenna channel configuration.
- On the secondary antenna, the SV channel count is from at N to N+(M-1), where M is the number of channels in the secondary antenna SV channel configuration.
- Even though L-Band channels cannot be configured with the **ASSIGN** command, they are included when determining N.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ASSIGN header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	channel	0 to n-1, where n is the maximum number of GNSS channels in the current channel configuration		Desired SV channel number where channel 0 is the first SV channel. The last channel depends on the model configuration.	Ulong	4	H
3	state	Refer to <i>Table 18: Channel State</i> on the next page		Set the SV channel state. If a value is not given, the default of ACTIVE is used when the additional optional parameters are entered.	Enum	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	prn	Refer to <i>PRN Numbers</i> on page 52		Optional satellite PRN number. A value must be entered unless the state parameter is IDLE or AUTO.	Ulong	4	H+8
5	Doppler	-100 000 to 100 000 Hz		Current Doppler offset of the satellite. (default=0) <b>Note:</b> Satellite motion, receiver antenna motion and receiver clock frequency error must be included in the calculation of Doppler frequency.	Long	4	H+12
6	Doppler window	0 to 10 000 Hz		Error or uncertainty in the Doppler estimate above. (default=4500) <b>Note:</b> This is a $\pm$ value. Example: 500 for $\pm$ 500 Hz	Ulong	4	H+16

Table 18: Channel State

Binary	ASCII	Description
0	IDLE	Set the SV channel to not track any satellites
1	ACTIVE <sup>1</sup>	Set the SV channel active (default)
2	AUTO	Tell the receiver to automatically assign PRN numbers to channels

<sup>1</sup>A PRN number is required when using the ACTIVE channel state in this command.

## 2.10 ASSIGNALL

### *Assigns all channels of a satellite system*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



The **ASSIGNALL** command should only be used by advanced users.

This command is used to override the automatic satellite channel algorithms with manual instructions for all channels in a satellite system.

**Message ID: 28**

#### Abbreviated ASCII Syntax:

```
ASSIGNALL [system][state][prn [Doppler [Doppler window]]]
```

#### ASCII Example 1:

```
ASSIGNALL GLONASS IDLE
```

In example 1, all GLONASS channels are idled, essentially stopping the receiver from tracking GLONASS.

#### ASCII Example 2:

```
ASSIGNALL GLONASS AUTO
```

In example 2, all GLONASS channels are enabled in auto mode. This enables the receiver to automatically assign channels to track the available GLONASS satellites.



This command is the same as **ASSIGN** except that it affects all SV channels of the specified satellite system.



These command examples are only applicable to specific receiver models.

If the system field is used with this command and the receiver has no channels configured with that satellite system, the command is rejected.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ASSIGNALL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	system	See <i>Table 19: Satellite System</i> below		Satellite system to assign. If no value is specified, the value defaults to ALL.	Enum	4	H
3	state	Refer to <i>Table 18: Channel State</i> on page 78)		Set the SV channel state.	Enum	4	H+4
4	prn	Refer to <i>PRN Numbers</i> on page 52		Optional satellite PRN code. A value must be entered if the state parameter is neither IDLE or AUTO.	Ulong	4	H+8
5	Doppler	-100 000 to 100 000 Hz		Current Doppler offset of the satellite. (default=0) <b>Note:</b> Satellite motion, receiver antenna motion and receiver clock frequency error must be included in the calculation of Doppler frequency.	Long	4	H+12
6	Doppler window	0 to 10 000 Hz		Error or uncertainty in the Doppler estimate above. (default=4500) <b>Note:</b> This is a $\pm$ value Example, 500 for $\pm$ 500 Hz	Ulong	4	H+16

Table 19: Satellite System

Binary	ASCII	Description
3	ALL	All systems
99	GPS	GPS system
100	SBAS	SBAS system
101	GLONASS	GLONASS system
102	GALILEO	GALILEO system
103	BEIDOU	BeiDou system
104	QZSS	QZSS system
105	NAVIC	NavIC system





GLONASS SVs cannot be assigned if there is no information on GLONASS frequencies and matching slot numbers.

## 2.11 ASSIGNLBANDBEAM

### Configure L-Band tracking

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command assigns TerraStar or Veripos beams to the L-Band channels based on the defined L-Band assignment option.



Logging the **ASSIGNLBANDBEAM** command may not display the correct values. To access the actual beam name, frequency and baud rate values, log the **LBANDTRACKSTAT** log (see page 684) or if the beam name is known, log the **LBANDBEAMTABLE** log (see page 682) and find the associated frequency and baud rate.

**Message ID:** 1733

#### Abbreviated ASCII Syntax:

```
ASSIGNLBANDBEAM [option] [name] [frequency] [baudrate] [Dopplerwindow]
```

#### Factory Default:

```
ASSIGNLBANDBEAM idle
```

#### ASCII Examples:

```
ASSIGNLBANDBEAM auto
```

```
ASSIGNLBANDBEAM 98W
```

```
ASSIGNLBANDBEAM manual 98w 1545865000 1200
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Command header	ASSIGNLBANDBEAM header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Option	Assignment option (see <i>Table 20: L-Band Assignment Option</i> on the next page) (manual=default)	Enum	4	H
3	Name	Beam name (empty string=default)	Char[8]	8	H+4
4	Frequency	Beam frequency in Hz (0=default)	Ulong	4	H+12
5	Baud rate	Data baud rate (0=default)	Ulong	4	H+16
6	Doppler window	Doppler window to search (6000=default)	Ulong	4	H+20

**Table 20: L-Band Assignment Option**

ASCII	Binary	Description
IDLE	0	Idle all L-Band channels
AUTO	1	<p>The receiver searches for multiple L-Band beams on the L-Band channels based on AUTO selection criteria.</p> <p>If the receiver position is known, the AUTO selection criteria is a ranking of granted access L-Band beams by descending elevation angle.</p> <p>If the receiver position is not known, the AUTO selection criteria is a ranking of granted access L-Band beams in the order they appear in the stored beam table (see the <b>LBANDBEAMTABLE</b> log on page 682).</p>
MANUAL	2	The receiver assigns the specified beam on the first L-BAND channel and makes the other L-BAND channels IDLE.

## 2.12 AUTH

### *Authorization code for different model*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to add or remove authorization codes from the receiver. Authorization codes are used to authorize models of software for a receiver. Models control the functionality the receiver provides. The receiver is capable of keeping track of 24 authorization codes at one time. The **MODEL** command (see page 266) can then be used to switch between authorized models. The **VALIDMODELS** command (see page 979) lists the current available models in the receiver. The **AUTHCODES** log (see page 491) lists all authorization codes entered into the receiver. This simplifies the use of multiple software models on the same receiver.

If there is more than one valid model in the receiver, the receiver either uses the model of the last auth code entered via the **AUTH** command or the model that was selected by the **MODEL** command, whichever was done last. Adding an authorization code or using the **MODEL** command causes an automatic reset of the receiver. Removing an authorization code does not cause a reset.



Removing an authorization code will cause the receiver to permanently lose this information.

**Message ID: 49**

**Abbreviated ASCII Syntax:**

```
AUTH [state] part1 part2 part3 part4 part5 model [date]
```

**Input Example:**

```
AUTH add T48JF2,W25DBM,JH46BJ,2WGHMJ,8JW5TW,G2SR0RCCR,101114
```

```
AUTH erase_table PW5W2B,WW5TM9,WW2PCZ,WW3M4H,WW4HPG,ERASE_AUTH
```






When you are ready to upgrade from one model to another, call 1-800-NOVATEL to speak with our Customer Support/Sales Personnel, who can provide the authorization code that unlocks the additional features of your GNSS receiver. This procedure can be performed at your work site and takes only a few minutes.

Receiver models can also be downgraded. This is a two step handshaking process and is best performed in a location with e-mail access.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	AUTH header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	state	See <i>Table 21: AUTH Command State</i> on the next page		Authorization code function to perform	Enum	4	H
3	part1	6 character ASCII string		Authorization code section 1	String [max. 16]	Variable <sup>1</sup>	H+4
4	part2	6 character ASCII string		Authorization code section 2	String [max. 16]	Variable <sup>1</sup>	H+20
5	part3	6 character ASCII string		Authorization code section 3	String [max. 16]	Variable <sup>1</sup>	H+36
6	part4	6 character ASCII string		Authorization code section 4	String [max. 16]	Variable <sup>1</sup>	H+52
7	part5	6 character ASCII string		Authorization code section 5	String [max. 16]	Variable <sup>1</sup>	H+68
8	model	Alpha numeric	Null terminated	Model name of the receiver	String [max. 16]	Variable <sup>1</sup>	H+84
9	date	Numeric	Null terminated	Expiry date entered as yymmdd in decimal	String [max 7]	Variable <sup>1</sup>	Variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 21: AUTH Command State

ASCII	Binary	Description
REMOVE	0	<p>Remove the authorization code from the system</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  For this parameter, the Part1-Part5 fields can be entered as 0 0 0 0 0, and only the model name entered. </div>
ADD	1	Add the authorization code to the system (default)
ADD_DOWNLOAD	4	Add the authorization code to the system (Deprecated: Use ADD instead)
ERASE_TABLE	7	<p>Erase all authorization codes from the system. Requires a special authorization code to prevent against accidental erasing.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The special authorization code required for this option is: PW5W2B,WW5TM9,WW2PCZ,WW3M4H,WW4HPG,ERASE_AUTH </div>
CLEAN_TABLE	8	<p>Remove all invalidated authorization codes from the system.</p> <p>When an authorization code is removed, it is simply invalidated and so it still uses one of the 24 spaces reserved for authorization codes in the receiver. Use the CLEAN_TABLE option to free up the spaces from removed authorization codes.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The special authorization code required for this option is: 4DR69H,G369W8,34MNJJ,5NHXCJ,GW7C75,CLEAN_AUTH </div>

## 2.13 AUTOSURVEY

### *Survey for accurate position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

The **AUTOSURVEY** command uses position averaging to automatically determine the position for a base station.

When the **AUTOSURVEY** command is sent, the receiver starts position averaging. The position averaging continues until a specified accuracy level is met or until the specified survey time expires. When position averaging is complete, the calculated position is saved as the fix position for the base station. This calculated position is then used when transmitting differential corrections to the rover.



The **AUTOSURVEY** command computes the base position in WGS84.



If the **FIX** command is entered by a user, the **SAVECONFIG** command must then be issued to save to NVM. If the **FIX** command is issued by the **AUTOSURVEY** feature, the **SAVECONFIG** command does not need to be issued, because it is automatically saved to NVM.

On subsequent power ups or resets, an **AUTOSURVEY** runs to determine if the base station has moved. As the **AUTOSURVEY** runs, the average position calculated is compared to the saved fix position. If the average position is within the **AUTOSURVEY** tolerance setting, the receiver assumes it has not moved and uses the previously saved fix position. If the average position is outside of the **AUTOSURVEY** tolerance setting, the receiver assumes it has moved and will continue calculating a position average until the accuracy level is met or until the specified survey time expires.

The surveyed positions saved using the **AUTOSURVEY** command can be viewed using the **SAVEDSURVEYPOSITIONS** log on page 910. Surveyed positions can be added or deleted using the **SURVEYPOSITION** command on page 420.

**Message ID: 1795**

#### **Abbreviated ASCII Syntax:**

```
AUTOSURVEY control [time] [accuracy] [tolerance] [save_nvm] [position_id]
```

#### **Input Example:**


In the following example, the receiver is set up to survey its position for up to 24 hours or until the averaged position accuracy is 10 cm. On subsequent power ups at the same location, the survey will terminate as soon as the receiver determines the position is within 4 m of its surveyed position. Once the receiver has fixed its position, it will transmit RTCMV3 corrections over COM2.

```
SERIALCONFIG COM2 115200 N 8 1 N ON
INTERFACEMODE COM2 NONE RTCMV3 OFF
LOG COM2 RTCM1004 ONTIME 1
LOG COM2 RTCM1012 ONTIME 1
LOG COM2 RTCM1006 ONTIME 10
LOG COM2 RTCM1033 ONTIME 10
```

```
LOG COM2 RTCM1019 ONTIME 120
AUTOSURVEY ENABLE 1440 .1 4
SAVECONFIG
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	AUTOSURVEY header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	control	disable	0	Disables the self-survey feature and halts any self-survey related activity (default = disable)	Enum	4	H
		enable	1	Enables the self-survey feature			
3	time	1 - 6000 minutes		Maximum amount of time to perform self-survey (default = 1440 minutes)	Ulong	4	H+4
4	accuracy	0 - 100 metres		Desired horizontal standard deviation (default = 0.1 metres)	Float	4	H+8
5	tolerance	3 - 100 metres		Maximum distance between calculated position and saved position. During the self-survey, if the distance between the calculated position and the previously surveyed position is less than this value, the previous position is used. (default = 10 metres)	Float	4	H+12
6	save_nvm	OFF	0	Do not save position in NVM	Enum	4	H+16
		ON	1	Save position in NVM (default = ON)			



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	position_id	4 character string		<p>ID for the saved position. If the ID is not specified or if the ID is entered as "AUTO", receiver automatically generates a unique ID for the position.</p> <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;">  This ID is the ID for the surveyed position, not the station ID set using the <b>DGPSTXID</b> command on page 138. </div>	String[5]	5 <sup>1</sup>	H+20

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.14 BASEANTENNAPCO

### *Sets the PCO model of the base receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use the **BASEANTENNAPCO** command to set the Phase Center Offsets (PCO) for a given frequency on the remote base receiver from which this receiver is receiving corrections. The Offsets are defined as North, East and Up from the Antenna Reference Point to the Frequency Phase Center in millimetres.

**Message ID: 1415**

#### Abbreviated ASCII Syntax:

```
BASEANTENNAPCO Frequency NorthOffset EastOffset UpOffset [CorrectionType
[StationId]]
```

#### ASCII Example:

```
BASEANTENNAPCO GPSL1 0.61 1.99 65.64
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BASEANTENNAPCO header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Frequency	See Table 22: Frequency Type on the next page		The frequency that the phase center offsets are valid for.	Enum	4	H
3	NorthOffset			NGS standard Phase Center North Offset in millimetres.	Double	8	H+4
4	EastOffset			NGS standard Phase Center East Offset in millimetres.	Double	8	H+12
5	UpOffset			NGS standard Phase Center Up Offset in millimetres.	Double	8	H+20
6	CorrectionType	See Table 65: RTK Source Type on page 349		Correction type (default = AUTO)	Enum	4	H+28

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	StationID	Char [8] or ANY		ID string for the base station (default = ANY)	Char	8	H+32

Table 22: Frequency Type

Value	Name	Description
0	GPSL1	GPS L1 frequency
1	GPSL2	GPS L2 frequency
2	GLONASSL1	GLONASS L1 frequency
3	GLONASSL2	GLONASS L2 frequency
5	GPSL5	GPS L5 frequency
7	GALILEOE1	Galileo E1 frequency
8	GALILEOE5A	Galileo E5a frequency
9	GALILEOE5B	Galileo E5b frequency
10	GALILEOALTBOC	Galileo AltBOC frequency
11	BEIDOU B1	BeiDou B1 frequency
12	BEIDOU B2	BeiDou B2 frequency
13	QZSSL1	QZSS L1 frequency
14	QZSSL2	QZSS L2 frequency
15	QZSSL5	QZSS L5 frequency

## 2.15 BASEANTENNAPCV

### *Sets the PCV model of the base receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use the **BASEANTENNAPCV** command to set the Phase Center Variation (PCV) for a given frequency on the remote base receiver from which this receiver is receiving corrections. The Phase Center Variation entries follow the NGS standard and correspond to the phase elevation at 5 degree increments starting at 90 degrees and decreasing to 0.

**Message ID:** 1416

#### Abbreviated ASCII Syntax:

```
BASEANTENNAPCV Frequency [PCVArray [CorrectionType [StationId]]]
```

#### ASCII Example:

```
BASEANTENNAPCV GPSL1 0.00 -0.020 -0.07 -0.15 -0.24 -0.34 -0.43 -0.51 -0.56 -
0.61 -0.65 -0.69 -0.69 -0.62 -0.44 -0.13 0.28 0.70 1.02
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BASEANTENNAPCV header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Frequency	See Table 22: Frequency Type on the previous page		The frequency for which the phase center variations are valid.	Enum	4	H
3	PCVArray			NGS standard 19 element array of phase center variations, in millimetres, in 5 degree elevation increments from 90 to 0. Defaults to zero for all elevation increments.	Double [19]	152	H+4
4	CorrectionType	See Table 65: RTK Source Type on page 349		Correction type (default = AUTO)	Enum	4	H+156
5	StationID	Char [8] or ANY		ID string (default = ANY)	Char	8	H+160

## 2.16 BASEANTENNATYPE

### *Sets the antenna type of the base receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use the **BASEANTENNATYPE** command to set the type of antenna type connected to the base receiver from which this receiver is receiving RTK corrections. There are two sources of antenna information:

- An internal table  
The firmware contains a set of predefined antenna and radome types taken from the IGS ANTEX file. Refer to *Table 23: Antenna Type* on the next page and *Table 24: Radome Type* on page 103 for the antennas currently supported.
- User-defined antennas  
User-defined antenna types can be entered using the **ANTENNATYPE** command (see page 72).

The **THISANTENNATYPE** command (see page 428) is used to set the type of antenna being used with this receiver.

**Message ID: 1419**

#### Abbreviated ASCII Syntax:

```
BASEANTENNATYPE AntennaType [RadomeType] [CorrectionType] [StationId]
```

#### ASCII Examples:

```
BASEANTENNATYPE NOV702
```

```
BASEANTENNATYPE USER_ANTENNA_1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BASEANTENNATYPE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	AntennaType	See <i>Table 23: Antenna Type</i> on the next page or <i>Table 17: User-Defined Antenna Type</i> on page 74		Antenna type	Enum	4	H
3	RadomeType	See <i>Table 24: Radome Type</i> on page 103		Radome type (default = NONE)	Enum	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	CorrectionType	See Table 65: RTK Source Type on page 349		Correction type (default = AUTO)	Enum	4	H+8
5	StationID	Char [8] or ANY		Base station ID (default = ANY)	Char	8	H+12



The latest information can be obtained from the National Geodetic Survey (NGS) site [www.ngs.noaa.gov/ANTCAL](http://www.ngs.noaa.gov/ANTCAL).

**Table 23: Antenna Type**

Value	Name	Description
0	NONE	No antenna model
2	AUTO	Determine the antenna model from the RTK corrections (Not valid for THISANTENNATYPE)
3	AERAT2775_43	
4	AOAD_M_B	
5	AOAD_M_T	AOAD/M_T
6	AOAD_M_TA_NGS	AOAD/M_TA_NGS
7	APSAPS-3	
8	ASH700228A	
9	ASH700228B	
10	ASH700228C	
11	ASH700228D	
12	ASH700228E	
13	ASH700699.L1	
14	ASH700700.A	
15	ASH700700.B	
16	ASH700700.C	
17	ASH700718A	
18	ASH700718B	
19	ASH700829.2	
20	ASH700829.3	

Value	Name	Description
21	ASH700829.A	
22	ASH700829.A1	
23	ASH700936A_M	
24	ASH700936B_M	
25	ASH700936C_M	
26	ASH700936D_M	
27	ASH700936E	
28	ASH700936E_C	
29	ASH700936F_C	
30	ASH701008.01B	
31	ASH701073.1	
32	ASH701073.3	
33	ASH701933A_M	
34	ASH701933B_M	
35	ASH701933C_M	
36	ASH701941.1	
37	ASH701941.2	
38	ASH701941.A	
39	ASH701941.B	
40	ASH701945B_M	
41	ASH701945C_M	
42	ASH701945D_M	
43	ASH701945E_M	
44	ASH701945G_M	
45	ASH701946.2	
46	ASH701946.3	
47	ASH701975.01A	
48	ASH701975.01AGP	
49	JAV_GRANT-G3T	


Value	Name	Description
50	JAV_RINGANT_G3T	
51	JAVRINGANT_DM	
52	JNSMARANT_GGD	
53	JPLD/M_R	
54	JPLD/M_RA_SOP	
55	JPSLEGANT_E	
56	JPSODYSSEY_I	
57	JPSREGANT_DD_E	
58	JPSREGANT_SD_E	
59	LEIAR10	
60	LEIAR25	
61	LEIAR25.R3	
62	LEIAR25.R4	
63	LEIAS05	
64	LEIAX1202GG	
65	LEIAS10	
66	LEIAX1203+GNSS	
67	LEIAT202+GP	
68	LEIAT202-GP	
69	LEIAT302+GP	
70	LEIAT302-GP	
71	LEIAT303	
72	LEIAT502	
73	LEIAT503	
74	LEIAT504	
75	LEIAT504GG	
76	LEIATX1230	
77	LEIATX1230+GNSS	
78	LEIATX1230GG	




Value	Name	Description
79	LEIAX1202	
80	LEIGG02PLUS	
81	LEIGS08	
82	LEIGS09	
83	LEIGS12	
84	3S-02-TSADM	
85	3S-02-TSATE	
86	LEIGS15	
87	LEIMNA950GG	
88	LEISR299_INT	
89	LEISR399_INT	
90	LEISR399_INTA	
91	MAC4647942	
92	MPL_WAAS_2224NW	
93	MPL_WAAS_2225NW	
94	MPLL1_L2_SURV	
95	NAVAN2004T	
96	NAVAN2008T	
97	NAX3G+C	
98	NOV_WAAS_600	
99	NOV501	
100	NOV501+CR	
101	NOV502	
102	NOV502+CR	
103	NOV503+CR	
104	NOV531	
105	NOV531+CR	
106	NOV600	
107	NOV702	

Value	Name	Description
108	NOV702GG	
109	NOV750.R4	
110	SEN67157596+CR	
111	SOK_RADIAN_IS	
112	SOK502	
113	SOK600	
114	SOK702	
115	SPP571212238+GP	
116	STXS9SA7224V3.0	
117	TOP700779A	
118	TOP72110	
119	TPSCR.G3	
120	TPSCR3_GGD	
121	TPSCR4	
122	TPSG3_A1	
123	TPSHIPER_GD	
124	TPSHIPER_GGD	
125	TPSHIPER_LITE	
126	TPSHIPER_PLUS	
127	TPSLEGANT_G	
128	TPSLEGANT2	
129	TPSLEGANT3_UHF	
130	TPSODYSSEY_I	
131	TPSPG_A1	
132	TPSPG_A1+GP	
133	TRM14177.00	
134	TRM14532.00	
135	TRM14532.10	
136	TRM22020.00+GP	

Value	Name	Description
137	TRM22020.00-GP	
138	TRM23903.00	
139	TRM27947.00+GP	
140	TRM27947.00-GP	
141	TRM29659.00	
142	TRM33429.00+GP	
143	TRM33429.00-GP	
144	TRM33429.20+GP	
145	TRM39105.00	
146	TRM41249.00	
147	TRM41249USCG	
148	TRM4800	
149	TRM55971.00	
150	TRM57970.00	
151	TRM57971.00	
152	TRM5800	
153	TRM59800.00	
154	TRM59800.80	
155	TRM59900.00	
156	TRMR8_GNSS	
157	TRMR8_GNSS3	
158	ASH701023.A	
159	CHCC220GR	
160	CHCC220GR2	
161	CHCX91+S	
162	GMXZENITH10	
163	GMXZENITH20	
164	GMXZENITH25	
165	GMXZENITH25PRO	

Value	Name	Description
166	GMXZENITH35	
167	JAVRINGANT_G5T	
168	JAVTRIUMPH_1M	
169	JAVTRIUMPH_1MR	
170	JAVTRIUMPH_2A	
171	JAVTRIUMPH_LSA	
172	JNSCR_C146-22-1	
173	JPSREGANT_DD_E1	
174	JPSREGANT_DD_E2	
175	JPSREGANT_SD_E1	
176	JPSREGANT_SD_E2	
177	LEIAR20	
178	LEIGG03	
179	LEIGS08PLUS	
180	LEIGS14	
181	LEIICG60	
182	NOV533+CR	
183	NOV703GGG.R2	 This value may also be used for the GPS-713-GGG-N and GPS-713-GGGL-N antennas.
184	NOV750.R5	
185	RNG80971.00	
186	SEPCHOKE_B3E6	
187	SEPCHOKE_MC	
188	STXS10SX017A	
189	STXS8PX003A	
190	STXS9PX001A	
191	TIAPENG2100B	
192	TIAPENG2100R	
193	TIAPENG3100R1	

Value	Name	Description
194	TIAPENG3100R2	
195	TPSCR.G5	
196	TPSG5_A1	
197	TPSPN.A5	
198	TRM55970.00	
199	TRMR10	
200	TRMR4-3	
201	TRMR6-4	
202	TRMR8-4	
203	TRMR8S	
204	TRMSPS985	
205	AERAT1675_120	
206	ITT3750323	
207	NOV702GGL	
208	NOV704WB	
209	ARFAS1FS	
210	CHAPS9017	
211	CHCI80	
212	GMXZENITH15	
213	HXCCGX601A	
214	IGAIG8	
215	LEICGA60	
216	LEIGS15.R2	
217	LEIGS16	
218	MVEGA152GNSSA	
219	SEPALTUS_NR3	
220	SJTTL111	
221	SOKGCX3	
222	SOKSA500	

Value	Name	Description
223	STHCR3-G3	
224	STXS9I	
225	TPSCR.G5C	
226	TPSHIPER_HR	
227	TPSHIPER_HR+PS	
228	TRM105000.10	
229	TRM115000.00	
230	TRM115000.10	
231	TRMR2	
232	TWIVP6000	
233	TWIVP6050_CONE	
234	JAVTRIUMPH_2A+G	
235	JAVTRIUMPH_2A+P	
236	LEIGS18	
237	LEIGG04PLUS	
238	STXS800	
239	STXS800A	
240	NOV850	 This value may also be used for the GNSS-802, GNSS-802L, GNSS-804 and GNSS-804L antennas.
241	TRM159800.00	
242	TRM159900.00	
243	LEIGG04	
244	LEIICG70	
245	JAV_GRANT-G3T+G	
246	JAVGRANT_G5T+GP	
247	JAVTRIUMPH_3A	
248	TPSHIPER_VR	
249	TRM59800.00C	
250	TRM59800.99	

Value	Name	Description
251	TRMR10-2	
252	TRMR12	
253	TRMSPS986	
254	TWIVC6050	
255	TWIVC6150	
256	GMXZENITH60	
257	ASH701945B.99	
258	LEIAS11	
1001	USER_ANTENNA_1	User defined antenna type 1
1002	USER_ANTENNA_2	User defined antenna type 2
1003	USER_ANTENNA_3	User defined antenna type 3
1004	USER_ANTENNA_4	User defined antenna type 4
1005	USER_ANTENNA_5	User defined antenna type 5

**Table 24: Radome Type**

Value	Name
0	NONE
1	SPKE
2	SNOW
3	SCIS
4	SCIT
5	OLGA
6	PFAN
7	JVDM
8	LEIT
9	LEIC
10	LEIS
11	MMAC
12	NOVS
13	TPSH

<b>Value</b>	<b>Name</b>
14	CONE
15	TPSD
16	TCWD
17	UNAV
18	TZGD
19	CHCD
20	JAVC
21	LEIM
22	NOVC
23	ARFC
24	HXCS
25	JVGR
26	STHC
27	DUTD
28	JAVD
29	JVSD



## 2.17 BDSECUTOFF

### *Sets elevation cut-off angle for BeiDou satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the elevation cut-off angle for tracked BeiDou satellites. The receiver does not start automatically searching for a BeiDou satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the BDSECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle; it could be used in these situations:

- The antenna is at a high altitude, and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **BDSECUTOFF** command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command on page 156 to set the cut-off angle for all other systems.



For dual antenna receivers, this command applies to both the primary and secondary antennas.

**Message ID: 1582**

**Abbreviated ASCII Syntax:**

```
BDSECUTOFF angle
```

**Factory Default:**

```
BDSECUTOFF 5.0
```

**ASCII Example:**

```
BDSECUTOFF 10.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BDSECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.18 BESTVELTYPE

*Sets the velocity used in the BESTVEL and GPVTG logs*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command configures the source of the velocity that is output in the BESTVEL and GPVTG logs. Set the type to something other than BESTPOS when an unchanging velocity source with specific characteristics is needed.

The Doppler velocity is the highest-availability, lowest-latency velocity available from the receiver. Due to its low latency, it is also the noisiest velocity.



When the **RADARCONFIG** command (see page 323) is enabled, the **BESTVELTYPE** mode is set to **DOPPLER**.

**Message ID:** 1678

**Abbreviated ASCII Syntax:**

```
BESTVELTYPE mode
```

**Factory Default:**

```
BESTVELTYPE bestpos
```

**ASCII Example:**

```
BESTVELTYPE doppler
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BESTVELTYPE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	BESTPOS	1	Use the velocity from the same positioning filter that is being used to fill BESTPOS and GPGGA.	Enum	4	H
		DOPPLER	2	Always fill BESTVEL and GPVTG using Doppler-derived velocities.			

## 2.19 BLUETOOTHCONTROL

### *Configure Bluetooth on the receiver*

Platform: SMART2-B, SMART2-TB

The BLUETOOTHCONTROL command is used to set the Bluetooth operating mode.



Bluetooth support for Apple products requires that a custom application be developed.

**Message ID: 2265**

**Abbreviated ASCII Syntax:**

```
BLUETOOTH switch
```

**Factory Default:**

```
BLUETOOTHCONTROL ENABLE
```

**Examples:**

```
BLUETOOTHCONTROL ENABLE
```

```
BLUETOOTHCONTROL DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	BLUETOOTHCONTROL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Switch	DISABLE	0	Power off Bluetooth module	Enum	4	H+4
		ENABLE	1	Power on Bluetooth module and configure to support Bluetooth Serial Port Profile			
		LOAD	2	Load the Bluetooth firmware <sup>1</sup>			

<sup>1</sup>The LOAD switch should not normally be necessary. In rare situations, it might be possible for the Bluetooth module firmware to become corrupt. The LOAD switch can be used to force the Bluetooth image to be reloaded onto the module to correct the situation. Once the firmware is loaded onto the module, Bluetooth functionality is enabled automatically.

## 2.20 CANCONFIG

### Configure CAN ports

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use the CANCONFIG command to configure the hardware parameters of the CAN ports.

**Message ID: 884**

#### Abbreviated ASCII Syntax:

```
CANCONFIG port switch [speed]
```

#### Factory Default:

```
CANCONFIG CAN1 OFF 250K
```

```
CANCONFIG CAN2 OFF 250K
```

#### ASCII Example:

```
CANCONFIG CAN1 OFF 500K
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CANCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	CAN1	1	Physical CAN port ID <b>Note:</b> The OEM7500 supports the CAN2 port only. <b>Note:</b> SMART2 does not support the CAN2 port	Enum	4	H
		CAN2	2				
3	switch	ON	1	Sets the port to be On or Off the CAN bus	Enum	4	H+4
		OFF	0				
4	speed	See <i>Table 25: CAN Port Speed</i> on the next page		Physical CAN port speed (bits per second) (default = 250K	Enum	4	H+8



The CAN port must be set to OFF (using `CANCONFIG <port> OFF`) before the port speed can be changed.

**Table 25: CAN Port Speed**

<b>ASCII Value</b>	<b>Binary Value</b>
10K	0
20K	1
50K	2
100K	3
125K	4
250K	5
500K	6
1M	7

## 2.21 CCOMCONFIG

### Configure the CAN COM port

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Bind a CAN communication port to a J1939 node (see **J1939CONFIG** command on page 241) and specify the CAN protocol, PGN, priority and address for messages transmitted and received over the CCOM port.



This command is not supported on the SMART2.

**Message ID: 1902**

#### Abbreviated ASCII Syntax:

```
CCOMCONFIG port node protocol [pgn [priority [address]]]
```

#### Factory Default:

```
CCOMCONFIG ccom1 node1 J1939 61184 7 fe
CCOMCONFIG ccom2 node2 J1939 61184 7 fe
CCOMCONFIG ccom3 node1 J1939 126720 7 fe
CCOMCONFIG ccom4 none none 0 0 0
CCOMCONFIG ccom5 none none 0 0 0
CCOMCONFIG ccom6 none none 0 0 0
```

#### ASCII Example :

```
ccomconfig ccom1 node1 j1939 1792 6 1b
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CCOMCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	port	CCOM1	38	Name of CCOM port.	Enum	4	H
		CCOM2	39				
		CCOM3	40				
		CCOM4	41				
		CCOM5	42				
		CCOM6	43				
3	node	NODE1	1	The J1939 node to use. This binds a CCOM port to the CAN NAME/address associated with the node.	Enum	4	H+4
		NODE2	2				
4	protocol	See <i>Table 26: CAN Protocol</i> on the next page		CAN transport protocol to use.	Enum	4	H+8
5	pgn	0 - 131071		<p>Any valid PGN as defined by the J1939 protocol.</p> <p>All messages transmitted over this CCOM port will contain this PGN value.</p> <p>Only messages with this PGN will be received on this CCOM port.</p> <p><b>Note:</b> This value is ignored if the protocol is NMEA2000.</p>	Ulong	4	H+12
6	priority	0-7		<p>Default CAN message priority for transmitted messages. (Priority 0 is the highest priority)</p> <p><b>Note:</b> This value is ignored if the protocol is NMEA2000.</p>	Uchar	1	H+16

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	address	00 – FF		<p><b>00 – FD:</b> Transmit and receive messages to/from this address only.</p> <p><b>FE:</b> Transmit and receive message to/from the address of the first message received.</p> <p><b>FF:</b> Broadcast messages and receive messages from all addresses.</p> <p><b>Note:</b> This value is ignored if the protocol is NMEA2000.</p>	Hex	1	H+17

Table 26: CAN Protocol

Binary	ASCII	Description
2	J1939	J1939 single packet
3	NMEA2000	NMEA2000 (single packet, multi-packet, fast packet)
5	ISO11783	ISO 11783 transport protocol

## 2.22 CLOCKADJUST

### *Enables clock adjustments*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

All oscillators have some inherent drift. By default, the receiver attempts to steer the receiver's clock to accurately match GPS reference time. Use the **CLOCKADJUST** command to disable this function. The **TIME** log can then be used to monitor clock drift.



The **CLOCKADJUST** command should only be used by advanced users.



1. If the **CLOCKADJUST** command is ENABLED and the receiver is configured to use an external reference frequency (set in the **EXTERNALCLOCK** command (see page 168) for an external clock - TCXO, OCXO, RUBIDIUM, CESIUM, or USER), then the clock steering process takes over the VARF output pins and may conflict with a previously entered **EVENTOUTCONTROL** command (see page 165).
2. When using the **EXTERNALCLOCK** and **CLOCKADJUST** commands together, issue the **EXTERNALCLOCK** command (see page 168) first to avoid losing satellites.
3. When disabled, the range measurement bias errors continue to accumulate with clock drift.
4. Pseudorange, carrier phase and Doppler measurements may jump if the **CLOCKADJUST** mode is altered while the receiver is tracking.
5. When disabled, the time reported on all logs may be offset from GPS reference time. The 1PPS output may also be offset. The amount of this offset may be determined from the **TIME** log (see page 965).
6. A discussion on GPS reference time may be found in *GPS Reference Time Status* on page 53.

**Message ID:** 15

**Abbreviated ASCII Syntax:**

```
CLOCKADJUST switch
```

**Factory Default:**

```
CLOCKADJUST ENABLE
```

**ASCII Example:**

```
CLOCKADJUST DISABLE
```



The **CLOCKADJUST** command can be used to calibrate an internal oscillator. Disable the **CLOCKADJUST** mode in order to find out what the actual drift is from the internal oscillator. Watch the **CLOCKMODEL** log to see the drift rate and adjust the oscillator until the drift stops.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CLOCKADJUST header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Disallow adjustment of internal clock	Enum	4	H
		ENABLE	1	Allow adjustment of internal clock			

## 2.23 CLOCKCALIBRATE

### *Adjusts clock steering parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to adjust the control parameters of the clock steering loop. The receiver must be enabled for clock steering before these values can take effect. Refer to the **CLOCKADJUST** command on page 115 to enable or disable clock steering.

To disable the clock steering process, issue the **CLOCKADJUST DISABLE** command.

The current values used by the clock steering process are listed in the **CLOCKSTEERING** command (see page 550).



The values entered using the **CLOCKCALIBRATE** command are saved to non-volatile memory (NVM). To restore the values to their defaults, the **FRESET CLKCALIBRATION** command must be used. Issuing FRESET without the CLKCALIBRATION parameter will not clear the values (see **FRESET** command on page 193 for more details).

**Message ID: 430**

#### **Abbreviated ASCII Syntax:**

```
CLOCKCALIBRATE [mode] [period] [pulsewidth] [slope] [bandwidth]
```

#### **ASCII Example:**

```
CLOCKCALIBRATE AUTO
```



The receiver by default steers its INTERNAL VCTCXO but can be commanded to control an EXTERNAL reference oscillator. Use the **EXTERNALCLOCK** command (see page 168) to configure the receiver to use an external reference oscillator. If the receiver is configured for an external reference oscillator and configured to adjust its clock, then the clock steering loop attempts to steer the external reference oscillator through the use of the VARF signal. Note that the clock steering control process conflicts with the manual **EVENTOUTCONTROL** command (see page 165) and **FREQUENCYOUT** command (see page 190). It is expected that the VARF signal is used to provide a stable reference voltage by the use of a filtered charge pump type circuit (not supplied).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CLOCKCALIBRATE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	SET	0	Sets the period, pulsewidth, slope and bandwidth values into NVM for the currently selected steered oscillator. (INTERNAL or EXTERNAL)	Enum	4	H
		AUTO	1	Forces the receiver to do a clock steering calibration to measure the slope (change in clock drift rate with a 1 bit change in pulse width) and required pulsewidth to zero the clock drift rate. After the calibration, these values along with the period and bandwidth are entered into NVM and are then used from this point forward on the selected oscillator.			
		OFF	2	Terminates a calibration process currently underway. (default)			
3	period	0 to 262144		Signal period in 10 ns steps. Frequency Output = $100,000,000 / \text{Period}$ (default=11000)	Ulong	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	pulsewidth			<p>Sets the initial pulse width that should provide a near zero drift rate from the selected oscillator being steered. The valid range for this parameter is 10% to 90% of the period. If this value is not known, (in the case of a new external oscillator) then it should be set to ½ the period and the mode should be set to AUTO to force a calibration. (default=6600)</p>	Ulong	4	H+8
5	slope			<p>This value should correspond to how much the clock drift changes with a 1 bit change in the pulsewidth m/s/bit. The default values for the slope used for the INTERNAL and EXTERNAL clocks is -2.0 and -0.01 respectively. If this value is not known, then its value should be set to 1.0 and the mode should be set to AUTO to force a calibration. Once the calibration process is complete and using a slope value of 1.0, the receiver should be recalibrated using the measured slope and pulsewidth values (see the <b>CLOCKSTEERING</b> log on page 550). This process should be repeated until the measured slope value remains constant (less than a 5% change). (default=0.774)</p>	Float	4	H+12

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	bandwidth			This is the value used to control the smoothness of the clock steering process. Smaller values result in slower and smoother changes to the receiver clock. Larger values result in faster responses to changes in oscillator frequency and faster start up clock pull in. The default values are 0.03 and 0.001 Hz respectively for the INTERNAL and EXTERNAL clocks. (default=0.03)	Float	4	H+16



## 2.24 CLOCKOFFSET

### *Adjusts for delay in 1PPS output*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to remove a delay in the PPS output. The PPS signal is delayed from the actual measurement time due to two major factors:

- A delay in the signal path from the antenna to the receiver
- An intrinsic delay through the RF and digital sections of the receiver

The second delay is automatically accounted for by the receiver using a nominal value determined for each receiver type. However, since the delay from the antenna to the receiver cannot be determined by the receiver, an adjustment cannot automatically be made. The **CLOCKOFFSET** command can be used to adjust for this delay.



For dual antenna receivers, a **CLOCKOFFSET\_1** command should be issued to adjust for delay on the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the message is for the second antenna. To issue an ASCII message, enter **CLOCKOFFSETA\_1**.

**Message ID:** 596

**Abbreviated ASCII Syntax:**

```
CLOCKOFFSET offset
```

**Factory Default:**

```
CLOCKOFFSET 0
```

**ASCII Example:**

```
CLOCKOFFSET -15
```



There may be small variances in the delays for each cable or card. The **CLOCKOFFSET** command can be used to characterize each setup. For example, for a cable with a delay of 10 ns, the offset can be set to -10 to remove the delay from the PPS output.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CLOCKOFFSET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

---

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	offset	±2000		Specifies the offset in nanoseconds	Long	4	H

## 2.25 CNOUPDATE

### Sets the C/No update rate

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the C/No update rate.

**Message ID:** 849

#### Abbreviated ASCII Syntax:

```
CNOUPDATE rate
```

#### Factory Default:

```
CNOUPDATE default
```

#### ASCII Example:

```
CNOUPDATE 20Hz
```



Use the **CNOUPDATE** command for higher resolution update rate of the C/No measurements of the incoming GNSS signals. By default, the C/No values are calculated at approximately 4 Hz but this command allows you to increase that rate to 20 Hz.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	CNOUPDATE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	rate	DEFAULT	0	Default 5 Hz C/No update rate	Enum	4	H
		20HZ	1	20 Hz C/No update rate			

## 2.26 COMCONTROL

### *Controls the serial port hardware control lines*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720

This command is typically used to control the hardware RTS control line of the serial communication (COM) ports; or, for SPAN applications, to output a 1PPS timing signal on a COM TX port. The TOGGLEPPS mode of this command is typically used to supply a timing signal to a host PC computer by using the RTS line. The PULSEPPSHIGH mode of this command is typically used to supply the timing pulse for SPAN applications by using the TX line. The accuracy of controlling the COM control signals is better than 900  $\mu$ s. The other modes are typically used to control custom peripheral devices.



1. If handshaking is disabled, any of these modes can be used without affecting regular serial communications through the selected COM port. However, if handshaking is enabled, it may conflict with handshaking of the selected COM port, causing unexpected results.
2. To use the COM2 flow control signals, COM5 must be disabled. See OEM7600, OEM7700 and OEM7720 Multiplexed Port in the [OEM7 Installation and Operation User Manual](#) for more information.

**Message ID: 431**

#### **Abbreviated ASCII Syntax:**

```
COMCONTROL [port] [signal] [control]
```

#### **Factory Default:**

```
COMCONTROL COM1 TX DEFAULT
COMCONTROL COM2 TX DEFAULT
COMCONTROL COM3 TX DEFAULT
COMCONTROL COM4 TX DEFAULT
COMCONTROL COM5 TX DEFAULT
```

#### **ASCII Example 1:**

```
INTERFACEMODE COM5 NONE NONE OFF (OEM7600, OEM7700 and OEM7720 only)
SERIALCONFIG COM2 9600 N 8 1 CTS (to enable handshaking)
COMCONTROL COM2 RTS FORCELOW
```

#### **ASCII Example 2:**

To set a break condition on COM1:

```
COMCONTROL COM1 TX FORCEHIGH
```

A break condition remains in effect until it is cleared. To clear a break condition on COM1:

```
COMCONTROL COM1 TX DEFAULT
```

or

```
COMCONTROL COM1 TX FORCELOW
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	COMCONTROL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	COM1	1	Serial port to control.	Enum	4	H
		COM2	2				
		COM3	3				
		COM4	19				
		COM5	31				
3	signal	RTS	0	COM signal to control. The controllable COM signals are RTS and TX.	Enum	4	H+4
		TX	2				
4	control	See <i>Table 27: Control Options</i> below		Control applied to the signal	Enum	4	H+8

Table 27: Control Options

Binary	ASCII	Description
0	DEFAULT	Disables this command and returns the COM signal to its default state
1	FORCEHIGH	Immediately forces the signal high <sup>1</sup>
2	FORCELOW	Immediately forces the signal low <sup>2</sup>
3	TOGGLE	Immediately toggles the current state of the signal
4	TOGGLEPPS	Toggles the state of the selected signal within 900 $\mu$ s after each 1PPS event. The state change of the signal lags the 1PPS by an average value of 450 $\mu$ s. The delay of each pulse varies by a uniformly random amount less than 900 $\mu$ s
5	PULSEPPSLOW	Pulses the line low at a 1PPS event and to high 1 ms after it.
6	PULSEPPSHIGH	Pulses the line high for 1 ms at the time of a 1PPS event

<sup>1</sup>High on RS232; Low on UART<sup>2</sup>Low on RS232; High on UART

Table 28: Port, Signal and Control Availability on OEM7 Receivers

	OEM719	OEM729	OEM7600	OEM7700	OEM7720
<b>COM1 TX</b>	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH
<b>COM2 TX</b>	N/A	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH
<b>COM2 RTS</b>	N/A	DEFAULT FORCEHIGH FORCELOW TOGGLE TOGGLEPPS PULSEPPSLOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW TOGGLE TOGGLEPPS PULSEPPSLOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW TOGGLE TOGGLEPPS PULSEPPSLOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW TOGGLE TOGGLEPPS PULSEPPSLOW PULSEPPSHIGH
<b>COM3 TX</b>	N/A	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH
<b>COM4 TX</b>	N/A	N/A	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH
<b>COM5 TX</b>	N/A	N/A	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH	DEFAULT FORCEHIGH FORCELOW PULSEPPSHIGH

## 2.27 CONFIGCODE

### Configuration Code

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Configuration Codes can be used to make permanent changes to a receiver. These are configurations that persist even after non-volatile memory is cleared.



A configuration code is different from an auth code, which controls the functionality the receiver firmware provides.



This command writes to and reads from non-volatile memory. Therefore, it is required that the receiver maintain power throughout the execution of this command. Loss of power during this critical step could result in an unrecoverable error.

**Message ID: 1041**

#### Abbreviated ASCII Syntax:

```
CONFIGCODE [action] data1 data2 data3 data4 data5 description
```

#### Input Example:

```
CONFIGCODE ERASE_TABLE WJ4HDW,GM5Z99,T2M7DP,KG2T8T,KF7GKR, TABLECLEAR
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	CONFIGCODE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	action	REMOVE	0	Invalidate the specified configuration code.	Enum	4	H
		ADD	1	Add the specified configuration code. (default)			
		ADD_NO_RESET	3	Add the specified configuration code without resetting the receiver.			
3	data1			First data block	String [15]	Max. 16	H+4
4	data2			Second data block	String [15]	Max. 16	Variable
5	data3			Third data block	String [15]	Max. 16	Variable
6	data4			Fourth data block	String [15]	Max. 16	Variable
7	data5			Fifth data block	String [15]	Max. 16	Variable
8	description			Description of the config code	String [15]	Max. 16	Variable



For the fields with a format of **String [15]** (fields 3 to 8):

- The string must end with one or more '\0'.
- The number of valid characters should not be more than 14.
- The total number of characters (including padding '\0's) must be 4 byte aligned and not more than 16 bytes.



## 2.28 DATADECODESIGNAL

### *Enable/Disable navigation data decoding for GNSS signal*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to enable or disable framing and decoding of the navigation message for each GNSS signal. When disabled, the receiver will no longer output raw frame data, ephemeris or almanac data from that signal. Signals which do not yet have the built in capability to output raw frame data are not configurable. Note that if a primary signal such as GPSL1CA is disabled, it may cause the receiver to no longer function normally because this signal's data is essential for setting receiver time and computing positions.

The default setting for each GNSS signal, and which signals can be configured, is available in *Table 29: GNSS Signal Default and Configurability* below. The table also lists if the signal's navigation message is used to compute the satellite position. For the binary value and a longer description for each signal, see *Table 30: Signal Type* on page 131.

**Table 29: GNSS Signal Default and Configurability**

Signal	Primary Signal	Default	Configurable	Used for satellite positioning
GPSL1C	No	Disabled	No	No
GPSL1CA	Yes	Enabled	Yes	Yes
GPSL2Y	No	Disabled	No	No
GPSL2C	No	Disabled	Yes	No
GPSL2P	No	Disabled	No	No
GPSL5	No	Disabled	Yes	No
GLOL1CA	Yes	Enabled	Yes	Yes
GLOL2CA	No	Disabled	No	No
GLOL2P	No	Disabled	No	No
GLOL3	No	Disabled	No	No
SBASL1	No	Enabled	Yes	Yes
SBASL5	No	Enabled	Yes	Yes
GALE1	Yes	Enabled	Yes	Yes
GALE5A	No	Enabled	Yes	No
GALE5B	No	Enabled	Yes	Yes
GALALTB0C	No	Disabled	No	No
GALE6B	No	Enabled	Yes	No
GALE6C	No	Enabled	Yes	No

Signal	Primary Signal	Default	Configurable	Used for satellite positioning
BDSB1C	No	Enabled	Yes	Yes
BDSB1D1	Yes	Enabled	Yes	Yes
BDSB1D2	Yes	Enabled	Yes	Yes
BDSB2A	No	Enabled	Yes	Yes
BDSB2BI	No	Disabled	No	No
BDSB2D1	No	Disabled	No	No
BDSB2D2	No	Disabled	No	No
BDSB3D1	No	Disabled	No	No
BDSB3D2	No	Disabled	No	No
QZSSL1C	No	Disabled	No	No
QZSSL1CA	Yes	Enabled	Yes	Yes
QZSSL2C	No	Disabled	Yes	No
QZSSL5	No	Disabled	Yes	No
QZSSL6	No	Disabled	No	No
NAVICL5SPS	Yes	Enabled	Yes	Yes

**Message ID: 1532****Abbreviated ASCII Syntax:**

```
DATADECODESIGNAL signaltype switch
```

**Abbreviated ASCII Example:**

```
DATADECODESIGNAL GPSL2C enable
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DATADECODESIGNAL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	signal type	See Table 30: Signal Type below		GNSS Signal Type	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	switch	Disable	0	Enable or disable the data decoding	Enum	4	H+4
		Enable	1				

Table 30: Signal Type

Value (Binary)	Signal (ASCII)	Description
33	GPSL1CA	GPS L1 C/A-code
47	GPSL1CP	GPS L1C P-code
68	GPSL2Y	GPS L2 P(Y)-code
69	GPSL2C	GPS L2 C/A-code
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
2662	GLOL3	GLONASS L3
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10466	GALE5A	Galileo E5A
10499	GALE5B	Galileo E5B
10532	GALALTBOC	Galileo ALT-BOC
10565	GALE6C	Galileo E6C
10572	GALE6B	Galileo E6B
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2I with D1 navigation data
12804	BDSB2D2	BeiDou B2I with D2 navigation data
12877	BDSB3D1	BeiDou B3 with D1 navigation data
12880	BDSB3D2	BeiDou B3 with D2 navigation data
12979	BDSB1C	BeiDou B1C

<b>Value (Binary)</b>	<b>Signal (ASCII)</b>	<b>Description</b>
13012	BDSB2A	BeiDou B2a
13077	BDSB2BI	BeiDou B2b (I)
14753	QZSSL1CA	QZSS L1 C/A-code
14760	QZSSL1CP	QZSS L1C P-code
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5
14891	QZSSL6P	QZSS L6P
19073	NAVICL5SPS	NavIC L5 SPS

## 2.29 DATUM

### *Select a datum transformation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to select the ellipsoid used for converting Cartesian co-ordinates to geodetic (latitude, longitude and height) and the transformation applied to positions calculated by the receiver before they are output.

The datum transformation selected by this command will be applied to **all** solutions calculated by the receiver. Solutions based on corrections, like RTK or pseudorange differential, will naturally be in the datum of the corrections; for example, RTK networks in North America will transmit corrections that position users in NAD83. Uncorrected solutions, meanwhile, will be in WGS84 or a datum closely-aligned to it. If different solutions are in different datums then a single transformation will not work in all cases. Care must therefore be taken when using this command.

With RTK, it is normally intended to position in the datum of the base station. A transformation applied to the natural solution will shift the solution out of the base station datum. Consequently, this command is generally not suitable for use with RTK.

Use of the factory default datum of WGS84 will result in no transformation being applied to any solution. User defined datums can also be used, after being created using the **USERDATUM** command (see page 456) or **USEREXPDATUM** command (see page 458).

Datum transformations matter mostly for precise solutions like PPP and RTK, where the difference between datums is large relative to the solution accuracy. As already outlined, RTK solutions are, typically, already in the desired datum. Therefore, in most cases only PPP solutions will need transforming. To transform only the PPP solution, use the **OUTPUTDATUM** command on page 284.



Do not mix the **DATUM USER** and **OUTPUTDATUM** commands.

**Message ID:** 160

**Abbreviated ASCII Syntax:**

DATUM datum

**Factory Default:**

DATUM WGS84

**ASCII Example:**

DATUM USER

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DATUM header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	datum	WGS84	61	World Geodetic System 84	Enum	4	H
		USER	63	User defined datum set by the <b>USERDATUM</b> command (see page 456) and <b>USEREXPDATUM</b> command (see page 458).			

## 2.30 DATUMTRANSFORMATION

### *Set user-defined datum transformation parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to add or remove user-defined datum transformations. Each entry converts from one geodetic datum in the **GEODETICDATUMS** list to another. Generally, "From" is the native service datum for one of the position types, such as ITRF2014 for TerraStar or Veripos, and "To" is a desired output datum. Entries are indexed by the pair "From" "To".



The **OUTPUTDATUM** command (see page 284) must be sent for any user-defined transformations to take effect.

The receiver firmware includes a default set of datum transformations. These default transformations cannot be deleted, but can be overwritten with user-defined parameters. A default transformation that has been overwritten can be reverted to the default parameters by deleting the transformation. User-defined transformations will persist a factory reset; all user-defined transformations can be removed by sending **FRESET USER\_DATUM\_TRANSFORMATION** (see the **FRESET** command on page 193).

The transformation used by this command is the 7 parameter Helmert transformation:

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + (1 + s) \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Where:

**X, Y, Z** are the coordinates in the input datum

**X<sub>s</sub>, Y<sub>s</sub>, Z<sub>s</sub>** are the coordinates in the output datum

**T<sub>x</sub>, T<sub>y</sub>, T<sub>z</sub>** are the translation parameters corrected for translation rate (metres)

**R<sub>x</sub>, R<sub>y</sub>, R<sub>z</sub>** are the rotation parameters corrected for rotation rate (radians)

**s** is the scale difference corrected for scale rate (unitless)

A given parameter P is corrected for rate:

$$P(t) = P(t_r) + \dot{P}(t_r) (t - t_r)$$

Where:

**t<sub>r</sub>** is the reference epoch of the parameters

**t** is the current epoch

**P(t<sub>r</sub>)** is the parameter at the reference epoch

**Ḑ(t<sub>r</sub>)** is the rate of the parameter at the reference epoch

The entry contains the reference epoch, value of each parameter at the reference epoch and rate of each parameter at the reference epoch. Both the user-defined and default transformations are reported by the **DATUMTRANSFORMATIONS** log (see page 553).

**Message ID: 2297**





Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	tx			Translations at the reference epoch (metres)	Float	4	Variable
7	ty				Float	4	Variable
8	tz				Float	4	Variable
9	rx			Rotations at the reference epoch (milliarcseconds)	Float	4	Variable
10	ry				Float	4	Variable
11	rz				Float	4	Variable
12	scale			Scale difference at the reference epoch (parts per billion)	Float	4	Variable
13	tx_vel			Translation rates (metres/year)	Float	4	Variable
14	ty_vel				Float	4	Variable
15	tz_vel				Float	4	Variable
16	rx_vel			Rotation rates (milliarcsecond/year)	Float	4	Variable
17	ry_vel				Float	4	Variable
18	rz_vel				Float	4	Variable
19	scale_vel			Scale difference rate (parts per billion/year)	Float	4	Variable

## 2.31 DGPSTXID

### Sets DGPS station ID

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the station ID value for the receiver when it is transmitting corrections. This allows for the easy identification of which base station was the source of the data.

For example, if you want to compare RTCM and RTCMV3 corrections, you would be easily able to identify their base stations by first setting their respective DGPSTXID values.



If **DGPSTXID AUTO** is sent, the Base Station ID String (field 3) is mandatory, but any ID entered is always replaced by **ANY**.

**Message ID:** 144

**Abbreviated ASCII Syntax:**

DGPSTXID type ID

**Factory Default:**

DGPSTXID auto ANY

**ASCII Examples:**

DGPSTXID RTCM 2 - using an RTCM type and ID

DGPSTXID CMR 30 - using a CMR type and ID

DGPSTXID CMR ANY - using the default CMR ID

DGPSTXID RTCA d36d - using an RTCA type and ID

DGPSTXID RTCMV3 2050 - using an RTCMV3 type and ID

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DGPSTXID header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	type	RTCM	0	See Table 65: RTK Source Type on page 349	Enum	4	H
		RTCA	1				
		CMR	2				
		AUTO	10				
		RTCMV3	13				
		NOVATELX	14				
3	ID	Char[5]		Base Station ID String or ANY	Char[5]	8	H+4

## 2.32 DIFFCODEBIASCONTROL

*Enables/disables satellite differential code biases*



The **DIFFCODEBIASCONTROL** command has been deprecated. To disable differential code biases, use the **SETDIFFCODEBIASES** command (see page 395) to set the biases to zero.

**Message ID: 913**

## 2.33 DLLTIMECONST

### *Sets carrier smoothing*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the amount of carrier smoothing performed on the code measurements. An input value of 100 corresponds to approximately 100 seconds of smoothing. Upon issuing the command, the locktime (amount of continuous tracking in seconds) for all tracking satellites is reset to zero and each code smoothing filter is restarted. You must wait for at least the length of smoothing time for the new smoothing constant to take full effect. The optimum setting for this command depends on the application.



1. This command may not be suitable for every GNSS application.
2. When using DLLTIMECONST in differential mode with the same receivers, the same setting should be used at both the base and rover station. If the base and rover stations use different types of receivers, it is recommended that you use the command default value at each receiver (DLLTIMECONST <signaltype> 100).
3. There are several considerations when using the DLLTIMECONST command:
  - The attenuation of low frequency noise (multipath) in pseudorange measurements
  - The effect of time constants on the correlation of phase and code observations
  - The rate of “pulling-in” of the code tracking loop (step response)
  - The effect of ionospheric divergence on carrier smoothed pseudorange (ramp response)
4. To get unsmoothed pseudorange measurements, choose 0 as the time constant.

The primary reason for applying carrier smoothing to the measured pseudoranges is to mitigate the high frequency noise inherent in all code measurements. Adding more carrier smoothing by increasing the DLLTIMECONST value filters out lower frequency noise, including some multipath frequencies.

There are also some adverse effects of higher DLLTIMECONST values on some performance aspects of the receiver. Specifically, the time constant of the tracking loop is directly proportional to the DLLTIMECONST value and affects the degree of dependence between the carrier phase and pseudorange information. Carrier phase smoothing of the code measurements (pseudoranges) is accomplished by introducing data from the carrier tracking loops into the code tracking system. Phase and code data, collected at a sampling rate greater than about 3 time constants of the loop, are correlated (the greater the sampling rate, the greater the correlation). This correlation is not relevant if only positions are logged from the receiver, but is an important consideration if the data is combined in some other process such as post-mission carrier smoothing. Also, a narrow bandwidth in a feedback loop impedes the ability of the loop to track step functions. Steps in the pseudorange are encountered during initial lock-on of the satellite and when working in an environment conducive to multipath. A low DLLTIMECONST value allows the receiver to effectively adapt to these situations.



Also, increased carrier smoothing may cause problems when satellite signals are strongly affected by the ionosphere. The rate of divergence between the pseudoranges and phase-derived ranges is greatest when a satellite is low in the sky since the GPS signal must travel through a much “thicker” ionosphere. The tracking error of the receiver is greatest at these times when a lot of carrier smoothing is implemented. In addition, changing periods of ionospheric activity (diurnal changes and the 11-year cycle) influences the impact of large DLLTIMECONST values. It is important to realize that the advantages of carrier smoothing do not come without some trade off in receiver performance. The factory default DLLTIMECONST value of 100 was selected as an optimal compromise of the above considerations. For the majority of applications, this default value should be appropriate. However, the flexibility exists to adjust the parameter for specific applications by users who are familiar with the consequences.

### Message ID: 1011

#### Abbreviated ASCII Syntax:

```
DLLTIMECONST signaltype timeconst
```

#### Factory Defaults:

```
DLLTIMECONST <signaltype> 100
```

#### Example:

```
DLLTIMECONST GPSL2C 100
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DLLTIMECONST header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	signal type	See <i>Table 31: Signal Type</i> below		Signal type	Enum	4	H
3	time const			Time constant (seconds)	Ulong	4	H+4

**Table 31: Signal Type**

Value (Binary)	Signal (ASCII)	Description
33	GPSL1CA	GPS L1 C/A-code
47	GPSL1CP	GPS L1C P-code
68	GPSL2Y	GPS L2 P(Y)-code

Value (Binary)	Signal (ASCII)	Description
69	GPSL2C	GPS L2 C/A-code
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
2662	GLOL3	GLONASS L3
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10466	GALE5A	Galileo E5A
10499	GALE5B	Galileo E5B
10532	GALALTBOC	Galileo ALT-BOC
10565	GALE6C	Galileo E6C
10572	GALE6B	Galileo E6B
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2I with D1 navigation data
12804	BDSB2D2	BeiDou B2I with D2 navigation data
12877	BDSB3D1	BeiDou B3 with D1 navigation data
12880	BDSB3D2	BeiDou B3 with D2 navigation data
12979	BDSB1C	BeiDou B1C
13012	BDSB2A	BeiDou B2a
13077	BDSB2BI	BeiDou B2b (I)
14753	QZSSL1CA	QZSS L1 C/A-code
14760	QZSSL1CP	QZSS L1C P-code
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5
14891	QZSSL6P	QZSS L6P
19073	NAVICL5SPS	NavIC L5 SPS

## 2.34 DNSCONFIG

### *Manually configures Ethernet DNS servers*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command is part of the Ethernet set up. It is used to configure the Domain Name Servers (DNS) so that host names can be used instead of IP addresses.



The **DNSCONFIG** command configures a DNS server for the Ethernet interface, ETHA.

The **DNSCONFIG** command will fail if the IP address for the Ethernet interface, ETHA, is configured to use DHCP. Ensure the IP address for the Ethernet interface is configured to use a static IP address before entering the **DNSCONFIG** command.

When using DHCP, the DNS server received using DHCP is used and the DNS server configured by **DNSCONFIG** is ignored.

**Message ID:** 1244

#### Abbreviated ASCII Syntax:

```
DNSCONFIG DNSServers IP
```

#### Factory Default:

```
DNSCONFIG 0
```

#### ASCII Example:

```
DNSCONFIG 1 192.168.1.5
```

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DNSCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	DNSServers	0	0	Number of DNS servers	Enum	4	H
		1	1	If this field is set to 0, an IP address is not required.			



---

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
3	IP	ddd.ddd. ddd.ddd		IP address of primary DNS server	String [16]	variable <sup>1</sup>	H+4

---

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.35 DOPPLERWINDOW

### *Change the behavior of the doppler search*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Change the behavior of the doppler search for automatic channel assignments in the receiver. This command is used where the SV doppler may be outside of the normal receiver doppler search window of +/-16000 Hz.



The **DOPPLERWINDOW** command should only be used by advanced users.

**Message ID:** 1512

#### Abbreviated ASCII Syntax:

```
DOPPLERWINDOW system setting doppler_window
```

#### Factory Default:

```
DOPPLERWINDOW ALL AUTO 0
```

#### ASCII Example:

```
DOPPLERWINDOW GPS USER 20000
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Value	Binary Offset
1	Command header	–	–	DOPPLERWINDOW header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	system	See <i>Table 32: Satellite System</i> on the next page		Satellite system to configure.	Enum	4	H
3	setting	AUTO	0	Doppler window mode	Enum	4	H+4
		USER	1				
4	doppler_window	0 to 50000		Doppler window to search in Hz	Ulong	4	H+8
5	Reserved				Long	4	H+12

**Table 32: Satellite System**

<b>System</b>	<b>Value</b>
ALL	3
GPS	99
GLONASS	101
SBAS	100
GALILEO	102
BEIDOU	103
QZSS	104
NAVIC	105

## 2.36 DUALANTENNAALIGN

### *Dual Antenna ALIGN configuration*

**Platform:** OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2, CPT7

Use this command to enable or disable ALIGN and configure the ALIGN operation rates on a dual antenna receiver.

**Message ID:** 1761

**Abbreviated ASCII Syntax:**

```
DUALANTENNAALIGN switch [obsrate] [posrate]
```

**Factory Default:**

```
DUALANTENNAALIGN enable 1 1
```

**Example:**

```
DUALANTENNAALIGN enable 10 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	DUALANTENNAALIGN header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	Switch	DISABLE	0	Disable the dual antenna ALIGN feature	Enum	4	H
		ENABLE	1	Enable the dual antenna ALIGN feature			
3	obsrate	1, 2, 4, 5, 10, 20		Rate in Hz at which heading output is required (default = 1 Hz)	Ulong	4	H+4
4	posrate	1, 2, 4, 5, 10, 20		Rate in Hz at which MASTERPOS output is required (default = 1 Hz)	Ulong	4	H+8



Dual antenna ALIGN rates (*obsrate* and *posrate*) are limited to the maximum position rate allowed by the receiver model.

## 2.37 DYNAMICS

### *Tunes receiver parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to adjust the receiver dynamics to that of an application. It is used to optimally tune receiver parameters.

The **DYNAMICS** command adjusts the Tracking State transition time out value of the receiver, see *Table 157: Tracking State* on page 808. When the receiver loses the position solution, see *Table 86: Solution Status* on page 523, it attempts to steer the tracking loops for fast reacquisition (5 s time-out by default). The **DYNAMICS** command adjusts this time-out value, effectively increasing the steering time. The three states AIR, LAND or FOOT set the time-out to 5, 10 or 20 seconds respectively.



The **DYNAMICS** command should only be used by advanced users. The default of AUTO should **not** be changed except under very specific conditions.

**Message ID:** 258

**Abbreviated ASCII Syntax:**

DYNAMICS settings

**Factory Default:**

DYNAMICS AUTO

**Example:**

DYNAMICS FOOT

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	DYNAMICS header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	settings	See <i>Table 33: User Dynamics</i> on the next page		Receiver dynamics based on the current environment	Enum	4	H

**Table 33: User Dynamics**

Binary	ASCII	Description
0	AIR	Receiver is in an aircraft or a land vehicle, for example a high speed train, with velocity greater than 110 km/h (30 m/s). This is also the most suitable dynamic for a jittery vehicle at any speed.
1	LAND	Receiver is in a stable land vehicle with velocity less than 110 km/h (30 m/s).
2	FOOT	Receiver is being carried by a person with velocity less than 11 km/h (3 m/s).
3	AUTO	Receiver monitors dynamics and adapts behavior accordingly



Qualifying North American Solar Challenge cars annually weave their way through 1000's of miles between the US and Canada. GNSS keeps them on track through many intersections on secondary highways and gives the Calgary team constant intelligence on the competition's every move. In this case, with average speeds of 46 miles/hour and at times a jittery vehicle, AIR is the most suitable dynamic.

## 2.38 ECHO

### *Sets port echo*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set a port to echo.

**Message ID:** 1247

**Abbreviated ASCII Syntax:**

```
ECHO [port] echo
```

**Factory Default:**

```
ECHO COM1 OFF
```

```
ECHO COM2 OFF
```

```
ECHO COM3 OFF (not supported on OEM719)
```

```
ECHO COM4 OFF (OEM7600, OEM7700 and OEM7720 only)
```

```
ECHO COM5 OFF (OEM7600, OEM7700 and OEM7720 only)
```

```
ECHO USB1 OFF
```

```
ECHO USB2 OFF
```

```
ECHO USB3 OFF
```

```
ECHO ICOM1 OFF (not supported on OEM719)
```

```
ECHO ICOM2 OFF (not supported on OEM719)
```

```
ECHO ICOM3 OFF (not supported on OEM719)
```

```
ECHO ICOM4 OFF (not supported on OEM719)
```

```
ECHO ICOM5 OFF (not supported on OEM719)
```

```
ECHO ICOM6 OFF (not supported on OEM719)
```

```
ECHO ICOM7 OFF (not supported on OEM719)
```

```
ECHO SCOM1 OFF
```

```
ECHO SCOM2 OFF
```

```
ECHO SCOM3 OFF
```

```
ECHO SCOM4 OFF
```

**ASCII Example:**

```
ECHO COM1 ON
```

```
ECHO ON
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ECHO header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See Table 34: <i>Communications Port Identifiers</i> below		Port to configure (default = THISPORT)	Enum	4	H
3	echo	OFF	0	Sets port echo to off	Enum	4	H+4
		ON	1	Sets port echo to on			

**Table 34: Communications Port Identifiers**

ASCII Port Name	Binary Value
ALL	8
BT1	33
CCOM1	38
CCOM2	39
CCOM3	40
CCOM4	41
CCOM5	42
CCOM6	43
COM1	1
COM2	2
COM3	3
COM4	19
COM5	31
COM6	32
COM7	34
COM8	35
COM9	36



ASCII Port Name	Binary Value
COM10	37
ETH1	20
FILE	7
ICOM1	23
ICOM2	24
ICOM3	25
ICOM4	29
ICOM5	46
ICOM6	47
ICOM7	48
IMU	21
NCOM1	26
NCOM2	27
NCOM3	28
NOPORT	0
SCOM1	49
SCOM2	50
SCOM3	51
SCOM4	52
THISPORT	6
USB1	13
USB2	14
USB3	15
WCOM1	30

## 2.39 ECUTOFF

### *Sets satellite elevation cut-off for GPS Satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the elevation cut-off angle for tracked GPS satellites. The receiver does not start automatically searching for a GPS satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the ECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle; it could be used in these situations:

- The antenna is at a high altitude, and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **ECUTOFF** command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command (see page 156) to set the cut-off angle for any system.



For dual antenna receivers, this command applies to both the primary and secondary antennas.



A low elevation satellite is a satellite the receiver tracks just above the horizon. Generally, a satellite is considered low elevation if it is between 0 and 15 degrees above the horizon.

There is no difference between the data transmitted from a low elevation satellite and that transmitted from a higher elevation satellite. However, differences in the signal path of a low elevation satellite make their use less desirable. Low elevation satellite signals have more error due to the increased amount of atmosphere they must travel through. In addition, signals from low elevation satellites don't fit the assumption that a signal travels in air nearly the same as in a vacuum. As such, using low elevation satellites in the solution results in greater position inaccuracies.

The elevation cut-off angle is specified with **ECUTOFF** to ensure that noisy, low elevation satellite data below the cut-off is not used in computing a position. If post-processing data, it is still best to collect all data (even that below the cut-off angle). Experimenting with different cut-off angles can then be done to provide the best results. In cases where there are not enough satellites visible, a low elevation satellite may actually help in providing a useful solution.

**Message ID: 50**

**Abbreviated ASCII Syntax:**

ECUTOFF angle

**Factory Default:**

ECUTOFF 5.0

**ASCII Example:**

ECUTOFF 10.0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.40 ELEVATIONCUTOFF

### *Sets the elevation cut-off angle for tracked satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **ELEVATIONCUTOFF** command is used to set the elevation cut-off angle for tracked satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when the satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (refer to the **ASSIGN** command on page 76).

In either case, satellites below the elevation cut-off angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **ELEVATIONCUTOFF** command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



This command combines the following commands into one convenient command: **ECUTOFF**, **GLOECUTOFF**, **GALECUTOFF**, **QZSSECUTOFF**, **SBASECUTOFF**, **BDSECUTOFF** and **NAVICECUTOFF**.



For dual antenna receivers, this command applies to both the primary and secondary antennas.



A low elevation satellite is a satellite the receiver tracks just above the horizon. Generally, a satellite is considered low elevation if it is between 0 and 15 degrees above the horizon.

There is no difference between the data transmitted from a low elevation satellite and that transmitted from a higher elevation satellite. However, differences in the signal path of a low elevation satellite make their use less desirable. Low elevation satellite signals have more error due to the increased amount of atmosphere they must travel through. In addition, signals from low elevation satellites don't fit the assumption that a signal travels in air nearly the same as in a vacuum. As such, using low elevation satellites in the solution results in greater position inaccuracies.

The elevation cut-off angle is specified with the **ELEVATIONCUTOFF** command to ensure that noisy, low elevation satellite data below the cut-off is not used in computing a position. If post-processing data, it is still best to collect all data (even that below the cutoff angle). Experimenting with different cut-off angles can then be done to provide the best results. In cases where there are not enough satellites visible, a low elevation satellite may actually help in providing a useful solution.

**Message ID: 1735**

**Abbreviated ASCII Syntax:**

ELEVATIONCUTOFF Constellation Angle [Reserved]

**Factory default:**

ELEVATIONCUTOFF ALL 5.0 0

**ASCII Example:**

ELEVATIONCUTOFF GPS 5

ELEVATIONCUTOFF ALL 5

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ELEVATIONCUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	Constellation	GPS	0	Sets the cut-off angle for GPS Constellation satellites only.	Enum	4	H
		GLONASS	1	Sets the cut-off angle for GLONASS constellation satellites only.			
		SBAS	2	Sets the cut-off angle for SBAS constellation satellites only.			
		GALILEO	5	Sets the cut-off angle for Galileo constellation satellites only.			
		BEIDOU	6	Sets the cut-off angle for BeiDou constellation satellites only.			
		QZSS	7	Sets the cut-off angle for QZSS constellation satellites only.			
		NAVIC	9	Sets the cut-off angle for NavIC constellation satellites only.			
		ALL	32	Sets the cut-off angle for all satellites regardless of the constellation.			
3	Angle	±90.0 degrees		Elevation cut-off angle relative to the horizon.	Float	4	H+4
4	Reserved	0		Reserved Field (optional)	Ulong	4	H+8

## 2.41 ETHCONFIG

### *Configures Ethernet physical layer*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command is used to configure the Ethernet physical layer.

**Message ID: 1245**

#### Abbreviated ASCII Syntax:

```
ETHCONFIG interface_name [speed] [duplex] [crossover] [power_mode]
```

#### Factory Default:

```
ETHCONFIG etha auto auto auto powerdown (OEM7 receiver cards)
```

```
ETHCONFIG etha auto auto auto auto (PwrPak7)
```

#### ASCII Example:

```
ETHCONFIG etha 100 full mdix normal
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ETHCONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	interface_name	ETHA	2	Name of the Ethernet interface	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	speed	AUTO	1	Auto-negotiate speed (default) AUTO is the recommended value for the speed parameter. If setting speed to AUTO, duplex must be set to AUTO at the same time otherwise a "parameter 3 out of range" error occurs.	Enum	4	H+4
		10	2	Force 10BaseT			
		100	3	Force 100BaseT			
4	duplex	AUTO	1	Auto-negotiate duplex (default) If setting duplex to AUTO, speed must be set to AUTO at the same time otherwise a "parameter 3 out of range" error occurs.	Enum	4	H+8
		HALF	2	Force half duplex			
		FULL	3	Force full duplex			
5	crossover	AUTO	1	Auto-detect crossover (default)	Enum	4	H+12
		MDI	2	Force MDI (straight through)			
		MDIX	3	Force MDIX (crossover)			



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	power_mode	AUTO	1	Energy detect mode (default for PwrPak7)	Enum	4	H+16
		POWERDOWN	2	Soft power down mode (default for OEM7 receiver cards)			
		NORMAL	3	Normal mode			



The crossover parameter is ignored on OEM7 receivers, as the hardware automatically detects the cable connection and configures the interface for proper communication. For backwards compatibility, the crossover options are still accepted, but have no functional impact.

## 2.42 EVENTINCONTROL

### *Controls Event-In input triggers*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command controls up to four Event-In input triggers. Each input can be used as an event strobe.

When used as an event strobe, an accurate GPS time or position is applied to the rising or falling edge of the input event pulse (refer to the **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** log on page 699, **MARKPOS**, **MARK2POS**, **MARK3POS** and **MARK4POS** log on page 696 or **MARK1PVA**, **MARK2PVA**, **MARK3PVA** and **MARK4PVA** log on page 1131). Each input strobe is usually associated with a separate device, therefore different solution output lever arm offsets can be applied to each strobe. When used as an Event Input Trigger, it is possible to overwhelm the receiver with a very high rate of input events that impacts the performance of the receiver. For this reason, the receiver internally throttles the rate at which it responds to input events. The limit is 200 Hz.

**Message ID: 1637**



#### **Abbreviated ASCII Syntax:**

```
EVENTINCONTROL mark switch [polarity] [t_bias] [t_guard]
```

#### **ASCII Example:**

```
EVENTINCONTROL MARK1 ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	EVENTINCONTROL header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mark	MARK1	0	Choose which Event-In Mark to change. This value must be specified.	Enum	4	H
		MARK2	1	 MARK3 is available only on OEM7600, OEM7700, OEM7720, PwrPak7, CPT7 and CPT7700 receivers.			
		MARK3	2				
		MARK4	3	 MARK4 is available only on OEM7600, OEM7700 and OEM7720 receivers.			
3	switch	DISABLE	0	Disables Event Input	Enum	4	H+4
		EVENT	1	Enables Event Input			
		ENABLE	3	A synonym for the EVENT option (for compatibility with previous releases)			
4	polarity	NEGATIVE	0	Negative polarity (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity			
5	t_bias	default: 0 minimum: -999,999,999 maximum: 999,999,999		A constant time bias in nanoseconds can be applied to each event pulse. Typically this is used to account for a transmission delay.	Long	4	H+12

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	t_guard	default: 4 minimum: 2 maximum: 3,599,999		The time guard specifies the minimum number of milliseconds between pulses. This is used to coarsely filter the input pulses.	Ulong	4	H+16

## 2.43 EVENTOUTCONTROL

### *Control Event Out properties*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This command configures up to seven Event Out output strobes. The event strobes toggle between 3.3 V and 0 V. The pulse consists of two periods: one active period followed by a not active period. The start of the active period is synchronized with the top of the GNSS time second and the signal polarity determines whether the active level is 3.3 V or 0 V. The not active period immediately follows the active period and has the alternate voltage.



The outputs that are available vary according to the platform.



A 100 MHz clock is used internally to create these output signals. As a result, all period values are limited to 10 ns steps.



The EVENTOUT outputs cannot synchronize with GPS time until the receiver reaches FINESTEERING time status. As the receiver transitions to GPS time, there may be additional, unexpected pulses on the EVENTOUT signals.

**Message ID: 1636**





#### **Abbreviated ASCII Syntax:**



```
EVENTOUTCONTROL mark switch [polarity] [active_period] [non_active_period]
```

#### **ASCII Example:**

```
EVENTOUTCONTROL MARK3 ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	EVENTOUTCONTROL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mark	MARK1	0	Choose which Event Out Mark to change. This value must be specified.	Enum	4	H
		MARK2	1	 On OEM719 and OEM729 receivers, only MARK1 is available.			
		MARK3	2	 On OEM7600, OEM7700 and OEM7720 receivers, only MARK1 through MARK4 are available.			
		MARK4	3	 MARK4 is not available on the CPT7 and CPT7700.			
		MARK5	4	 MARK3 defaults to a PPS type signal on the CPT7 and CPT7700.			
		MARK6	5				
		MARK7	6				
3	switch	DISABLE	0	Disables the Event output	Enum	4	H+4
		ENABLE	1	Enables the Event output			
4	polarity	NEGATIVE	0	Negative polarity (active = 0V) (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity (active = 3.3V)			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	active_period		default: 500,000,000 minimum: 10 maximum: 999,999,990	Active period of the Event Out signal in nanoseconds. 10 ns steps must be used.  If the value entered is not a multiple of 10, it will be rounded down to the nearest 10 ns.	Ulong	4	H+12
6	non_active_period		default: 500,000,000 minimum: 10 maximum: 999,999,990	Non-active period of the Event Out signal in nanoseconds. 10 ns steps must be used.  If the value entered is not a multiple of 10, it will be rounded down to the nearest 10 ns.	Ulong	4	H+16



The sum of the active period and inactive period should total 1,000,000,000 ns. If the total exceeds one full second, both the active and inactive period duration will be as given and the EVENTOUT pulses will no longer be synchronized with the 1PPS output pulse.

Alternately, the sum of the active and inactive periods may be less than 1,000,000,000 ns, but should divide evenly into 1,000,000,000 ns. For example, if the active period is 150,000,000 and the inactive period is 50,000,000, the sum of the periods is 200,000,000 ns which divides evenly into one full second.

If the sum is less than one full second and does not divide evenly into one full second, the active and inactive period duration will be as given and the EVENTOUT pulses will no longer be synchronized with the 1PPS output pulse.

A 100 MHz clock is used internally to create these output signals. As a result, all period values are limited to 10 ns steps.

## 2.44 EXTERNALCLOCK

### *Sets external clock parameters*

**Platform:** OEM729

The **EXTERNALCLOCK** command is used to enable the OEM7 card to operate with an optional external oscillator. You are able to optimally adjust the clock model parameters of these receivers for various types of external clocks.



1. This command affects the interpretation of the CLOCKMODEL log.
2. If the **EXTERNALCLOCK** command is enabled and set for an external clock (TCXO, OCXO, RUBIDIUM, CESIUM or USER) and the **CLOCKADJUST** command (see page 115) is ENABLED, then the clock steering process takes over the VARF output pins and may conflict with a previously entered **EVENTOUTCONTROL** command (see page 165). If clocksteering is not used with the external oscillator, the clocksteering process must be disabled by using the **CLOCKADJUST disable** command.
3. When using the **EXTERNALCLOCK** command and **CLOCKADJUST** command (see page 115) together, issue the **EXTERNALCLOCK** command first to avoid losing satellites.

There are three steps involved in using an external oscillator:

1. Follow the procedure outlined in the [OEM7 Installation and Operation User Manual](#) to connect an external oscillator to the OEM7.
2. Using the **EXTERNALCLOCK** command, select a standard oscillator and its operating frequency.
3. Using the **CLOCKADJUST** command (see page 115), disable the clocksteering process if external clocksteering is not used.

An unsteered oscillator can be approximated by a three-state clock model, with two states representing the range bias and range bias rate, and a third state assumed to be a Gauss-Markov (GM) process representing the range bias error generated from satellite clock dither. The third state is included because the Kalman filter assumes an (unmodeled) white input error. The significant correlated errors produced by satellite clock dither are obviously not white and the Markov process is an attempt to handle this kind of short term variation.

The internal units of the new clock model's three states (offset, drift and GM state) are metres, metres per second and metres. When scaled to time units for the output log, these become seconds, seconds per second and seconds, respectively.

The user has control over 3 process noise elements of the linear portion of the clock model. These are the  $h_0$ ,  $h_{-1}$  and  $h_{-2}$  elements of the power law spectral density model used to describe the frequency noise characteristics of oscillators:

$$S_y(f) = \frac{h_{-2}}{f^2} + \frac{h_{-1}}{f} + h_0 + h_1 f + h_2 f^2$$

where  $f$  is the sampling frequency and  $S_y(f)$  is the clock's power spectrum. Typically only  $h_0$ ,  $h_{-1}$ , and  $h_{-2}$  affect the clock's Allan variance and the clock model's process noise elements.

Before using an optional external oscillator, several clock model parameters must be set. There are default settings for a Voltage-Controlled Temperature-Compensated Crystal Oscillator (VCTCXO), Ovenized Crystal Oscillator (OCXO), Rubidium and Cesium standard, which are given in *Table 35: Clock Type* on page 170. You may alternatively choose to supply customized settings.





The **EXTERNALCLOCK** command configures whether the receiver uses its own internal temperature-compensated crystal oscillator or that of an external oscillator as a frequency reference. It also sets which clock model is used for an external oscillator.

To force the OEM7 to use the internal oscillator, use the **EXTERNALCLOCK disable** command and physically disconnect the external oscillator input. Do not use the EXTERNALCLOCK OCXO, CESIUM, RUBIDIUM or USER parameters if there is no external oscillator connected to the OEM7.

### Message ID: 230

#### Abbreviated ASCII Syntax:

```
EXTERNALCLOCK clocktype [freq] [h0 [h-1 [h-2]]]
```

#### Factory Default:

```
EXTERNALCLOCK disable
```

#### ASCII Examples:

```
EXTERNALCLOCK USER 10MHZ 1.0167e-23 6.87621e-25 8.1762e-26
```

```
EXTERNALCLOCK TCXO 5MHZ
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	EXTERNALCLOCK header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	clocktype	See <i>Table 35: Clock Type</i> on the next page		Clock type	Enum	4	H
3	freq	5MHz	1	Optional frequency. If a value is not specified, the default is 5 MHz	Enum	4	H+4
		10MHz	2				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	$h_0$	1.0 e-35 to 1.0 e-18		Optional timing standards. These fields are only valid when the USER clocktype is selected. Do not use h values with VCTCXO, OCXO, CESIUM or RUBIDIUM clock types. The h values for these options are fixed, see <i>Table 36: Pre-Defined Values for Oscillators</i> below (default=0.0)	Double	8	H+8
5	$h_{-1}$	1.0 e-35 to 1.0 e-18			Double	8	H+16
6	$h_{-2}$	1.0 e-35 to 1.0 e-18			Double	8	H+24

Table 35: Clock Type

ASCII	Binary	Description
DISABLE	0	Turns the external clock input off, reverts back to the on-board VCTCXO. When used in a binary command, use the parameter defaults (i.e. freq=1, $h_0=0$ , $h_{-1}=0$ , $h_{-2}=0$ ).
TCXO	1	Sets the predefined values for a VCTCXO
OCXO	2	Sets the predefined values for an OCXO
RUBIDIUM	3	Sets the predefined values for a rubidium oscillator
CESIUM	4	Sets the predefined values for a cesium oscillator
USER	5	Defines custom process noise elements

Table 36: Pre-Defined Values for Oscillators

Clock Type	$h_0$	$h_{-1}$	$h_{-2}$
VCTCXO	1.0 e-21	1.0 e-20	1.0 e-20
OCXO	2.51 e-26	2.51 e-23	2.51 e-22
Rubidium	1.0 e-23	1.0 e-22	1.3 e-26
Cesium	2.0 e-20	7.0 e-23	4.0 e-29

## 2.45 FILEAUTOTRANSFER

*Enables/Disables automatic file transfer*

Platform: PwrPak7



PwrPak7M variants do not support this command.

Use this command to configure the automatic transfer function from internal memory to an external USB stick. If the mode is set to COPY or MOVE, all log files, except the file currently being logged to, will be automatically transferred to a USB stick when the USB stick is inserted.

This command will transfer all recorded log files to the USB stick provided the USB stick has enough free space to hold all the data. **Files too large to fit in the remaining space on the USB stick are skipped.**

The command must be issued before the USB stick is inserted. If the command is not issued first, the USB stick must be removed and reinserted to trigger the auto transfer.

The status of the transfer can be viewed by logging the **FILETRANSFERSTATUS** log (see page 566).

A transfer in progress can be canceled by issuing the **FILETRANSFER CANCEL** command.

The settings for this command can be saved using the **SAVECONFIG** command (see page 366).

**Message ID: 2135**

**Abbreviated ASCII Syntax:**

```
FILEAUTOTRANSFER [TransferMode]
```

**ASCII Example:**

```
FILEAUTOTRANSFER COPY
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	FILEAUTOTRANSFER header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	TransferMode	1	OFF	Automatic copy/move is disabled (default)	Ulong	4	H
		2	COPY	Automatically copies all files			
		3	MOVE	Automatically copies all files and then deletes them from internal memory after a successful copy			



For the fastest transfer of files to an external memory stick, it is recommended that logging to a file be stopped.

## 2.46 FILECONFIG

### *Open or close a log file*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



PwrPak7M variants do not support this command.

To record logs, log requests are sent to the FILE port. Before the logs sent to the FILE port can be saved in a file, the file must be opened using the **FILECONFIG** command.

When configured to be open, a log file will be opened when the active file media is ready and has sufficient space. Once a log file is opened, any logs requested for the FILE port are recorded to the file.

Use the **FILESTATUS** log (see page 560) to determine the state of the log file.



The file media is separately configured:

- On receiver cards, this is always USBSTICK, which is the only media available.
- On the PwrPak7, the active file media is configured using the **FILEMEDIACONFIG** command on page 176.

When a file is opened, the file name is automatically generated based on the following format:

<PSN>\_<INDEX>.LOG

where:

- <PSN> is the PSN of the receiver
- <INDEX> is a number from 1 to 511.

The lowest number that produces an unused file name is selected. If there is no such number available, the **FILESTATUS** log (see page 560) will report an error.

The number is not zero-padded (i.e. the sequence is as follows: 1,2, ... ,9,10,11,12, ... ,99,100, ... , 510,511).

When a file is closed and the receiver has a valid time, the file is renamed based on the following format:

<PSN>\_<UTC Date>\_<UTC Time>.LOG

where:

- <PSN> is the PSN of the receiver
- <UTC\_Date> is the UTC date in the format YYYY-MM-DD
- <UTC\_Time> is the UTC time in the format HH-MM-SS

Example file name: NOV12001200A\_2017-01-10\_12-14-34.LOG

When a file is closed, but the receiver does not have a valid time, the file is left with its automatically generated name.

**Other Notes:**

- The FILE port represents the internal logging to flash memory. It has a NOVATEL Interface Mode - output only, no input is possible.
- Only logs that are published after the log file is open are recorded.
- Only one log file can be open at a time.
- Logs requested to the FILE port are still produced even if the log file is closed; however the logs are not recorded. (This is similar to requesting logs to COM4 when there's no cable on COM4.) If a new log file is opened, recording of the previously requested logs continues with the new file.
- When a file is closed, the log file is renamed to the format <PSN>\_<UTC Date>\_<UTC Time>.LOG, where the UTC time is the time when the file is closed. If the time is not available, the file is not renamed. If there is already a file with the intended name, the file is not renamed.
- After closing a file, the file system will be flushed to ensure that all data is written to the media.
- A disk is considered "full" when it has <= 10 MB of free space. This buffer is left in place to allow the system time and space to open up a new file if required.

**Message ID: 2116****Abbreviated ASCII Syntax:**

```
FILECONFIG FileOperation
```

**Factory Default:**

```
FILECONFIG CLOSE
```

**Example:**

```
FILECONFIG OPEN
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FILECONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	FileOperation	OPEN	1	Open (create) a new logging file	Enum	4	H
		CLOSE	2	Close the logging file			

## 2.47 FILEDELETE

**Deletes files from the currently selected mass storage device**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



PwrPak7M variants do not support this command.

Use this command to delete a single file, or use the wild card symbol (\*) to delete all files, from the logging directory of the currently selected file media. This command will not delete a file if it is currently open for logging. Use the **FILESTATUS** log (see page 560) to determine the state of the log file.



The wild card symbol deletes all files in the directory. It cannot be used to delete a subset of the files in the directory. For example, the command **FILEDELETE \*.LOG** will be rejected by the receiver.

The file media is separately configured:

- On receiver cards, the file media is always USBSTICK, which is the only media available.
- On enclosure products, the active file media is configured using a product-specific command, such as **FILEMEDIACONFIG** command (see page 176).

The list of files stored on the currently selected file media can be retrieved using the **FILELIST** log on page 558.

**Message ID: 2190**

**Abbreviated ASCII Syntax:**

```
FILEDELETE FileName
```

**Example:**

```
FILEDELETE NMNE17130016A_2017-12-11_18-17-06.LOG    – Delete the file
NMNE17130016A_2017-12-11_18-17-06.LOG
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Command header	FILEDELETE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).		H	0
2	FileName	Name of file to delete, or the wild card symbol (*)	String (Max 128)	variable <sup>1</sup>	H

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.48 FILEMEDIACONFIG

### *Specify the file media*

**Platform:** PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this command.

Use this command to specify which storage media is used for File operations.

To determine what storage device is currently being used for File operations, log this command. For example:

```
LOG FILEMEDIACONFIG
```



On OEM7 receiver cards, the file media is always USBSTICK, which is the only media available. On PwrPak7, CPT7 and CPT7700 products, the active file media is configured using the **FILEMEDIACONFIG** command.

**Message ID:** 2117

#### **Abbreviated ASCII Syntax:**

```
FILEMEDIACONFIG MassStorage
```

#### **ASCII Example:**

```
FILEMEDIACONFIG INTERNAL_FLASH    – Use internal flash as the media
FILEMEDIACONFIG USBSTICK          – Use a USB stick as the media
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FILEMEDIACONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	MassStorage	0	DEFAULT_NO_STORAGE	Disable logging to any media. Makes the internal storage accessible to a connected computer. (CPT7 and CPT7700 only)			
		1	USBSTICK	Use a USB stick as the mass storage device. (Not supported on CPT7 or CPT7700)	Enum	4	H
		2	INTERNAL_FLASH	Use Internal storage as the mass storage device.			

## 2.49 FILEROTATECONFIG

### *Set the maximum size and duration of a log file*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



PwrPak7M variants do not support this command.

Use this command to configure the maximum size and duration for a log file. This command also configures the action taken when the log file media is full.

A file rotation is when a new file is opened, the currently opened file is closed and logging on the FILE port is rerouted to this new file. There is no data loss during this process and individual logs within the file are not spread between log files.

**Message ID: 2133**

#### Abbreviated ASCII Syntax:

```
FILEROTATECONFIG [MaxFileTime] [MaxFileSize] [DiskFullAction]
```

#### Factory Default:

```
FILEROTATECONFIG 0 4096 STOP
```

#### Example:

```
FILEROTATECONFIG 2 4096 STOP
```

The file is left open for 2 hours or until the file size reaches 4096 MB. When the log file media is full, the file is closed.

```
FILEROTATECONFIG 4 4096 OVERWRITE
```

The file is left open for 4 hours or until the file size reaches 4096 MB. When the log media file is full, the oldest file on the log media file will be deleted.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FILEROTATECONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	MaxFileTime	0 to 24		<p>Maximum number of hours to leave a file open before triggering a file rotation.</p> <p>Set to 0 for no maximum time.</p> <p>Maximum value is 24.</p> <p>Default is 0.</p>	Ushort	2	H
3	MaxFileSize	1 to 4096		<p>Maximum number of mega bytes (MB) for the file size. A file rotation is triggered when the file is within 1 MB of this size.</p> <p>Maximum value is 4096 MB</p> <p>Default is 4096 MB (4 GB).</p>	Ushort	2	H+2
4	DiskFullAction	See Table 37: Disk Full Action below		Action to take when the log file media is full.	Enum	4	H+4

Table 37: Disk Full Action

Binary	ASCII	Description
0	STOP	<p>Stops logging when the file media has 1 MB of free space or less or the file media has reached 1500 files.</p> <p>Default is STOP.</p>
1	OVERWRITE	<p>Deletes the oldest log file when the file media has 10 MB of free space or less or the file media has exceeded 1500 files.</p> <p>To be selected for deletion a file must satisfy these requirements:</p> <ul style="list-style-type: none"> <li>The file must use the <b>FILECONFIG</b> command (see page 173) file name format.</li> <li>The &lt;PSN&gt; value must match the current receiver.</li> </ul> <p>File age is determined using the <b>FILECONFIG</b> command (see page 173) file name format.</p> <ul style="list-style-type: none"> <li>Temporary files (i.e. those with an &lt;INDEX&gt; value) are considered oldest. Such files will be sorted by their &lt;INDEX&gt; value with lower values considered older.</li> <li>Non-temporary files will be sorted by the date reported in the file format.</li> </ul>

## 2.50 FILETRANSFER

### *Copy files from internal memory*

Platform: PwrPak7



PwrPak7M variants do not support this command.

Use this command to copy files from internal memory to a USB stick. This command can also be used to cancel the file transfer in progress.



This command returns a response immediately to show that the copy/move operation started. However, the actual transfer of files will take some time. Use the **FILETRANSFERSTATUS** log (see page 566) to monitor the status of the file transfer.

To view the names of the files in memory, log the **FILELIST** log (see page 558).

**Message ID: 2109**

#### Abbreviated ASCII Syntax:

```
FILETRANSFER FileTransferOperation <FileName>
```

#### ASCII Examples:

```
FILETRANSFER COPY ALL      – Copies all files on internal memory
```

```
FILETRANSFER MOVE BMHR16460033T_2017-3-16_21-18-48.log
```

```
FILETRANSFER CANCEL      – Cancels file transfer operation
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	FILETRANSFER header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	FileTransferOperation	1	COPY	Copy the file	Enum	4	H
		2	MOVE	Copy the file and then delete file from internal memory			
		3	CANCEL	Cancels the file transfer currently in progress			
3	FileName			The name of the file to be moved or copied. To move or copy all of the files on internal memory, use ALL.	String	Variable	H+4



When a **FILETRANSFER CANCEL ALL** command is issued, the file currently being transferred, and any pending files, are not transferred to the destination media. Any files already transferred are unaffected.

## 2.51 FIX

### *Constrains to fixed height or position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to fix height or position to the input values. For various applications, fixing these values can assist in improving acquisition times and accuracy of position or corrections. For example, fixing the position is a requirement for differential base stations as it provides the reference position to base the differential corrections from.



1. It is strongly recommended that the FIX POSITION entered be accurate to within a few metres. This level of accuracy can be obtained from a receiver using single point positioning once 5 or 6 satellites are being tracked.
2. FIX POSITION should only be used for base station receivers. Applying FIX POSITION to a rover switches it from RTK mode to a fixed position mode. Applying FIX POSITION to the rover does not speed up ambiguity resolution.
3. Any setting other than FIX POSITION disables output of differential corrections unless the **MOVINGBASESTATION** command (see page 268) is set to ENABLE.
4. You can fix the position of the receiver using latitude, longitude and height in Mean Sea Level (MSL) or ellipsoidal parameters depending on the UNDULATION setting. The factory default for the **UNDULATION** command (see page 446) setting is EGM96, where the height entered in the **FIX** command is set as MSL height. If you change the UNDULATION setting to USER 0, the height entered in the **FIX** command is set as ellipsoidal height (refer to *Table 38: FIX Parameters* on the next page).

Error checking is performed on the entered fixed position by the integrity monitor. Depending on the result of this check, the position can be flagged with the following statuses.

- SOL\_COMPUTED: The entered position has been confirmed by measurement.
- PENDING: Insufficient measurements are available to confirm the entered position.
- INTEGRITY\_WARNING: First level of error when an incorrect position has been entered. The fixed position is off by approximately 25-50 metres.
- INVALID\_FIX: Second level of error when an inaccurate position has been entered. The fixed position is off by a gross amount.



An incorrectly entered fixed position will be flagged either INTEGRITY\_WARNING or INVALID\_FIX. This will stop output of differential corrections or RTK measurements and can affect the clock steering and satellite signal search. Checks on the entered fixed position can be disabled using the **RAIMMODE** command (see page 326).

**Message ID: 44**

**Abbreviated ASCII Syntax:**

```
FIX type [param1 [param2 [param3]]]
```

**Factory Default:**

```
FIX none
```

**ASCII Example:**

```
FIX none
```

```
FIX HEIGHT 4.567
```

```
FIX position 51.116 -114.038 1065.0
```



In order to maximize the absolute accuracy of RTK rover positions, the base station coordinates must be fixed to their known position using the **FIX POSITION [lat][lon][hgt]** command.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FIX header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	type	See Table 39: Fix Types on the next page		Fix type	Enum	4	H
3	param1	See Table 38: FIX Parameters below		Parameter 1	Double	8	H + 4
4	param2			Parameter 2	Double	8	H + 12
5	param3			Parameter 3	Double	8	H + 20

Table 38: FIX Parameters

ASCII Type Name	Parameter 1	Parameter 2	Parameter 3
AUTO	Not used	Not used	Not used
HEIGHT	Default MSL height <sup>1</sup> (-1000 to 20000000 m)	Not used	Not used
NONE	Not used	Not used	Not used
POSITION	Lat (-90 to 90 degrees) where a '-' sign denotes south and a '+' sign denotes north	Lon (-360 to 360 degrees) where a '-' sign denotes west and a '+' sign denotes east	Default MSL height <sup>1</sup> (-1000 to 20000000 m)

<sup>1</sup>See also Note #4 above.



For a discussion on height, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

Table 39: Fix Types

ASCII Name	Binary Value	Description
NONE	0	Unfix. Clears any previous FIX commands
AUTO	1	Configures the receiver to fix the height at the last calculated value if the number of satellites available is insufficient for a 3-D solution. This provides a 2-D solution. Height calculation resumes when the number of satellites available allows a 3-D solution
HEIGHT	2	<p>Configures the receiver in 2-D mode with its height constrained to a given value. This command is used mainly in marine applications where height in relation to mean sea level may be considered to be approximately constant. The height entered using this command is referenced to the mean sea level, see the <b>BESTPOS</b> log on page 520 (is in metres). The receiver is capable of receiving and applying differential corrections from a base station while fix height is in effect. The fix height command overrides any previous <b>FIX HEIGHT</b> or <b>FIX POSITION</b> command.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  This command only affects pseudorange corrections and solutions.         </div>
POSITION	3	<p>Configures the receiver with its position fixed. This command is used when it is necessary to generate differential corrections.</p> <p>For both pseudorange and differential corrections, this command must be properly initialized before the receiver can operate as a GNSS base station. Once initialized, the receiver computes differential corrections for each satellite being tracked. The computed differential corrections can then be output to rover stations using the RTCMV3 differential corrections data log format. See Transmitting and Receiving Corrections in the <a href="#">OEM7 Installation and Operation User Manual</a>, <a href="#">PwrPak7 Installation and Operation User Manual</a>, <a href="#">SMART7 Installation and Operation User Manual</a> or <a href="#">CPT7 Installation and Operation User Manual</a> for information about using the receiver for differential applications.</p> <p>The values entered into the fix position command should reflect the precise position of the base station antenna phase center. Any errors in the fix position coordinates directly bias the corrections calculated by the base receiver.</p> <p>The receiver performs all internal computations based on WGS84 and the <b>DATUM</b> command (see page 133) is defaulted as such. The datum in which you choose to operate (by changing the <b>DATUM</b> command (see page 133)) is internally converted to and from WGS84. Therefore, all differential corrections are based on WGS84, regardless of your operating datum.</p> <p>The <b>FIX POSITION</b> command overrides any previous <b>FIX HEIGHT</b> or <b>FIX POSITION</b> command settings.</p>



## 2.52 FORCEGALE6CODE

### *Force receiver to track Galileo E6C or E6B signal*

**Platform:** OEM719, OEM729, OEM7700, PwrPak7, CPT7700

Use this command to force Galileo E6 channels to track E6B or E6C.

**Message ID:** 2222

**Abbreviated ASCII Syntax:**

```
FORCEGALE6CODE E6codetype
```

**Factory Default:**

```
FORCEGALE6CODE E6B
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FORCEGALE6CODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	E6codetype	E6B	0	Galileo E6 code type (default = E6B)	Enum	4	H
		E6C	1				

## 2.53 FORCEGLOL2CODE

### *Forces receiver to track GLONASS satellite L2 P or L2 C/A code*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to force the receiver to track GLONASS satellite L2 P-code or L2 C/A code. This command has no effect if the channel configuration contains both GLONASS L2 P and L2 C/A channels.

**Message ID:** 1217

#### Abbreviated ASCII Syntax:

```
FORCEGLOL2CODE L2type
```

#### Factory Default:

```
FORCEGLOL2CODE default
```

#### ASCII Example:

```
FORCEGLOL2CODE p
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FORCEGLOL2CODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	L2type	P	1	L2 P-code or L2 Precise code	Enum	4	H
		C	2	L2 C/A code or L2 Coarse/Acquisition code			
		DEFAULT	3	Set to channel default			

The following table lists which L2 signal is tracked based on the channel configuration and the setting used for the L2type parameter.

**Table 40: Signals Tracked – Channel Configuration and L2type Option**

Channel Configuration for L2 Signal	L2type Setting		
	P	C	DEFAULT
L2	P	C	P

<b>L2type Setting</b>			
<b>Channel Configuration for L2 Signal</b>	<b>P</b>	<b>C</b>	<b>DEFAULT</b>
L2C	P	C	C
L2PL2C	Both	Both	Both

## 2.54 FORCEGPSL2CODE

### *Forces receiver to track GPS satellite L2 P or L2C code*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to force the receiver to track GPS L2 P-code or L2C code. AUTO tells the receiver to use L2C code type if available and L2 P-code if L2C code is not available. This command has no effect if the channel configuration contains both GPS L2 P and L2 C channels.

**Message ID:** 796

#### Abbreviated ASCII Syntax:

```
FORCEGPSL2CODE L2type
```

#### Factory Default:

```
FORCEGPSL2CODE default
```

#### ASCII Example:

```
FORCEGPSL2CODE p
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FORCEGPSL2CODE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	L2type	See Table 41: GPS L2 Code Type below		GPS L2 code type	Enum	4	H

**Table 41: GPS L2 Code Type**

Binary	ASCII	Description
0	AUTO	Receiver uses the L2C if available and L2 P otherwise. An exception is when the receiver is doing RTK positioning. In that case, AUTO changes the L2 code type being tracked to match the L2 code type found in the base station corrections, which ensures the greatest number of satellites are used in the solution.
1	P	L2 P-code or L2 Precise code
2	C	L2C code or L2 Civilian code
3	DEFAULT	Set to channel default

The following table lists which L2 signal is tracked based on the channel configuration and the setting used for the L2type parameter.

**Table 42: Signals Tracked – Channel Configuration and L2type Option**

Channel Configuration for L2 Signal	L2type Setting			
	Auto	P	C	DEFAULT
L2	C if available, P(Y) otherwise	P(Y)	C	P(Y)
L2C	C if available, P(Y) otherwise	P(Y)	C	C
L2P	C if available, P(Y) otherwise	P(Y)	C	P(Y)
L2AUTO	C if available, P(Y) otherwise	P(Y)	C	C if available, P(Y) otherwise
L2PL2C	Both	Both	Both	Both

## 2.55 FREQUENCYOUT

### *Sets output pulse train available on VARF*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This command is used to set the output pulse train available on the Variable Frequency (VARF) or EVENT\_OUT1 pin. The output waveform is coherent with the 1PPS output, see the usage note and *Figure 4: Pulse Width and 1PPS Coherency* on the next page.



The **FREQUENCYOUT** command is obsolete. Use the **EVENTOUTCONTROL** command (see page 165) to set the output pulse train on the VARF and EVENT\_OUT pins.



If the **CLOCKADJUST** command (see page 115) is ENABLED and the receiver is configured to use an external reference frequency (set in the **EXTERNALCLOCK** command (see page 168) for an external clock - TCXO, OCXO, RUBIDIUM, CESIUM, or USER), then the clock steering process takes over the VARF output pins and may conflict with a previously entered **FREQUENCYOUT** command.



*Figure 4: Pulse Width and 1PPS Coherency* on the next page shows how the chosen pulse width is frequency locked but not necessarily phase locked when using ENABLE option. To synchronize the phase, use ENABLESYNC option.



The EVENTOUT outputs cannot synchronize with GPS time until the receiver reaches FINESTEERING time status. As the receiver transitions to GPS time, there may be additional, unexpected pulses on the EVENTOUT signals.

**Message ID: 232**

#### **Abbreviated ASCII Syntax:**

```
FREQUENCYOUT [switch] [pulsewidth] [period]
```

#### **Factory Default:**

```
FREQUENCYOUT disable
```

#### **ASCII Example:**

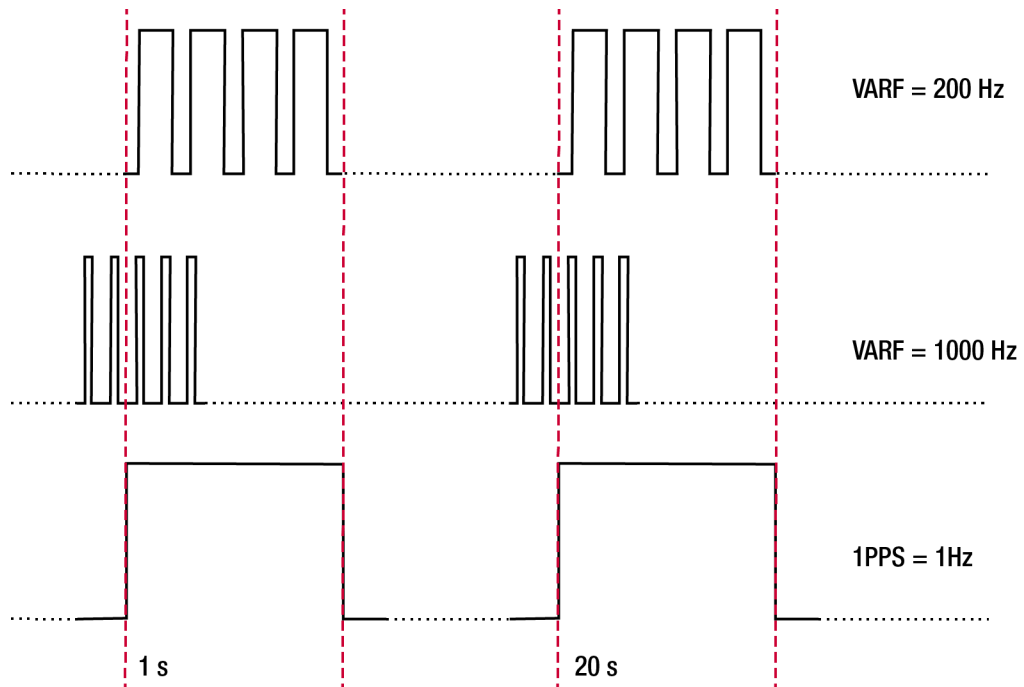
```
FREQUENCYOUT ENABLE 50000 100000
```

This example generates a 50% duty cycle 1 kHz square wave.



Signal integrity will begin to degrade when generating a clock frequency greater than 10 MHz. It is not recommended to generate a clock frequency greater than 20 MHz.

**Figure 4: Pulse Width and 1PPS Coherency**



When using ENABLE option, the VARF and 1PPS are not necessarily in phase as described in *Figure 4: Pulse Width and 1PPS Coherency* above. To align the phase of the VARF with the 1PPS, use the ENABLESYNC option and the VARF phase will be synchronized to the leading edge of the 1PPS pulse. Note that if the VARF and 1PPS frequencies are not even multiples of each other, this may cause the VARF to have a shorter cycle pulse prior to each 1PPS pulse. 1PPS is not affected.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FREQUENCYOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	switch	DISABLE	0	Disable causes the output to be fixed low. (if NONE specified, defaults to DISABLE)	Enum	4	H
		ENABLE	1	Enables customized frequency output.			
		ENABLESYNC	2	Enables customized frequency output synchronized to PPS.			
3	pulsewidth	(0 to 1073741823)		Number of 10 ns steps for which the output is high.  Duty cycle = pulsewidth / period. If pulsewidth is greater than or equal to the period, the output is a high DC signal. If pulsewidth is 1/2 the period, then the output is a square wave. (default = 0)	Ulong	4	H+4
4	period	(0 to 1073741823)		Signal period in 10 ns steps.  Frequency Output = 100,000,000 / Period  (default = 0)	Ulong	4	H+8



## 2.56 FRESET

### *Clears selected data from NVM and reset*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to clear data which is stored in non-volatile memory. Such data includes the almanac, ephemeris, and any user specific configurations. The commands, ephemeris, almanac, and L-Band related data, excluding the subscription information, can be cleared by using the STANDARD target. The receiver is forced to reset.



**FRESET STANDARD** (which is also the default) causes most commands, ephemeris, GNSS and almanac data previously saved to NVM to be erased.



The **FRESET STANDARD** command will erase all user settings. You should know your configuration (by requesting the **RXCONFIG** log on page 876) and be able to reconfigure the receiver before you send the **FRESET** command.

**Message ID:** 20

**Abbreviated ASCII Syntax:**

```
FRESET [target]
```

**Input Example:**

```
FRESET COMMAND
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	FRESET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	target	See <i>Table 43: FRESET Target</i> on the next page		What data is to be reset by the receiver (default = STANDARD)	Enum	4	H



If you are receiving no data or random data from your receiver, try the following before contacting NovAtel:

- Verify that the receiver is tracking satellites by logging the **TRACKSTAT** log (see page 969) and checking that the receiver is tracking at least four satellites.
- Check the integrity and connectivity of power, antenna and data cables
- Verify the baud rate settings of the receiver and terminal device (your PC, data logger or laptop)
- Switch COM ports
- Issue the **FRESET** command.

**Table 43: FRESET Target**

Binary	ASCII	Description
0	STANDARD	Resets commands (except <b>CLOCKCALIBRATION</b> and <b>MODEL</b> ). Resets the stored ephemeris and almanac from all satellite systems. Resets all L-Band related data except for the subscription information. Does not reset the Ethernet settings or stored Profile configurations. (default)
1	COMMAND	Resets the stored commands saved by the <b>SAVECONFIG</b> command (see page 366)
2	GPSALMANAC	Resets the stored GPS almanac
3	GPSEPHEM	Resets the stored GPS ephemeris
4	GLOEPHEM	Resets the stored GLONASS ephemeris
5	MODEL	Resets the currently selected model
10	USERDATA	Resets the user data saved using the <b>NVMUSERDATA</b> command (see page 283)
11	CLKCALIBRATION	Resets the parameters entered using the <b>CLOCKCALIBRATE</b> command (see page 117)
20	SBASALMANAC	Resets the stored SBAS almanac
21	LAST_POSITION	Resets the position using the last stored position
31	GLOALMANAC	Resets the stored GLONASS almanac
39	GALFNAV_EPH	Resets the stored GALFNAV ephemeris
40	GALINAV_EPH	Resets the stored GALINAV ephemeris
45	GALFNAV_ALM	Resets the stored GALFNAV almanac
46	GALINAV_ALM	Resets the stored GALINAV almanac

Binary	ASCII	Description
52	PROFILEINFO	Resets the stored profile configurations (see the <b>PROFILE</b> command on page 315)
54	QZSSALMANAC	Resets the QZSS almanac
55	QZSSEPHemeris	Resets the QZSS ephemeris
57	BDSALMANAC	Resets the BeiDou almanac
58	BDSEPHemeris	Resets the BeiDou ephemeris
60	USER_ACCOUNTS	Resets the admin password to the default (the receiver PSN) (see the <b>SETADMINPASSWORD</b> command on page 388)
64	ETHERNET	Resets the stored Ethernet settings saved by the <b>SAVEETHERNETDATA</b> command (see page 367)
85	SRTK_ SUBSCRIPTIONS	Resets the Secure RTK Subscription data stored on the rover receiver
87	NAVICEPHEMERIS	Resets the NavIC ephemeris
88	NAVICALMANAC	Resets the NavIC almanac
93	USER_DATUM	Resets the user datums (see the <b>GEODETI CDATUM</b> command on page 203)
94	USER_DATUM_ TRANSFORMATION	Resets the user datums transformations (see the <b>DATUMTRANSFORMATION</b> command on page 135)

## 2.57 GALECUTOFF

### *Sets elevation cut-off angle for Galileo satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the elevation cut-off angle for tracked Galileo satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they were manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the GALECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and thus look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **GALECUTOFF** because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command (see page 156) to set the cut-off angle for any system.



For dual antenna receivers, this command applies to both the primary and secondary antennas.

**Message ID:** 1114

**Abbreviated ASCII Syntax:**

```
GALECUTOFF angle
```

**Factory Default:**

```
GALECUTOFF 5.0
```

**ASCII Example:**

```
GALECUTOFF 10.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GALECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.58 GENERATEALIGNCORRECTIONS

### Configure ALIGN Master

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to configure the ALIGN Master and starts sending out ALIGN corrections through the specified port. This command is like sending the following commands to the Master, assuming the use of a serial port and default ALIGN corrections:

```
unlogall [port]
fix none
movingbasestation enable
interfacemode [port] novatel rtca
serialconfig [port] [baud] N 8 1 N ON
log [port] rtcaobs3 ontime [rate = 1/ obsregrate]
log [port] rtcarefext ontime [rate = 1/ refextregrate]
```

**Message ID: 1349**

#### Abbreviated ASCII Syntax:

```
GENERATEALIGNCORRECTIONS port [baud] [obsregrate] [refextregrate]
[interfacemode]
```

#### ASCII Example:

```
GENERATEALIGNCORRECTIONS COM2 230400 10 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GENERATEALIGN CORRECTIONS header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	port	See Table 34: Communications Port Identifiers on page 152		Port identifier (default = THISPORT)	Enum	4	H
3	baud	9600, 19200, 38400, 57600, 115200, 230400 or 460800		Communication baud rate (bps) (default = 9600)	Ulong	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	obsreqrate	1, 2, 4, 5, 10, 20, 50 or 100		RTCAOBS3 data rate in Hz (default = 1)	Ulong	4	H+8
5	refextreqrate	0, 1, 2, 4, 5, 10, 20, 50 or 100		RTCAREFEXT data rate in Hz (default = 1)	Ulong	4	H+12
6	interfacemode	RTCA	3	Correction interface mode (default = RTCA)	Enum	4	H+16
		NOVATELX	35				

## 2.59 GENERATEDIFFCORRECTIONS

*Sends a preconfigured set of differential corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to configure the receiver to send a preconfigured set of differential pseudorange corrections.

**Message ID: 1296**

**Abbreviated ASCII Syntax:**

```
GENERATEDIFFCORRECTIONS mode port
```

**ASCII Example:**

```
GENERATEDIFFCORRECTIONS rtc com2
```

Preconfigured set of differential corrections sent when RTCM:

```
RTCM1 ontime 1
RTCM31 ontime 1
RTCM3 ontime 10
```

Preconfigured set of differential corrections sent when RTCA:

```
RTCA1 ontime 1
RTCAREF ontime 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GENERATEDIFF CORRECTIONS header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	mode	RTCM	2	Serial port interface mode identifier. See <i>Table 46: Serial Port Interface Modes</i> on page 219	Enum	4	H
		RTCA	3				
3	port	See <i>Table 70: COM Port Identifiers</i> on page 385		Port to configure	Enum	4	H+4



## 2.60 GENERATERTKCORRECTIONS

*Sends a preconfigured set of RTK corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to configure the receiver to send a preconfigured set of RTK (carrier phase) corrections.

**Message ID: 1260**

**Abbreviated ASCII Syntax:**

```
GENERATERTKCORRECTIONS mode port
```

**ASCII Example:**

```
GENERATERTKCORRECTIONS rtcmv3 com2
```

Preconfigured set of differential corrections sent when RTCM:

```
RTCM1819 ontime 1
RTCM3 ontime 10
RTCM22 ontime 10
RTCM23 ontime 60
RTCM24 ontime 60
```

Preconfigured set of differential corrections sent when RTCMV3:

```
RTCM1004 ontime 1
RTCM1012 ontime 1
RTCM1006 ontime 10
RTCM1008 ontime 10
RTCM1033 ontime 10
```

Preconfigured set of differential corrections sent when RTCA:

```
RTCAOBS2 ontime 1
RTCAREF ontime 10
```

Preconfigured set of differential corrections sent when CMR:

```
CMROBS ontime 1
CMRGLOBS ontime 1
CMRREF ontime 10
```

Preconfigured set of differential corrections sent when NOVATELX:

```
NOVATELXOBS ontime 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GENERATERTK CORRECTIONS header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	mode	RTCM	2	Serial port interface mode identifier. For more information, see <i>Table 46: Serial Port Interface Modes</i> on page 219	Enum	4	H
		RTCA	3				
		CMR	4				
		MSM	5				
		MSM12	6				
		MSM3	7				
		MSM4	8				
		MSM5	9				
		MSM6	10				
		MSM7	11				
		RTCMV3	14				
		NOVATELX	35				
3	port	See <i>Table 70: COM Port Identifiers</i> on page 385		Port to configure	Enum	4	H+4



For information about the RTCM, RTCA and CMR messages, refer to the official standards document for those messages.

## 2.61 GEODETICDATUM

### *Set user-defined datum parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to add or remove user-defined datums. These are used by the **OUTPUTDATUM** command (see page 284) to select an entry from the **DATUMTRANSFORMATIONS** log (see page 553). They also define the parameters to convert between geodetic and Cartesian coordinates.



The **OUTPUTDATUM** command (see page 284) must be sent for any user-defined datums to be output.

The receiver firmware includes a default set of datums. These default datums cannot be deleted, but can be overwritten with user-defined parameters. A default datum that has been overwritten can be reverted to the default parameters by deleting the datum. User-defined datums will persist a factory reset; all user-defined datums can be removed by sending **FRESET USER\_DATUM** (see the **FRESET** command on page 193). Both the user-defined and default datums are reported by the **GEODETICDATUMS** log (see page 588).

Datum entries are indexed by name, which must be unique. The name “USER” is reserved for the **USEREXPDATUM** entry. “ECEF” is reserved as a generic Earth-Centered Earth-Fixed datum. “UNKNOWN” is also reserved.

The EPSG Geodetic Parameter Dataset ([www.epsg-registry.org/](http://www.epsg-registry.org/)) is a commonly-referenced database of coordinate reference systems and coordinate transformations. The EPSG code in this log is intended to correspond to the “GeodeticDatum” data type within that database. If you are using a custom datum, the ESPG code should be 0.

**Message ID: 2295**

#### **Abbreviated ASCII Syntax:**

```
GEODETICDATUM switch name epsg_code anchor semimajor_axis flattening
```

#### **Factory Default:**

```
GEODETICDATUM none
```

#### **ASCII Example:**

Save a new geodetic datum to NVM:

```
GEODETICDATUM save NAD83 (MA11) 1118 EARTH_FIXED 6378137.0 298.257222101
```

Delete an existing geodetic datum from NVM:

```
GEODETICDATUM delete NAD83 (MA11)
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	GEODETICDATUM header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	switch	SAVE	1	Save the datum.	Enum	4	H
		DELETE	2	Delete the datum.			
3	name			Name of the datum.	Char [32]	variable	H+4
4	epsg_code			The EPSG code.	Ulong	4	variable
5	anchor	See Table 44: Datum Anchors below		The datum type.	Enum	4	variable
6	semimajor_axis	6300000.000 – 6400000.000		Semi-major axis of the datum's ellipsoid (metres)	Double	8	variable
7	flattening	290.000 – 305.000		Inverse flattening of the datum's ellipsoid (unitless)	Double	8	variable

Table 44: Datum Anchors

Binary	ASCII	Description
0	UNKNOWN	The anchor is unknown.
1	EARTH_FIXED	The datum is not anchored to a specific tectonic plate (e.g., ITRF2005, WGS84).
2	PLATE_FIXED	The datum is anchored to a tectonic plate (e.g., NAD83, ETRF89).

## 2.62 GGAQUALITY

### *Customizes the GPGGA GPS quality indicator*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to customize the NMEA GPGGA GPS quality indicator. See also the **GPGGA** log on page 610.



This command also changes the GPS quality indicator in the NMEA2000 CAN logs.

**Message ID: 691**

#### Abbreviated ASCII Syntax:

```
GGAQUALITY #entries pos_type quality
```

#### Input Example 1:

```
GGAQUALITY 1 waas 2
```

Makes the WAAS solution type show 2 as the quality indicator.

#### Input Example 2:

```
GGAQUALITY 2 waas 2 NARROW_FLOAT 3
```

Makes the WAAS solution type show 2 and the NARROW\_FLOAT solution type show 3, as their quality indicators.

#### Input Example 3:

```
GGAQUALITY 0
```

Sets all the quality indicators back to the default.



Some solution types, see *Table 87: Position or Velocity Type* on page 524, share a quality indicator. For example, converged PPP and NARROW\_FLOAT all share an indicator of 5. This command can be used to customize an application to have unique indicators for each solution type. Sets all the quality indicators back to the default. Refer to *Table 113: GPS Quality Indicators* on page 612.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GGAQUALITY header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	#entries	0-20		The number of position types that are being remapped (20 max)	Ulong	4	H
3	pos_type	See <i>Table 87: Position or Velocity Type</i> on page 524		The position type that is being remapped	Enum	4	H+4
4	quality	See <i>Table 113: GPS Quality Indicators</i> on page 612		The remapped quality indicator value that will appear in the GPGGA log for this position type	Ulong	4	H+8
...	Next solution type and quality indicator set, if applicable				Variable		

## 2.63 GLIDEINITIALIZATIONPERIOD

### *Configures the GLIDE initialization period*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the initialization period for Relative PDP (GLIDE) when pseudorange measurements are used more heavily. During the initialization period, the PDP output position is not as smooth as during full GLIDE operation, but it helps to get better absolute accuracy at the start. The longer this period is, the better the absolute accuracy that can be attained. The maximum period that can be set through **GLIDEINITIALIZATIONPERIOD** is 1200 seconds.

**Message ID: 1760**

#### **Abbreviated ASCII Syntax:**

```
GLIDEINITIALIZATIONPERIOD initialization
```

#### **Factory Default:**

```
GLIDEINITIALIZATIONPERIOD 300
```

#### **ASCII Example:**

```
GLIDEINITIALIZATIONPERIOD 100
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	GLIDEINITIALIZATION PERIOD header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	initialization	0 -1200 s		Initialization period for GLIDE in seconds	Double	8	H

## 2.64 GLOECUTOFF

### *Sets GLONASS satellite elevation cut-off*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the elevation cut-off angle for tracked GLONASS satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they were manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the GLOECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction
- However, for GLONASS it is not recommended to use negative values because a receiver should not track both antipodal satellites that are broadcasting on the same frequency.



Care must be taken when using **GLOECUTOFF** because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command (see page 156) to set the cut-off angle for any system.



For dual antenna receivers, this command applies to both the primary and secondary antennas.

**Message ID: 735**

**Abbreviated ASCII Syntax:**

```
GLOECUTOFF angle
```

**Factory Default:**

```
GLOECUTOFF 5.0
```

**ASCII Example:**

```
GLOECUTOFF 0
```



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	GLOECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.65 HDTOUTTHRESHOLD

### *Controls GPHDT log output*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to control the output of the NMEA **GPHDT** log (see page 629). It sets a heading standard deviation threshold. Only heading information with a standard deviation less than this threshold can be output into a GPHDT message.

**Message ID: 1062**

#### **Abbreviated ASCII Syntax:**

```
HDTOUTTHRESHOLD thresh
```

#### **Factory Default:**

```
HDTOUTTHRESHOLD 2.0
```

#### **ASCII Example:**

```
HDTOUTTHRESHOLD 12.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	HDTOUTTHRESHOLD header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	thresh	0.0 - 180.0		Heading standard deviation threshold (degrees)	Float	4	H

## 2.66 HEADINGOFFSET

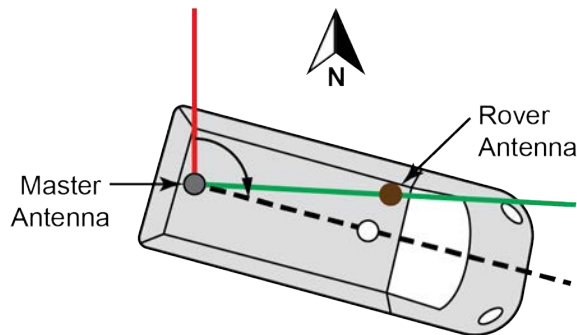
### *Adds heading and pitch offset values*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to add an offset in the heading and pitch values of the **HEADING2** log (see page 646) and **GPHDT** log (see page 629).

The unmodified heading value represents the angle from True North of the base to rover vector in a clockwise direction. In some installations, it may not be possible to place the rover antenna in the desired location, for instance to match the forward-facing direction of the vehicle.

**Figure 5: HEADINGOFFSET Example**



In the example above, the rover antenna (in brown) is offset from the forward direction of travel and the provided heading will look as if the land vehicle is "slipping". A Heading offset can help to account for that difference between heading output and actual course over ground. It is up to the user to accurately determine the offset to be applied to suit their scenario.

**Message ID: 1082**

#### **Abbreviated ASCII Syntax:**

```
HEADINGOFFSET headingoffsetindeg [pitchoffsetindeg]
```

#### **Factory Default:**

```
HEADINGOFFSET 0 0
```

#### **ASCII Example:**

```
HEADINGOFFSET 2 -1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	HEADINGOFFSET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	headingoffsetindeg	-180.0 - 180.0		Offset added to heading output (degrees). Default=0	Float	4	H
3	pitchoffsetindeg	-90.0 - 90.0		Offset added to pitch output (degrees). Default=0	Float	4	H+4

## 2.67 ICOMCONFIG

### *Configures IP virtual COM port*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command is used for Ethernet set up and to configure the transport/application layer of the configuration.



Access to the ICOM ports can be restricted by turning on ICOM security using the **IPSERVICE** command (see page 225).

**Message ID: 1248**

#### **Abbreviated ASCII Syntax:**

```
ICOMCONFIG [port] protocol [endpoint [bindinterface]]
```

#### **Factory Default:**

```
ICOMCONFIG ICOM1 TCP :3001
ICOMCONFIG ICOM2 TCP :3002
ICOMCONFIG ICOM3 TCP :3003
ICOMCONFIG ICOM4 TCP :3004
ICOMCONFIG ICOM5 TCP :3005
ICOMCONFIG ICOM6 TCP :3006
ICOMCONFIG ICOM7 TCP :3007
```

#### **ASCII Example:**

```
ICOMCONFIG ICOM1 TCP :2000 All
```




Due to security concerns, configuring and enabling ICOM ports should only be done to receivers on a closed system, that is, board-to-board. NovAtel is not liable for any security breaches that may occur if not used on a closed system.

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ICOMCONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	port	THISPORT	6	Name of the port (default = THISPORT).	Enum	4	H
		ICOM1	23				
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46				
		ICOM6	47				
		ICOM7	48				
3	protocol	See Table 45: ICOMCONFIG Protocol on the next page		Protocol used by the ICOM port.	Enum	4	H+4
4	endpoint	Host:Port For example: 10.0.3.1:8000 mybase.com:3000		Endpoint to wait on, or to connect to where host is a host name or IP address and port is the TCP/UDP port number. If host is blank, act as a server.	String [80]	variable <sup>1</sup>	H+8

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
5	bindInterface	ALL	1	Not supported. Set to <i>ALL</i> for future compatibility. (default)	Enum	4	H+88

Table 45: ICOMCONFIG Protocol

Binary	ASCII	Description
1	DISABLED	Will disable the service
2	TCP	Use TCP The TCP protocol option provides a one to one connection between an OEM receiver and a client.
3	UDP	Use UDP
4	TCP_SERVER	<p>Allow multiple TCP connections</p> <p>The TCP_SERVER protocol option provides one to many connections so several clients can connect to the same ICOM port. All clients receive the same OEM receiver output, including command responses. When an ICOM port is configured as a TCP_SERVER, it is suggested to not issue commands on the ICOM port or have responses disabled (see <b>INTERFACEMODE</b> command on the next page).</p> <p>There is a maximum limit of 20 client connections over all ICOM ports configured as TCP_SERVER.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  Before setting the ICOM port protocol to TCP_SERVER, use the <b>ETHCONFIG</b> command (see page 159) to set the <b>power_mode</b> of the Ethernet port to <b>powerdown</b>. After the ICOM configuration is complete, set the power_mode back to Normal. </div>

## 2.68 INTERFACEMODE

### *Sets receive or transmit modes for ports*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to specify what type of data a particular port on the receiver can transmit and receive. The receive type tells the receiver what type of data to accept on the specified port. The transmit type tells the receiver what kind of data it can generate. For example, to accept RTCMV3 differential corrections, set the receive type on the port to RTCMV3.

It is also possible to disable or enable the generation or transmission of command responses for a particular port. Disabling of responses is important for applications where data is required in a specific form and the introduction of extra bytes may cause problems, i.e., RTCA, RTCM, RTCMV3 or CMR. Disabling a port prompt is also useful when the port is connected to a modem or other device that responds with data the RECEIVER does not recognize.



For applications running in specific interface modes, see *Table 46: Serial Port Interface Modes* on page 219, set the appropriate interface modes before sending or receiving corrections. It is important that the port interface mode matches the data being received on that port. Mismatches between the interface mode and received data can result in CPU overloads.

When INTERFACEMODE port NONE NONE OFF is set, the specified port is disabled from interpreting any input or output data. Therefore, no commands or differential corrections are decoded by the specified port. When GENERIC is set for a port, it is also disabled but data can be passed through the disabled port and be output from an alternative port using the pass-through logs PASSCOM, PASSAUX and PASSUSB. See *PASSCOM, PASSAUX, PASSUSB, PASSETH1, PASSICOM, PASSNCOM* on page 737 for details on these logs along with Pass-Through Logging in the [OEM7 Installation and Operation User Manual](#). [PwrPak7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#) for information about pass-through logging. See also the **SERIALCONFIG** command on page 383. If you intend to use the **SERIALCONFIG** command (see page 383), ensure you do so before the **INTERFACEMODE** command on each port. The **SERIALCONFIG** command (see page 383) can remove the **INTERFACEMODE** command setting if the baud rate is changed after the interface mode is set. You should also turn break detection off using the **SERIALCONFIG** command (see page 383) to stop the port from resetting because it is interpreting incoming bits as a break command. If such a reset happens, the Interface mode will be set back to the default NOVATEL mode for both input and output.

### 2.68.1 SPAN Systems

The INTERFACEMODE of the receiver is also configured for the serial port dedicated to the IMU. This mode changes automatically upon sending a **CONNECTIMU** command (see page 1001) and the change is reflected when logging this command. This is normal operation.



When the **CONNECTIMU** command (see page 1001) is used to configure the IMU connected to the receiver, the correct interface mode for the IMU port is automatically set. The IMU port should not be altered using the **INTERFACEMODE** command in normal operation. Doing so may result in the loss of IMU communication.

**Message ID: 3**

**Abbreviated ASCII Syntax:**



```
INTERFACEMODE [port] rxtype txtype [responses]
```

**Factory Default:**

```
INTERFACEMODE COM1 NOVATEL NOVATEL ON
INTERFACEMODE COM2 NOVATEL NOVATEL ON
INTERFACEMODE COM3 NOVATEL NOVATEL ON
INTERFACEMODE AUX NOVATEL NOVATEL ON
INTERFACEMODE USB1 NOVATEL NOVATEL ON
INTERFACEMODE USB2 NOVATEL NOVATEL ON
INTERFACEMODE USB3 NOVATEL NOVATEL ON
INTERFACEMODE ICOM1 NOVATEL NOVATEL ON
INTERFACEMODE ICOM2 NOVATEL NOVATEL ON
INTERFACEMODE ICOM3 NOVATEL NOVATEL ON
INTERFACEMODE ICOM4 NOVATEL NOVATEL ON
INTERFACEMODE ICOM5 NOVATEL NOVATEL ON
INTERFACEMODE ICOM6 NOVATEL NOVATEL ON
INTERFACEMODE ICOM7 NOVATEL NOVATEL ON
INTERFACEMODE NCOM1 RTCMV3 NONE OFF
INTERFACEMODE NCOM2 RTCMV3 NONE OFF
INTERFACEMODE NCOM3 RTCMV3 NONE OFF
INTERFACEMODE CCOM1 NOVATELBINARY NOVATELBINARY ON
INTERFACEMODE CCOM2 NOVATELBINARY NOVATELBINARY ON
INTERFACEMODE CCOM3 AUTO NOVATEL OFF
INTERFACEMODE CCOM4 AUTO NOVATEL OFF
INTERFACEMODE CCOM5 AUTO NOVATEL OFF
INTERFACEMODE CCOM6 AUTO NOVATEL OFF
INTERFACEMODE SCOM1 NOVATEL NOVATEL ON
INTERFACEMODE SCOM2 NOVATEL NOVATEL ON
INTERFACEMODE SCOM3 NOVATEL NOVATEL ON
INTERFACEMODE SCOM4 NOVATEL NOVATEL ON
```

**ASCII Example 1:**

```
INTERFACEMODE COM1 RTCMV3 NOVATEL ON
```

**ASCII Example 2:**

```
INTERFACEMODE COM2 MRTCA NONE
```



Are NovAtel receivers compatible with others on the market?

All GNSS receivers output two solutions: position and time. The manner in which they output them makes each receiver unique. Most geodetic and survey grade receivers output the position in electronic form (typically RS-232), which makes them compatible with most computers and data loggers. All NovAtel receivers have this ability. However, each manufacturer has a unique way of formatting the messages. A NovAtel receiver is not directly compatible with a Trimble or Ashtech receiver (which are also incompatible with each other) unless everyone uses a standard data format.

However, there are several standard data formats available. For position and navigation output there is the NMEA format. Real-time differential corrections use RTCM or RTCA format. For receiver code and phase data RINEX format is often used. NovAtel and all other major manufacturers support these formats and can work together using them. The NovAtel format measurement logs can be converted to RINEX using the utilities provided in NovAtel Convert.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	INTERFACEMODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See Table 34: <i>Communications Port Identifiers</i> on page 152		Serial port identifier (default = THISPORT) <b>Note:</b> On the SMART2, the INTERFACEMODE command cannot be used with CCOM ports.	Enum	4	H
3	rxtype	See Table 46: <i>Serial Port Interface Modes</i> on the next page		Receive interface mode	Enum	4	H+4
4	txtype	See Table 46: <i>Serial Port Interface Modes</i> on the next page		Transmit interface mode	Enum	4	H+8
5	responses	OFF	0	Turn response generation off	Enum	4	H+12
		ON	1	Turn response generation on (default)			

Table 46: Serial Port Interface Modes

Binary Value	ASCII Value	Description
0	NONE	The port accepts/generates nothing. The port is disabled.
1	NOVATEL	The port accepts/generates NovAtel commands and logs. The port also accepts/generates NOVATELX corrections.
2	RTCM	The port accepts/generates RTCM corrections.
3	RTCA	The port accepts/generates RTCA corrections.
4	CMR	The port accepts/generates CMR corrections.
5	Reserved	
6	Reserved	
7	IMU	This port supports communication with a NovAtel supported IMU.
8	RTCMNOCR	When RTCMNOCR is used as the <i>txtype</i> , the port generates RTCM corrections without the CR/LF appended. When RTCMNOCR is used as the <i>rxtype</i> , the port accepts RTCM corrections with or without the CR/LF appended.
9	Reserved	
10	TCOM1	<p>INTERFACEMODE tunnel modes. To configure a full duplex tunnel, configure the baud rate on each port. Once a tunnel is established, the baud rate does not change. Special characters, such as a BREAK condition, do not route across the tunnel transparently and the serial port is altered, see the <b>SERIALCONFIG</b> command on page 383. Only serial ports may be in a tunnel configuration: COM1, COM2, COM3 or AUX may be used.</p> <p>For example, configure a tunnel at 115200 bps between COM1 and AUX:</p> <pre>SERIALCONFIG AUX 115200 SERIALCONFIG COM1 115200 INTERFACEMODE AUX TCOM1 NONE OFF INTERFACEMODE COM1 TAUX NONE OFF</pre> <p>The tunnel is fully configured to receive/transmit at a baud rate of 115200 bps.</p>
11	TCOM2	
12	TCOM3	
13	TAUX <sup>1</sup>	
14	RTCMV3	

<sup>1</sup>Only available on specific models.

Binary Value	ASCII Value	Description
15	NOVATELBINARY	The port only accepts/generates binary messages. If an ASCII command is entered when the mode is set to binary only, the command is ignored. Only properly formatted binary messages are responded to and the response is a binary message.
16-17	Reserved	
18	GENERIC	The port accepts/generates nothing. The <b>SEND</b> command (see page 380) or <b>SENDHEX</b> command (see page 382) from another port generate data on this port. Any incoming data on this port can be seen with PASSCOM logs on another port, see <b>PASSCOM</b> , <b>PASSAUX</b> , <b>PASSUSB</b> , <b>PASSETH1</b> , <b>PASSICOM</b> , <b>PASSNCOM</b> log on page 737.
19	IMARIMU	This port supports communication with an iMAR IMU.
20	MRTCA	The port accepts/generates Modified Radio Technical Commission for Aeronautics (MRTCA) corrections.
21-22	Reserved	
23	KVHIMU	This port supports communication with a KVH CG5100 IMU.
24-26	Reserved	
27	AUTO	For auto-detecting different RTK correction formats and incoming baud rate (over serial ports). The change of baud rate will not appear when <b>SERIALCONFIG</b> is logged as this shows the saved baud rate for that port.
28-34	Reserved	
35	NOVATELX	The port accepts/generates NOVATELX corrections.
36-40	Reserved	
41	KVH1750IMU	The port supports communication with a KVH 1750 or KVH P-1750 IMU.
42	VERIPOS	This port supports Veripos, TerraStar and Oceanix PPP corrections.
43-45	Reserved	
46	TCCOM1	CCOM1 Tunnel
47	TCCOM2	CCOM2 Tunnel
48	TCCOM3	CCOM3 Tunnel
49	NOVATELMINBINARY	NovAtel binary message with a minimal header. Only available for CCOM ports.
50	TCCOM4	CCOM4 Tunnel
51	TCCOM5	CCOM5 Tunnel

Binary Value	ASCII Value	Description
52	TCCOM6	CCOM6 Tunnel
53-57	Reserved	
60	TSCOM1	SCOM1 Tunnel
61	TSCOM2	SCOM2 Tunnel
62	TSCOM3	SCOM3 Tunnel
63	TSCOM4	SCOM4 Tunnel
64	LUA	Lua stdin/stdout/stderr. Use the LUA PROMPT command to set this Interface Mode.
65	HONEYWELLIMU	This port supports communication with a Honeywell HG4930 IMU

## 2.69 IONOCONDITION

### *Sets ionospheric condition*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to change the level of ionosphere activity that is assumed by the RTK positioning algorithms.



Only advanced users should use this command.

**Message ID:** 1215

**Abbreviated ASCII Syntax:**

```
IONOCONDITION mode
```

**Factory Default:**

```
IONOCONDITION AUTO
```

**ASCII Example:**

```
IONOCONDITION normal
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	IONOCONDITION header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).		H	
2	mode	quiet	0	Receiver assumes a low level of ionosphere activity	Enum	4	H
		normal	1	Receiver assumes a medium level of ionosphere activity			
		disturbed	2	Receiver assumes a high level of ionosphere activity			
		auto	10	Receiver monitors the ionosphere activity and adapts behavior accordingly			

## 2.70 IPCONFIG

### *Configures network IP settings*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command is used to configure static/dynamic TCP/IP properties for the Ethernet connection.



In addition to configuring an IP address and netmask for the interface, this command also includes a gateway address.



For certain applications when **STATIC** IP address is used, it is necessary to specify a DNS server using the **DNSCONFIG** command (see page 144) in order to use host names.

**Message ID:** 1243

#### Abbreviated ASCII Syntax:

```
IPCONFIG [interface_name] address_mode [IP_address [netmask [gateway]]]
```

#### Factory Default:

```
IPCONFIG ETHA DHCP
```

#### ASCII Examples:

```
IPCONFIG ETHA STATIC 192.168.74.10 255.255.255.0 192.168.74.1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	IPCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	interface_name	ETHA	2	Name of the Ethernet interface (default = ETHA)	Enum	4	H
3	address_mode	DHCP	1	Use Dynamic IP address	Enum	4	H+4
		STATIC	2	Use Static IP address			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	IP_address	ddd.ddd.ddd.ddd (For example: 10.0.0.2)		IP Address-decimal dot notation	String [16]	variable <sup>1</sup>	H+8
5	netmask	ddd.ddd.ddd.ddd (For example: 255.255.255.0)		Netmask-decimal dot notation	String [16]	variable <sup>1</sup>	H+24
6	gateway	ddd.ddd.ddd.ddd (For example: 10.0.0.1)		Gateway-decimal dot notation	String [16]	variable <sup>1</sup>	H+40

---

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.



## 2.71 IPSERVICE

### Configure availability of networks ports/services

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

Use the **IPSERVICE** command to configure the availability of specific network ports/services. When disabled, the service does not accept incoming connections.



On most OEM7 receivers, the FTP Server is disabled by default. The exception is the PwrPak7 which has FTP enabled by default.



We have found two problems in the Microsoft® FTP clients contained within the Internet Explorer® and Edge browsers which make them unsuitable for retrieving files from a NovAtel receiver. When using a Windows® computer to transfer files off a NovAtel receiver, we suggest using a 3rd party FTP client.

**Message ID:** 1575

#### Abbreviated ASCII Syntax:

```
IPSERVICE IPService switch
```

#### Factory Default:

```
IPSERVICE WEB_SERVER DISABLE (OEM719 and OEM7500)
```

```
IPSERVICE WEB_SERVER ENABLE (OEM729, OEM7600 OEM7700 and OEM7720)
```

```
IPSERVICE SECURE_ICOM DISABLE
```



#### ASCII Example:

```
IPSERVICE FTP_SERVER ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	IPSERVICE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	ipservice	See Table 47: IP Services on the next page		IP service	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	switch	DISABLE	0	Disable the IP service specified.	Enum	4	H+4
		ENABLE	1	Enable the IP service specified.			

Table 47: IP Services

Value (Binary)	IP Service (ASCII)	Description
0	NO_PORT	No port
1	FTP_SERVER	<p>FTP server port.</p> <p>For most OEM7 receivers the default = DISABLE. For the PwrPak7 the default = ENABLE.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  A username and password is required to access the FTP server. The default username is <i>admin</i>. The default password for receiver cards is the receiver PSN. The default password for the PwrPak7 or CPT7 is the enclosure PSN. To change the default password, see the <b>SETADMINPASSWORD</b> command on page 388. </div>
2	WEB_SERVER	<p>Web server port</p> <p>For most OEM7 receivers the default = ENABLE. For the OEM7500 and OEM719 the default = DISABLE.</p>
3	SECURE_ICOM	<p>Enables or disables security on ICOM ports.</p> <p>When security is enabled, a login is required as part of the connection process (see the <b>LOGIN</b> command on page 253).</p> <p>Default = DISABLE</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  Security in this sense means users must supply a name and password before being allowed to enter commands on the ICOM ports. It does not mean there is data encryption </div>
4	SOFTLOADFILE	<p>Enables or disables SOFTLOADFILE.</p> <p>By default the receiver will not allow the <b>SOFTLOADFILE</b> command (see page 409) to operate until it is enabled using the <b>IPSERVICE</b> command.</p>

## 2.72 ITBANDPASSCONFIG

### *Enable and configure bandpass filter on receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to apply a bandpass filter at a certain frequency to mitigate interference in the pass band of GNSS signals. The **ITBANDPASSBANK** log (see page 663) provides information on the allowable configuration settings for each frequency band. The bandpass filter is symmetrical in nature, which means that specifying one cutoff frequency will apply a cutoff on both the low side and high side of the signal's center frequency. Only one filter can be applied for each signal.



To enable a filter on the signal received from the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), use the **ITBANDPASSCONFIG\_1** command.



On OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7 receivers, any filter enabled for GPS L2 or GLONASS L2 on the secondary antenna will be applied to both GPS L2 and GLONASS L2. For this reason, care must be taken to avoid attenuating the signals with a bandpass filter that is too narrow in bandwidth. The recommended maximum lower cutoff frequency is 1221 MHz. The recommended minimum upper cutoff frequency is 1254 MHz.



When applying a filter to a signal using the MEC correlator, a brief interruption in tracking may occur on affected channels.



This command is available only on some software models. Contact your NovAtel Sales representative ([novatel.com/contactus/sales-offices](http://novatel.com/contactus/sales-offices)) to obtain interference mitigation functionality.

**Message ID: 1999**

#### **Abbreviated ASCII Syntax:**

```
ITBANDPASSCONFIG frequency switch [cutofffrequency]
```

#### **ASCII Example:**

```
ITBANDPASSCONFIG gpsl5 enable 1165.975
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITBANDPASSCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	frequency	See Table 54: <i>Frequency Types</i> on page 238		Set the frequency band on which to apply the filter	Enum	4	H
3	switch	DISABLE	0	Disable filter	Enum	4	H+4
		ENABLE	1	Enable filter			
4	cutofffrequency			Cut off frequency for band pass filter (MHz). (default = 0) Refer to <b>ITBANDPASSBANK</b> log (see page 663) for the allowable values.	Float	4	H+8

## 2.73 ITDETECTCONFIG

### *Enable interference detection on receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to enable or disable interference detection on the receiver. It is applicable to both Spectral Analysis Detection and Statistical Analysis Detection at the same time. Detection can be enabled on all RF paths, only one RF path (L1, L2, or L5), or no RF paths. By default, only the RF paths connecting to the first antenna are enabled.



To start interference detection on the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), use the **ITDETECTCONFIG\_1** command. The interference detection results for the secondary antenna are shown in the **ITDETECTSTATUS\_1** log.

Enabling interference detection on both antennas slows down interference detection. In most installations, the primary and secondary antennas will be impacted by the same interference signal. Therefore, enabling interference detection on both antennas will slow down detection and may provide only limited benefits.

**Message ID: 2143**

#### **Abbreviated ASCII Syntax:**

```
ITDETECTCONFIG RFPATH [reserved1] [reserved2] [reserved3]
```

#### **Factory Default:**

```
ITDETECTCONFIG all
```

#### **ASCII Example:**

```
ITDETECTCONFIG L1
```

```
ITDETECTCONFIG none
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITDETECTCONFIG header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	RFPATH	See <i>Table 48: RF Path Selection</i> below		RF path selected for detection. By default, all paths are turned on. The receiver will cycle through all active paths.	Enum	4	H
3	reserved1	0		Reserved parameter	Ulong	4	H+4
4	reserved2	0		Reserved parameter	Ulong	4	H+8
5	reserved3	0		Reserved parameter	Ulong	4	H+12

Table 48: RF Path Selection

ASCII Value	Binary Value	Description
NONE	0	Turn off detection on all paths
ALL	1	Turn on detection on all paths (cycle through all active paths)
L1	2	Turn on detection only on L1 path
L2	3	Turn on detection only on L2 path
L5	4	Turn on detection only on L5 path

## 2.74 ITFRONTENDMODE

### *Configure the front end mode settings*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to configure the front end mode for the L1, L2 and L5 RF paths to use the default third-order CIC mode or HDR (High Dynamic Range) mode. The HDR mode is used in an interference environment to obtain best interference rejection in general. However, the power consumption will increase in this mode.



To configure the front end mode settings for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), use the **ITFRONTENDMODE\_1** command.



This command is available only on some software models. Contact your NovAtel Sales representative ([novatel.com/contactus/sales-offices](http://novatel.com/contactus/sales-offices)) to obtain interference mitigation functionality.

**Message ID: 2039**

#### **Abbreviated ASCII Syntax:**

```
ITFRONTENDMODE frequency mode
```

#### **Factory Default**

```
ITFRONTENDMODE L1 cic3
```

```
ITFRONTENDMODE L2 cic3
```

```
ITFRONTENDMODE LBAND cic3
```

```
ITFRONTENDMODE L5 cic3
```

#### **ASCII Example:**

```
ITFRONTENDMODE L1 hdr
```



On the OEM7500, the default mode for all frequency bands is HDR.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITFRONTENDMODE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	frequency	See <i>Table 49: Frequency Bands</i> below		Set the frequency band for adjustment	Enum	4	H
3	mode	See <i>Table 50: Mode</i> below		Select the desired mode	Enum	4	H+4

**Table 49: Frequency Bands**

Binary Value	ASCII Value	Description
2	L1	Selects the L1 frequency
3	L2	Selects the L2 frequency
4	LBAND	Selects the L-Band frequency
5	L5	Selects the L5 frequency

**Table 50: Mode**

Binary Value	ASCII Value	Description
0	CIC3	3rd order CIC (CIC3) mode (default)
1	HDR	High Dynamic Range (HDR) mode



## 2.75 ITPROGFILTCONFIG

### *Enable and configure filtering on the receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to set the programmable filter to be either a notch filter or a bandpass filter to mitigate interference in the pass band of GNSS signals. The notch filter is used to attenuate a very narrow band of frequencies (specified by the notch width) around the configured center frequency.

The bandpass filter is symmetrical in nature, which means that specifying one cutoff frequency will apply a cutoff on both the low side and high side of the spectrum center frequency.

The **ITPROGFILTBANK** log (see page 671) provides information on the allowable configuration settings for the programmable filter (i.e. the allowable settings for the notch filter and bandpass filter) for each frequency band. Only one filter can be applied for each frequency.



To enable a filter on the signal received from the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), use the **ITPROGFILTCONFIG\_1** command.



When applying a filter to a signal using the MEC correlator, a brief interruption in tracking may occur on affected channels.



This command is available only on some software models. Contact your NovAtel Sales representative ([novatel.com/contactus/sales-offices](http://novatel.com/contactus/sales-offices)) to obtain interference mitigation functionality.

**Message ID: 2000**

#### **Abbreviated ASCII Syntax:**

```
ITPROGFILTCONFIG frequency filterid switch [filtermode] [cutofffreq]
[notchwidth]
```

#### **ASCII Example:**

```
ITPROGFILTCONFIG gpsl1 pf0 enable notchfilter 1580 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITPROGFILTCONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	frequency	See <i>Table 54: Frequency Types</i> on page 238		Set the frequency band on which to apply the filter	Enum	4	H
3	filterid	See <i>Table 51: Programmable Filter ID</i> below		Select the filter ID to use	Enum	4	H+4
4	switch	DISABLE	0	Disable the filter	Enum	4	H+8
		ENABLE	1	Enable the filter			
5	filtermode	See <i>Table 52: Programmable Filter Mode</i> on the next page		Configure the type of filter to use (default = NONE)	Enum	4	H+12
6	cutofffreq			Center frequency for notch filter or cut off frequency for bandpass filter (MHz).  Refer to <b>ITPROGFILTBANK</b> log (see page 671) for the allowable values.  (default = 0)	Float	4	H+16
7	notchwidth			Notch width (MHz).  Refer to <b>ITPROGFILTBANK</b> log (see page 671) for the allowable values.  (default = 0)	Float	4	H+20

Table 51: Programmable Filter ID

Binary Value	ASCII Value	Description
0	PF0	Programmable Filter 0
1	PF1	Programmable Filter 1

**Table 52: Programmable Filter Mode**

<b>Binary Value</b>	<b>ASCII Value</b>	<b>Description</b>
0	NOTCHFILT	Configure the filter as a notch filter
1	BANDPASSFIL	Configure the filter as a bandpass filter
2	NONE	Turn off filter If the switch parameter is set to ENABLED while the filtermode parameter is set to NONE, the system will return a parameter out of range message.

## 2.76 ITSPECTRALANALYSIS

### *Enable and configure spectral analysis on receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to view the spectrum in a range of frequencies. The **ITSPECTRALANALYSIS** command enables and configures the spectral analysis. The spectrum is viewed by plotting the PSD samples in the **ITPSDFINAL** log (see page 676).



Decreasing the update period or increasing the FFT size will impact receiver idle time. The idle time should be monitored to prevent adverse effects on receiver performance.



Due to the high volume of data, a higher bandwidth medium, such as USB or Ethernet, is recommended when monitoring signals using the Interference Toolkit.



To enable and configure spectral analysis on the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), use the **ITSPECTRALANALYSIS\_1** command.

**Message ID:** 1967

#### **Abbreviated ASCII Syntax:**

```
ITSPECTRALANALYSIS mode [frequency] [updateperiod] [FFTsize] [timeavg]
[integration_window]
```

#### **Factory Default:**

```
ITSPECTRALANALYSIS off
```

#### **ASCII Example:**

```
ITSPECTRALANALYSIS predecimation gps11 500 1K 20 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITSPECTRALANALYSIS header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	See Table 53: Data Sources for PSD Samples below		Set the view mode.	Enum	4	H
3	frequency	See Table 54: Frequency Types on the next page		Set the frequency band to view.	Enum	4	H+4
4	updateperiod	50 to 100000		The spectrum update rate in milliseconds. The update period is limited by the FFT size chosen. For 32k the minimum update period is 100 ms and for 64k the minimum update period is 200 ms. (default = 1000)	Ulong	4	H+8
5	FFTsize	See Table 55: FFT Sizes on page 239		The frequency resolution of the spectrum. (default = 1k)	Enum	4	H+12
6	timeavg	0 to 50		Time averaging window in seconds. 0 means no time averaging. (default = 10)	Ulong	4	H+16
7	integration window	1 to 1024		The integration window size of FFT samples. 1 means no integration. (default = 5)	Ulong	4	H+20

Table 53: Data Sources for PSD Samples

Binary Value	ASCII Value	Description
0	OFF	Disable spectral analysis
1	PREDECIMATION	Perform spectrum analysis on the pre-decimated spectrum. This can be used to see a wide view of the spectrum for an RF path (L1, L2 or L5).
2	POSTDECIMATION	Perform spectrum analysis on the post-decimated spectrum. This is narrower than pre-decimation and is used to see the spectrum for a given signal.

Binary Value	ASCII Value	Description
3	POSTFILTER	Perform spectrum analysis on the post-filtered spectrum. This can be used when either bandpass or notch filters have been enabled to see the spectrum after the filters are applied.



The post-filter spectrum is not available for the Galileo AltBOC frequency. Only the pre-decimation and post-decimation spectrums are available for Galileo AltBOC.

**Table 54: Frequency Types**

Binary Value	ASCII Value	Description
0	GPSL1	GPS L1 frequency
1	GPSL2	GPS L2 frequency
2	GLONASSL1	GLONASS L1 frequency
3	GLONASSL2	GLONASS L2 frequency
4	Reserved	
5	GPSL5	GPS L5 frequency
6 <sup>1</sup>	LBAND	Inmarsat L-Band frequency
7	GALILEOE1	Galileo E1 frequency
8	GALILEOE5A	Galileo E5A frequency
9	GALILEOE5B	Galileo E5B frequency
10	GALILEOALTBOC	Galileo AltBOC frequency
11	BEIDOUB1	BeiDou B1 frequency
12	BEIDOUB2	BeiDou B2I frequency
13	QZSSL1	QZSS L1 frequency
14	QZSSL2	QZSS L2 frequency
15	QZSSL5	QZSS L5 frequency
16	QZSSL6	QZSS L6 frequency
17	GALILEOE6	Galileo E6 frequency
18	BEIDOUB3	BeiDou B3 frequency
19	GLONASSL3	GLONASS L3 frequency

<sup>1</sup>Must first enable L-Band using the ASSIGNLBANDBEAM command.

Binary Value	ASCII Value	Description
20	NAVICL5	NavIC L5 frequency
21	BEIDOU B1C	BeiDou B1C frequency
22	BEIDOU B2A	BeiDou B2a frequency
23	BEIDOU B2B	BeiDou B2b frequency

Table 55: FFT Sizes

Binary Value	ASCII Value	Description
0	1K	1K FFT, 1024 samples
1	2K	2K FFT, 2048 samples
2	4K	4K FFT, 4096 samples
3	8K	8K FFT, 8192 samples
4	16K	16K FFT, 16384 samples
5	32K	32K FFT, 32768 samples
6	64K	64K FFT, 65536 samples



The 64k FFT is not available in post-decimation or post-filter modes.

## 2.77 ITWARNINGCONFIG

### Configure the sensitivity level for the Jammer Detected bit

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to configure the sensitivity level for setting the Jammer Detected bit in the Receiver Status word in the **RXSTATUS** log (see page 878). Warning level 0 is the most sensitive, while level 3 is the least sensitive. Warning level 0 means the Jammer Detected bit can be set even when minor interference is detected. By default, the warning level is set at 3.



On dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), the **ITWARNINGCONFIG** setting applies to both antenna inputs.

**Message ID:** 2289

#### Abbreviated ASCII Syntax

```
ITWARNINGCONFIG WarningLevel [Reserved]
```

#### Factory Default:

```
ITWARNINGCONFIG 3
```

#### ASCII Example:

```
ITWARNINGCONFIG 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	ITWARNINGCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	WarningLevel	0		Detection warning sensitivity level 0 = most sensitive 3 = least sensitive	Ulong	4	H
		1					
		2					
		3					
3	Reserved	0		Reserved	Ulong	4	H+4



## 2.78 J1939CONFIG

### Configure CAN network-level parameters

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to configure the CAN J1939 network-level parameters (NAME, etc).

Issuing this command may initiate a CAN 'Address Claim' procedure. The status of the node and address claim are reported in the **J1939STATUS** log (see page 680).

Once a "node" is configured using **J1939CONFIG**, and the "port" is configured to ON using **CANCONFIG** "port" ON, J1939CONFIG "node" cannot be entered again until the "port" is configured to "OFF" using **CANCONFIG** "port" OFF. (See the **CANCONFIG** command on page 110)

**Message ID: 1903**

#### Abbreviated ASCII Syntax:

```
J1939CONFIG node port [pref_addr [alt_addr_range_start] [alt_addr_range_end]
[mfgcode] [industry] [devclass] [devinstance] [func] [funcinstance]
[ECUinstance]]
```



#### Factory Default:

```
J1939CONFIG NODE1 CAN1 1C 0 FD 305 2 0 0 23 0 0
J1939CONFIG NODE2 CAN2 1C 0 FD 305 2 0 0 23 0 0
```

#### ASCII Example :

```
J1939CONFIG NODE1 CAN1 AA 0 FD 305 2 0 0 23 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	J1939CONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	node	NODE1	1	Identifies the J1939 Node (i.e., CAN NAME)  SMART2 does not support NODE2.	Enum	4	H
		NODE2	2				
3	port	CAN1	1	Physical CAN port to use  SMART2 does not support the CAN2 port.	Enum	4	H+4
		CAN2	2				
4	pref_addr	0x0 - 0xFD		Preferred CAN address. The receiver attempts to claim this address (default = 0x0)	Ulong	4	H+8
5	alt_addr_range_start	0x0 - 0xFD		When the pref_addr cannot be claimed, the receiver attempts to claim an address from this range. (default: 0x0)	Ulong	4	H+12
6	alt_addr_range_end	0x0 - 0xFD		End of alternative address range. (default: 0xFD)	Ulong	4	H+16
7	mfgcode	0-2047		NAME: Manufacturer Code. Refer to ISO 11783-5. (default: 0)	Ulong	4	H+20
8	industry	0 - 7		NAME: Industry Group (default: 2)	Ulong	4	H+24
9	devclass	0 - 127		NAME: Device Class (default: 0)	Ulong	4	H+28

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
10	devinstance	0 - 15		NAME: Device Class Instance (default: 0)	Ulong	4	H+32
11	func	0 - 255		NAME: Function (default: 23)	Ulong	4	H+36
12	funcinstance	0 - 31		NAME: Function instance (default: 0)	Ulong	4	H+40
13	ECUinstance	0 - 7		NAME: ECU Instance (default: 0)	Ulong	4	H+44



Due to current limitations in the CAN stack, NODE1 can only be associated with CAN1 and NODE2 can only be associated with CAN2. A mismatch combination results in an 'invalid parameter' error.



Node statistics are reported in the **J1939STATUS** log (see page 680).

## 2.79 LOCKOUT

*Prevents the receiver from using a satellite*



The **LOCKOUT** command has been deprecated. To prevent the receiver from using a satellite in the solution computations, use the **LOCKOUTSATELLITE** command on the next page.

**Message ID: 137**

## 2.80 LOCKOUTSATELLITE

### *Prevents use of a satellite*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command locks a specified satellite out from being used in solution computations. This command only locks out single satellites and must be repeated for each satellite. See also the **UNLOCKOUTSATELLITE** command on page 450.

#### Abbreviated ASCII Syntax:

```
LOCKOUTSATELLITE system prn
```

**Message ID: 2415**

#### ASCII Example:

```
LOCKOUTSATELLITE gps 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	LOCKOUTSATELLITE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	system	See <i>Table 124: Satellite System</i> on page 654		The satellite system for the satellite being locked out.	Enum	4	H
3	prn	See <i>1.6.1 PRN Numbers</i> on page 52		Unique identifier for the satellite being locked out.	Ulong	4	H+4

## 2.81 LOCKOUTSYSTEM

*Prevents the receiver from using a system*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to prevent the receiver from using all satellites in a system in the solution computations.



The **LOCKOUTSYSTEM** command does not prevent the receiver from tracking an undesirable satellite.

This command must be repeated for each system to be locked out. See also the **UNLOCKOUTSYSTEM** command on page 451 and **UNLOCKOUTALL** command on page 449.

**Message ID:** 871

**Abbreviated ASCII Syntax:**

```
LOCKOUTSYSTEM system
```

**Factory Defaults:**

```
LOCKOUTSYSTEM sbas
```

```
LOCKOUTSYSTEM navic
```



The **LOCKOUTSYSTEM** command removes one or more systems from the solution while leaving other systems available.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	LOCKOUTSYSTEM header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	system	See <i>Table 124: Satellite System</i> on page 654		A single satellite system to be locked out	Enum	4	H

## 2.82 LOG

### *Requests logs from the receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Many different types of data can be logged using different methods of triggering the log events. Every log element can be directed to any combination of the receiver's ports. The ontime trigger option requires the addition of the period parameter. See *Logs* on page 481 for further information and a complete list of data log structures. The LOG command tables in this section show the binary format followed by the ASCII command format.

The optional parameter [hold] prevents a log from being removed when the **UNLOGALL** command (see page 454), with its defaults, is issued. To remove a log which was invoked using the [hold] parameter requires the specific use of the **UNLOG** command (see page 452). To remove all logs that have the [hold] parameter, use the **UNLOGALL** command (see page 454) with the held field set to 1.

The [port] parameter is optional. If [port] is not specified, [port] is defaulted to the port that the command was received on.



1. The OEM7 family of receivers can handle 80 simultaneous log requests. If an attempt is made to log more than 80 logs at a time, the receiver responds with an Insufficient Resources error. Note that **RXSTATUSEVENTA** logs are requested on most ports by default and these logs count against the 80.
2. The user is cautioned that each log requested requires additional CPU time and memory buffer space. Too many logs may result in lost data and low CPU idle time. Receiver overload can be monitored using the idle-time field and buffer overload bits of the Receiver Status in any log header.
3. Only the **MARKPOS**, **MARK2POS**, **MARK3POS** and **MARK4POS** log (see page 696), **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** log (see page 699) and 'polled' log types are generated, on the fly, at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.
4. Use the ONNEW trigger with the **MARKPOS**, **MARK2POS**, **MARK3POS** and **MARK4POS** log (see page 696) and **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** log (see page 699).
5. Polled log types do not all allow fractional offsets.
6. If ONTIME trigger is used with asynchronous logs, the time stamp in the log does not necessarily represent the time the data was generated but rather the time when the log is transmitted.
7. Published logs are not placed in a queue if there is no physical or virtual connection when the log is generated. Thus, a log requested ONNEW or ONCHANGED that is in SAVECONFIG may not be received if it is published before connections are made. This can happen if there's no cable connected or if the communication protocol has not been established yet (e.g. CAN, Ethernet, USB, etc).
8. PwrPak7M variants do not support the FILE port.

**Message ID:** 1

**Abbreviated ASCII Syntax:**

```

LOG [port] message ONNEW
LOG [port] message ONCHANGED
LOG [port] message ONTIME period [offset [hold]]
LOG [port] message ONNEXT
LOG [port] message ONCE
LOG [port] message ONMARK

```

**Factory Default:**

```

LOG COM1 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG COM2 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG COM3 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG USB1 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG USB2 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG USB3 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM1 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM2 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM3 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM4 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM5 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM6 RXSTATUSEVENTA ONNEW 0 0 HOLD
LOG ICOM7 RXSTATUSEVENTA ONNEW 0 0 HOLD

```

**Abbreviated ASCII Example 1:**

```
LOG COM1 BESTPOS ONTIME 7 0.5 HOLD
```

The above example shows **BESTPOS** logging to com port 1 at 7 second intervals and offset by 0.5 seconds (output at 0.5, 7.5, 14.5 seconds and so on). The [hold] parameter is set so that logging is not disrupted by the **UNLOGALL** command (see page 454).

To send a log once, the trigger option can be omitted.

**Abbreviated ASCII Example 2:**

```
LOG COM1 BESTPOS ONCE
```



Using the NovAtel Connect utility there are two ways to initiate data logging from the receiver's serial ports. Either enter the **LOG** command in the *Console* window or use the interface provided in the *Logging Control* window. Ensure the Power Settings on the computer are not set to go into Hibernate or Standby modes. Data is lost if one of these modes occurs during a logging session.

**2.82.1 Binary**

Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	LOG (binary) header	See <i>Table 4: Binary Message Header Structure</i> on page 38	This field contains the message header	-	H	0



Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	port	See <i>Table 5: Detailed Port Identifier</i> on page 40	Output port	Enum	4	H
3	message	Any valid message ID	Message ID of the log to output	Ushort	2	H+4
4	message type	Bits 0-4 = Measurement source <sup>1</sup> Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit ( <i>Message Responses</i> on page 49) 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7
6	trigger	See <i>Table 56: Binary Log Triggers</i> on the next page		Enum	4	H+8

<sup>1</sup>Bits 0-4 are used to indicate the measurement source. For dual antenna receivers, if bit 0 is set, the log is from the secondary antenna.

Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	period	Valid values for the high rate logging are 0.01, 0.02, 0.05, 0.1, 0.2, 0.25 and 0.5. For logging slower than 1 Hz any integer value is accepted	Log period (for ONTIME trigger) in seconds If the value entered is lower than the minimum measurement period, the command will be rejected. Refer to Receiver Technical Specifications on the OEM7 Documentation Portal ( <a href="https://docs.novatel.com/OEM7">docs.novatel.com/OEM7</a> ) for the maximum raw measurement rate to calculate the minimum period.	Double	8	H+12
8	offset	Offset for period (ONTIME trigger) in seconds.	A valid value is any integer (whole number) smaller than the period. For example, to log data at 1 second, after every minute, set the period to 60 and the offset to 1. These decimal values, on their own, are also valid: 0.1, 0.2, 0.25 or 0.5, as well as any multiple of the maximum logging rate defined by the receiver model. The offset cannot be smaller than the minimum measurement period supported by the model.	Double	8	H+20
9	hold	0 = NOHOLD	Allow log to be removed by the <b>UNLOGALL</b> command (see page 454)	Enum	4	H+28
		1 = HOLD	Prevent log from being removed by the default <b>UNLOGALL</b> command (see page 454)			

Table 56: Binary Log Triggers

Binary Value	Description
0 = ONNEW	Does not output current message but outputs when the message is updated (not necessarily changed)
1 = ONCHANGED	Outputs the current message and then continues to output when the message is changed

Binary Value	Description
2 = ONTIME	Output on a time interval
3 = ONNEXT	Output only the next message
4 = ONCE	Output only the current message (default). If no message is currently present, the next message is output when available.
5 = ONMARK	<p>Output when a pulse is detected on the mark 1 input, MK1I</p> <p>Refer to Receiver Technical Specifications on the OEM7 Documentation Portal (<a href="https://docs.novatel.com/OEM7">docs.novatel.com/OEM7</a>) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.*</p> <p>Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the <b>MARKPOS, MARK2POS, MARK3POS and MARK4POS</b> log on page 696 and the <b>MARK1TIME, MARK2TIME, MARK3TIME and MARK4TIME</b> log on page 699.</p>

### 2.82.2 ASCII

Field	Field Name	ASCII Value	Description	Format
1	LOG (ASCII) header	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII or ASCII respectively	-
2	port	<i>Table 5: Detailed Port Identifier on page 40</i>	Output port (default = THISPORT)	Enum
3	message	Any valid message name, with an optional A or B suffix	Message name of log to output	Char [ ]
4	trigger	See <i>Table 57: ASCII Log Triggers</i> on the next page		Enum
5	period	Any positive double value larger than the receiver's minimum raw measurement period	Log period (for ONTIME trigger) in seconds (default = 0) If the value entered is lower than the minimum measurement period, the command will be rejected. Refer to Receiver Technical Specifications on the OEM7 Documentation Portal ( <a href="https://docs.novatel.com/OEM7">docs.novatel.com/OEM7</a> ) for the maximum raw measurement rate to calculate the minimum period.	Double

Field	Field Name	ASCII Value	Description	Format
6	offset	Any positive double value smaller than the period	Offset for period (ONTIME trigger) in seconds (default = 0) For example, if you want to log data, at 1 second after every minute, set the period to 60 and the offset to 1.	Double
7	hold	NOHOLD	To be removed by the <b>UNLOGALL</b> command (see page 454) (default)	Enum
		HOLD	Prevent log from being removed by the default <b>UNLOGALL</b> command (see page 454)	

Table 57: ASCII Log Triggers

ASCII Value	Description
ONNEW	Output when the message is updated (not necessarily changed)
ONCHANGED	Output when the message is changed
ONTIME	Output on a time interval
ONNEXT	Output only the next message
ONCE	Output only the current message (default)
ONMARK	<p>Output when a pulse is detected on the mark 1 input, MK1I</p> <p>Refer to Receiver Technical Specifications on the OEM7 Documentation Portal (<a href="https://docs.novatel.com/OEM7">docs.novatel.com/OEM7</a>) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.</p> <p>Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the <b>MARKPOS, MARK2POS, MARK3POS and MARK4POS</b> log on page 696 and the <b>MARK1TIME, MARK2TIME, MARK3TIME and MARK4TIME</b> log on page 699.</p>

## 2.83 LOGIN

### *Start a secure ICOM/SCOM connection to the receiver*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

When ICOM/SCOM ports have security enabled (see the **IPSERVICE** command on page 225), a session to the ICOM/SCOM port can be established but commands are refused until a valid **LOGIN** command is issued. Both the UserName and Password are required. The **LOGIN** command checks the supplied credentials against known UserNames/Passwords and determines if the login is successful or not. A successful login permits the secured ICOM/SCOM command interpreter to accept further commands and returns OK. An unsuccessful login does not release the secured ICOM/SCOM command interpreter and returns Login Failed.

Entering a **LOGIN** command on any command port other than the ICOM/SCOM port has no effect, regardless of whether the UserName/Password is correct. In this case, the appropriate response (OK or Login Failed) is returned, but there is no effect on the command interpreter.



When security is enabled, access to the port is restricted unless a valid name and password are supplied. It does not mean there is data encryption enabled. Username is case-insensitive and password is case-sensitive.

**Message ID: 1671**

#### **Abbreviated ASCII Syntax:**

```
LOGIN [commport] UserName Password
```

#### **ASCII Example:**

```
LOGIN ADMIN ADMINPASSWORD
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	LOGIN header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	commport	ICOM1	23	The ICOM or SCOM port to log into.	Enum	4	H
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46	This is an optional parameter.			
		ICOM6	47	If no value is entered, logs in to the ICOM port currently being used. (default=THISPORT)			
		ICOM7	48				
		SCOM1	49				
		SCOM2	50				
		SCOM3	51				
SCOM4	52						
3	username			Provide the user name for the login command. The user name is not case sensitive.	String [32]	variable <sup>1</sup>	H+4
4	password			Provide the password for the user name. The password is case sensitive.	String [28]	variable <sup>1</sup>	variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.84 LOGOUT

**End a secure ICOM/SCOM session started using the LOGIN command**

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

Use the **LOGOUT** command to sign out of an ICOM/SCOM connection after a user has successfully logged in using the **LOGIN** command. After the sending the **LOGOUT** command, the ICOM/SCOM connection will not accept further commands, other than a new LOGIN command. The session itself is not ended. This only applies to ICOM/SCOM ports that have had security enabled (see the **IPSERVICE** command on page 225).

**Message ID: 1672**

**Abbreviated ASCII Syntax:**

```
LOGOUT [commport]
```

**ASCII Example:**

```
LOGOUT
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	LOGOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	commport	ICOM1	23	The ICOM or SCOM port from which to log out. This is an optional parameter. If no value is entered, logs out from the ICOM/SCOM port currently being used.	Enum	4	H
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46				
		ICOM6	47				
		ICOM7	48				
		SCOM1	49				
		SCOM2	50				
		SCOM3	51				
		SCOM4	52				

## 2.85 LUA

### Configure Lua Interpreter

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to configure the execution of the Lua interpreter on the receiver. Scripts that appear within the **LUAFILELIST** log (see page 690) can be executed by the Lua interpreter.

**Message ID:** 2049

#### Abbreviated ASCII Syntax:

```
LUA option [LuaInterpreterArguments]
```

#### Abbreviated ASCII Example:

```
lua start "printarguments.lua 1 2 3 4 5"
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	LUA header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	START	1	Start the Lua interpreter in the background. The file descriptors stdout, stdin and stderr will not be accessible outside the receiver.	Enum	4	H
		PROMPT	2	Start the Lua interpreter in interactive mode and connect stdout, stdio and stderr to the port on which the command was entered. The INTERFACEMODE of that port will be changed to LUA for both RX and TX.			



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	LuaInterpreter Arguments	STRING		String containing Lua interpreter options including the name of the script file to run and arguments to pass to the script.  This string must be enclosed in quotes if it contains any spaces.  String arguments within the field <b>must</b> be enclosed by single quotes.	String [400]	Variable	H+4

The format of the Lua Interpreter Arguments is as follows as adapted from the standard Lua 5.3 interpreter:

```
[options] [script [args]]
```

Available options are:

```
-e stat execute string 'stat'
```

```
-i enter interactive mode after executing 'script'.
```

(This is added to the arguments when using the PROMPT option of the LUA command)

```
-l name require library 'name'
```

## 2.86 MAGVAR

### Sets a magnetic variation correction

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The receiver computes directions referenced to True North (also known as geodetic north). The Magnetic Variation Correction command (MAGVAR) is used to navigate in agreement with magnetic compass bearings. The correction value entered here causes the "bearing" field of the navigate log to report bearing in degrees Magnetic. The receiver computes the magnetic variation correction when using the auto option. See *Figure 6: Illustration of Magnetic Variation and Correction* below.

The receiver calculates values of magnetic variation for given values of latitude, longitude and time using the International Geomagnetic Reference Field (IGRF) 2020 spherical harmonic coefficients and IGRF time corrections to the harmonic coefficients. (IGRF-2020 is also referred to as IGRF-13.) The model is intended for use up to the year 2025. Furthermore, the receiver will compute for years beyond 2025, but accuracy may be further reduced depending on the behavior of the magnetic pole.

**Message ID: 180**

#### Abbreviated ASCII Syntax:

```
MAGVAR type [correction [std dev]]
```

#### Factory Default:

```
MAGVAR correction 0 0
```

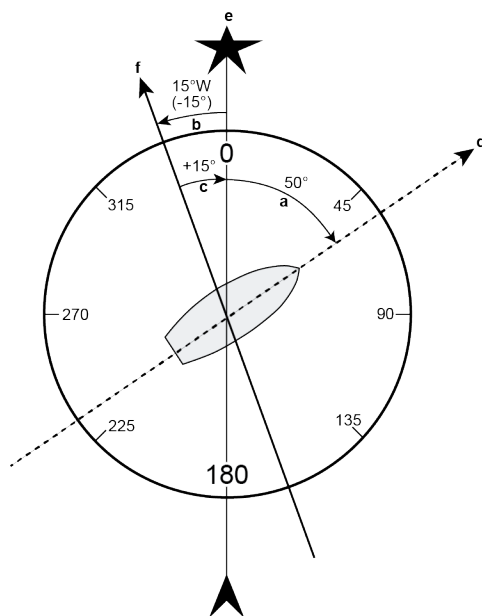
#### ASCII Example 1:

```
MAGVAR AUTO
```

#### ASCII Example 2:

```
MAGVAR CORRECTION 15 0
```

**Figure 6: Illustration of Magnetic Variation and Correction**



Ref	Description
a	True Bearing
b	Local Magnetic Variation
c	Local Magnetic Variation Correction (Inverse of magnetic variation)
a + c	Magnetic Bearing
d	Heading: 50° True, 65° Magnetic
e	True North
f	Local Magnetic North



Magnetic North refers to the location of the Earth's Magnetic North Pole. Its position is constantly changing in various cycles over centuries, years and days. These rates of change vary and are not well understood. However, we are able to monitor the changes.

True North refers to the earth's spin axis, that is, at 90° north latitude or the location where the lines of longitude converge. The position of the spin axis does not vary with respect to the Earth.

The locations of these two poles do not coincide. Thus, a relationship is required between these two values for users to relate GNSS bearings to their compass bearings. This value is called the magnetic variation correction or declination.

GNSS does not determine where Magnetic North is nor do the satellites provide magnetic correction or declination values. However, OEM7 receivers store this information internally in look up tables so that when you specify that you want to navigate with respect to Magnetic North, this internal information is used. These values are also available from various information sources such as the United States Geological Survey (USGS). The USGS produces maps and has software which enables the determination of these correction values. By identifying your location (latitude and longitude), you can obtain the correction value. Refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	MAGVAR header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	type	AUTO	0	Use IGRF corrections	Enum	4	H
		CORRECTION	1	Use the correction supplied			
3	correction	± 180.0 degrees		Magnitude of correction (Required field if type = Correction)	Float	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	std_dev	± 180.0 degrees		Standard deviation of correction (default = 0)	Float	4	H+8

## 2.87 MARKCONTROL

### *Controls processing of mark inputs*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This command is used to control the processing of the mark inputs. Using this command, the mark inputs can be enabled or disabled, polarity can be changed and a time offset and guard against extraneous pulses can be added.

The MARKxPOS and MARKxTIME logs have their outputs (and extrapolated time tags) pushed into the future (relative to the mark input (MKI) event) by the amount entered into the time bias field. In almost all cases, this value is set to 0, which is also the default setting (see *MARKPOS*, *MARK2POS*, *MARK3POS* and *MARK4POS* on page 696 and *MARK1TIME*, *MARK2TIME*, *MARK3TIME* and *MARK4TIME* on page 699).

**Message ID: 614**

#### Abbreviated ASCII Syntax:

```
MARKCONTROL signal [switch [polarity [timebias [timeguard]]]]
```

#### Factory Default:

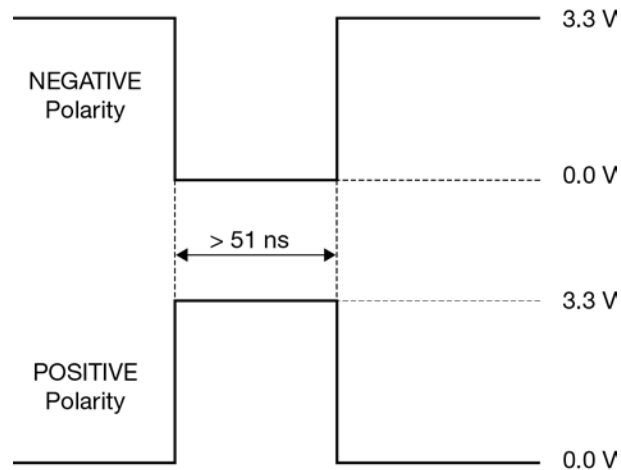
```
MARKCONTROL MARK1 ENABLE
```

```
MARKCONTROL MARK2 ENABLE
```

#### ASCII Example:

```
MARKCONTROL MARK1 ENABLE NEGATIVE 50 100
```

**Figure 7: TTL Pulse Polarity**



If using an external device, such as a camera, connect the device to the receiver's I/O port. Use a cable that is compatible with both the receiver and the device. A MARKIN pulse can be a trigger from the device to the receiver. See also the **MARKPOS**, **MARK2POS**, **MARK3POS** and **MARK4POS** command on page 696 and the **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** command on page 699.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	MARKCONTROL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	signal	MARK1	0	Specifies which mark input the command should be applied to. Set to MARK1 for the Event1 input, MARK2 for Event2, MARK3 for Event3 and MARK4 for Event4. All of the mark inputs have 10 K pull-up resistors to 3.3 V and are leading edge triggered.  MARK3 and MARK4 are available only on the OEM7600, OEM7700 and OEM7720.	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				
3	switch	DISABLE	0	Disables or enables processing of the mark input signal for the input specified. If DISABLE is selected, the mark input signal is ignored. (default = ENABLE)	Enum	4	H+4
		ENABLE	1				
4	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be received on the mark input. See <i>Figure 7: TTL Pulse Polarity</i> on the previous page for more information. (default= NEGATIVE)	Enum	4	H+8
		POSITIVE	1				
5	timebias	Any valid long value		Optional value to specify an offset, in nanoseconds, to be applied to the time the mark input pulse occurs. (default =0)	Long	4	H+12

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	timeguard			<p>Optional field to specify a time period, in milliseconds, during which subsequent pulses after an initial pulse are ignored.</p> <p>See Receiver Technical Specifications on the OEM7 Documentation Portal (<a href="http://docs.novatel.com/OEM7">docs.novatel.com/OEM7</a>) for the maximum raw measurement rate to determine the minimum period. If the value entered is lower than the minimum measurement period, the value is ignored and the minimum period is used.</p>	Ulong	4	H+16

## 2.88 MEDIAFORMAT

### *Format the media for PwrPak7*

**Platform:** PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this command.

Formats the specified media as FAT32, using enclosure specific cluster size and other parameters.



Only the internal flash memory can be formatted.



Entering this command results in complete loss of all data stored on the media.

Entering this command initiates the format operation. An error is reported if formatting could not be initiated, for example due to the media being disconnected.

Formatting progress can be observed using the **FILESYSTEMSTATUS** log on page 564, which will report *BUSY*, followed by *MOUNTED*.



The receiver may reboot in the process.

**Message ID: 2128**

**Abbreviated ASCII Syntax:**

```
MEDIAFORMAT MassStorage
```

**Example:**

```
MEDIAFORMAT INTERNAL_FLASH
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	MEDIAFORMAT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	MassStorage	INTERNAL_ FLASH	4	Format the internal memory in the enclosure.	Enum	4	H

## 2.89 MODEL

### *Switches to a previously authorized model*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to switch the receiver between models previously added with the **AUTH** command (see page 84). When the **MODEL** command is issued, the receiver saves the specified model as the active model. The active model is now used on every subsequent start up. The **MODEL** command causes an automatic reset.

Use the **VALIDMODELS** log (see page 979) to output a list of available models on the receiver. Use the **VERSION** log (see page 984) to output the active model. Use the **AUTHCODES** log (see page 491) to output a list of the auth codes present on the receiver.



If the **MODEL** command is used to switch to an expired model, the receiver will reset and enter into an error state. Switch to a valid model to continue.

**Message ID:** 22

**Abbreviated ASCII Syntax:**

```
MODEL model
```

**Input Example:**

```
MODEL DDNRNNTBN
```



NovAtel uses the term models to refer to and control different levels of functionality in the receiver firmware. For example, a receiver may be purchased with an L1 only capability and be easily upgraded at a later time to a more feature intensive model, like L1/L2 dual-frequency. All that is required to upgrade is an authorization code for the higher model and the **AUTH** command (see page 84). Reloading the firmware or returning the receiver for service to upgrade the model is not required. Upgrades are available from [novatel.com/support](http://novatel.com/support).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	MODEL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

---

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	model	Max 16 character null-terminated string (including the null)		Model name	String [max 16]	Variable <sup>1</sup>	H

---

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.90 MOVINGBASESTATION

### *Enables the use of a moving base station*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to enable or disable a receiver from transmitting corrections without a fixed position.

The moving base function allows you to obtain a centimetre level xyz baseline estimate when the base station and possibly the rover are moving. It is very similar to normal RTK, with one stationary base station and a moving rover (refer to Transmitting and Receiving Corrections in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#)). The BSLNXYZ log is an asynchronous 'matched' log that can be logged with the onchanged trigger to provide an accurate baseline between the base and rover.

Due to the latency of the reference station position messages, the following logs are not recommended to be used when in moving baseline mode: BESTXYZ, GPGST, MARKPOS, MARK2POS, MATCHEDPOS, MATCHEDXYZ, RTKPOS and RTKXYZ. The position error of these logs could exceed 100 m, depending on the latency of the reference station position message. If a rover position is required during moving basestation mode, then PSRPOS is recommended.

The **MOVINGBASESTATION** command must be used to allow the base to transmit messages without a fixed position.



1. Use the PSRPOS position log at the rover. It provides the best accuracy and standard deviations when the MOVINGBASESTATION mode is enabled.
2. This command supports RTCM V3 operation.
3. RTCM V3 support includes GPS + GLONASS operation.

**Message ID: 763**

#### **Abbreviated ASCII Syntax:**

```
MOVINGBASESTATION switch
```

#### **Factory Default:**

```
MOVINGBASESTATION disable
```

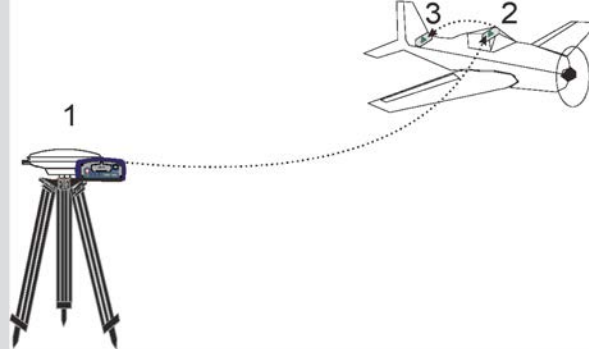
#### **ASCII Example:**

```
MOVINGBASESTATION ENABLE
```



Consider the case where there is a fixed base, an airplane flying with a moving base station near its front and a rover station at its tail end. Corrections can be sent between the receivers in a 'daisy chain' effect, where the fixed base station sends corrections to the moving base station, which in turn can send corrections to the rover.

**Figure 8: Moving Base Station 'Daisy Chain' Effect**



When using this method, the position type is only checked at the fixed base station. Moving base stations will continue to operate under any conditions.

This command is useful for moving base stations doing RTK positioning at sea. A rover station is used to map out local areas (for marking shipping lanes, hydrographic surveying and so on), while the base station resides on the control ship. The control ship may not move much (parked at sea), but there is a certain amount of movement due to the fact that it is floating in the ocean. By using the **MOVINGBASESTATION** command, the control ship is able to use RTK positioning and move to new survey sites.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	MOVINGBASESTATION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Do not transmit corrections without a fixed position.	Enum	4	H
		ENABLE	1	Transmit corrections without a fixed position.			

## 2.91 NAVICECUTOFF

### *Sets elevation cut-off angle for NavIC satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to set the elevation cut-off angle for tracked NavIC satellites. The receiver does not start automatically searching for a NavIC satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the NAVICECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle; it could be used in these situations:

- The antenna is at a high altitude, and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **NAVICECUTOFF** command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command on page 156 to set the cut-off angle for all other systems.



For dual antenna receivers, this command applies to both the primary and secondary antennas.

**Message ID:** 2134

**Abbreviated ASCII Syntax:**

```
NAVICECUTOFF angle
```

**Factory Default:**

```
NAVICECUTOFF 5.0
```

**ASCII Example:**

```
NAVICECUTOFF 10.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	NAVICECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.92 NMEABEIDOUTALKER

### *Sets the NMEA BeiDou talker ID*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to alter the behavior of the NMEA BeiDou talker ID. The default behavior is to use the legacy BeiDou Talker ID of 'BD' to maintain backward compatibility.

**Message ID: 2258**

#### **Abbreviated ASCII Syntax:**

```
NMEABEIDOUTALKER id
```

#### **Factory Default:**

```
NMEABEIDOUTALKER BD
```

#### **ASCII Example:**

```
NMEABEIDOUTALKER GB
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	NMEABEIDOUTALKER header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	id	GB	0	Use the NMEA ID	Enum	4	H
		BD	1	Use the legacy ID			



## 2.93 NMEAFORMAT

### *Customize NMEA output*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use the NMEAFORMAT command to customize the NMEA GPGGA and GPGGALONG output.



Modifying the NMEA output will make it not compliant with the NMEA standard.

**Message ID: 1861**

#### **Abbreviated ASCII Syntax:**

```
NMEAFORMAT field format
```

#### **Factory Default:**

```
NMEAFORMAT GGA_LATITUDE 9.4
NMEAFORMAT GGA_LONGITUDE 10.4
NMEAFORMAT GGA_ALTITUDE .2
NMEAFORMAT GGALONG_LATITUDE 12.7
NMEAFORMAT GGALONG_LONGITUDE 13.7
NMEAFORMAT GGALONG_ALTITUDE .3
```

#### **Example:**

The following settings increase the precision of the GPGGA latitude and longitude fields:

```
NMEAFORMAT GGA_LATITUDE 11.6
NMEAFORMAT GGA_LONGITUDE 12.6
```

The following settings decrease the precision of the GPGGALONG latitude and longitude fields:

```
NMEAFORMAT GGALONG_LATITUDE 11.6
NMEAFORMAT GGALONG_LONGITUDE 12.6
```

The following setting stops the undulation fields of the GPGGALONG log being filled, making a log like the GPGGARTK log that was in NovAtel's OEM6 firmware:

```
NMEAFORMAT GGALONG_UNDULATION !0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	NMEAFORMAT header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Field	GGA_ LATITUDE	0	GPGGA latitude field	Enum	4	H
		GGA_ LONGITUDE	1	GPGGA longitude field			
		GGA_ ALTITUDE	2	GPGGA altitude (height) field			
		GGA_ UNDULATION	3	GPGGA undulation field			
		GGALONG_ LATITUDE	10	GPGGALONG latitude field			
		GGALONG_ LONGITUDE	11	GPGGALONG longitude field			
		GGALONG_ ALTITUDE	12	GPGGALONG altitude (height) field			
		GGALONG_ UNDULATION	13	GPGGALONG undulation field			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	Format	Char[8]		<p>The Format field has a syntax similar to the printf function commonly found in programming languages. The format is:</p> <p style="padding-left: 40px;">!x.y</p> <p>Where:</p> <p><b>y</b> is the number of digits to display after the decimal point</p> <p><b>x</b> sets the minimum field width including the decimal point. X is optional if ! is not used. If the value requires fewer digits than x, leading zeros are added to the output.</p> <p><b>!</b> forces the field width to x. ! is optional. If a value exceeds the permitted width, the value will be saturated. If ! is used, y must be less than x.</p> <p>Examples (GGA_LATITUDE):</p> <p>.5 = 5106.98120</p> <p>2.3 = 5106.981</p> <p>7.1 = 05107.0</p> <p>!7.2 = 5106.98</p> <p>!7.3 = 999.999</p>	Char[8]	8	H+4

## 2.94 NMEATALKER

### *Sets the NMEA talker ID*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to alter the behavior of the NMEA talker ID. The talker is the first 2 characters after the \$ sign (e.g. \$XXGST) in the log header of the NMEA logs. The logs affected by the **NMEATALKER** command are GPHDT, GPGLL, GPGRS, GPGSA, GPGST, GPGSV, GPRMC, and GPVTG.



On SPAN systems, the GPGGA position is always based on the position solution from the BESTPOS log which incorporate GNSS + INS solutions as well.

When **NMEATALKER** is set to **GP**, the NMEA messages include information only about the GPS satellites that have a 'GP' talker solution, even when the receiver is tracking multi-constellation and with SPAN enabled.

When **NMEATALKER** is set to **AUTO**, the NMEA Talker ID changes as per the constellations enabled on the receiver. When there is one constellation in the solution and SPAN is disabled, the NMEA Talker ID is output as: **GP** for GPS, **GL** for GLONASS, **GA** for Galileo, and **GB/BD** for BeiDou (see **NMEABEIDOUTALKER** command on page 272). If more than one constellation is in the solution, the NMEA Talker ID is output as **GN** with the exception of GPGSV.



With SPAN enabled, the logs change their IDs to **IN** in single and multi-constellation use cases, except the GPGSA, GPGRS, GPHDT logs, which report a **GN** talker ID, and GPGSV.



For the **GPGSV** message with a single constellation in the solution, the output is its respective talker ID. When it is a multi-constellation solution, multiple sentences are output with their respective talker IDs. For example, if both GPS and GLONASS satellites are in the solution, two different sentences with **GP** and **GL** are output.

**Message ID: 861**

**Abbreviated ASCII Syntax:**

```
NMEATALKER id
```

**Factory Default:**

```
NMEATALKER gp
```

**ASCII Example:**

```
NMEATALKER auto
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	NMEATALKER header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	ID	GP	0	GPS (GP) only	Enum	4	H
		AUTO	1	GPS, GLONASS, Galileo, BeiDou, NavIC and/or QZSS			



The NMEATALKER command only affects NMEA logs that are capable of a GPS output. For example, GLMLA is a GLONASS-only log and the output will always use the GL talker. *Table 58: NMEA Talkers* below shows the NMEA logs and whether they use GPS (GP), GLONASS (GL), Galileo (GA), BeiDou (GB/BD), NavIC (GI), QZSS (GQ) or combined (GN) talkers with NMEATALKER AUTO.

**Table 58: NMEA Talkers**

Log	Talker IDs
GLMLA	GL
GPALM	GP
GPGGA	GP
GPGLL	GA or GB/BD or GI or GL or GN or GP or GQ
GPGRS	GA or GB/BD or GI or GL or GN or GP or GQ
GPGSA	GA or GB/BD or GI or GL or GN or GP or GQ
GPGST	GA or GB/BD or GI or GL or GN or GP or GQ
GPGSV	GA or GB/BD or GI or GL or GP or GQ
GPRMB	GA or GB/BD or GI or GL or GN or GP or GQ
GPRMC	GA or GB/BD or GI or GL or GN or GP or GQ
GPVTG	GA or GB/BD or GI or GL or GN or GP or GQ
GPZDA	GP



Which BeiDou NMEA Talker ID, GB or BD, is used in NMEA logs is determined by the setting in the **NMEABEIDOUTALKER** command on page 272.

## 2.95 NMEAVERSION

### Sets the NMEA Version for Output

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to set the output version of NMEA messages.

**Message ID: 1574**

#### Abbreviated ASCII Syntax:

```
NMEAVERSION Version
```

#### Factory Defaults:

```
NMEAVERSION V31
```

#### ASCII Example:

```
NMEAVERSION V41
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	NMEAVERSION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Version	V31	0	NMEA messages will be output in NMEA version 3.01 format.	Enum	4	H
		V41	1	NMEA messages will be output in NMEA version 4.11 format.			

## 2.96 NTRIPCONFIG

### *Configures NTRIP*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command sets up and configures NTRIP communication.

**Message ID: 1249**

#### Abbreviated ASCII Syntax:

```
NTRIPCONFIG port type [protocol [endpoint [mountpoint [username [password
[bindinterface]]]]]]
```



Mountpoint, username and password are all set up on the caster.

#### Factory Default:

```
NTRIPCONFIG ncom1 disabled
NTRIPCONFIG ncom2 disabled
NTRIPCONFIG ncom3 disabled
NTRIPCONFIG ncomX disabled
```

#### ASCII Example:

```
NTRIPCONFIG ncom1 client v1 :2000 calg0
```

#### ASCII example (NTRIP client):

```
NTRIPCONFIG ncom1 client v2 192.168.1.100:2101 RTCM3 calgaryuser calgarypwd
```

#### ASCII example (NTRIP server):

```
NTRIPCONFIG ncom1 server v1 192.168.1.100:2101 RTCM3 "" casterpwd
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	NTRIPCONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	port	NCOM1	26	Name of the port see <i>Table 34: Communications Port Identifiers</i> on page 152	Enum	4	H
		NCOM2	27				
		NCOM3	28				
3	type	DISABLED	1	NTRIP type	Enum	4	H+4
		CLIENT	2				
		SERVER	3				
4	protocol	DISABLED	1	Protocol (default V1)	Enum	4	H+8
		V1	2				
		V2	3				
5	endpoint	Max 80 character string		Endpoint to wait on or to connect to where host is a hostname or IP address and port is the TCP/UDP port number (default = 80)	String [80]	variable <sup>1</sup>	H+12
6	mountpoint	Max 80 character string		Which mount point to use	String [80]	variable <sup>1</sup>	variable
7	user name	Max 30 character string		Login user name	String [30]	variable <sup>1</sup>	variable
8	password	Max 30 character string		Password	String [30]	variable <sup>1</sup>	variable
9	bindInterface	ALL (default)	1	Not supported. Set to <i>ALL</i> for future compatibility.	Enum	4	variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.



## 2.97 NTRIPSOURCETABLE

### Set NTRIP caster endpoints

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command is used to set the NTRIP caster endpoints to be used for the **SOURCETABLE** log (see page 954).

**Message ID:** 1343

#### Abbreviated ASCII Syntax:

```
NTRIPSOURCETABLE endpoint [reserved1] [reserved2]
```

#### Factory Default:

```
NTRIPSOURCETABLE none
```

#### ASCII Example:

```
NTRIPSOURCETABLE base.station.novatel.ca:1001
```

```
NTRIPSOURCETABLE 192.168.1.100:1001
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-		NTRIPSOURCETABLE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Endpoint	Max 80 character string		Endpoint, in format of host:port, to connect to where the host is a hostname or IP address and port is the TCP/IP port number	String[80]	variable <sup>1</sup>	H
3	Reserved1	Reserved		Reserved	Ulong	4	variable
4	Reserved2	Reserved		Reserved	Ulong	4	variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.98 NVMRESTORE

### *Restores NVM data after an NVM failure*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to restore Non-Volatile Memory (NVM) data after a NVM Fail error. This failure is indicated by bit 15 of the receiver error word being set (see also **RXSTATUS** command on page 878 and **RXSTATUSEVENT** command on page 891). If corrupt NVM data is detected, the receiver remains in the error state and continues to flash an error code on the Status LED until the **NVMRESTORE** command is issued (refer to Built-In Status Tests in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for further explanation).

After entering the **NVMRESTORE** command and resetting the receiver, the communications link may have to be reestablished at a different baud rate from the previous connection.

**Message ID: 197**

#### **Abbreviated ASCII Syntax:**

NVMRESTORE



The possibility of NVM failure is extremely remote, however, if it should occur it is likely that only a small part of the data is corrupt. This command is used to remove the corrupt data and restore the receiver to an operational state. The data lost could be the user configuration, almanac, model or other reserved information.

## 2.99 NVMUSERDATA

### *Write User Data to NVM*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command writes the data provided in the data array to NVM. This data can be retrieved by issuing the command **LOG NVMUSERDATA**.

The user data is maintained through power cycles and a standard **FRESET** command (see page 193). To clear the user data, use the **FRESET USERDATA** command.



The user data may be deleted if the **NVMRESTORE** command (see page 282) is sent. **NVMRESTORE** should be used with caution and is meant for use only in the event of a NVM receiver error.

**Message ID:** 1970

#### **Abbreviated ASCII Syntax:**

NVMUSERDATA #bytes data

Field	Field Type	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	NVMUSERDATA header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	#bytes	-	Number of bytes of data to follow	Ulong	4	H
3	data	-	User input data up to a maximum of 2000 bytes. Data is entered in hexadecimal values with no separators between the values. For example, 1a2b3c4e	Uchar	2000	H+4

## 2.100 OUTPUTDATUM

### *Choose the datum to output*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to select the geodetic datum and co-ordinate epoch in which solutions are output. If not set, the factory default value is the service datum and service epoch.

For either a plate-fixed or earth-fixed output datum, using a fixed epoch will compensate for plate motion such that you get repeatable co-ordinates over time for a location. Compensation is controlled using the **TECTONICSCOMPENSATIONSOURCE** command on page 423 and reported by the **TECTONICSCOMPENSATION** log (see page 957). The **TECTONICSCOMPENSATION** log will warn when operating in a region where tectonics compensation may be imprecise.

In order for solutions to be transformed to the requested datum, the datum selected must be in the list of datums shown in the **GEODETICDATUMS** log (see page 588).

To view the current datum, use the **BESTDATUMINFO** log (see page 517).

**Message ID: 1144**

#### **Abbreviated ASCII Syntax:**

```
OUTPUTDATUM name epoch_option epoch
```

#### **Factory Default:**

```
OUTPUTDATUM SERVICE SERVICE_EPOCH 0
```

#### **ASCII Example:**

To output co-ordinates in NAD83(CSRs), 2018:

```
OUTPUTDATUM NAD83(CSRs) FIXED_EPOCH 2018.000
```

To output co-ordinates in ITRF2014, 2010.00:

```
OUTPUTDATUM ITRF2014 FIXED_EPOCH 2010.000
```



The **OUTPUTDATUM** command only accepts a non-zero epoch if the **FIXED\_EPOCH** option is specified. The earliest fixed epoch is 1900.



Do not mix the **OUTPUTDATUM** command and the **DATUM USER** commands.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	OUTPUTDATUM header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	name			The datum output.	Char[32]	variable	H
3	epoch_option	See <i>Table 59: Epoch Options</i> below		Epoch type to enter coordinates in.	Enum	4	variable
4	epoch	0 = unknown (default), else >=1900.00		Co-ordinate epoch (decimal year). Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable

Table 59: Epoch Options

Binary	ASCII	Description
0	SERVICE_EPOCH	Output coordinates in the epoch of the corrections (default)
1	FIXED_EPOCH	Output coordinates in the specified epoch
2	CURRENT_EPOCH	Output coordinates in the current epoch

## 2.101 PDPFILTER

### *Enables, disables or resets the PDP filter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to enable, disable or reset the Pseudorange/Delta-Phase (PDP) filter. The main advantages of the PDP implementation are:

- Smooths a jumpy position
- Bridges outages in satellite coverage (the solution is degraded from normal but there is at least a reasonable solution without gaps)



Enable the PDP filter to output the PDP solution in the **BESTPOS** log (see page 520), **BESTVEL** log (see page 535) and *NMEA Standard Logs* on page 728.

Refer to PDP and GLIDE Configurations in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for information on configuring your receiver for PDP or GLIDE operation.

### 2.101.1 GLIDE Position Filter

GLIDE is a mode of the PDP filter that optimizes the position for consistency over time rather than absolute accuracy. This is ideal in clear sky conditions where the user needs a tight, smooth and consistent output. The GLIDE filter works best with SBAS. The PDP filter is smoother than a least squares solution but is still noisy in places. The GLIDE filter produces a very smooth solution with relative rather than absolute position accuracy. There should typically be less than 1 centimetre difference in error from epoch to epoch. GLIDE also works in single point and DGPS modes. See also the **PDPMODE** command on page 288 and the **PDPPOS** log on page 748, **PDPVEL** log on page 752 and **PDPXYZ** log on page 753.



Refer also to the application note [APN-038: Pseudorange/Delta-Phase \(PDP\) and GLIDE Filters](#), available at [novatel.com/support/support-materials/application-notes](http://novatel.com/support/support-materials/application-notes).

**Message ID:** 424

**Abbreviated ASCII Syntax:**

```
PDPFILTER switch
```

**Factory Default:**

```
PDPFILTER disable
```

**ASCII Example:**

```
PDPFILTER enable
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PDPFILTER header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Disable the PDP filter.	Enum	4	H
		ENABLE	1	Enable the PDP filter.			
		RESET	2	Reset the PDP filter. A reset clears the filter memory so that the PDP filter can start over.			

## 2.102 PDPMODE

### *Selects the PDP mode and dynamics*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to select the mode and dynamics of the PDP filter.



The **PDPFILTER ENABLE** command (see the **PDPFILTER** command on page 286) must be entered before the **PDPMODE** command.

It is recommended that the ionotype be left at AUTO when using either normal mode PDP or GLIDE. See also the **SETIONOTYPE** command on page 397.

**Message ID:** 970

#### Abbreviated ASCII Syntax:

```
PDPMODE mode dynamics
```

#### Factory Default:

```
PDPMODE normal auto
```

#### ASCII Example:

```
PDPMODE relative dynamic
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PDPMODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	NORMAL	0	In RELATIVE or GLIDE mode, GLIDE performance is optimized to obtain a consistent error in latitude and longitude over time periods of 15 minutes or less, rather than to obtain the smallest absolute position error. See also <i>GLIDE Position Filter</i> on page 286 for GLIDE mode additional information.  GLIDE mode is the same as RELATIVE mode.	Enum	4	H
		RELATIVE	1				
		GLIDE	3				
3	dynamics	AUTO	0	Auto detect dynamics mode	Enum	4	H+4
		STATIC	1	Static mode			
		DYNAMIC	2	Dynamic mode			

## 2.103 PGNCONFIG

### Configure NMEA2000 PGNs.

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to configure the PGNs of the proprietary NMEA 2000 fast-packet messages the OEM7 receivers produce.

The receiver must be reset after issuing a **SAVECONFIG** command (see page 366) for all the configuration changes to take affect.

**Message ID: 1892**

#### Abbreviated ASCII Syntax:

```
PGNCONFIG message_id pgn priority
```

#### Factory Default:

```
PGNCONFIG INSPVACMP 130816 7
PGNCONFIG INSPVASDCMP 130817 7
```

#### ASCII Example:

```
PGNCONFIG INSPVACMP 129500 3
```

This example sets the INSPVACMP message to PGN 129500 with priority 3.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PGNCONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	message_id	INSPVACMP	1889	NovAtel message ID	Ulong	4	H
		INSPVASDCMP	1890				
3	pgn	0 to 4294967295		PGN to use for message_id	Ulong	4	H+4
4	priority	0 - 7		CAN priority to use	Uchar	1	H+8

## 2.104 \$PMDT

### Configure Terrain Compensation

**Platform:** SMART7-I, SMART7-W, SMART2-TB

Use this NMEA style command to configure the Terrain Compensation feature. The Terrain Compensation filter must be enabled using the **TILTCOMPENSATIONCONTROL** command (see page 429) for the Terrain Compensation feature to function.



The maximum antenna height allowed is 500 inches (12.7 metres).



The **TILTFILTER** command (see page 430) and **TILTZERO** command (see page 432) can also be used to configure the Terrain Compensation filter. Either method will successfully configure the Terrain Compensation feature, however only the single-stage zeroing procedure is supported with the **\$PMDT** command. For best performance, two-stage zeroing using the **TILTZERO** command (see page 432) is recommended.

**Message ID: 2308**




#### Syntax

```
$PMDT, a, x, x, x, x, x.x*hh [CR] [LF]
```

#### ASCII Example:

```
$PMDT, u, , 1, 1*78
```

Field	Structure	Description	Format	Symbol	Example
1	\$PMDT	Command header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	\$PMDT	\$PMDT
2	function	The function to perform, where: l = Set level state u = Set/Get antenna height x = Save level and height values r = Clear the saved level and height values Required field.	Char	a	,u
3	reserved	Reserved field, leave empty.	–	x	,

Field	Structure	Description	Format	Symbol	Example
4	antenna height feet	Antenna height in feet. Will be combined with the <i>antenna height inches</i> field.   Cannot be combined with the <i>antenna height metres</i> field.	Integer	x	,1
5	antenna height inches	Antenna height in inches. Will be combined with the <i>antenna height feet</i> field.   Cannot be combined with the <i>antenna height metres</i> field.	Integer	x	,1
6	antenna height metres	Antenna height in metres.   Cannot be combined with the <i>antenna height feet</i> and <i>antenna height inches</i> fields.	Float	x.x	,2.2
7	*hh	NMEA check sum.		*hh	*78
8	[CR][LF]	Sentence terminator.			[CR][LF]



For more information about configuring the Terrain Compensation feature using the \$PMDT command, see Configuring Terrain Compensation using NMEA Command in the [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

## 2.105 POSAVE

### *Implements base station position averaging*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command implements position averaging for base stations. Position averaging continues for a specified number of hours or until the estimated averaged position error is within specified accuracy limits. Averaging stops when the time limit or the horizontal standard deviation limit or the vertical standard deviation limit is achieved. When averaging is complete, the **FIX POSITION** command is automatically invoked. See the **FIX** command on page 182.

If differential logging is initiated, then issue the **POSAVE** command followed by the **SAVECONFIG** command (see page 366). The receiver averages positions after every power on or reset. It then invokes the **FIX POSITION** command to enable it to send differential corrections.

**Message ID:** 173

#### Abbreviated ASCII Syntax:

```
POSAVE state [maxtime [maxhstd [maxvstd]]]
```

#### Factory Default:

```
POSAVE off
```

#### ASCII Example 1:

```
POSAVE on 24 1 2
```

#### ASCII Example 2:

```
POSAVE OFF
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	POSAVE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	state	ON	1	Enable position averaging	Enum	4	H
		OFF	0	Disable position averaging			
3	maxtime	0.01 - 100 hours		Maximum amount of time that positions are to be averaged (default=0.01)	Float	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	maxhstd	0 - 100 m		Desired horizontal standard deviation (default = 0.0)	Float	4	H+8
5	maxvstd	0 - 100 m		Desired vertical standard deviation (default = 0.0)	Float	4	H+12



The **POSAVE** command can be used to establish a new base station, in any form of survey or RTK data collection, by occupying a site and averaging the position until either a certain amount of time has passed or position accuracy has reached a user-specified level. User-specified requirements can be based on time or horizontal or vertical quality of precision.

## 2.106 POSTIMEOUT

### *Sets the position time out*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the time out value for the position calculation in seconds.

In position logs, for example **BESTPOS** log (see page 520) or **PSRPOS** log (see page 774), when the position time out expires, the Position Type field is set to NONE. Other field values in these logs remain populated with the last available position data. Also, the position is no longer used in conjunction with the almanac to determine what satellites are visible.

**Message ID:** 612

#### Abbreviated ASCII Syntax:

```
POSTIMEOUT sec
```

#### Factory Default:

```
POSTIMEOUT 600
```

#### ASCII Example:

```
POSTIMEOUT 1200
```



When performing data collection in a highly dynamic environment (for example, urban canyons or in high speed operations), you can use **POSTIMEOUT** to prevent the receiver from outputting calculated positions that are too old. Use **POSTIMEOUT** to force the receiver position type to NONE. This ensures that the position information being used in the **BESTPOS** log (see page 520) or **PSRPOS** log (see page 774) is based on a recent calculation. All position calculations are then recalculated using the most recent satellite information.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	POSTIMEOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	sec	0-86400		Time out in seconds	Ulong	4	H

## 2.107 PPPBASICCONVERGEDCRITERIA

### *Configures decision for PPP Basic convergence*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **PPPBASICCONVERGEDCRITERIA** command sets the threshold that determines if the solution has converged for lower accuracy PPP solutions. These are the PPP solutions reported with the PPP\_BASIC and PPP\_BASIC\_CONVERGING position types.



The convergence threshold for high-accuracy PPP solutions (reported with PPP and PPP\_CONVERGING position types) is set using the **PPPCONVERGEDCRITERIA** command (see page 298).



Relaxing the convergence threshold shortens the time before a PPP solution is reported as converged. However, it does not alter solution behavior. During the initial PPP solution period, the positions can have decimetre error variation. Only relax the convergence threshold if the application can tolerate higher solution variability.

**Message ID: 1949**

#### **Abbreviated ASCII Syntax:**

```
PPPBASICCONVERGEDCRITERIA criteria tolerance
```

#### **Factory Default:**

```
PPPBASICCONVERGEDCRITERIA horizontal_stddev 0.60
```

#### **ASCII Example:**

```
PPPBASICCONVERGEDCRITERIA total_stddev 0.45
```



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPBASIC CONVERGED CRITERIA header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Criteria	TOTAL_STDDEV	1	Use the total, 3D, standard deviation	Enum	4	H
		HORIZONTAL_STDDEV	2	Use the horizontal, 2D, standard deviation			
3	Tolerance			Tolerance (m)	Float	4	H+4

## 2.108 PPPCONVERGEDCRITERIA

### *Configures decision for PPP convergence*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **PPPCONVERGEDCRITERIA** command sets the threshold that determines if the solution has converged for high-accuracy PPP solutions. These are the PPP solutions reported with the PPP and PPP\_ CONVERGING position types.



The convergence threshold for lower accuracy PPP solutions (reported with PPP\_BASIC and PPP\_BASIC\_CONVERGING position types) is set using the **PPPBASICCONVERGEDCRITERIA** command (see page 296).



Relaxing the convergence threshold shortens the time before a PPP solution is reported as converged. However, it does not alter solution behavior. During the initial PPP solution period, the positions can have decimetre error variation. Only relax the convergence threshold if the application can tolerate higher solution variability.

**Message ID: 1566**

#### **Abbreviated ASCII Syntax:**

```
PPPCONVERGEDCRITERIA criteria tolerance
```

#### **Factory Default:**

```
PPPCONVERGEDCRITERIA horizontal_stddev 0.32
```

#### **ASCII Example:**

```
PPPCONVERGEDCRITERIA total_stddev 0.15
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPCONVERGED CRITERIA header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	Criteria	TOTAL_ STDDEV	1	Use the total, 3D, standard deviation	Enum	4	H
		HORIZONTAL_ STDDEV	2	Use the horizontal, 2D, standard deviation			
3	Tolerance			Tolerance (m)	Float	4	H+4

## 2.109 PPPDYNAMICS

### *Sets the PPP dynamics mode*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command configures the dynamics assumed by the PPP filter. AUTO detects the antenna dynamics and adapts filter operation accordingly.



The automatic dynamics detection may be fooled by very slow, “creeping” motion, where the antenna consistently moves less than 2 cm/s. In such cases, the mode should explicitly be set to DYNAMIC.

**Message ID: 1551**

#### Abbreviated ASCII Syntax:

```
PPPDYNAMICS mode
```

#### Factory Default:

```
PPPDYNAMICS dynamic
```

#### ASCII Example:

```
PPPDYNAMICS auto
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPDYNAMICS header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Mode	AUTO	0	Automatically determines dynamics mode	Enum	4	H
		STATIC	1	Static mode			
		DYNAMIC	2	Dynamic mode			

## 2.110 PPPDYNAMICSEED

### *Seed the PPP filter in any platform motion state*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command enables seeding of the PPP engine regardless of the receiver motion state. Accurate seeds can be used to improve initial PPP convergence and re-convergence following signal outages.

The seed position given by the **PPPDYNAMICSEED** command must be in a datum consistent with the PPP corrections that are in use. For TerraStar corrections, the datum is ITRF2014. The dynamic seed's time must refer to receiver time and cannot be more than 15 seconds in the past. A valid PPP solution (the **PPPPOS** log (see page 758) solution status is SOL\_COMPUTED) must have been computed for the same epoch as the seed in order for the seed to be used.

See the **PPPSEED** command on page 304 for stationary-only seeding and for other control over seeding.

**Message ID: 2071**

#### Abbreviated ASCII Syntax:

```
PPPDYNAMICSEED week seconds latitude longitude height northing_std_dev
easting_std_dev height_std_dev [northing_easting_covariance] [northing_
height_covariance] [easting_height_covariance]
```

#### Example :

```
PPPDYNAMICSEED 1817 247603 51.2086442297 -113.9810263055 1071.859 0.02 0.02
0.04
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPDYNAMICSEED header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).		H	0
2	week	0-9999		GPS Week number	Ulong	4	H
3	seconds	0-604800		Number of seconds into GPS week	Ulong	4	H+4
4	latitude	±90		Latitude (degrees)	Double	8	H+8
5	longitude	±180		Longitude (degrees)	Double	8	H+16
6	height	> -2000.0		Ellipsoidal height (metres)	Double	8	H+24

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	northing_std_dev			Northing standard deviation (metres)	Float	4	H+32
8	easting_std_dev			Easting standard deviation (metres)	Float	4	H+36
9	height_std_dev			Ellipsoidal height standard deviation (metres)	Float	4	H+40
10	northing_easting_covariance			Covariance between northing and easting components (metres)	Float	4	H+44
11	northing_height_covariance			Covariance between northing and height components (metres)	Float	4	H+48
12	easting_height_covariance			Covariance between easting and height components (metres)	Float	4	H+52

## 2.111 PPPRESET

### *Reset the PPP filter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command resets the PPP filter. After a reset, the PPP filter is restored to its initial state and PPP convergence will start over.



If deletion of the NVM-saved PPP seed information is also required, then a **PPPSEED CLEAR** command must be applied before the PPPRESET command. See the **PPPSEED** command on the next page.

**Message ID:** 1542

#### Abbreviated ASCII Syntax:

```
PPPRESET [Option]
```

#### ASCII Example :

```
PPPRESET
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Bytes	Binary Format	Binary Offset
1	Command header	-	-	PPPRESET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Option	FILTER	1	Reset the PPP filter. (default = FILTER)	4	Enum	H
		ALL	10	Reset the PPP filter and Corrections			

## 2.112 PPPSEED

### *Control the seeding of the PPP filter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **PPPSEED** command controls the seeding of the PPP filter. Accurate position seeding can accelerate PPP convergence.

**PPPSEED SET** is used to explicitly specify a seed position. The seed position must be in a datum consistent with the PPP corrections that will be used. For TerraStar corrections, this is ITRF2014. The **PPPSEED SET** command can only be used to give seed positions for stationary platforms. If the platform is moving, use the **PPPDYNAMICSEED** command (see page 301).



Caution must be exercised when using **PPPSEED SET**. While a good seed position can accelerate convergence, a bad seed position hurts performance. In some cases, a bad seed can prevent a solution from ever converging to a correct position. In other cases, a bad seed might be rejected immediately. In still other cases, the filter might operate with it for a time period only to reject it later. In this case, the filter position is partially reset, with a corresponding discontinuity in the PPP position.

**PPPSEED STORE** and **RESTORE** are intended to simplify seeding in operations where the antenna does not move between power-down and power-up. For example, in agricultural operations a tractor might be stopped in a field at the end of a day and then re-started the next day in the same position. Before the receiver is powered-down, the current PPP position could be saved to NVM using the **PPPSEED STORE** command, and then that position applied as a seed after power-up using **PPPSEED RESTORE**.

**PPPSEED AUTO** automates the STORE and RESTORE process. When this option is used, the PPP filter automatically starts using the stopping position of the previous day. For this command to work, the **PPPDYNAMICS** command (see page 300) setting must be AUTO so that the receiver can determine when it is static, or the filter must explicitly be told it is static using **PPPDYNAMIC STATIC**. Additionally, in order for the receiver to recall the saved seed, the **PPPSEED AUTO** command should be saved to NVM using the **SAVECONFIG** command (see page 366).

**Message ID: 1544**

#### **Abbreviated ASCII Syntax:**

```
PPPSEED option [latitude] [longitude] [height] [northing_std_dev] [easting_std_dev] [height_std_dev]
```

#### **ASCII Example:**

```
PPPSEED set 51.11635322441 -114.03819311672 1064.5458 0.05 0.05 0.05
```



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPSEED header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	CLEAR	0	Resets the stored seed, and prevents any auto seeding from occurring.	Enum	4	H
		SET	1	Immediately apply the specified co-ordinates as a seed position.			
		STORE	2	Store the current PPP position in NVM for use as a future seed.			
		RESTORE	3	Retrieve and apply a seed position that was previously saved in NVM via the STORE or AUTO options.			
		AUTO	4	Automatically store and restore PPP seed positions.			
3	latitude	±90		Latitude (degrees)	Double	8	H+4
4	longitude	±180		Longitude (degrees)	Double	8	H+12
5	height	> -2000.0		Ellipsoidal height (metres)	Double	8	H+20
6	northing_std_dev			Northing standard deviation (metres)	Float	4	H+28
7	easting_std_dev			Easting standard deviation (metres)	Float	4	H+32

---

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
8	height_std_dev			Ellipsoidal height standard deviation (metres)	Float	4	H+36
9	Reserved				Float	4	H+40

## 2.113 PPPSOURCE

*Specifies the PPP correction source*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command determines what corrections the PPP filter can use.

**PPPSOURCE NONE** disables PPP output.

**Message ID:** 1707

**Abbreviated ASCII Syntax:**

```
PPPSOURCE source
```

**Factory Default:**

```
PPPSOURCE auto
```

**ASCII Example:**

```
PPPSOURCE none
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPSOURCE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	source	NONE	0	Disable PPP output	Enum	4	H
		TERRASTAR	1	Only accept TerraStar PPP corrections			
		VERIPOS	2	Only accept Veripos PPP corrections			
		AUTO	100	Automatically select and use the best corrections			

## 2.114 PPPTIMEOUT

### *Sets the maximum age of the PPP corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the maximum age of the corrections that can be used to generate a PPP solution.

By default, the delay is 360 seconds. If the received corrections are older than the delay time set, the receiver will not produce a PPP solution.

For example, if L-Band tracking (which is providing us the PPP corrections) is lost for some reason, the receiver will continue generating a PPP solution using last available corrections until the age of corrections is older than the delay time. Note that using older corrections degrades the PPP solution quality.

**Message ID: 1560**

#### **Abbreviated ASCII Syntax:**

```
PPPTIMEOUT delay
```

#### **Factory Default:**

```
PPPTIMEOUT 360
```

#### **ASCII Example:**

```
PPPTIMEOUT 120
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPPTIMEOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	delay	5 to 900 s		Maximum corrections age	Ulong	4	H

## 2.115 PPSCONTROL

### *Controls the PPS output*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command provides a method for controlling the polarity, period and pulse width of the PPS output on the OEM7. The PPS output can also be disabled using this command.



This command is used to setup the PPS signal coming from the receiver. For example, to take measurements such as temperature or pressure, in synch with your GNSS data, the PPS signal can be used to trigger measurements in other devices.

The leading edge of the 1 PPS pulse is always the trigger/reference. For example:

```
PPSPCONTROL ENABLE NEGATIVE
```

generates a normally high, active low pulse with the falling edge as the reference, while:

```
PPSPCONTROL ENABLE POSITIVE
```

generates a normally low, active high pulse with the rising edge as the reference.

The pulse width is user-adjustable. The adjustable pulse width feature supports triggers/systems that need longer, or shorter, pulse widths than the default to register the pulse enabling a type of GPIO line for manipulation of external hardware control lines.

The switch states allow more control over disabling/enabling the PPS. The ENABLE\_FINETIME switch prevents the PPS from being enabled until FINE or FINESTEERING time status has been reached. The ENABLE\_FINETIME\_MINUTEALIGN switch is similar to ENABLE\_FINETIME with caveat that the PPS will still not be enabled until the start of the next 60 seconds (a 1 minute modulus) after FINE or FINESTEERING time status has been reached.



If the value of a field shared with PPSPCONTROL2 is changed in PPSPCONTROL, the value of that field is also changed in PPSPCONTROL2. For example, if the polarity is changed using the PPSPCONTROL command, the polarity is also changed in PPSPCONTROL2 command.

**Message ID: 613**

**Abbreviated ASCII Syntax:**

```
PPSPCONTROL [switch [polarity [period [pulsewidth]]]]
```

**Factory Default:**

```
PPSPCONTROL enable negative 1.0 1000
```

**ASCII Example:**

```
PPSPCONTROL enable positive 0.5 2000
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPSCONTROL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Disable the PPS	Enum	4	H
		ENABLE	1	Enable the PPS (default)			
		ENABLE_FINETIME	2	Enable the PPS only when FINE or FINESTEERING time status has been reached			
		ENABLE_FINETIME_MINUTEALIGN	3	Enable the PPS only when FINE or FINESTEERING time status has been reached AND the start of the next 60 seconds (1 minute modulus) has occurred			
3	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be generated on the PPS output. See <i>Figure 7: TTL Pulse Polarity</i> on page 261 for more information (default=NEGATIVE)	Enum	4	H+4
		POSITIVE	1				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	period	0.05, 0.1, 0.2, 0.25, 0.5, 1.0, 2.0, 3.0,...20.0		Optional field to specify the period of the pulse, in seconds (default=1.0)	Double	8	H+8
5	pulsewidth	Any positive value less than or equal to half the period		Optional field to specify the pulse width of the PPS signal in microseconds. This value should always be less than or equal to half the period. (default=1000)	Ulong	4	H+16

## 2.116 PPSCONTROL2

**Controls polarity, period, pulse width and estimated error limit of the PPS output**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **PPSCONTROL2** command provides a method for controlling the polarity, period, pulse width, and estimated error limit of the PPS output on the OEM7. The PPS output can also be disabled using this command.

This command is identical to the **PPSCONTROL** command (see page 309) with the addition of a new parameter that represents the Estimated Error Limit.



If the value of a field shared with PPSCONTROL is changed in PPSCONTROL2, the value of that field is also changed in PPSCONTROL. For example, if the polarity is changed using the PPSCONTROL2 command, the polarity is also changed in PPSCONTROL command.

The estimated error limit sets an allowable  $\pm$  range for the clock offset. The PPS output is only enabled when the clock offset is within this range.

**Message ID: 1740**

**Abbreviated ASCII Syntax:**

```
PPSCONTROL2 [switch [polarity [period [pulsewidth [estimatederrorlimit]]]]]
```

**Factory default:**

```
PPSCONTROL2 enable negative 1.0 1000 0
```

**ASCII Example:**

```
PPSCONTROL2 enable_finetime positive 0.5 2000 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PPSCONTROL2 header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	switch	DISABLE	0	Disable the PPS	Enum	4	H
		ENABLE	1	Enable the PPS (default)			
		ENABLE_FINETIME	2	Enable the PPS only when FINE or FINESTEERING time status has been reached			
		ENABLE_FINETIME_MINUTEALIGN	3	Enable the PPS only when FINE or FINESTEERING time status has been reached AND the start of the next 60 seconds (1 minute modulus) has occurred			
3	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be generated on the PPS output. See <i>Figure 7: TTL Pulse Polarity</i> on page 261 for more information (default = NEGATIVE).	Enum	4	H+4
		POSITIVE	1				
4	period	0.05, 0.1, 0.2, 0.25, 0.5, 1.0, 2.0, 3.0,...20.0		Optional field to specify the period of the pulse in seconds (default = 1.0).	Double	8	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	pulse width	Any value less than or equal to half the pulse period in microseconds.		Optional field to specify the pulse width of the PPS signal in microseconds. This value should always be less than or equal to half the period. (default = 1000).	Ulong	4	H+16
6	estimated error limit	0 to 2147483647 in nanoseconds		Optional field to specify the $\pm$ estimated error limit (in nanoseconds) for the clock offset (default = 0). The PPS output is only enabled when the clock offset is within this limit.  An estimated error limit of 0 removes the estimated error limit restraint on the PPS.	Long	4	H+20

## 2.117 PROFILE

### *Profile in Non-Volatile Memory (NVM)*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to configure multiple profiles in the NVM at receiver startup. The output is in the **PROFILEINFO** log (see page 767).

**Message ID: 1411**

#### Abbreviated ASCII Syntax:

```
PROFILE Option Name [command]
```

#### ASCII Examples:

```
PROFILE create Base
PROFILE createelement Base "log com1 versiona"
PROFILE createelement Base "serialconfig com2 115200"
PROFILE createelement Base "log com2 rtcml004 ontime 1"
PROFILE activate Base
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PROFILE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Option	Refer to <i>Table 60: Profile Option</i> on the next page		Profile options	Enum	4	H
3	Name			Profile name	String [Max 20]	variable <sup>1</sup>	H+4
4	Command			Profile command	String [Max 200]	variable <sup>1</sup>	variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

**Restrictions:**

1. Only supports up to a maximum of 9 profiles.
2. Only supports up to a maximum of 20 commands per profile.
3. Only supports up to a maximum of 200 characters long for each command.
4. Only supports up to a maximum of 1500 characters for all commands in one profile.
5. If one of the profiles is activated, the **SAVECONFIG** functionality is disabled.
6. All profiles are deleted by a **FRESET PROFILEINFO** command (see the **FRESET** command on page 193).
7. The receiver resets after a profile is activated.
8. Some commands optionally accept a port parameter and will default to **THISPORT** if no port is provided (e.g. **LOG** command). Since the commands in a profile are not sent from a port **THISPORT** is undefined in this case. When adding such commands to a profile, be sure to specify the port for the command rather than letting the command use the default, which may result in incorrect behavior.
9. Commands that lead to a reset of the receiver are rejected by the **PROFILE** command (see page 315).

**Table 60: Profile Option**

Binary	ASCII	Description
0	Reserved	
1	CREATE	Create a profile
2	DELETE	Delete an existing profile
3	CREATEELEMENT	Create an element in an existing profile
4	DELETEELEMENT	Delete an existing element in an existing profile
5	ACTIVATE	Activate an existing profile
6	DEACTIVATE	Deactivate a running profile

## 2.118 PSRDIFFSOURCE

### *Sets the pseudorange differential correction source*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command determines what differential corrections the PSR and PDP (GLIDE) filters can use.

Differential corrections can be generated from RTK corrections. **PSRDIFFSOURCE RTK** enables this, allowing the PSR and PDP solutions to benefit from RTK corrections. An RTK-capable model is not required for this capability.

**PSRDIFFSOURCE NONE** stops any corrections from being used; the PSR and PDP solutions will revert to being uncorrected.



All types except NONE may revert to SBAS (if enabled) or SINGLE position types.

**Message ID:** 493

#### Abbreviated ASCII Syntax:

```
PSRDIFFSOURCE type [id]
```

#### Factory Default:

```
PSRDIFFSOURCE AUTO ANY
```

#### ASCII Examples:

1. Enable only SBAS:

```
PSRDIFFSOURCE SBAS
SBASCONTROL ENABLE AUTO
```

2. Enable differential from RTCM, with a fall-back to SBAS:

```
PSRDIFFSOURCE RTCM ANY
SBASCONTROL ENABLE AUTO
```

3. Disable all corrections:

```
PSRDIFFSOURCE NONE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PSRDIFFSOURCE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	type	See Table 61: <i>PSRDIFFSOURCE</i> Type below		Correction type	Enum	4	H
3	id	Char[4] or ANY		Base station ID (default = ANY)	Char[4]	8	H+4

Table 61: PSRDIFFSOURCE Type

Binary	ASCII	Description
0	RTCM	Use RTCM version 2 corrections from the given ID if available or SBAS otherwise. RTCM ID: $0 \leq \text{RTCM ID} \leq 1023$ or ANY to accept all IDs.
1	RTCA	Use RTCA corrections from the given ID if available or SBAS otherwise. RTCA ID: A four-character string containing only alpha (a-z) or numeric characters (0-9) or ANY to accept all IDs.
5	SBAS	Use SBAS corrections such as WAAS, EGNOS and MSAS in the region they are available and applicable to. There are regions that are not covered by any SBAS satellite and so the solution will revert to uncorrected. The ID is ignored for this correction type.
6	RTK	Use differential corrections that are generated from RTK corrections if available or SBAS otherwise. The correction type used is determined by the setting of the <b>RTKSOURCE</b> command (see page 348).
10	AUTO	Use any available correction source. If both differential and SBAS sources are available, differential will be preferred.
11	NONE	Disables all differential correction types. Solution will revert to uncorrected.

## 2.119 PSRDIFFSOURCE TIMEOUT

*Sets pseudorange differential correction source timeout*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

When multiple differential correction sources are available, this command allows the user to set a time in seconds, that the receiver will wait before switching to another differential source, if corrections from the original source are lost.

**Message ID:** 1449

**Abbreviated ASCII Syntax:**

```
PSRDIFFSOURCE TIMEOUT option [timeout]
```

**Factory Default:**

```
PSRDIFFSOURCE TIMEOUT AUTO
```

**ASCII Example:**

```
PSRDIFFSOURCE TIMEOUT auto
PSRDIFFSOURCE TIMEOUT set 180
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PSRDIFFSOURCE TIMEOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	AUTO	1	Use AUTO or SET to set the time	Enum	4	H
		SET	2				
3	timeout	0 to 3600 sec		Specify the timeout (default=0)	Ulong	4	H+4

## 2.120 PSRDIFFTIMEOUT

**Sets maximum age of pseudorange differential data**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the maximum age of pseudorange differential correction data to use when operating as a rover station. Received pseudorange differential correction data, older than the specified time, is ignored. This time out period also applies to differential corrections generated from RTK corrections.



The RTCA Standard for scat-i stipulates that the maximum age of differential correction messages cannot be greater than 22 seconds. Therefore, for RTCA rover users, the recommended PSRDIFF delay setting is 22.

**Message ID: 1450**

**Abbreviated ASCII Syntax:**

```
PSRDIFFTIMEOUT delay
```

**Factory Default:**

```
PSRDIFFTIMEOUT 300
```

**ASCII Example:**

```
PSRDIFFTIMEOUT 60
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	PSRDIFFTIMEOUT header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	delay	2 to 1000 s		Maximum pseudorange differential age	Ulong	4	H



## 2.121 QZSSECUTOFF

### *Sets QZSS satellite elevation cutoff*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the elevation cut-off angle for tracked QZSS satellites. The receiver does not start automatically searching for a QZSS satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the **ASSIGN** command on page 76).

In either case, satellites below the QZSSECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle; it could be used in these situations:

- The antenna is at a high altitude, and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Care must be taken when using **QZSSECUTOFF** command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.



Use the **ELEVATIONCUTOFF** command (see page 156) to set the cut-off angle for any system.



For dual antenna receivers, this command applies to both the primary and secondary antennas.

**Message ID: 1350**

**Abbreviated ASCII Syntax:**

```
QZSSECUTOFF angle
```

**Factory Default:**

```
QZSSECUTOFF 5.0
```

**ASCII Example**

```
QZSSECUTOFF 10.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	QZSSECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90 degrees		Elevation cutoff angle relative to the horizon	Float	4	H

## 2.122 RADARCONFIG

### Configure the Emulated Radar Output

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to configure the Emulated Radar (ER) output.



The ER signal is output on the **VARF** or **EVENT\_OUT1** pin of the receiver.



When **RADARCONFIG** is enabled, the **BESTVELTYPE** command (see page 107) mode is set to **DOPPLER**.

**Message ID:** 1878

#### Abbreviated ASCII Syntax:

```
RADARCONFIG switch [freq_step [update_rate [resp_mode [threshold]]]]
```

#### Factory Default:

```
radarconfig disable
```

#### ASCII Example:

```
radarconfig enable 26.11 5hz 2 3.5
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RADARCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Disables radar emulation	Enum	4	H
		ENABLE	1	Enables radar emulation			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	freq_step	10.06 16.32 26.11 28.12 34.80 36.11		Frequency step per kilometre per hour. (default = 36.11 Hz/kph)	Double	8	H+4
4	update_rate	1HZ	1	Rate at which the output frequency is adjusted (default = 10HZ) <sup>1</sup>	Enum	4	H+12
		2HZ	2				
		5HZ	3				
		10HZ	4				
		20HZ	5				
5	resp_mode	See <i>Table 62: Response Modes</i> below		Specify how responsive radar emulation is to changes in velocity (Default = 500) <sup>1</sup>	Integer	4	H+16
6	threshold	2 to 50 kph		The speed threshold at which to switch between response mode 1000 and response mode 500. The threshold is only applicable when the response mode is set to 2. (default = 5 kph)	Double	8	H+20

Table 62: Response Modes

Mode	Description
1	Immediate. This results in the lowest latency at the cost of higher noise.
2	Automatically switch between 1000 and 500 depending on speed. When speed is below the Threshold parameter, use Response Mode 500. Otherwise, use Response Mode 1000.
500	Signal is minimally smoothed resulting in low latency but increased noise.

<sup>1</sup>The number of samples used for smoothing depends on both the update\_rate and resp\_mode parameters. For instance, if the update\_rate is 5 Hz and the resp\_mode is 2000 ms, the number of samples used will be 10.

Mode	Description
1000	Output signal is smoothed over a smaller window resulting in less latency than 2000 and less noise than 500.
2000	Output signal is smoothed to reduce noise at the cost of higher latency.

## 2.123 RAIMMODE

### *Configures RAIM mode*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to configure Receiver Autonomous Integrity Monitoring (RAIM) operation. This command uses RTCA MOPS characteristics which defines the positioning accuracy requirements for airborne lateral navigation (LNAV) and vertical navigation (VNAV) at 3 stages of flight:

1. En route travel
2. Terminal (within range of air terminal)
3. Non-precision approach

In order to ensure that the required level of accuracy is available in these phases of flight, MOPS requires the computation of protection levels (HPL and VPL). MOPS has the following definitions that apply to NovAtel's RAIM feature:

**Horizontal Protection Level (HPL)** is a radius of the circle in the horizontal plane. Its center is at the true position, that describes the region, assured to contain the indicated horizontal position. It is the horizontal region where the missed alert and false alert requirements are met using autonomous fault detection.

**Vertical Protection Level (VPL)** is half the length of the segment on the vertical axis. Its center is at the true position, that describes the region, assured to contain the indicated vertical position when autonomous fault detection is used.

**Horizontal Alert Limit (HAL)** is a radius of the circle in the horizontal plane. Its center is at the true position, that describes the region, required to contain the indicated horizontal position with the required probability.

**Vertical Alert Limit (VAL)** is half the length of the segment on the vertical axis. Its center is at the true position, that describes the region, required to contain the indicated vertical position with certain probability.

**Probability of False Alert ( $P_{fa}$ )** is a false alert defined as the indication of a positioning failure, when a positioning failure has not occurred (as a result of false detection). A false alert would cause a navigation alert.

### 2.123.1 Detection strategy

NovAtel's RAIM detection strategy uses the weighted Least-Squares Detection (LSA) method. This method computes a solution using a LSA and is based on the sum of squares of weighted residuals. It is a comparison between a root sum of squares of residuals and a decision threshold to determine a pass/fail decision.

### 2.123.2 Isolation strategy

NovAtel RAIM uses the maximum residual method. Logically it is implemented as a second part of Fault Detection and Exclusion (FDE) algorithm for LSA detection method. Weighted LSA residuals are standardized individually and the largest residual is compared to a decision threshold. If it is more than the threshold, the observation corresponding to this residual is declared faulty.

**Message ID: 1285**

**Abbreviated ASCII Syntax:**

```
RAIMMODE mode [hal [val [pfa]]]
```

**Factory Default:**

```
RAIMMODE default
```

**Input Example:**

```
RAIMMODE user 100 100 0.01
```

```
RAIMMODE terminal
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RAIMMODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	MODE	See <i>Table 63: RAIM Mode Types</i> below			Enum	4	H
3	HAL	$5 \leq HAL \leq 9999.99$		Horizontal alert limit (m) (Default = 0.0)	Double	8	H+4
4	VAL	$5 \leq VAL \leq 9999.99$		Vertical alert limit (m) (Default = 0.0)	Double	8	H+12
5	PFA	$(P_{fa}) = 1e^{-7} \leq P_{fa} \leq 0.25$		Probability of false alert (Default = 0.0)	Double	8	H+20

**Table 63: RAIM Mode Types**

Binary	ASCII	Description
0	DISABLE	Do not do integrity monitoring of least squares solution
1	USER	User will specify alert limits and probability of false alert
2	DEFAULT	Use NovAtel RAIM (default)
3	APPROACH	Default numbers for non-precision approach navigation modes are used - HAL = 556 m (0.3 nm), VAL = 50 m for LNAV/VNAV
4	TERMINAL	Default numbers for terminal navigation mode are used - HAL = 1 nm, no VAL requirement
5	ENROUTE	Default numbers for enroute navigation mode are used - HAL = 2 nm, no VAL requirement

## 2.124 REFERENCESTATIONTIMEOUT

### *Sets timeout for removing previously stored base stations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command sets how long the receiver will retain RTK base station co-ordinates. Shorter durations might be required if the receiver is operating in a VRS RTK network that recycles base station IDs quickly.

**Message ID:** 2033

#### Abbreviated ASCII Syntax:

```
REFERENCESTATIONTIMEOUT option [timeout]
```

#### Factory Default:

```
REFERENCESTATIONTIMEOUT AUTO
```

#### ASCII Example:

```
REFERENCESTATIONTIMEOUT SET 90
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	REFERENCESTATIONTIMEOUT header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	option	AUTO	1	Sets the Timeout to 90 seconds <sup>1</sup>  The Timeout field is optional for AUTO and has no effect	Enum	4	H
		SET	2	Must set the timeout value using the Timeout field  0 is not accepted when using the SET option			
3	timeout	1 to 3600 s		Specify the time	Ulong	4	H+4

<sup>1</sup>This behavior is subject to change.



## 2.125 RESET

### *Performs a hardware reset*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command performs a hardware reset. The receiver configuration reverts either to the factory default, if no user configuration was saved or the last **SAVECONFIG** settings. Refer to the **FRESET** command on page 193 and **SAVECONFIG** command on page 366.

The optional delay field is used to set the number of seconds the receiver is to wait before resetting.

**Message ID:** 18

#### Abbreviated ASCII Syntax:

```
RESET [delay]
```

#### Input Example

```
RESET 30
```



The **RESET** command can be used to erase any unsaved changes to the receiver configuration.

Unlike the **FRESET** command on page 193, the **RESET** command does not erase data stored in the NVM, such as Almanac and Ephemeris data.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RESET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	delay (0-60)			Seconds to wait before resetting (default = 0)	Ulong	4	H

## 2.126 RFINPUTGAIN

### *Configure the Calibrated Antenna Gain (CAG)*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to select the mode (AUTO or MANUAL) of setting the CAG for the purpose of interference detection.

If auto mode is used, the receiver will automatically compute the CAG at start up. In this case it is assumed that the receiver is powered up with its antenna connected and no interference is present.



If the antenna is changed, either reset the receiver or reissue this command to allow receiver to recompute the CAG.

If manual mode is used, the CAG input by the user is used by the receiver to detect interference.

The CAG is defined to be the cascaded RF gain before receiver input plus LNA noise figure (NF), counting active antenna LNA gain, in-line amplifier, RF cable or distribution loss prior to receiver input connector.

A typical GNSS active antenna (of reasonable quality) has a noise figure of ~2dB (dominated by the LNA in an active antenna).

$\text{RFINPUTGAIN} = \text{Cascaded Gain before receiver} + \text{LNA NF}$



**For advanced users.**

If using this command in manual mode, the antenna gain must be accurately measured when the system is not experiencing any interference. If an erroneous CAG is injected, the interference detection performance can be degraded.

**Message ID: 1658**

**Abbreviated ASCII Syntax:**

```
RFINPUTGAIN RFPATH [mode] [CAG]
```

**Factory Default:**

```
RFINPUTGAIN L1 AUTO
```

```
RFINPUTGAIN L2 AUTO
```

```
RFINPUTGAIN L5 AUTO
```

**ASCII Example:**

```
RFINPUTGAIN L1 MANUAL 30
```

```
RFINPUTGAIN L2 30
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RFINPUTGAIN header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	RFPATH	L1	2	RF path selection	Enum	4	H
		L2	3				
		L5	5				
3	mode	AUTO	0	Calibrated Antenna Gain (CAG) mode.	Enum	4	H+4
		MANUAL	1	Default = MANUAL			
4	CAG	0.0-100.0		Calibrated Antenna Gain value If the mode is MANUAL, a value for CAG must be entered.	Float	4	H+8

## 2.127 RTKANTENNA

**Specifies L1 phase center (PC) or ARP and enables/disables PC modeling**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to specify whether to use L1 phase center or Antenna Reference Point (ARP) positioning.

There is also an option to apply phase center variation modeling. If there are any conditions that make a selected mode impossible, the solution status in the position log will indicate an error or warning.

L1 ARP offsets and L2 ARP offsets can be entered using the **BASEANTENNAPCO** command on page 90 and **THISANTENNAPCO** command on page 426. Phase center variation parameters can be entered using the **BASEANTENNAPCV** command on page 92 and **THISANTENNAPCV** command on page 427.

Error states occur if either the rover does not have the necessary antenna information entered or the base is not sending sufficient information to work in the requested mode. An example of these error conditions is:

- Position reference to the ARP is requested but no rover antenna model is available

**Message ID: 858**

**Abbreviated ASCII Syntax:**

```
RTKANTENNA posref pcv
```

**Factory Default:**

```
RTKANTENNA unknown disable
```

**ASCII Example:**

```
RTKANTENNA arp enable
```



This command is used for high precision RTK positioning allowing application of antenna offset and phase center variation parameters.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKANTENNA header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	posref	L1PC	0	L1 phase center position reference	Enum	4	H
		ARP	1	ARP position reference			
		UNKNOWN	2	Unknown position reference			
3	pcv	DISABLE	0	Disable PCV modeling	Enum	4	H+4
		ENABLE	1	Enable PCV modeling			
4	Reserved				Bool	4	H+8
5	Reserved				Bool	4	H+12

## 2.128 RTKASSIST

### *Enable or disable RTK ASSIST*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command enables or disables RTK ASSIST.

RTK ASSIST uses L-Band-delivered corrections to enable RTK operation to continue for extended durations if RTK corrections are lost. In order to use RTK ASSIST, a receiver with L-Band tracking capability and RTK ASSIST capability is needed. The duration of RTK ASSIST operation can be limited using the **RTKASSISTTIMEOUT** command (see page 335).

When active, RTK ASSIST is shown in the RTKPOS and BESTPOS extended solution status field (see *Table 90: Extended Solution Status* on page 527). The active status and further details on the RTK ASSIST status are available through the **RTKASSISTSTATUS** log on page 859.



For reliable RTK ASSIST performance, the RTK base station position must be within 1 metre of its true WGS84 position.

**Message ID:** 1985

**Abbreviated ASCII Syntax:**

```
RTKASSIST switch
```

**Factory Default:**

```
RTKASSIST enable
```

**ASCII Example:**

```
RTKASSIST disable
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKASSIST header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	switch	DISABLE	0	Disable RTK ASSIST	Enum	4	H
		ENABLE	1	Enable RTK ASSIST			

## 2.129 RTKASSISTTIMEOUT

### Set the maximum RTK ASSIST duration

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command sets how long the receiver will report an RTK solution when RTK is being maintained by RTK ASSIST. The maximum permitted duration of RTK ASSIST operation is determined by the subscription and receiver model. Values less than the subscription limit can be set using the **RTKASSISTTIMEOUT** command.



When RTK ASSIST is active, the **RTKTIMEOUT** command is disregarded. The maximum time that RTK will continue past an RTK corrections outage is controlled by **RTKASSISTTIMEOUT**.

**Message ID:** 2003

#### Abbreviated ASCII Syntax:

```
RTKASSISTTIMEOUT limit_type [limit_value]
```

#### Factory Default:

```
RTKASSISTTIMEOUT SUBSCRIPTION_LIMIT
```

#### ASCII Example:

```
RTKASSISTTIMEOUT USER_LIMIT 900
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKASSISTTIMEOUT header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	limit_type	SUBSCRIPTION_LIMIT	0	Use maximum permitted duration limit.	Enum	4	H
		USER_LIMIT	1	The maximum RTK ASSIST duration is user set, up to the limit permitted by the subscription and model.			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	limit_value			Time out value in seconds. Only valid for the USER_LIMIT Limit Type.	Ulong	4	H+4



## 2.130 RTKDYNAMICS

### *Sets the RTK dynamics mode*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to specify how the receiver looks at the data. There are three modes: STATIC, DYNAMIC and AUTO. The STATIC mode forces the RTK software to treat the rover station as though it were stationary.

DYNAMIC mode forces the software to treat the rover as though it were in motion. If the receiver is undergoing very slow, steady motion (<2.5 cm/s for more than 5 seconds), use DYNAMIC mode (as opposed to AUTO) to prevent inaccurate results and possible resets.



For reliable performance, the antenna should not move more than 1-2 cm when in STATIC mode.

**Message ID:** 183

#### Abbreviated ASCII Syntax:

```
RTKDYNAMICS mode
```

#### Factory Default:

```
RTKDYNAMICS dynamic
```

#### ASCII Example:

```
RTKDYNAMICS static
```



Use the STATIC option to decrease the time required to fix ambiguities and reduce the amount of noise in the position solution. If STATIC mode is used when the antenna is not static, the receiver will have erroneous solutions and unnecessary RTK resets.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKDYNAMICS header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	AUTO	0	Automatically determines dynamics mode	Enum	4	H
		STATIC	1	Static mode			
		DYNAMIC	2	Dynamic mode			

## 2.131 RTKINTEGCRITERIA

### *Report inaccurate fixed-integer RTK positions with float solution type*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command forces a fixed-integer RTK position to be reported as float if the estimated solution standard deviation exceeds a threshold.

Normally, a fixed-integer solution is very accurate. However, in some rarely-occurring situations, even a fixed-integer solution can become inaccurate; for example, if the DOP is high due to satellites not being visible. In such cases, the accuracy of the RTK solution might be worse than what is customarily expected from a fixed-integer solution. The **RTKINTEGCRITERIA** command changes the solution type of these high standard deviation integer solutions to their float equivalent. `NARROW_INT`, for instance, becomes `NARROW_FLOAT`. Depending on the `GGAQUALITY` command setting, this will also impact the NMEA GGA quality flag.

**Message ID: 2070**

#### Abbreviated ASCII Syntax:

```
RTKINTEGCRITERIA criteria threshold
```

#### Factory Default:

```
RTKINTEGCRITERIA TOTAL_STDDEV 1.0
```

#### ASCII Example:

```
RTKINTEGCRITERIA HORIZONTAL_STDDEV 0.25
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKINTEGCRITERIA header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	criteria	TOTAL_STDDEV	1	Test the threshold against the estimated total, 3D, standard deviation	Enum	4	H
		HORIZONTAL_STDDEV	2	Test the threshold against the estimated horizontal standard deviation			
3	threshold	0.01 m and higher		Estimated solution standard deviation (m) required for solution to be reported as integer	Float	4	H+4

## 2.132 RTKMATCHEDTIMEOUT

**Sets RTK filter reset time after corrections are lost**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command sets the length of time the receiver continues to use the last RTK correction data once the corrections stop. Once this time is reached, the RTK filter is reset.

**Message ID: 1447**

**Abbreviated ASCII Syntax:**

```
RTKMATCHEDTIMEOUT timeout
```

**ASCII Example:**

```
RTKMATCHEDTIMEOUT 180
```

**Factory Default**

```
RTKMATCHEDTIMEOUT 300
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKMATCHEDTIMEOUT header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	timeout	1 to 3600 s		Time out period	Ulong	4	H

## 2.133 RTKNETWORK

*Specifies the RTK network mode*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Network RTK uses permanent base station installations, allowing kinematic GNSS users to achieve centimetre accuracies, without the need of setting up a GNSS base station, at a known site. This command sets the RTK network mode for a specific network. For more details on Network RTK, refer to the application note [APN-041 Network RTK](#).

**Message ID:** 951

**Abbreviated ASCII Syntax:**

```
RTKNETWORK mode [network#]
```

**Factory Default:**

```
RTKNETWORK AUTO
```

**Input Example:**

```
RTKNETWORK imax
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKNETWORK header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	mode	<i>Table 64: Network RTK Mode below</i>		RTK network mode. The factory default is auto where the receiver switches to the first available network RTK source	Enum	4	H
3	network#	0 to 4294967295		Specify a number for the network (default = 0)	Ulong	4	H+4

**Table 64: Network RTK Mode**

Binary	ASCII	Description
0	DISABLE	Single reference station RTK mode. All received network RTK corrections are ignored.

Binary	ASCII	Description
1-4	Reserved	
5	VRS	<p>The Virtual Reference Station (VRS) or Virtual Base Station (VBS) idea introduced by Trimble, is that a base station is artificially created in the vicinity of a rover receiver. All baseline length dependent errors, such as abnormal troposphere variation, ionospheric disturbances and orbital errors, are reduced for this VRS. The rover receiving VRS information has a lower level of these errors than a distant base station. The VRS is calculated for a position, supplied by the rover during communication start up, with networking software. The VRS position can change if the rover is far away from the initial point. The format for sending the rover's position is standard NMEA format. Most rovers receive VRS data, for a calculated base station, within a couple of metres away.</p> <p>The VRS approach requires bi-directional communication for supplying the rover's position to the networking software.</p>
6	IMAX	<p>The iMAX idea, introduced by Leica Geosystems, is that networking software corrections, based on the rover's position, are calculated as with VRS. However, instead of calculating the base station observations for the provided position or another position closer to the base station, original observation information is corrected with the calculated corrections and broadcast. VRS works so that although the rover is unaware of the errors the VRS is taking care of, there still might be ionospheric remains in the base station observations. iMAX provides actual base station position information. The rover may assume the base station is at a distance and open its settings for estimation of the remaining ionospheric residuals. The iMAX method may trigger the rover to open its settings further than required, since the networking software removes at least part of the ionospheric disturbances. However, compared to VRS above, this approach is safer since it notifies the rover when there might be baseline length dependent errors in the observation information. iMAX requires bi-directional communication to the networking software for supplying the base station observation information.</p>
7	FKP	<p>The FKP method delivers the information from a base station network to the rover. No precise knowledge of the rover's position is required for providing the correct information. The corrections are deployed as gradients to be used for interpolating to the rover's actual position.</p>
8	MAX	<p>The basic principle of the master-auxiliary concept is to provide, in compact form, as much of the information from the network and the errors it is observing to the rover as possible. With more information about the state and distribution of the dispersive and non-dispersive errors across the network, the rover is able to use more intelligent algorithms in the determination of its position solution. Each supplier of reference station software will have their own proprietary algorithms for modeling or estimating these error sources. The rover system can decide to use or to neglect the network RTK information, depending on its own firmware algorithm performance.</p>
9	Reserved	
10	AUTO	<p>Default value, assume single base. If network RTK corrections are detected then the receiver will switch to the appropriate mode. iMAX and VRS can only be detected using RTCMV3, however, it is not possible to distinguish between iMAX or VRS. If iMAX or VRS is detected, then iMAX will be assumed.</p>

## 2.134 RTKPORTMODE

*Assigns the port for RTK and ALIGN messages*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7



This command only applies to receivers with both RTK and ALIGN enabled.

A rover receiver with RTK and ALIGN enabled can receive RTK and ALIGN corrections at the same time. However, the two different sources (RTK and ALIGN) must be sent to different ports.

Use the RTKPORTMODE command to route correction feeds to different ports. RTK and ALIGN can be routed to any user specified ports.

Failing to specify the mode for the incoming source could cause unexpected behavior of RTK or ALIGN.



Ports configured using the RTKPORTMODE command must also be configured using the **INTERFACEMODE** command (see page 216).

**Message ID: 1936**

**Abbreviated ASCII Syntax:**

```
RTKPORTMODE [port] mode
```

**Factory Default:**

```
RTKPORTMODE COM1 RTK
RTKPORTMODE COM2 RTK
RTKPORTMODE COM3 RTK
RTKPORTMODE COM4 RTK
RTKPORTMODE COM5 RTK
RTKPORTMODE COM6 RTK
RTKPORTMODE ICOM1 RTK
RTKPORTMODE ICOM2 RTK
RTKPORTMODE ICOM3 RTK
RTKPORTMODE ICOM4 RTK
RTKPORTMODE ICOM5 RTK
RTKPORTMODE ICOM6 RTK
RTKPORTMODE ICOM7 RTK
RTKPORTMODE NCOM1 RTK
RTKPORTMODE NCOM2 RTK
RTKPORTMODE NCOM3 RTK
RTKPORTMODE USB1 RTK
```



```

RTKPORTMODE USB2 RTK
RTKPORTMODE USB3 RTK
RTKPORTMODE WCOM1 RTK
RTKPORTMODE BT1 RTK
RTKPORTMODE AUX RTK
RTKPORTMODE CCOM1 RTK
RTKPORTMODE CCOM2 RTK
RTKPORTMODE CCOM3 RTK
RTKPORTMODE CCOM4 RTK
RTKPORTMODE CCOM5 ALIGN
RTKPORTMODE CCOM6 RTK

```

**ASCII Example:**

```

RTKPORTMODE COM2 RTK
RTKPORTMODE COM3 ALIGN

```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKPORTMODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Port	See Table 34: <i>Communications Port Identifiers</i> on page 152		Port identifier (default = THISPORT)	Enum	4	H
3	Mode	RTK	0	Mode for this port	Enum	4	H+4
		ALIGN	1				

## 2.135 RTKQUALITYLEVEL

### Sets an RTK quality mode

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to select an RTK quality mode.

**Message ID:** 844

#### Abbreviated ASCII Syntax:

```
RTKQUALITYLEVEL mode
```

#### Factory Default:

```
RTKQUALITYLEVEL normal
```

#### ASCII Example:

```
RTKQUALITYLEVEL extra_safe
```



The **EXTRA\_SAFE** mode is needed in areas where the signal is partially blocked and the position solution in **NORMAL** mode shows **NARROW\_INT** even though the real position solution is out by several metres. Using **EXTRA\_SAFE** in these environments means the solution will be slower getting to **NARROW\_INT** but it is less likely to be erroneous.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKQUALITYLEVEL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	mode	NORMAL	1	Set the RTK quality level mode to Normal RTK	Enum	4	H
		EXTRA_SAFE	4	Set the RTK quality level mode to Extra Safe RTK			

## 2.136 RTKRESET

### *Reset the RTK filter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command resets the RTK filter and causes the AdVanceRTK filter to undergo a complete reset, forcing the system to restart the ambiguity resolution calculations.

**Message ID:** 2082

#### **Abbreviated ASCII Syntax:**

```
RTKRESET [Switch]
```

#### **Example :**

```
RTKRESET
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Byte	Binary Offset
1	Command header	-	-	RTKRESET header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Switch	FILTER	1	Reset the RTK filter. This is an optional parameter	Enum	4	H

## 2.137 RTKSOURCE

### *Sets the RTK correction source*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command determines what corrections the RTK filter can use. This is useful when the receiver is receiving corrections from multiple base stations.

**RTKSOURCE NONE** stops any corrections from being used and immediately disables RTK and RTK ASSIST.

**Message ID: 494**

#### Abbreviated ASCII Syntax:

```
RTKSOURCE type [id]
```

#### Factory Default:

```
RTKSOURCE AUTO ANY
```

#### ASCII Examples:

1. Use a specific format:

```
RTKSOURCE RTCMV3
```

2. Use a specific format and base station:

```
RTKSOURCE RTCMV3 1234
```

3. Disable RTK:

```
RTKSOURCE NONE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKSOURCE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	type	See Table 65: <i>RTK Source Type</i> on the next page		Correction type	Enum	4	H
3	id	Char[4] or ANY		Base station ID (default = ANY)	Char[4]	8	H+4

Table 65: RTK Source Type

Binary	ASCII	Description
0	RTCM	Use only RTCM version 2 corrections from the given ID. ID: $0 \leq \text{RTCM ID} \leq 1023$ <b>or</b> ANY to accept all IDs
1	RTCA	Use only RTCA corrections from the given ID. ID: A four-character string containing only alpha (a-z) or numeric characters (0-9) <b>or</b> ANY to accept all IDs
2	CMR	Use only CMR corrections from the given ID. ID: $0 \leq \text{CMR ID} \leq 31$ <b>or</b> ANY to accept all IDs
10	AUTO	Use any correction type received. If multiple types are received, the first one received is used. If entered, the ID must be ANY.
11	NONE	Disable RTK
13	RTCMV3	Use only RTCM version 3 corrections from the given ID. ID: $0 \leq \text{RTCMV3 ID} \leq 4095$ <b>or</b> ANY to accept all IDs.
14	NOVATELX	Use only NovAtel proprietary corrections from the given ID. ID: A four-character string containing alpha (a-z) or numeric characters (0-9) <b>or</b> ANY to accept all IDs.

## 2.138 RTKSOURCETIMEOUT

### Sets RTK correction source timeout

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

When multiple RTK correction sources are available, this command allows the user to set a time, in seconds, that the receiver will wait before switching to another RTK correction source if corrections from the original source are lost.

**Message ID:** 1445

#### Abbreviated ASCII Syntax:

```
RTKSOURCETIMEOUT option [timeout]
```

#### Factory Default:

```
RTKSOURCETIMEOUT AUTO
```

#### ASCII Example:

```
RTKSOURCETIMEOUT auto
RTKSOURCETIMEOUT set 180
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKSOURCETIMEOUT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	AUTO	1	Sets the timeout according to network type or other self-detected conditions. Timeout field is optional for AUTO and has no effect	Enum	4	H
		SET	2	Sets the timeout to the value entered in the <i>timeout</i> field.			
3	timeout	1 to 3600 s (maximum)		Specify the time 0 is not accepted if <i>SET</i> is entered in the <i>option</i> field (default=0 for the AUTO option)	Ulong	4	H+4

## 2.139 RTKSVENTRIES

### *Sets number of satellites in corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command sets the number of satellites (at the highest elevation) that are transmitted in the RTK corrections from a base station receiver. This is useful when the amount of bandwidth available for transmitting corrections is limited.

**Message ID:** 92

**Abbreviated ASCII Syntax:**

```
RTKSVENTRIES number
```

**Factory Default:**

```
RTKSVENTRIES 24
```

**ASCII Example:**

```
RTKSVENTRIES 7
```



GPS devices have enabled many transit and fleet authorities to provide Automatic Vehicle Location (AVL). AVL systems track the position of individual vehicles and relay that data back to a remote dispatch location that can store or better utilize the information. Consider the implementation of an AVL system within a police department, to automatically log and keep track of the location of each cruiser. Typically a fleet uses a 9600 bps connection where AVL data is relayed back to headquarters. The limited bandwidth of the radio must be shared amongst the AVL and other systems in multiple cruisers.

When operating with a low baud rate radio transmitter (9600 or lower), especially over a long distance, the AVL system could limit the number of satellites for which corrections are sent using the **RTKSVENTRIES** command.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKSVENTRIES header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	number	4-24		The number of SVs to be transmitted in correction messages	Ulong	4	H

## 2.140 RTKTIMEOUT

### *Sets maximum age of RTK data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command is used to set the maximum age of RTK data to use when operating as a rover station. RTK data received that is older than the specified time is ignored.



When RTK ASSIST is active, the **RTKTIMEOUT** command is disregarded. The maximum time that RTK will continue past an RTK corrections outage is controlled by the settings in the **RTKASSISTTIMEOUT** command (see page 335).

**Message ID:** 910

#### **Abbreviated ASCII Syntax:**

```
RTKTIMEOUT delay
```

#### **Factory Default:**

```
RTKTIMEOUT 60
```

#### **ASCII Example (rover):**

```
RTKTIMEOUT 20
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKTIMEOUT header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	delay	5 to 60 s		Maximum RTK data age	Ulong	4	H



## 2.141 RTKTRACKINGCONTROL

### *Automatic selection of tracking signal type based on incoming base station observations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Controls automatic selection of tracking signal type based on incoming base station observations. This command only applies to single GPS/GLO L2 channel configurations that can auto switch between L2P and L2C. The **RTKTRACKINGCONTROL** command setting takes precedence over the **FORCEGPSL2CODE** command (see page 188) and **FORCEGLOL2CODE** command (see page 186). If this command is set to AUTO, the receiver will always match signal types to incoming corrections.

**Message ID: 1351**

**Abbreviated ASCII Syntax:**

```
RTKTRACKINGCONTROL [Mode]
```

**Factory Default:**

```
RTKTRACKINGCONTROL AUTO
```

**ASCII Example:**

```
RTKTRACKINGCONTROL DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	RTKTRACKINGCONTROL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Mode	Disable	0	Disable automatic selection of tracking signal	Enum	4	H
		Auto	1	Automatically match the signal type to the incoming corrections.  Default = Auto			

## 2.142 SATEL4CONFIG

### *Configures the Satel 400 MHz radio*

**Platform:** SMART7 with RELAY7



Refer to the Satel Radio user documentation for complete and detailed instructions regarding the correct operation of the radio.

This command is used to configure the Satel 400 MHz radio. Settings in this command are stored directly on the radio and not saved through the **SAVECONFIG** command on page 366.

The receiver responds to the command immediately, but the radio settings will take some time to be applied. Log **SATELSTATUS** **onchanged** to receive a notification that configuration is complete. See the **SATELSTATUS** log on page 902.



- Not all compatibility modes are available on every radio.
- The **radio\_behavior** setting determines whether the receive or transmit frequency will be changed.
- If the TRIMTALK protocol is used, the base transmitter type (TRIMBLE or PACCREST) must be specified.
- If the **tx\_power** value entered does not correspond to one of the power levels supported by the radio module, the output power is set to the nearest possible value.



Do not interrupt power while the radio is being configured.

**Message ID: 2215**

#### **Abbreviated ASCII Syntax:**

```
SATEL4CONFIG <radio_behavior> <tx_power> <freq> <compatibility> <base_type>
```

#### **ASCII Example:**

```
SATEL4CONFIG RX 0 438000000 7 TRIMBLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATEL4CONFIG header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	radio_behavior	RX	0	Configure for receive	Enum	4	H
		TX	1	Configure for transmit			
3	tx_power	0 - 35000		Transmit power (mW)	UInt	4	H+4
4	freq	403000000 - 473000000		Frequency (Hz)	UInt	4	H+8
5	compatibility	1-20 Refer to <i>Table 67: Compatibility Mode</i> on the next page		Radio compatibility mode	UInt	4	H+12
6	base_type	0-5 Refer to <i>Table 66: Base Type</i> below		Base transmitter type (required for TRIMTALK protocol only)	Enum	4	H+16
7	crc			32-bit CRC (ASCII and Binary only)	Hex	4	H+20

Table 66: Base Type

Binary	ASCII	Description
0	PACCREST	PacCrest transmitter
4	TRIMBLE	Trimble transmitter
9	NONE	Unspecified base type

Table 67: Compatibility Mode

Binary	Protocol	Modulation	Link Rate (bps) / Channel Spacing (kHz)	FEC
1	Pacific Crest compatible	GMSK	4800 / 12.5	ON
2	Pacific Crest compatible	GMSK	4800 / 12.5	OFF <sup>1</sup>
3	Pacific Crest compatible	4FSK	9600 / 12.5	ON
4	Pacific Crest compatible	4FSK	9600 / 12.5	OFF <sup>1</sup>
5	Pacific Crest FST compatible	4FSK	9600 / 12.5	ON
6	Pacific Crest FST compatible	4FSK	9600 / 12.5	OFF <sup>1</sup>
7	Trimtalk compatible	GMSK	4800 / 12.5	
8	Trimtalk compatible	GMSK	8000 <sup>2</sup>	
9	Satel 3AS	4FSK	9600 / 12.5	ON
10	Satel 3AS	4FSK	9600 / 12.5	OFF
11	Pacific Crest compatible	GMSK	9600 / 25.0	ON
12	Pacific Crest compatible	GMSK	9600 / 25.0	OFF <sup>1</sup>
13	Pacific Crest compatible	4FSK	19200 / 25.0	ON
14	Pacific Crest compatible	4FSK	19200 / 25.0	OFF <sup>1</sup>
15	Pacific Crest FST compatible	4FSK	19200 / 25.0	ON
16	Pacific Crest FST compatible	4FSK	19200 / 25.0	OFF <sup>1</sup>
17	Trimtalk compatible	GMSK	9600 / 25.0	
18	Trimtalk compatible	GMSK	16000 <sup>2</sup>	
19	Satel 3AS	4FSK	19200 / 25.0	ON
20	Satel 3AS	4FSK	19200 / 25.0	OFF



Base Type is only required for receiving TRIMTALK protocol.

<sup>1</sup>The Satel M3-R4 and EASy radio modules do not support disabling FEC for compatibility modes.

<sup>2</sup>The Satel M3-R4 and EASy radio modules do not support Trimtalk 8000 and 16000 link rates.

## 2.143 SATEL9CONFIG

### *Configures the Satel 900 MHz radio*

**Platform:** SMART7 with RELAY7



Refer to the Satel Radio user documentation for complete and detailed instructions regarding the correct operation of the radio.

This command is used to configure the Satel 900 MHz radio. Settings in this command are stored directly on the radio and not saved through the **SAVECONFIG** command on page 366. Commands can be sent immediately after each other with the last command issued is applied.

Log **SATELSTATUS** *onchanged* to send a notification configuration is complete. See the **SATELSTATUS** log on page 902.



If modem mode is set to P2MP\_SLAVE or P2MP\_RX\_SLAVE , the receive frequency is changed. If modem mode is set to P2MP\_MASTER, the transmit frequency is changed.



Do not interrupt power while the radio is being configured.

**Message ID: 2217**

#### Abbreviated ASCII Syntax:

```
SATEL9CONFIG <modem_mode> <freq_key> <nid> <minpkt> <maxpkt> <retrytimeout>
<subnet> <repeaters> <packetrepeat> <txpwr>
```

#### ASCII Example:

```
SATEL9CONFIG P2MP_MASTER 1 4024 8 9 10 00 TRUE 1 1000
SATEL9CONFIG P2MP_RX_SLAVE 1 4024 8 9 10 00 FALSE 1 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATEL9CONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	modem_mode	P2MP_MASTER	2	Radio role in the network	Enum	4	H
		P2MP_SLAVE	3				
		P2MP_RX_SLAVE	8				
3	freq_key	0-14		Frequency key for hop table	Uint	4	H+4
4	nid	0-4095		Network ID	Uint	4	H+8
5	minpkt	0-9		Minimum packet size	Uint	4	H+12
6	maxpkt	0-9		Maximum packet size	Enum	4	H+16
7	retrytimeout	8-255		Number of failed attempts to receive packets before connection drops	Uint	4	H+20
8	subnet	0x00, 0xFF		Multi-point subnet 0x00=Roaming 0xFF=Disabled	Uint	4	H+24
9	repeaters	FALSE	0	Boolean to specify if repeaters are present in the network	Bool	4	H+28
		TRUE	1				
10	packetrepeat	0-9		Number of times the master repeats packets	Uint	4	H+32
11	txpwr	0-1000		Transmitter output power (mW)	Uint	4	H+36
12	crc			32-bit CRC (ASCII and Binary only)	Hex	4	H+40

## 2.144 SATEL9CONFIGL

### *Configures the 900 MHz radio network*

**Platform:** SMART7 with RELAY7

The SATEL9CONFIGL command is used to configure the Satel 900 MHz radio network settings using a simplified Leica channel identifier.

**Message ID:** 2218

#### Abbreviated ASCII Syntax:

```
SATEL9CONFIGL <modem_mode> <channel> <retrytimeout> <subnet> <repeaters>
<packetrepeat> <txpwr>
```

#### ASCII Example:

```
SATEL9CONFIGL P2MP_MASTER 2 10 00 TRUE 1 1000
SATEL9CONFIGL P2MP_RX_SLAVE 4 10 FF FALSE 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATEL9CONFIGL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	modem_mode	P2MP_MASTER	2	Radio roll in network	Enum	4	H
		P2MP_SLAVE	3				
		P2MP_RX_SLAVE	8				
3	channel	1-9		Virtual channel	Uint	4	H+4
4	retrytimeout	8-255		Number of failed attempts to receive packets before connection is dropped	Uint	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	subnet	0x00, 0xFF		Multi-point subnet 0x00=Roaming, 0xFF=Disabled	Uint	4	H+12
6	repeaters	FALSE	0	Boolean to specify if repeaters are present in the network	Bool	4	H+16
		TRUE	1				
7	packetrepeat	0-9		Number of times master repeats packets	Uint	4	H+20
8	txpwr	0-1000		Transmitter output power (mW)	Uint	4	H+24
9	crc			32-bit CRC (ASCII and Binary only)	Hex	4	H+28



## 2.145 SATEL9CONFIGN

### *Configures the 900 MHz radio network*

**Platform:** SMART7 with RELAY7

The SATEL9CONFIGN command is used to configure the Satel 900 MHz radio network settings using a simplified Novariant channel identifier.

**Message ID:** 2219

#### Abbreviated ASCII Syntax:

```
SATEL9CONFIGN <modem_mode> <channel> <retrytimeout> <subnet> <repeaters>
<packetrepeat> <txpwr>
```

#### ASCII Example:

```
SATEL9CONFIGN P2MP_MASTER 32 10 00 TRUE 1 1000
SATEL9CONFIGN P2MP_RX_SLAVE 32 10 00 FALSE 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATEL9CONFIGN header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	modem_mode	P2MP_MASTER	2	Radio roll in network	Enum	4	H
		P2MP_SLAVE	3				
		P2MP_RX_SLAVE	8				
3	channel	1-3824		Virtual channel	Uint	4	H+4
4	retrytimeout	8-255		Number of failed attempts to receive packets before connection is dropped	Uint	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	subnet	0x00, 0xFF		Multi-point subnet 0x00=Roaming, 0xFF=Disabled	Uint	4	H+12
6	repeaters	FALSE	0	Boolean to specify if repeaters are present in the network	Bool	4	H+16
		TRUE	1				
7	packetrepeat	0-9		Number of times master repeats packets	Uint	4	H+20
8	txpwr	0-1000		Transmitter output power (mW)	Uint	4	H+24
9	crc			32-bit CRC (ASCII and Binary only)	Hex	4	H+28

## 2.146 SATELCONTROL

### *Controls radio operations*

**Platform:** SMART7 with RELAY7

The SATELCONTROL command is used to control the SATEL radio.

**Message ID:** 2203

**Abbreviated ASCII Syntax:**

SATELCONTROL <Action> <Value>

**ASCII Example:**

SATELCONTROL POWER ON

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATELCONTROL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	action	Refer to <i>Table 68: Turn Radio On, Off or Factory Reset</i> below		Action to perform	Enum	4	H
	value			Corresponding value for action	String [24] <sup>1</sup>	24	H+4
3	crc			32-bit CRC (ASCII and Binary only)	Hex	4	H+28

**Table 68: Turn Radio On, Off or Factory Reset**

Action	Value	Description
POWER	ON	Toggles radio power (if available)
	OFF	
FRESET	-	Resets the radio to factory default settings

<sup>1</sup>Refer to *Table 1: Field Type* on page 32 for detailed explanation of string variable length.

## 2.147 SATELDETECT

### *Detects the connected Satel radio*

**Platform:** SMART7 with RELAY7

This command is used to detect the Satel radio connected to the serial port. Detection is only required to configure the radio settings. If previously configured, the radio can provide RTK corrections without issuing the SATELDETECT command.

Logging **SATELSTATUS** command on page 902 onchanged provides a notification of the completion of detection. If no radio is detected at the default baud rate of 38400, the RIL will attempt at additional baud rates (115200, 9600). If a radio is detected, the baud rate of the radio will be changed to 38400. Upon detection, the radio identity (version block) and settings (**SATEL4INFO** command on page 897 and **SATEL9INFO** command on page 899) will be queried. Regardless of the presence of a radio, the final state of the given serial port will be baud rate=38400, data bits=8, parity bits=N, stop bits=1. Interfacemode will be returned to its setting before SATELDETECT was issued.

**Message ID:** 2204

#### Abbreviated ASCII Syntax:

```
SATELDETECT COMPort
```

#### ASCII Example:

```
SATELDETECT COM3
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATELDETECT header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	port	See Table 70: COM Port Identifiers on page 385		Port to attempt detection (default=COM3)	Enum	4	H

## 2.148 SATELSTARTUPDETECT

*Detects the Satel radio on the COM port during startup*

**Platform:** SMART7 with RELAY7

This command is used to detect the Satel radio connected to the COM port during system startup. Detection functions the same as if the **SATELDETECT** command on the previous page was issued. **SATELDETECT** automatically begins at the next startup when the **SATELSTARTUPDETECT** command is used.

The **SAVECONFIG** command on the next page must be issued to set this command parameters.

**Message ID:** 2221

**Abbreviated ASCII Syntax:**

```
SATELSTARTUPDETECT Switch COMPort
```

**ASCII Example:**

```
SATELSTARTUPDETECT Enable COM3
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SATELSTARTUPDETECT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	Enable	-	Switch to enable detection (default=Disable)	Enum	4	H
		Disable	-				
3	port	See Table 70: COM Port Identifiers on page 385		Port to attempt detection (default=COM3)	Enum	4	H+4

## 2.149 SAVECONFIG

### *Save current configuration in NVM*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

**SAVECONFIG** saves commands that were entered by a user to establish the desired receiver configuration on start up. This configuration is saved in Non-Volatile Memory (NVM) and is executed every time the receiver starts, whether by power-up or the **RESET** command on page 329. The commands that are saved as part of a configuration are typically **LOG** requests, **SERIALCONFIG** commands, **ICOMCONFIG** commands, a **FIX** command and others. Note that not all the commands can be saved using **SAVECONFIG**. All of the commands that set a configuration, including default commands for the receiver, are listed in the **RXCONFIG** log (see page 876). To clear the commands saved by **SAVECONFIG** and reset the receiver to the factory-set configuration, the **FRESET** command on page 193 can be used.



If using the **SAVECONFIG** command in NovAtel Connect, ensure that you have all windows other than the Console window closed. Otherwise, log requests used for the various windows are saved as well. This will result in unnecessary data being logged.

**Message ID: 19**

**Abbreviated ASCII Syntax:**

SAVECONFIG

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SAVECONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

## 2.150 SAVEETHERNETDATA

*Save the configuration data associated with an Ethernet interface*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

Saving the configuration data for an Ethernet interface allows the interface to start automatically at boot time and be configured with either a static IP address or to obtain an address using DHCP. The **SAVEETHERNETDATA** command saves the configuration for the interface previously entered using the **ETHCONFIG** command (see page 159), **IPCONFIG** command (see page 223) and **DNSCONFIG** command (see page 144). The configuration data that is saved will survive a **RESET** command (see page 329) and **FRESET** command (see page 193). To clear the Ethernet interface configuration data, the **FRESET ETHERNET** command is used. It is not necessary to issue the **SAVECONFIG** command (see page 366) to save the Ethernet interface configuration data. In fact, if **SAVECONFIG** is used to save the **ETHCONFIG**, **IPCONFIG** and **DNSCONFIG** commands, the configuration saved by **SAVEETHERNETDATA** will take precedence over the **SAVECONFIG** configuration.

**Message ID:** 1679

**Abbreviated ASCII Syntax:**

```
SAVEETHERNETDATA [Interface]
```

**ASCII Example:**

```
ETHCONFIG ETHA AUTO AUTO AUTO AUTO
IPCONFIG ETHA STATIC 192.168.8.11 255.255.255.0 192.168.8.1
DNSCONFIG 1 192.168.4.200
SAVEETHERNETDATA ETHA
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SAVEETHERNETDATA header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Interface	ETHA	2	The Ethernet interface to save the configuration data for. The default is ETHA.	Enum	4	H

Note that the configurations set using the **ICOMCONFIG** command (see page 213) and **NTRIPCONFIG** command (see page 279) are not saved by the **SAVEETHERNETDATA** command. The following factory default ICOM configurations can be used if Ethernet access to the receiver is required immediately after the receiver is **RESET** or **FRESET**.

```
ICOMCONFIG ICOM1 TCP :3001
ICOMCONFIG ICOM2 TCP :3002
ICOMCONFIG ICOM3 TCP :3003
ICOMCONFIG ICOM4 TCP :3004
ICOMCONFIG ICOM5 TCP :3005
ICOMCONFIG ICOM6 TCP :3006
ICOMCONFIG ICOM7 TCP :3007
```

See also the following commands:

- **ETHCONFIG** command on page 159
- **IPCONFIG** command on page 223
- **DNSCONFIG** command on page 144
- **FRESET** command on page 193



## 2.151 SBASCONTROL

### Sets SBAS test mode and PRN

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to dictate how the receiver tracks and uses correction data from Satellite Based Augmentation Systems (SBAS).

To enable the position solution corrections, issue the **SBASCONTROL ENABLE** command. The receiver does not, by default, attempt to track or use any SBAS signals satellites unless told to do so by the **SBASCONTROL** command. When in AUTO mode, if the receiver is outside the defined satellite system's corrections grid, it reverts to ANY mode and chooses a system based on other criteria.

The "testmode" parameter in the example provides a method to use a particular satellite even if it is currently operating in test mode. The recommended setting for tracking satellites operating in test mode is ZEROTOTWO. On a simulator, you may want to leave this parameter off or specify NONE explicitly.

When using the **SBASCONTROL** command to direct the receiver to use a specific correction type, the receiver begins to search for and track the relevant GEO PRNs for that correction type only.

The receiver can be forced to track a specific PRN using the **ASSIGN** command (see page 76). The receiver can also be forced to use the corrections from a specific SBAS PRN using the **SBASCONTROL** command.

Disable stops the corrections from being used.

**Message ID: 652**

#### Abbreviated ASCII Syntax:

```
SBASCONTROL switch [system] [prn] [testmode]
```


#### Factory Default:

```
SBASCONTROL disable
```

#### ASCII Example:

```
SBASCONTROL enable waas
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SBASCONTROL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	switch	DISABLE	0	Receiver does not use the SBAS corrections it receives (default)	Enum	4	H
		ENABLE	1	Receiver uses the SBAS corrections it receives			
3	system	See <i>Table 69: System Types</i> on the next page		Choose the SBAS the receiver will use	Enum	4	H+4
4	prn	0		Receiver uses any PRN (default)	Ulong	4	H+8
		120-158 and 183-187		Receiver uses SBAS corrections only from this PRN if tracked, otherwise any PRN			
				 Using the <b>ASSIGN</b> command (see page 76) to manually assign an SBAS channel is recommended if a specific PRN is desired.			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	testmode	NONE	0	Receiver interprets Type 0 messages as they are intended (as do not use) (default)	Enum	4	H+12
		ZEROTOTWO	1	Receiver interprets Type 0 messages as Type 2 messages			
		IGNOREZERO	2	Receiver ignores the usual interpretation of Type 0 messages (as do not use) and continues			

Table 69: System Types

ASCII	Binary	Description
NONE	0	Does not use any SBAS satellites (Default for SBASCONTROL DISABLE)
AUTO	1	Automatically determines satellite system to use and prevents the receiver from using satellites outside of the service area (Default for SBASCONTROL ENABLE)
ANY	2	Uses any and all SBAS satellites found
WAAS	3	Uses only WAAS satellites
EGNOS	4	Uses only EGNOS satellites
MSAS	5	Uses only MSAS satellites
GAGAN	6	Uses only GAGAN satellites
QZSS	7	Uses only QZSS L1S signals

## 2.152 SBASECUTOFF

### Sets SBAS satellite elevation cut-off

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the elevation cut-off angle for tracked SBAS satellites. The receiver does not start automatically searching for an SBAS satellite until it rises above the cut-off angle (when satellite position is known). Tracked SBAS satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the **ASSIGN** command on page 76).

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction



Use the **ELEVATIONCUTOFF** command (see page 156) to set the cut-off angle for any system.

**Message ID:** 1000

**Abbreviated ASCII Syntax:**

```
SBASECUTOFF angle
```

**Factory Default:**

```
SBASECUTOFF -5.0
```

**ASCII Example:**

```
SBASECUTOFF 10.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SBASECUTOFF header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

## 2.153 SBASTIMEOUT

### *Sets the SBAS position time out*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to set the amount of time the receiver remains in an SBAS position if it stops receiving SBAS corrections.

**Message ID: 1001**

#### Abbreviated ASCII Syntax:

```
SBASTIMEOUT mode [delay]
```

#### Factory Default:

```
SBASTIMEOUT auto
```

#### ASCII Example:

```
SBASTIMEOUT set 100
```



When the time out mode is AUTO, the time out delay is 180 s.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SBASTIMEOUT header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	mode	Reserved	0	Reserved	Enum	4	H
		AUTO	1	Set to the default value (180 s)			
		SET	2	Set the delay in seconds			
3	delay	2 to 1000 s		Maximum SBAS position age (default=180)	Double	8	H+4
4	Reserved				Double	8	H+12

## 2.154 SELECTCHANCONFIG

### Sets the channel configuration

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Some software models come with support for more than one channel configuration, which can be verified by logging **CHANCONFIGLIST** log (see page 543). The **SELECTCHANCONFIG** command is used to pick a different channel configuration. If a different channel configuration is selected via the **SELECTCHANCONFIG** command, the receiver resets and starts up with the new configuration. The Set in Use number in the **CHANCONFIGLIST** log (see page 543) changes as a result.



After a FRESET, the channel configuration is reset to the factory default value. For receivers with a positioning model bit of R, B or P, the default channel configuration is 3. For all other positioning models, the default is 1.



Subscription Managed Channel Configuration (SMCC) has precedence over **SELECTCHANCONFIG**. When the receiver is reset, any user selected channel configuration will be overwritten by the channel configuration chosen by SMCC.

**Message ID: 1149**

#### Abbreviated ASCII Syntax:

```
SELECTCHANCONFIG chanconfigsetting
```

#### Factory Default:

```
SELECTCHANCONFIG 3 (if the positioning model bit is R, B or P)
```

```
SELECTCHANCONFIG 1 (for all other positioning models)
```

#### ASCII Example:

```
SELECTCHANCONFIG 2
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SELECTCHANCONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	chanconfigsetting	1 to n where n is the number of channel configurations in the <b>CHANCONFIGLIST</b> log (see page 543)		Channel configuration to use	Ulong	4	H



Below is a use case example of the **SELECTCHANCONFIG** command. Abbreviated ASCII commands and logs are used to better illustrate the example.

1. LOG CHANCONFIGLIST to show what the channel configuration options are and which channel configuration set is being used.

```
<CHANCONFIGLIST COM1 0 76.5 UNKNOWN 0 11.470 024c0020 d1c0 32768
<   5 9
<       7
<       16 GPSL1L2PL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1
<       14 GLOL1L2
<       16 GALE1E5B
<       30 BEIDOU1B2B2B
<       5 LBAND
<       7
<       16 GPSL1L2
<       4 QZSSL1CAL2C
<       4 SBASL1
<       14 GLOL1L2
<       16 GALE1E5B
<       30 BEIDOU1B2B
<       5 LBAND
<       7
<       16 GPSL1L2PL2CL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1
<       14 GLOL1L2PL2C
<       16 GALE1E5AE5BALTB0C
<       30 BEIDOU1B2B2B
<       5 LBAND
<       8
<       16 GPSL1L2PL2CL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1L5
<       14 GLOL1L2PL2C
<       16 GALE1E5AE5BALTB0C
<       20 BEIDOU1B1CB2B2BB3
<       8 NAVICL5
<       5 LBAND
<       8
<       16 GPSL1L2PL2CL5L1C
<       4 QZSSL1CAL2CL5L1CL6
<       4 SBASL1L5
<       14 GLOL1L2PL2CL3
<       11 GALE1E5AE5BALTB0CE6
<       16 BEIDOU1B1CB2B2BB3
<       8 NAVICL5
<       5 LBAND
<       8
<       16 GPSL1L2PL2CL5
```





```

<      6 QZSSL1CAL2CL5
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<      9 GALE1E5AE5BALTBOCE6
<     24 BEIDOUB1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<     16 GPSL1L2PL2CL5
<      4 QZSSL1CAL2CL5
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<     16 GALE1E5AE5BALTBOCE6
<     16 BEIDOUB1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<     16 GPSL1L2PL2CL5
<      4 QZSSL1CAL2CL5L1CL6
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<     11 GALE1E5AE5BALTBOCE6
<     20 BEIDOUB1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<     16 GPSL1L2PL2CL5L1C
<      4 QZSSL1CAL2CL5L1C
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<     16 GALE1E5AE5BALTBOC
<     20 BEIDOUB1B1CB2B2B
<      8 NAVICL5
<      5 LBAND

```

2. There are nine options given for the model and the fifth channel configuration set is currently being used.
3. If the user would like to use the third channel configuration set enter, **SELECTCHANCONFIG 3** command.
4. The receiver receives the command and resets. At startup, the third channel configuration set is configured.
5. To verify that setting has changed, enter LOG CHANCONFIGLIST.

```

<CHANCONFIGLIST COM1 0 76.5 UNKNOWN 0 11.470 024c0020 d1c0 32768
<      3 9
<      7
<     16 GPSL1L2PL5
<      4 QZSSL1CAL2CL5
<      4 SBASL1
<     14 GLOL1L2
<     16 GALE1E5B

```



```

<          30 BEIDOU1B2B2B
<          5 LBAND
<          7
<          16 GPSL1L2
<          4 QZSSL1CAL2C
<          4 SBASL1
<          14 GLOL1L2
<          16 GALE1E5B
<          30 BEIDOU1B2B
<          5 LBAND
<          7
<          16 GPSL1L2PL2CL5
<          4 QZSSL1CAL2CL5
<          4 SBASL1
<          14 GLOL1L2PL2C
<          16 GALE1E5AE5BALTB0C
<          30 BEIDOU1B2B2B
<          5 LBAND
<          8
<          16 GPSL1L2PL2CL5
<          4 QZSSL1CAL2CL5
<          4 SBASL1L5
<          14 GLOL1L2PL2C
<          16 GALE1E5AE5BALTB0C
<          20 BEIDOU1B1CB2B2BB3
<          8 NAVICL5
<          5 LBAND
<          8
<          16 GPSL1L2PL2CL5L1C
<          4 QZSSL1CAL2CL5L1CL6
<          4 SBASL1L5
<          14 GLOL1L2PL2CL3
<          11 GALE1E5AE5BALTB0CE6
<          16 BEIDOU1B1CB2B2BB3
<          8 NAVICL5
<          5 LBAND
<          8
<          16 GPSL1L2PL2CL5
<          6 QZSSL1CAL2CL5
<          4 SBASL1L5
<          14 GLOL1L2PL2CL3
<          9 GALE1E5AE5BALTB0CE6
<          24 BEIDOU1B1CB2B2BB3
<          8 NAVICL5
<          5 LBAND
<          8
<          16 GPSL1L2PL2CL5
<          4 QZSSL1CAL2CL5
<          4 SBASL1L5
<          14 GLOL1L2PL2CL3
<          16 GALE1E5AE5BALTB0CE6
<          16 BEIDOU1B1CB2B2BB3

```



```
<      8 NAVICL5
<      5 LBAND
<      8
<     16 GPSL1L2PL2CL5
<      4 QZSSL1CAL2CL5L1CL6
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<     11 GALE1E5AE5BALTBOCE6
<     20 BEIDOUB1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<     16 GPSL1L2PL2CL5L1C
<      4 QZSSL1CAL2CL5L1C
<      4 SBASL1L5
<     14 GLOL1L2PL2CL3
<     16 GALE1E5AE5BALTBOC
<     20 BEIDOUB1B1CB2B2B
<      8 NAVICL5
<      5 LBAND
```

6. This log shows that the third set is selected. To further verify, enter **LOG TRACKSTAT** to show all the configured channels.

## 2.155 SEND

### *Sends an ASCII message to a COM port*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to send ASCII printable data from any of the COM or USB ports to a specified communications port. This is a one time command, therefore the data message must be preceded by the **SEND** command and followed by <CR> each time data is sent. If the data string contains delimiters (that is, spaces, commas, tabs and so on), the entire string must be contained within double quotation marks. Carriage return and line feed characters (for example, 0x0D, 0x0A) are appended to the sent ASCII data.

**Message ID: 177**

#### **Abbreviated ASCII Syntax:**

```
SEND [port] data
```

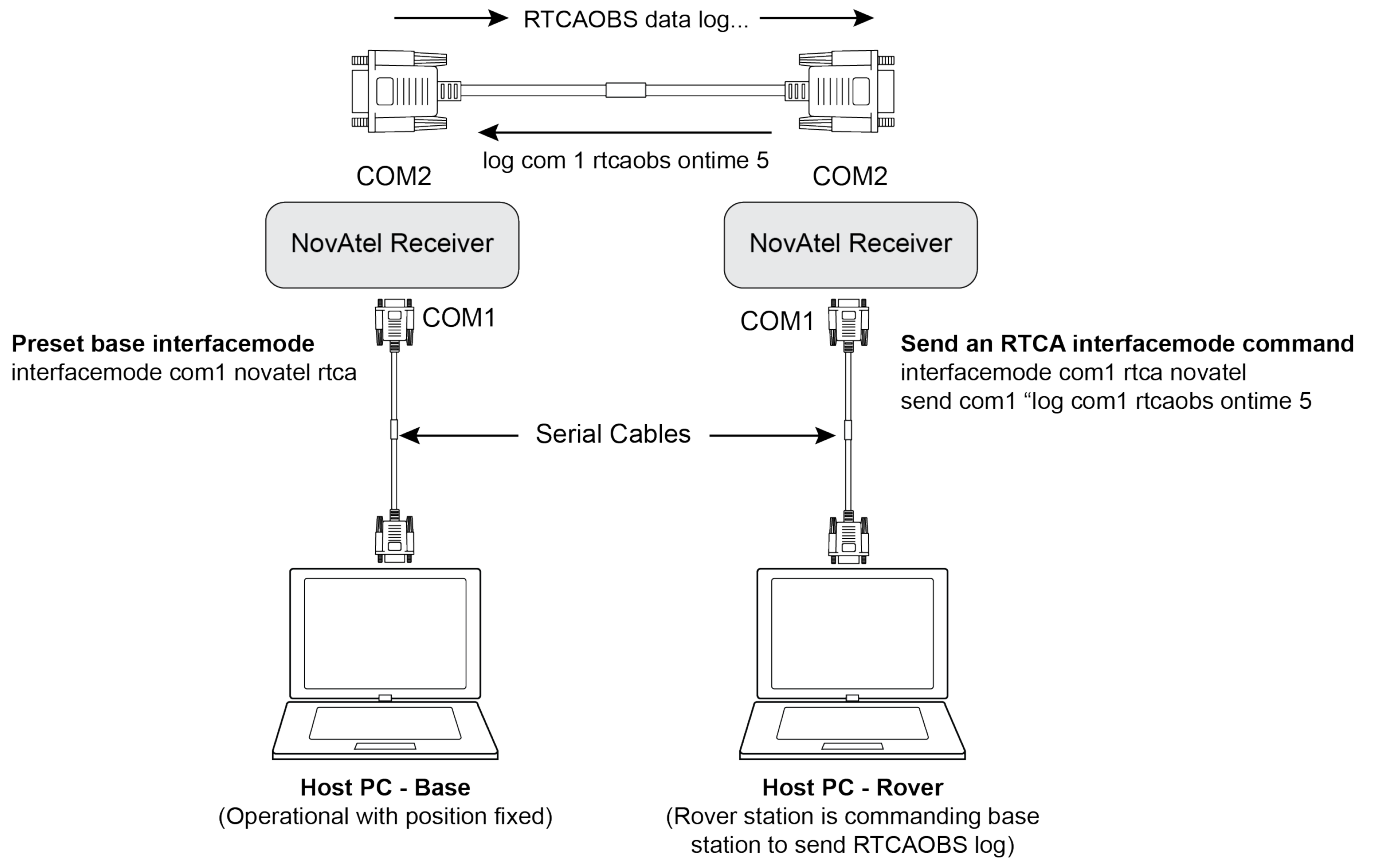
#### **ASCII Example**

```
SEND com1 "log com1 rtcaobs ontime 5"
```



**Scenario:** Assume you are operating receivers as base and rover stations. It could also be assumed that the base station is unattended but operational and you wish to control it from the rover station. From the rover station, you could establish the data link and command the base station receiver to send differential corrections.

**Figure 9: Using the SEND Command**



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SEND header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See Table 5: <i>Detailed Port Identifier</i> on page 40		Output port (default=THISPORT)	Enum	4	H
3	message	Max 100 character string (99 typed visible chars and a null char added by the firmware automatically)		ASCII data to send	String [max 100]	Variable <sup>1</sup>	H+4

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.156 SENDHEX

### *Send non-printable characters in hex pairs*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is like the **SEND** command (see page 380) except it is used to send non-printable characters expressed as hexadecimal pairs. Carriage return and line feed characters (for example, 0x0D, 0x0A) will not be appended to the sent data and so must be explicitly added to the data if needed.

**Message ID:** 178

#### Abbreviated ASCII Syntax:

```
SENDHEX [port] length data
```

#### Input Example:

```
SENDHEX COM1 6 143Ab5910D0A
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SENDHEX header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See Table 5: <i>Detailed Port Identifier</i> on page 40		Output port (default=THISPORT)	Enum	4	H
3	length	0 - 700		Number of hex pairs	Ulong	4	H+4
4	message	limited to a 700 maximum string (1400 pair hex). Even number of ASCII characters from set of 0-9, A-F. No spaces are allowed between pairs of characters		Data	String [max 700]	Variable <sup>1</sup>	H+8

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.157 SERIALCONFIG

### *Configures serial port settings*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to configure the receiver's asynchronous serial port communications drivers.



1. Also refer to the **ECHO** command on page 151.
2. The **SERIALCONFIG** command can be used as a log to confirm settings.
3. The entire content of the current log is sent before pausing due to the receipt of the XOFF character.

The current SERIALCONFIG port configuration can be reset to its default state by sending it two hardware break signals of 250 milliseconds each, spaced by fifteen hundred milliseconds (1.5 seconds) with a pause of at least 250 milliseconds following the second break. This will:

- Stop the logging of data on the current port (see the **UNLOGALL** command on page 454)
- Clear the transmit and receive buffers on the current port
- Return the current port to its default settings (see *Command Defaults* on page 59 for details)
- Set the interface mode to NovAtel for both input and output (see the **INTERFACEMODE** command on page 216)

This break detection can be disabled using the **SERIALCONFIG** command.



1. The **COMCONTROL** command (see page 124) may conflict with handshaking of the selected COM port. If handshaking is enabled, then unexpected results may occur.
2. Baud rates higher than 115,200 bps are not supported by standard PC hardware. Special PC hardware may be required for higher rates, including 230400 bps and 460800 bps. Avoid having COM ports of two receivers connected together using baud rates that do not match. Data transmitted through a port operating at a slower baud rate may be misinterpreted as break signals by the receiving port if it is operating at a higher baud rate because data transmitted at the lower baud rate is stretched relative to the higher baud rate. In this case, configure the receiving port to break detection disabled using the **SERIALCONFIG** command.



Use the **SERIALCONFIG** command before using the **INTERFACEMODE** command on each port. Turn break detection off using the **SERIALCONFIG** command to stop the port from resetting because it is interpreting incoming bits as a break command.

**Message ID: 1246**

#### **Abbreviated ASCII Syntax:**

```
SERIALCONFIG [port] baud [parity[databits[stopbits[handshaking[break]]]]]
```

#### **Factory Defaults:**

```
SERIALCONFIG COM1 9600 N 8 1 N ON
SERIALCONFIG COM2 9600 N 8 1 N ON
SERIALCONFIG COM3 9600 N 8 1 N ON
SERIALCONFIG COM4 9600 N 8 1 N ON
SERIALCONFIG COM5 9600 N 8 1 N ON
```

**ASCII Example:**

```
SERIALCONFIG com1 9600 n 8 1 n off
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SERIALCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See <i>Table 70: COM Port Identifiers</i> on the next page		Port to configure (default = THISPORT)	Enum	4	H
3	bps/ baud	2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 and 460800		Communication baud rate (bps).	Ulong	4	H+4
4	parity	See <i>Table 71: Parity</i> on the next page		Parity	Enum	4	H+8
5	data bits	7 or 8		Number of data bits (default = 8)	Ulong	4	H+12
6	stop bits	1 or 2		Number of stop bits (default = 1)	Ulong	4	H+16
7	handshake <sup>1</sup>	See <i>Table 72: Handshaking</i> on the next page		Handshaking	Enum	4	H+20
8	break	OFF	0	Disable break detection	Enum	4	H+24
		ON	1	Enable break detection (default)			

<sup>1</sup>The OEM719, CPT7 and SMART7 do not support hardware handshaking. Only transmit and receive lines exist for the OEM719, CPT7 and SMART7 ports.



**Table 70: COM Port Identifiers**

Binary	ASCII	Description	Applicable Receiver
1	COM1	COM port 1	OEM719, OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, SMART7
2	COM2	COM port 2	OEM719, OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, SMART7
3	COM3	COM port 3	OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, SMART7
6	THISPORT	The current COM port	OEM719, OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7
19	COM4	COM port 4	OEM7700, OEM7600, OEM7720
21	IMU	IMU COM port	dependent on hardware configuration
31	COM5	COM port 5	OEM7700, OEM7600, OEM7720
32	COM6	COM port 6	
33	BT1	Bluetooth COM port	dependent on hardware configuration
34	COM7	COM port 7	
35	COM8	COM port 8	
36	COM9	COM port 9	
37	COM10	COM port 10	

**Table 71: Parity**

Binary	ASCII	Description
0	N	No parity (default)
1	E	Even parity
2	O	Odd parity

**Table 72: Handshaking**

Binary	ASCII	Description
0	N	No handshaking (default)
1	XON	XON/XOFF software handshaking
2	CTS	CTS/RTS hardware handshaking

## 2.158 SERIALPROTOCOL

**Sets the protocol to be used by a serial port**

**Platform: OEM729, PwrPak7**

On some OEM7 receiver cards, selected ports can support either RS-232 or RS-422 signaling protocol. The default protocol is RS-232. The **SERIALPROTOCOL** command is used to select the protocol (RS-232 or RS-422) supported on the port.



RS-422/RS-232 selection is available only on COM1 of the OEM729 or COM1 and COM2 on the PwrPak7.

**Message ID: 1444**

**Abbreviated ASCII Syntax:**

```
SERIALPROTOCOL port protocol
```

**ASCII Example:**

```
SERIALPROTOCOL COM1 RS422
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SERIALPROTOCOL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	port	See <i>Table 73: Ports Supporting RS-422</i> on the next page		Select the COM port on which the protocol is being set. The port that can be entered depends on the hardware platform being used.	Enum	4	H
3	protocol	RS232	0	Set the port to use RS-232 protocol	Enum	4	H+4
		RS422	1	Set the port to use RS-422 protocol			



After switching a COM port from RS-232 to RS-422, send a carriage return (CR) on the newly configured port to flush the buffer prior to sending new commands on the port.

**Table 73: Ports Supporting RS-422**

<b>OEM7 Receiver Type</b>	<b>Allowable Ports</b>	<b>Binary Value</b>
OEM719	None	
OEM729	COM1	1
OEM7600	None	
OEM7700	None	
OEM7720	None	
PwrPak7	COM1	1
	COM2	2

## 2.159 SETADMINPASSWORD

### *Sets the administration password*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This command sets the administration password used to log into various web services.

- The administration password is required for Secure ICOM access.

The default admin password is the receiver PSN. For OEM7 enclosures, such as the PwrPak7, the default password is the enclosure PSN. The enclosure PSN is shown on the label on the bottom of the enclosure and in the ENCLOSURE line in the **VERSION** log (see page 984). The default password should be changed before connecting the receiver to a network.

**Message ID: 1579**

#### Abbreviated ASCII Syntax:

```
SETADMINPASSWORD oldpassword newpassword
```

#### Input example

```
SETADMINPASSWORD ABC123 XYZ789
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETADMINPASSWORD header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	OldPassword	Maximum 28 character string		Previous password.	String [28]	variable <sup>1</sup>	H
3	NewPassword	Maximum 28 character string		New password.	String [28]	variable <sup>1</sup>	variable



This password can be restored to default (the receiver PSN) by issuing the **FRESET USER\_ACCOUNTS** command (see *FRESET* on page 193).

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.160 SETAPPROXPOS

### *Sets an approximate position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets an approximate latitude, longitude and height in the receiver. Estimating these parameters, when used in conjunction with an approximate time (see the **SETAPPROXTIME** command on the next page), can improve satellite acquisition times and Time To First Fix (TTFF). For more information about TTFF and Satellite Acquisition, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

The horizontal position entered should be within 200 km of the actual receiver position. The approximate height is not critical and can normally be entered as zero. By default, if the receiver cannot calculate a valid position within 2.5 minutes of entering an approximate position, the approximate position is ignored.

The approximate position is not visible in any position logs. It can be seen by issuing a SETAPPROXPOS log.

**Message ID: 377**

#### Abbreviated ASCII Syntax:

```
SETAPPROXPOS lat lon height
```

#### Input Example:

```
SETAPPROXPOS 51.116 -114.038 0
```



For an example on the use of this command, refer to the **SETAPPROXTIME** command on the next page.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETAPPROXPOS header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Lat	± 90 degrees		Approximate latitude	Double	8	H
3	Lon	± 180 degrees		Approximate longitude	Double	8	H+8
4	Height	-1000 to +20000000 m		Approximate height	Double	8	H+16

## 2.161 SETAPPROXTIME

### *Sets an approximate GPS reference time*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets an approximate time in the receiver. The receiver uses this time as system time until a coarse time can be acquired. This can be used in conjunction with an approximate position (see the **SETAPPROXPOS** command on the previous page) to improve Time To First Fix (TTFF). For more information TTFF and Satellite Acquisition, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).



The time entered should be within 10 minutes of the actual GPS reference time. If the week number entered does not match the broadcast week number, the receiver resets once it is tracking.

**Message ID: 102**

#### **Abbreviated ASCII Syntax:**

```
SETAPPROXTIME week sec
```

#### **Input Example:**

```
SETAPPROXTIME 1930 501232
```



Upon power up, the receiver does not know its position or time and therefore cannot use almanac information to aid satellite acquisition. You can set an approximate GPS reference time using the **SETAPPROXPOS** command (see page 389).

Approximate time and position may be used in conjunction with a current almanac to aid satellite acquisition. See the table below for a summary of the OEM7 family commands used to inject an approximated time or position into the receiver:

Approximate	Command
Time	SETAPPROXTIME
Position	SETAPPROXPOS

Base station aiding can help in these environments. A set of ephemerides can be injected into a rover station by broadcasting the RTCAEPHEM message from a base station. This is also useful in environments where there is frequent loss of lock. GPS ephemeris is three frames long within a sequence of five frames. Each frame requires 6 s of continuous lock to collect the ephemeris data. This gives a minimum of 18 s and a maximum of 36 s continuous lock time or when no recent ephemerides (new or stored) are available.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETAPPROXTIME header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	week	0-9999		GPS reference week number	Ulong	4	H
3	sec	0-604800		Number of seconds into GPS reference week	Double	8	H+4

## 2.162 SETBASERECEIVERTYPE

### Sets base receiver type

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command allows the user to specify the base receiver type to aid GLONASS ambiguity fixing in RTK. It can be used as a substitute for RTCM1033 messages that contains the information on the base receiver type. This command should be issued to the Rover.



An incorrect base type setting can significantly impair ambiguity resolution.

**Message ID:** 1374

#### Abbreviated ASCII Syntax:

```
SETBASERECEIVERTYPE base_type
```

#### Factory Default:

```
SETBASERECEIVERTYPE unknown
```

#### ASCII Example:

```
SETBASERECEIVERTYPE novatel
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETBASERECEIVERTYPE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	base_type	unknown	0	Unknown Base	Enum	4	H
		novatel	1	NovAtel Base			
		trimble	2	Trimble Base			
		topcon	3	Topcon Base			
		magellan	4	Magellan Base			
		leica	5	Leica Base			



## 2.163 SETBESTPOSCRITERIA

### Sets selection criteria for BESTPOS

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to set the criteria for the **BESTPOS** log (see page 520) and choose between 2D and 3D standard deviation to obtain the best position from the **BESTPOS** log (see page 520). It also allows you to specify the number of seconds to wait before changing the position type. This delay provides a single transition that ensures position types do not skip back and forth.



The **SETBESTPOSCRITERIA** command is also used as the basis for the **UALCONTROL** command (see page 441) standard deviations.

**Message ID:** 839

#### Abbreviated ASCII Syntax:

```
SETBESTPOSCRITERIA type [delay]
```

#### Factory Default:

```
SETBESTPOSCRITERIA pos3d 0
```

#### Input Example:

```
SETBESTPOSCRITERIA pos2d 5
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETBESTPOSCRITERIA header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	type	See <i>Table 74: Selection Type</i> on the next page		Select a 2D or 3D standard deviation type to obtain the best position from the BESTPOS log	Enum	4	H
3	delay	0 to 100 s		Set the number of seconds to wait before changing the position type. Default=0	Ulong	4	H+4

**Table 74: Selection Type**

<b>ASCII</b>	<b>Binary</b>	<b>Description</b>
POS3D	0	3D standard deviation
POS2D	1	2D standard deviation

## 2.164 SETDIFFCODEBIASES

### *Sets satellite differential code biases*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2



Changing the biases may negatively affect positioning accuracy. NovAtel recommends that only advanced users modify the biases.

Use this command to set the differential code biases that correct pseudorange errors affecting the L1/L2 ionospheric corrections. Bias values are restricted to between -10 ns and +10 ns. A set of biases is included in the firmware and use of the biases is enabled by default.

The receiver uses the C/A code on L1 and the P code on L2 to calculate a dual-frequency ionospheric correction. However, the GNSS clock corrections are broadcast as if the P codes on both L1 and L2 are used to calculate this correction. The biases account for the differences between the P and C/A codes on L1 and improve the estimate of the ionospheric correction.

The biases are calculated by the International GNSS Service (IGS). Calculation details, analysis and results are available at <http://aiuws.unibe.ch/spec/dcb.php>. The most recent 30 day average bias values can be downloaded from [http://ftp.aiub.unibe.ch/CODE/CODE\\_FULL.DCB](http://ftp.aiub.unibe.ch/CODE/CODE_FULL.DCB).

**Message ID: 687**

#### **Abbreviated ASCII Syntax:**

```
SETDIFFCODEBIASES bias_type biases
```

#### **ASCII Example:**

```
<SETDIFFCODEBIASES COM1 2 91.0 UNKNOWN 0 0.470 02440020 365b 32768
<  GPS_C1P1 1.302 -1.326 1.360 1.649 1.357 1.586 0.776 -0.079 -0.123 0.888 -
0.321 0.718 0.527 -0.720 1.193 -1.331 0.828 -1.061 -2.497 -2.106 -1.979 -2.747
-0.254 1.202 -0.716 0.077 -0.180 -1.059 1.269 -0.481 0.734 1.516 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000
<SETDIFFCODEBIASES COM1 1 91.0 UNKNOWN 0 0.471 02440020 365b 32768
<  GLONASS_C1P1 -0.092 0.381 0.581 1.033 0.642 -0.561 0.794 0.899 0.380 -0.832
-0.358 -0.606 -2.181 0.023 1.135 0.346 0.009 0.384 -1.394 0.224 -0.022 -0.824 -
0.133 -0.437 0.000 0.608 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
<SETDIFFCODEBIASES COM1 0 91.0 UNKNOWN 0 0.473 02440020 365b 32768
<  GPS_C2P2 1.358 0.000 -0.381 0.000 -0.344 -0.707 0.306 -1.068 0.624 1.480
0.000 -0.401 0.000 0.000 -0.169 0.0 00 0.236 0.000 0.000 0.000 0.000 0.000
0.000 0.051 -0.711 1.082 -0.128 0.000 -0.101 -0.483 -0.630 -0.015 0.000 0.0 00
0.000 0.000 0.000 0.000 0.000 0.000
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETDIFFCODEBIASES header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	bias_type	GPS_C1P1	0	Code pair to which biases refer	Enum	4	H
		GPS_C2P2	1				
		GLONASS_C1P1	2				
3	biases	-10 to +10 ns		Array of 40 biases (ns)	Float[40]	160	H+4

## 2.165 SETIONOTYPE

### *Enables ionospheric models*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to set which ionospheric corrections model the receiver should use. If the selected model is not available, the receiver reverts to AUTO.



L1 only models automatically use SBAS ionospheric grid corrections, if available.

**Message ID:** 711

#### Abbreviated ASCII Syntax:

```
SETIONOTYPE model
```

#### Factory Default:

```
SETIONOTYPE auto
```

#### ASCII Example:

```
SETIONOTYPE Klobuchar
```



An ionotype of AUTO is recommended for PDP and GLIDE.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETIONOTYPE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	model	NONE	0	Do not use ionosphere modeling	Enum	4	H
		KLOBUCHAR	1	Use the Klobuchar model broadcast by GPS			
		GRID	2	Use the SBAS grid model			
		L1L2	3	Multi-frequency computed			
		AUTO	4	Automatically determine the ionospheric model to use			

## 2.166 SETNAV

### *Sets start and destination waypoints*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command permits entry of one set of navigation waypoints (see *Figure 10: Illustration of SETNAV Parameters* below). The origin (from) and destination (to) waypoint coordinates entered are considered on the ellipsoidal surface of the current datum (default wgs84). Once **SETNAV** has been set, monitor the navigation calculations and progress by observing messages in the **NAVIGATE** log (see page 725).

Track offset is the perpendicular distance from the great circle line drawn between the from lat-lon and to lat-lon waypoints. It establishes the desired navigation path or track, that runs parallel to the great circle line, which now becomes the offset track, and is set by entering the track offset value in metres. A negative track offset value indicates that the offset track is to the left of the great circle line track. A positive track offset value (no sign required) indicates the offset track is to the right of the great circle line track (looking from origin to destination). See *Figure 10: Illustration of SETNAV Parameters* below for clarification.

**Message ID: 162**

#### **Abbreviated ASCII Syntax:**

```
SETNAV fromlat fromlon tolat tolon trackoffset from-point to-point
```

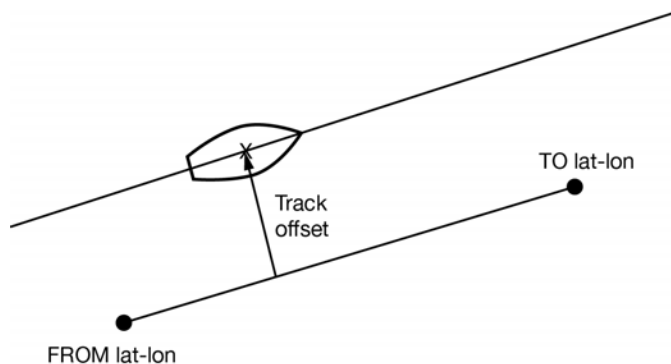
#### **Factory Default:**

```
SETNAV 90.0 0.0 90.0 0.0 0.0 from to
```

#### **ASCII Example:**

```
SETNAV 51.1516 -114.16263 51.16263 -114.1516 -125.23 FROM TO
```

**Figure 10: Illustration of SETNAV Parameters**



Consider the case of setting waypoints in a deformation survey along a dam. The surveyor enters the From and To point locations, on either side of the dam using the **SETNAV** command. They then use the NAVIGATE log messages to record progress and show where they are in relation to the From and To points.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETNAV header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	fromlat	± 90 degrees		Origin latitude in units of degrees/decimal degrees. A negative sign for South latitude. No sign for North latitude	Double	8	H
3	fromlon	± 180 degrees		Origin longitude in units of degrees/decimal degrees. A negative sign for West longitude. No sign for East longitude	Double	8	H+8
4	tolat	± 90 degrees		Destination latitude in units of degrees/decimal degrees	Double	8	H+16
5	tolon	± 180 degrees		Destination longitude in units of degrees/decimal degrees	Double	8	H+24
6	trackoffset	± 1000 km		Waypoint great circle line offset (in metres) establishes offset track. Positive indicates right of great circle line and negative indicates left of great circle line	Double	8	H+32
7	from-point	5 characters maximum		ASCII origin station name	String [max 5]	Variable <sup>1</sup>	H+40

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.



---

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
8	to-point	5 characters maximum		ASCII destination station name	String [max 5]	Variable <sup>1</sup>	Variable

## 2.167 SETROVERID

### Set ID for ALIGN rovers

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command sets the Rover ID output in the **ROVERPOS** log (see page 851), **HEADING2** log (see page 646), **ALIGNBSLNXYZ** log (see page 485) and **ALIGNBSLNENU** log (see page 483).

The default value for the ID is set using the last seven characters of the receiver PSN Number. For example, if the receiver PSN number is DMMU20240186H, the ID is set as Y86H, i.e., 24 is represented as Y and last three characters are filled in as is. The fourth and fifth last characters are ignored.



It is not guaranteed that each receiver will have a unique auto-generated ID. Use this command to set the ID in case the auto-generated ID overlaps with other rovers. It is the user's responsibility to ensure each receiver ID is unique (if they own multiple receivers). If the ID overlaps, use this command to set the ID.

**Message ID:** 1135

#### Abbreviated ASCII Syntax:

```
SETROVERID rovid
```

#### Factory Default:

If the receiver PSN is: DAB07170027

```
SETROVERID R027
```

#### Input Example

```
SETROVERID rovl
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETROVERID header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	rovid	4 Character String e.g., ROV1		ID String (maximum 4 characters plus NULL)	String[5]	5 <sup>1</sup>	H

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.168 SETTIMEBASE

### *Sets primary and backup systems for time base*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command configures the primary and backup steering system(s) for timing. The primary system is the system that the receiver steers the clock to. Upon startup, the primary system must be present long enough to steer the clock to be valid once, otherwise, the backup system cannot be used. The backup system is used whenever the primary system is not present.

**Message ID: 1237**

#### Abbreviated ASCII Syntax:

```
SETTIMEBASE primarysystem numbackups[system[timeout]]
```

#### Factory Default:

For GLONASS only receiver:

```
SETTIMEBASE glonass 0
```

For GPS capable receiver:

```
SETTIMEBASE gps 1 auto 0
```


For BeiDou only receiver:

```
SETTIMEBASE beidou 0
```

#### Input Example:

```
SETTIMEBASE gps 1 glonass 30
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETTIMEBASE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	primarysystem	<i>Table 75: System Used for Timing on the next page</i>		The primary system for steering the receiver clock	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	numbackups	0 or 4		<p>The number of records to follow.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  When more than one backup system is specified, the backup systems are selected according to numeric order. </div>	Ulong	4	H+4
4	system <sup>1</sup>	<i>Table 75: System Used for Timing below</i>		The system to be used for backup	Enum	4	H+8
5	timeout <sup>1</sup>	0 to +4294967295 (seconds)		Duration that the backup system is used to steer the clock. 0 means ongoing	Ulong	4	H+12

**Table 75: System Used for Timing**

Binary	ASCII
0	GPS <sup>2</sup>
1	GLONASS
2	GALILEO
3	BEIDOU
4	NAVIC
99	AUTO <sup>3</sup>

<sup>1</sup>The *system* and *timeout* fields can repeat.

<sup>2</sup>GPS setting includes QZSS satellites.

<sup>3</sup>AUTO is used only as a backup system (not available for primary system field).

## 2.169 SETTROPOMODEL

### *Sets Troposphere model*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the troposphere model used to correct ranges used in the PSRPOS and PDPPOS solutions.

**Message ID:** 1434

**Abbreviated ASCII Syntax:**

```
SETTROPOMODEL model
```

**Factory Default:**

```
SETTROPOMODEL auto
```

**Input Example:**

```
SETTROPOMODEL none
```



Disabling the troposphere model may negatively affect positioning accuracy. NovAtel recommends that only advanced users modify this setting.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETTROPOMODEL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	model	NONE	1	Do not apply any troposphere corrections	Enum	4	H
		AUTO	2	Automatically use an appropriate model			

## 2.170 SETUTCLEAPSECONDS

### *Sets future leap seconds*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command allows the user to force the UTC offset to be updated according to the input date.

Leap seconds will occur at the end of the UTC day specified. The receiver will use the leap second set by this command until a leap second can be obtained over the air.

**Message ID: 1150**

#### Abbreviated ASCII Syntax:

```
SETUTCLEAPSECONDS seconds [futureweek [futureday [futureseconds]]]
```

#### Input Example:

```
SETUTCLEAPSECONDS 18 1929 7 18
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETUTCLEAPSECONDS header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Seconds <sup>1</sup>	0 – 4294967295		Current UTC leap second	Ulong	4	H
3	Futureweek	0 – 10000		GPS Week when future leap seconds will take effect	Ulong	4	H+4
4	Futureday	1 – 7		Day of the week when future leap seconds will take effect	Ulong	4	H+8
5	Futureseconds	0 – 4294967295		Future leap second offset that will take effect at the end of the futuredaynumber of the futureweeknumber	Ulong	4	H+12

<sup>1</sup>This value will only be applied if the UTC status in the TIME log is not Valid.

## 2.171 SOFTLOADCOMMIT

### *Completes the SoftLoad process*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command completes the SoftLoad process by verifying the downloaded image and activating it. Refer to Updating Using SoftLoad Commands in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for more information about the SoftLoad process.

This command can only be sent to the receiver when the **SOFTLOADSTATUS** log (see page 951) reports READY\_FOR\_DATA.

After issuing the **SOFTLOADCOMMIT** command the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 300 seconds from the time the command was received by the receiver. If an error response is returned, consult the **SOFTLOADSTATUS** log on page 951 for more detail.

**Message ID:** 475

#### **Abbreviated ASCII Syntax:**

```
SOFTLOADCOMMIT
```

#### **Input Example:**

```
SOFTLOADCOMMIT
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SOFTLOADCOMMIT header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Reserved	-		Reserved. Set to 1 in the binary case	Enum	4	H

## 2.172 SOFTLOADDATA

*Sends firmware image data to the receiver for the SoftLoad process*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is only valid in binary mode.

This command is used to upload data to the receiver for the SoftLoad process. Refer to Updating Using SoftLoad Commands in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for more information about the SoftLoad process.

After each **SOFTLOADDATA** command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the **SOFTLOADSTATUS** log on page 951 for more detail.

This command can only be sent to the receiver after the **SOFTLOADSREC** command (see page 414) or **SOFTLOADSETUP** command (see page 412) have sent the content of the S0 records from the start of a firmware \*.shex file. In these cases, the **SOFTLOADSTATUS** log (see page 951) reports READY\_FOR\_SETUP or READY\_FOR\_DATA.

**Message ID: 1218**

**Abbreviated ASCII Syntax:**

Not applicable

Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	SOFTLOADDATA header NovAtel binary message header	-	H	0
2	offset	-	Offset of the data within the downloaded image	Ulong	4	H
3	data length	-	Number of bytes of data. This must match the number of bytes contained within the "data" field	Ulong	4	H+4
4	data	-	Incoming data up to a maximum of 4096 bytes	Uchar	4096	H+8



## 2.173 SOFTLOADFILE

*Updates firmware using file devices of the receiver*

Platform: PwrPak7



The **SOFTLOADFILE** command applies only to receivers with internal data storage.



To use the **SOFTLOADFILE** command, the option must be enabled using the **IPSERVICE** command (see page 225).

Receivers with storage for data logging can also store firmware images to be flashed onto the receiver. Using FTP, .shex or .sbin files are uploaded and the **SOFTLOADFILE** command programs a firmware file to be used on next boot. ASCII **.shex** files contain the required metadata to be used directly without further user input. However, binary **.sbin** files do not and require other SOFTLOAD commands to reset the process and set the Version, Platform, and Type values. See [docs.novatel.com/OEM7/Content/Firmware\\_Update/Updating\\_Using\\_SoftLoad.htm](https://docs.novatel.com/OEM7/Content/Firmware_Update/Updating_Using_SoftLoad.htm)



Use of binary images is recommended as binary images (.sbin) take significantly less time to flash than ASCII images (.shex). Binary image files are available on request from NovAtel Support.

**Message ID: 1302**

**Abbreviated ASCII Syntax:**

```
SOFTLOADFILE StorageDevice Filename [Reserved]
```

**ASCII Examples:**

```
SOFTLOADFILE INTERNAL_FLASH OM7MR0705RN0000.sbin
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	SOFTLOADFILE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	–	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	StorageDevice	USBSTICK	1	This allows .shex files to be flashed on to the receiver from an attached USB stick containing the files.	Enum	4	H
		INTERNAL_FLASH	4	The source device on which the firmware file is stored.			
3	Filename	Character string		File name of the firmware file. (128 characters maximum)	Char (max 128)	Variable <sup>1</sup>	H+4
4	Reserved				Int	4	H+132

### Binary File Example:

This example shows how to load the binary file EP7PR0705RN0000.sbin.

1. Upload the EP7PR0705RN0000.sbin firmware file via FTP.
2. Enter `LOG SOFTLOADSTATUS ONNEW`.
3. Enter `SOFTLOADRESET`.
4. Enter `SOFTLOADSETUP DATATYPE APP`.
5. Enter `SOFTLOADSETUP VERSION EP7PR0705RM0000` (version string).
6. Enter `SOFTLOADSETUP PLATFORM OEM7700` (must match receiver platform ID).
7. Enter `SOFTLOADFILE INTERNAL_FLASH EP7PR0705RN0000.sbin`.
8. Enter `RESET` when the SoftLoad status shows **COMPLETE**.

### ASCII File Example:

This example shows how to load the ASCII file EP7PR0705RN0000.shex.

1. Upload the EP7PR0705RN0000.shex firmware file via FTP.
2. Enter `LOG SOFTLOADSTATUS ONNEW`.
3. Enter `SOFTLOADFILE INTERNAL_FLASH EP7PR0705RN0000.shex`.
4. Enter `RESET` when the SoftLoad status shows **COMPLETE**.

<sup>1</sup>In the binary case, the string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.174 SOFTLOADRESET

### *Initiates a new SoftLoad process*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command restarts the SoftLoad process. Refer to Updating Using SoftLoad Commands in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for more information about the SoftLoad process.

The command does not affect the flash and does not reset the receiver.

The **SOFTLOADRESET** command can be issued at any time. If it is issued while a SoftLoad process is currently in progress then that process is terminated and a new one is started. After the **SOFTLOADRESET** command is processed the SOFTLOADSTATUS log will report a status of READY\_FOR\_SETUP.

After issuing the **SOFTLOADRESET** command the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 300 seconds from the time the command was received by the receiver. If an error response is returned, consult the **SOFTLOADSTATUS** log on page 951 for more detail.

**Message ID: 476**

#### Abbreviated ASCII Syntax:

```
SOFTLOADRESET
```

#### Input Example:

```
SOFTLOADRESET
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SOFTLOADRESET header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Reserved	-	-	Reserved. Set to 1 in the binary case	Enum	4	H

## 2.175 SOFTLOADSETUP

**Sends configuration information to the receiver for the SoftLoad process**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **SOFTLOADSETUP** command can be used in place of the **SOFTLOADSREC** command when sending S0 Records. This command is meant to be used if the user requires that the entire SoftLoad process be performed in binary, but can also be used in ASCII or abbreviated ASCII. The examples below are given in abbreviated ASCII for simplicity.

Refer to Updating Using SoftLoad Commands in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for more information about the SoftLoad process.

This command can only be sent to the receiver when the SOFTLOADSTATUS log reports READY\_FOR\_SETUP.

After each **SOFTLOADSETUP** command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the **SOFTLOADSTATUS** log on page 951 for more detail.

NovAtel S0 records use the following format: **S0~X~<<DATA>>**, where **X** is the Setup Type and **<<DATA>>** is a NULL terminated string. To convert from S0 record to the SOFTLOADSETUP command, convert the Setup Type to the appropriate Setup type enumeration, as described in *Table 76: Available Set Up Commands* on the next page, and copy the **<<DATA>>** string in to the Setup data string.

**Message ID: 1219**

**Abbreviated ASCII Syntax:**

```
SOFTLOADSETUP setuptype setupdata
```

**Input Example:**

```
SOFTLOADSETUP datatype "APP"
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SOFTLOADSETUP header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	Setuptype	See Table 76: Available Set Up Commands on the next page		The type of setup command	Enum	4	H
3	Setupdata	-	-	ASCII setup data string. See Table 76: Available Set Up Commands below for details on this data. This data can be pulled from the S0 records of the hex file being loaded onto the receiver. If the ASCII form of this command is used, this string must be enclosed in double quotes (" ")	String [512]	variable <sup>1</sup>	H+4

Table 76: Available Set Up Commands

Binary	ASCII	Description
1	Platform	Comma separated list of platforms supported by the data to be uploaded. This corresponds to S0~P~. For example, the S-Record S0~P~OEM729,OEM7700,OEM719, translates to SOFTLOADSETUP PLATFORM "OEM729,OEM7700,OEM719"
2	Version	Version of the data to be uploaded. This corresponds to S0~V~. For example, the S-Record S0~V~OMP070400RN0000, translates to SOFTLOADSETUP VERSION "OMP070400RN0000"
3	Datatype	Intended data block for the data to be uploaded. This corresponds to S0~T~. For example, the S-Record S0~T~APP, translates to SOFTLOADSETUP DATATYPE "APP"
4	Authcode	PSN and AUTH code for the data to be uploaded. The format is: PSN:AuthCode. Note that since there are commas within the AuthCode, double quotes must surround the PSN:AuthCode string. For example: SOFTLOADSETUP AUTHCODE "BFN10260115:T48JF2,W25DBM,JH46BJ,2WGHMJ,8JW5TW,G2SR0RCCR,101114"

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.176 SOFTLOADSREC

**Sends an S-Record to the receiver for the SoftLoad process**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to send S-Records to the receiver for the SoftLoad process. Refer to Updating Using SoftLoad Commands in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for more information about the SoftLoad process.

After each **SOFTLOADDATA** command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the **SOFTLOADSTATUS** log on page 951 for more detail.

This command can only be sent to the receiver when the SOFTLOADSTATUS log reports READY\_FOR\_SETUP or READY\_FOR\_DATA.

**Message ID: 477**

**Abbreviated ASCII Syntax:**

```
SOFTLOADSREC s-record
```

**Input Example:**

```
SOFTLOADSREC "S30900283C10FAA9F000EF"
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SOFTLOADSREC header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	s-record	-		ASCII S-Record string copites from firmware *.shex file	String [515]	variable <sup>1</sup>	H
3	Reserved	-	1	Reserved. Set to 1 in the binary case	Ulong	4	variable

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 2.177 STATUSCONFIG

### *Configures RXSTATUSEVENT mask fields*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to configure the various status mask fields in the **RXSTATUSEVENT** log (see page 891). These masks can modify whether various status fields generate errors or event messages when they are set or cleared.

Receiver Errors automatically generate event messages. These event messages are output in **RXSTATUSEVENT** log (see page 891). It is also possible to have status conditions trigger event messages to be generated by the receiver. This is done by setting/clearing the appropriate bits in the event set/clear masks. The set mask tells the receiver to generate an event message when the bit becomes set. Likewise, the clear mask causes messages to be generated when a bit is cleared. To disable all these messages without changing the bits, simply UNLOG the **RXSTATUSEVENT** log (see page 891) on the appropriate ports. Refer also to Built-In Status Tests in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

**Message ID: 95**

#### **Abbreviated ASCII Syntax:**

```
STATUSCONFIG type word mask
```

#### **Factory Default:**

```
STATUSCONFIG PRIORITY STATUS 0
STATUSCONFIG PRIORITY AUX1 0x00000008
STATUSCONFIG PRIORITY AUX2 0
STATUSCONFIG SET STATUS 0x00000000
STATUSCONFIG SET AUX1 0
STATUSCONFIG SET AUX2 0
STATUSCONFIG SET AUX4 0xFFFFFFFF
STATUSCONFIG CLEAR STATUS 0x00000000
STATUSCONFIG CLEAR AUX1 0
STATUSCONFIG CLEAR AUX2 0
STATUSCONFIG CLEAR AUX4 0
```

#### **ASCII Example:**

```
STATUSCONFIG SET STATUS 0028A51D
```



The receiver gives the user the ability to determine the importance of the status bits. In the case of the Receiver Status, setting a bit in the priority mask causes the condition to trigger an error. This causes the receiver to idle all channels, set the ERROR strobe line, flash an error code on the status LED, turn off the antenna (LNA power) and disable the RF hardware, the same as if a bit in the Receiver Error word is set. Setting a bit in an Auxiliary Status priority mask causes that condition to set the bit in the Receiver Status word corresponding to that Auxiliary Status.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	STATUSCONFIG header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	type	PRIORITY	0	Replace the Priority mask	Enum	4	H
		SET	1	Replace the Set mask			
		CLEAR	2	Replace the Clear mask			
3	word	STATUS	1	Receiver Status word	Enum	4	H+4
		AUX1	2	Auxiliary 1 Status word			
		AUX2	3	Auxiliary 2 Status word			
		AUX3	4	Auxiliary 3 Status word			
		AUX4	5	Auxiliary 4 Status word			
4	mask	8 digit hexadecimal		The hexadecimal bit mask	Ulong	4	H+8



## 2.178 STEADYLINE

### *Configures position mode matching*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

The STEADYLINE functionality helps mitigate the discontinuities that often occur when a GNSS receiver changes positioning modes. The effect is especially evident when a receiver transitions from an RTK position mode solution to a lower accuracy “fall back” solution, such as DGPS, WAAS+GLIDE or even autonomous GLIDE. Smooth transitions are particularly important for agricultural steering applications where sudden jumps may be problematic.

The STEADYLINE internally monitors the position offsets between all the positioning modes present in the receiver. When the receiver experiences a position transition, the corresponding offset is applied to the output position to limit a potential real position jump. When the original accurate position type returns, the STEADYLINE algorithm will slowly transition back to the new accurate position at a default rate of 0.005 m/s. This creates a smoother pass-to-pass relative accuracy at the expense of a possible degradation of absolute accuracy.

For example, a receiver can be configured to do both RTK and GLIDE. If this receiver has a fixed RTK position and experiences a loss of correction data causing the loss of the RTK solution it will immediately apply the offset between the two position modes and uses the GLIDE position stability to maintain the previous trajectory. Over time the GLIDE (or non-RTK) position will experience some drift. Once the RTK position is achieved again the receiver will start using the RTK positions for position stability and will slowly transition back to the RTK positions at a default rate of 0.005 m/s.

**Message ID: 1452**

#### **Abbreviated ASCII Syntax:**

```
STEADYLINE mode [transition_time]
```

#### **Factory Default:**

```
STEADYLINE disable
```

#### **ASCII Example:**

```
STEADYLINE prefer_accuracy 100
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	STEADYLINE header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	mode	See Table 77: STEADYLINE Mode on the next page		STEADYLINE mode	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	transition_time			Time over which solutions will transition in seconds. The minimum rate of change is 0.005 m/s regardless of this parameter.  Default = -1, which sets the parameter to the minimum value 0.005m/s.	Ulong	4	H+4

Table 77: STEADYLINE Mode

ASCII	Binary	Description
DISABLE	0	Disable STEADYLINE (default)
RESET	3	Reset the offsets, jump immediately to the current position and continue in the current mode.
PREFER_ACCURACY	4	Slowly transition to the new reference position when changing from less accurate reference positioning type to more accurate reference positioning type. Maintains the position offset calculated when changing from more accurate reference positioning type to a less accurate reference positioning type.

## 2.179 STEADYLINEDIFFERENTIALTIMEOUT

**Sets how long the receiver will report RTK/PPP after corrections are lost**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to set how long STEADYLINE will report RTK or PPP solutions after a loss of corrections. If able, STEADYLINE will report an RTK or PPP solution until this timeout expires or until the RTK/PPP timeout expires, whichever is higher.

For example:

- If the **RTKTIMEOUT** is 60 seconds and the **STEADYLINEDIFFERENTIALTIMEOUT** is 300 seconds, STEADYLINE will report an RTK solution for 300 seconds.
- If the **RTKTIMEOUT** is 60 seconds and the **STEADYLINEDIFFERENTIALTIMEOUT** is 30 seconds, STEADYLINE will report an RTK solution for 60 seconds.

**Message ID: 2002**

**Abbreviated ASCII Syntax:**

```
STEADYLINEDIFFERENTIALTIMEOUT timeout
```

**Factory Default:**

```
STEADYLINEDIFFERENTIALTIMEOUT 60
```

**ASCII Example:**

```
STEADYLINEDIFFERENTIALTIMEOUT 300
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	STEADYLINEDIFFERENTIALTIMEOUT header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	timeout	5 to 1200		Timeout period in seconds	Float	4	H

## 2.180 SURVEYPOSITION

### *Saves or deletes a surveyed position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this command to add or delete a surveyed position saved in the receiver NVM.

The surveyed positions added or deleted with this command are used in conjunction with the **AUTOSURVEY** command on page 87.

**Message ID: 1952**

#### **Abbreviated ASCII Syntax:**


```
SURVEYPOSITION option id [latitude] [longitude] [height] [tolerance]
```

#### **ASCII Examples:**

```
SURVEYPOSITION save auto 51.116 -114.038 1065.0 10.0
```

```
SURVEYPOSITION delete cal2
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	SURVEYPOSITION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	SAVE	1	Save the surveyed position in the receiver NVM	Enum	4	H
		DELETE	2	Delete the surveyed position from the receiver NVM			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	id	4 character string		<p>ID for the saved position</p> <p>When saving a position, "AUTO" can be entered and the receiver will automatically generate a unique ID for the position. "AUTO" cannot be used when deleting a position.</p> <p>To determine the ID for a saved position, use the <b>SAVEDSURVEYPOSITIONS</b> log on page 910.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  In the Binary case, the ID string must be null terminated and additional bytes of padding must be added to make the total length of the field 8 bytes. </div>	String[5]	8	H+4
4	latitude	-90 to 90		<p>Latitude of the position in degrees (default=0.0)</p> <p>A "-" sign denotes south and a "+" sign denotes north</p>	Double	8	H+12
5	longitude	-360 to 360		<p>Longitude of the position in degrees (default=0)</p> <p>A "-" sign denotes west and a "+" sign denotes east</p>	Double	8	H+20
6	height	-1000 to 20000000		<p>Mean Sea Level height of the position in metres (default=0.0)</p>	Double	8	H+28

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	tolerance	3 - 100		Position tolerance in metres (default=10.0) The maximum distance between the position calculated during an self-survey and the saved position. During the self-survey, if the distance between the calculated position and the previously surveyed position is less than this value, the previous position is used.	Double	8	H+36

## 2.181 TECTONICSCOMPENSATIONSOURCE

*Chooses the model to use for tectonics compensation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to enable tectonics (plate motion) compensation.

**Message ID: 2290**

**Abbreviated ASCII Syntax:**

```
TECTONICSCOMPENSATIONSOURCE model
```

**Factory Default:**

```
TECTONICSCOMPENSATIONSOURCE plate_motion_model
```

**ASCII Example:**

```
TECTONICSCOMPENSATIONSOURCE none
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	TECTONICS COMPENSATION SOURCE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	model	NONE	0	Do not use a tectonics model.	Enum	4	H
		PLATE_MOTION_MODEL	1	Automatically determine the plate the receiver is on and use the plate motion model of that plate.			
3	reserved				Char[32]	variable	H+4

## 2.182 TERRASTARAUTOCHANCONFIG

### *Enable channel configurations granted by eligible TerraStar subscriptions*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This command allows the receiver to use a subscription managed channel configuration while an eligible TerraStar or Oceanix subscription is active. This enables the receiver to make the best use of the subscription while computing PPP position solutions. Subscription managed multi-constellation, multi-frequency channel configurations give better accuracy and faster convergence time compared to a dual-constellation, dual-frequency channel configuration.



Subscription Managed Channel Configuration (SMCC) has precedence over **SELECTCHANCONFIG**. When the receiver is reset, any user selected channel configuration will be overwritten by the channel configuration chosen by SMCC.

To enable this feature, issue the **TERRASTARAUTOCHANCONFIG ENABLE** command. If an eligible subscription is active, as reflected in the **TERRASTARINFO** log (see page 959) or **OCEANIXINFO** log (see page 732), the channel configuration (see **CHANCONFIGLIST** log on page 543) will be modified on the next reboot.

Eligible subscriptions include: TerraStar-X, TerraStar-C PRO, RTK ASSIST PRO, RTK ASSIST, or Oceanix-H.

**Message ID: 2284**

#### Abbreviated ASCII Syntax:

```
TERRASTARAUTOCHANCONFIG switch
```

#### Factory Default:

```
TERRASTARAUTOCHANCONFIG ENABLE
```

#### ASCII Example:

```
TERRASTARAUTOCHANCONFIG ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	TERRASTARAUTOCHANCONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	switch	DISABLE	0	The receiver does not use the channel configuration granted by the TerraStar/Oceanix subscription.	Enum	4	H
		ENABLE	1	The receiver uses the channel configuration granted by the TerraStar/Oceanix subscription.			

## 2.183 THISANTENNAPCO

**Sets the PCO model of this receiver**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use the **THISANTENNAPCO** command to set the Phase Center Offsets (PCO) for the given frequency of this receiver. The Offsets are defined as North, East and Up from the Antenna Reference Point to the Frequency Phase Center in mm.

**Message ID:** 1417

**Abbreviated ASCII Syntax:**

```
THISANTENNAPCO Frequency[NorthOffset][EastOffset][UpOffset]
```

**ASCII Example:**

```
THISANTENNAPCO GPSL1 0.61 1.99 65.64
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	THISANTENNAPCO header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Frequency	See Table 22: <i>Frequency Type</i> on page 91		The frequency for which the phase center offsets are valid.	Enum	4	H
3	NorthOffset			NGS standard Phase Center North Offset (millimetres). <sup>1</sup>	Double	8	H+4
4	EastOffset			NGS standard Phase Center East Offset (millimetres). <sup>1</sup>	Double	8	H+12
5	UpOffset			NGS standard Phase Center Up Offset (millimetres). <sup>1</sup>	Double	8	H+20

<sup>1</sup>Enter values as per the NGS standards and tables to define which direction is plus or minus.

## 2.184 THISANTENNAPCV

**Sets the PCV model of this receiver**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use the **THISANTENNAPCV** command to set the Phase Center Variation (PCV) for the given frequency of this receiver. The Phase Center Variation entries follow the NGS standard and correspond to the phase elevation at 5 degree increments starting at 90 degrees and decreasing to 0.

**Message ID:** 1418

**Abbreviated ASCII Syntax:**

```
THISANTENNAPCV Frequency[PCVArray]
```

**ASCII Example:**

```
THISANTENNAPCV GPSL1 0.00 -0.020 -0.07 -0.15 -0.24 -0.34 -0.43 -0.51 -0.56 -
0.61 -0.65 -0.69 -0.69 -0.62 -0.44 -0.13 0.28 0.70 1.02
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	THISANTENNAPCV header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Frequency	See Table 22: <i>Frequency Type</i> on page 91		The frequency for which the phase center variations is valid.	Enum	4	H
3	PCVArray			NGS standard 19 Element array of Phase Center Variations for phase variation for 5 degree elevation increments starting at 90 degrees and decreasing to 0. The variances are entered in millimetres.	Double Array [19]	152	H+4

## 2.185 THISANTENNATYPE

*Sets the antenna type of this receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use the **THISANTENNATYPE** command to set the type of antenna being used with the receiver. There are two sources of antenna information:

- An internal table  
The firmware contains a set of predefined antenna and radome types taken from the IGS ANTEX file. Refer to *Table 23: Antenna Type* on page 94 and *Table 24: Radome Type* on page 103 for the antennas currently supported.
- User-defined antennas  
User-defined antenna types can be entered using the **ANTENNATYPE** command (see page 72).

The **BASEANTENNATYPE** command (see page 93) is used to set the RTK base antenna type.

**Message ID: 1420**

**Abbreviated ASCII Syntax:**

```
THISANTENNATYPE AntennaType [RadomeType]
```

**ASCII Examples:**

```
THISANTENNATYPE NOV702
```

```
THISANTENNATYPE USER_ANTENNA_1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	THISANTENNATYPE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	AntennaType	See <i>Table 23: Antenna Type</i> on page 94 or <i>Table 17: User-Defined Antenna Type</i> on page 74		Antenna type	Enum	4	H
3	RadomeType	See <i>Table 24: Radome Type</i> on page 103		Radome type (default = NONE)	Enum	4	H+4

## 2.186 TILTCOMPENSATIONCONTROL

### *Control the application of position corrections for vehicle roll and pitch*

**Platform:** SMART7-I, SMART7-W, SMART2-TB

Use this command to enable or disable the application of position corrections computed by the Terrain Compensation filter to output positions. If disabled, the Terrain Compensation filter will continue to run in the background but the reported position will be the antenna position.



The **TILTFILTER** command on the next page and **TILTZERO** command on page 432, or the **\$PMDT** command on page 291, are used to configure the Terrain Compensation feature.

**Message ID:** 2309

#### Abbreviated ASCII Syntax:

```
TILTCOMPENSATIONCONTROL switch
```

#### Abbreviated ASCII Example:

```
TILTCOMPENSATIONCONTROL ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	TILTCOMPENSATION CONTROL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	switch	DISABLE	0	Disable Terrain Compensation (Tilt) corrections.	Enum	4	H
		ENABLE	1	Enable Terrain Compensation (Tilt) corrections.			

## 2.187 TILTFILTER

### Configure the Terrain Compensation filter

**Platform:** SMART7-I, SMART7-W, SMART2-TB

Use this command to enable or disable estimation of the system pitch and roll by the Terrain Compensation filter and to set the height of the GNSS antenna above the vehicle control point. The Terrain Compensation filter must be enabled for the Terrain Compensation feature to function.



The **\$PMDT** command (see page 291) can also be used to enable and configure the Terrain Compensation filter.

**Message ID:** 1221

#### Abbreviated ASCII Syntax:

```
TILTFILTER switch antennaheight
```

#### Abbreviated ASCII Example:

```
TILTFILTER ENABLE 3.5
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	TILTFILTER header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	switch	DISABLE	0	Disable the Tilt filter.	Enum	4	H
		ENABLE	1	Enable the Tilt filter.			
3	antennaheight	0.0 – 12.7		Height of the GNSS antenna above the vehicle control point (metres). (default = 0.0)	Float	4	H+4
4	Reserved				Ulong	4	H+8
5	Reserved				Ulong	4	H+12



Refer to Terrain Compensation in the [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for information about mounting the SMART antenna and measuring the antenna height.

## 2.188 TILTZERO

### *Set the zero level and save Terrain Compensation settings to NVM*

**Platform:** SMART7-I, SMART7-W, SMART2-TB

Use this command to set the internal zero level for Terrain Compensation and to save the settings (including antenna height which is configured with the **TILTFILTER** command on page 430) to NVM.

Sending this command with the **ZERO** option will set the current roll and pitch as the zero level. Setting the zero level allows the Terrain Compensation filter to account for any mounting offset between the receiver and the vehicle or any inherent tilt of the vehicle. It will also observe the internal sensor biases to allow for better performance of the filter. Zeroing must be conducted while the vehicle is stationary on a flat surface. This option is used for single-stage zeroing.

Sending this command with the **NEW** and **ADD** options will set the zero level when performing two-stage zeroing.

Sending this command with the **SAVE** option will save the zero level and antenna height settings to NVM. These settings will be automatically used on following startups.

Sending this command with the **RESTORE** option will clear the saved settings from NVM and restore the factory defaults.



The **\$PMDT** command (see page 291) can also be used to zero the filter (single-stage leveling only) and save the settings to NVM.

**Message ID: 1222**

**Abbreviated ASCII Syntax:**

```
TILTZERO action
```

**ASCII Example:**

```
TILTZERO zero
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	TILTZERO header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	action	ZERO	0	Set the current orientation as zero (level) for single-stage zeroing.	Enum	4	H
		SAVE	1	Saves the current level setting and antenna height to NVM.			
		RESTORE	2	Clear the current Tilt filter settings saved in NVM and restore the factory defaults.			
		NEW	3	Begin a new two stage calibration.			
		ADD	4	Complete a two stage calibration and set the zero level.			



Use the **ZERO** option to perform a single-stage zeroing. Use the **NEW** and **ADD** options to perform a two-stage zeroing.

Refer to Leveling the Terrain Compensation Sensor in the [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for information about single-stage and two-stage zeroing.

## 2.189 TRACKSIGNAL

### *Enable or disable tracking of a signal*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Use this command to enable or disable tracking of a signal within the channel configuration signal list.



Primary signals (LBAND and L1/E1/B1I signals) can not be disabled using the **TRACKSIGNAL** command.



Channel configurations containing the B2 signal will either track B2I or B2a signals corresponding to the availability of BeiDou Phase II and Phase III satellites respectively. Due to this, using the BEIDOU B2 option will toggle tracking for both B2I and B2a signals.



By default, the receiver attempts to track all signals in the channel configuration without needing a **TRACKSIGNAL** command. The channel configuration can be determined from the **CHANCONFIGLIST** log (see page 543).

**Message ID:** 2311

**Abbreviated ASCII Syntax:**

```
TRACKSIGNAL signal track
```

**Abbreviated ASCII Example:**

```
TRACKSIGNAL GPSL2C DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	TRACKSIGNAL header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0


Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	signal	See <i>Table 78: Signal</i> below		The signal to enable or disable  The signal is expected to match a signal in the current channel configuration.	Enum	4	H
3	track	DISABLE	0	Disable tracking of the signal	Enum	4	H+4
		ENABLE	1	Enable tracking of the signal			

Table 78: Signal

Binary	ASCII
1	GPSL2
2	GPSL2P
3	GPSL2C
4	GPSL5
5	GPSL1C
6	SBASL5
7	GLOL2
8	GLOL2P
9	GLOL2C
10	GLOL3
11	GALE5A
12	GALE5B
13	GALALTBOC
14	GALE6
15	QZSSL2C
16	QZSSL5
17	QZSSL1C

Binary	ASCII
18	QZSSL6
19	BEIDOUB1C
20	BEIDOUB2
21	BEIDOUB3
22	BEIDOUB2B

## 2.190 TRACKSV

### *Overrides automatic satellite assignment criteria*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to override the automatic satellite/channel assignment for all satellites with manual instructions.

**Message ID:** 1326

#### Abbreviated ASCII Syntax:

```
TRACKSV system SVID condition
```

#### Factory Default:

```
GPS, GLONASS, GALILEO, QZSS, BeiDou and NavIC default = GOODHEALTH
SBAS default = ANYHEALTH
```

```
TRACKSV QZSS 198 NEVER
```

```
TRACKSV QZSS 202 NEVER
```



QZSS 198 and QZSS 202 are excluded because they are defined as test PRNs in the QZSS ICD.

#### Input Example:

```
TRACKSV GALILEO 0 ANYHEALTH
```



For dual antenna receivers, this command applies to both the primary and secondary antennas.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	TRACKSV header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	System	See <i>Table 124: Satellite System</i> on page 654		System that the SVID belongs to	Enum	4	H
3	SVID	Refer to <i>PRN Numbers</i> on page 52		Satellite SVID number "0" is allowed and applies to all SVIDs for the specified system type	Ulong	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	Condition	See <i>Table 79: TRACKSV Command Condition</i> below		Tracking condition	Enum	4	H+8

Table 79: TRACKSV Command Condition

Binary	ASCII	Description
1	NEVER	Never track this satellite
2	GOODHEALTH	Track this satellite if the health is indicated as healthy in both the almanac and ephemeris
3	ANYHEALTH	Track this satellite regardless of health status
4	ALWAYS	Always track this satellite

## 2.191 TUNNELESCAPE

### *Breaks out of an established tunnel*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The tunnel escape sequence feature allows you to break out of a tunnel between two ports by sending a predefined sequence of bytes through the tunnel in-line with the data stream.

Use the **TUNNELESCAPE** command to specify the tunnel escape sequence. The escape sequence is applied independently to all active tunnels. Use the **SAVECONFIG** command (see page 366) to save the escape sequence in case of a power cycle.

This command is used to define an escape sequence that, when detected in a byte stream between any two COM (or AUX) ports, resets the interface mode to **NOVATEL NOVATEL** on those ports. The baud rate and other port parameters remain unaffected.

The **TUNNELESCAPE** command accepts three parameters. The first is the *switch* parameter with **ENABLE** or **DISABLE** options. The second is the *length* parameter. It is a number from 1 to 8 and must be present if the switch parameter is set to **ENABLE**. The third parameter, *esc seq*, consists of a series of pairs of digits representing hexadecimal numbers, where the number of pairs are equal to the value entered for the second parameter. The series of hexadecimal pairs of digits represent the escape sequence. The receiver detects a sequence in a tunnel exactly as it was entered.

For example, the command **TUNNELESCAPE ENABLE 4 61626364** searches for the bytes representing “abcd” in a tunnel stream. **TUNNELESCAPE ENABLE 3 AA4412** searches for the NovAtel binary log sync bytes.

You must first set up a tunnel. For example, create a tunnel between COM1 and COM2 by entering **INTERFACEMODE COM1 TCOM2 NONE OFF**. The commands can be entered in any order.



1. All bytes, leading up to and including the escape sequence, pass through the tunnel before it is reset. Therefore, the escape sequence is the last sequence of bytes that passes through the tunnel. Configure the receiver to detect and interpret the escape sequence. For example, use this information to reset equipment or perform a shutdown process.
2. The receiver detects the escape sequence in all active tunnels in any direction.
3. Create tunnels using the **INTERFACEMODE** command (see page 216).

**Message ID: 962**

#### **Abbreviated ASCII Syntax:**

```
TUNNELESCAPE switch length escseq
```

#### **Factory Default:**

```
TUNNELESCAPE disable 0
```

#### **ASCII Example:**

```
TUNNELESCAPE enable 1 aa
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	TUNNELESCAPE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	H	0	-
2	switch	DISABLE	0	Enable or disable the tunnel escape mode	Enum	4	H
		ENABLE	1				
3	length	1 to 8		Specifies the number of hex bytes to follow	Ulong	4	H+4
4	escseq			Escape sequence where Hex pairs are entered without spaces, for example, AA4412	Uchar[8]	8	H+8



If using the **SAVECONFIG** command (see page 366) in NovAtel Connect, ensure all windows other than the *Console* window are closed. If open, NovAtel Connect also saves log commands used for its various windows. This results in unnecessary data being logged.



## 2.192 UALCONTROL

### Setup User Accuracy levels

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **UALCONTROL** command is used to define User Accuracy Levels. User accuracy levels are user defined standard deviations thresholds, used to determine solution acceptability. Issuing the **UALCONTROL** command causes the BESTPOS and GPGGA solution types to be controlled via the specified thresholds, rather than by the solution source or mode. The new solution types are described in the table below.

**Table 80: User Accuracy Level Supplemental Position Types and NMEA Equivalents**

Value	BESTPOS Position Type <sup>1</sup>	NMEA Equivalent <sup>2</sup>
70	OPERATIONAL	4
71	WARNING	5
72	OUT_OF_BOUNDS	1

1. As reported in the **BESTPOS** log (see page 520).  
2. Refers to the GPGGA quality indicator (see the **GPGGA** log on page 610 for details).

The **SETBESTPOSCRITERIA** command (see page 393) determines which standard deviations are compared against the provided thresholds.



UAL is useful for applications that rely upon specific solutions types being present in the BESTPOS or GPGGA logs. For example, if an agricultural steering system commonly requires an RTK fixed GPGGA solution type (4) to operate, and interruptions in RTK conventionally cause the GPGGA to switch to another solution type. This causes the steering system to disengage.

**Message ID: 1627**

**Abbreviated ASCII Syntax:**


```
UALCONTROL Action [Operational_limit] [Warning_limit]
```

**Factory Default:**

```
UALCONTROL disable
```

**ASCII Example:**

```
UALCONTROL enable 0.10 0.20
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UALCONTROL header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Action	DISABLE	0	Disables this feature	Enum	4	H
		ENABLE	1	Replace BESTPOS and GPGBA position types with OPERATIONAL, WARNING or OUT_OF_BOUNDS based on the entered standard deviations (refer to <i>Table 80: User Accuracy Level Supplemental Position Types and NMEA Equivalents</i> on the previous page)			
		CLEAR	2	Disable this feature and reset the entered standard deviations.			
3	Operational_Limit			Standard deviation in metres to report OPERATIONAL	Double	8	H+4
4	Warning_Limit			Standard deviation in metres to report WARNING  OUT_OF_BOUND reports when the standard deviation exceeds this value	Double	8	H+12

## 2.193 UNASSIGN

### *Unassigns a previously assigned channel*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command cancels a previously issued **ASSIGN** command (see page 76) and the SV channel reverts to automatic control (the same as **ASSIGN AUTO**).

**Message ID:** 29

#### Abbreviated ASCII Syntax:

```
UNASSIGN channel [state]
```

#### Input Example:

```
UNASSIGN 11
```



Issuing the **UNASSIGN** command to a channel that was not previously assigned by the **ASSIGN** command (see page 76) has no effect.



For dual antenna receivers:

- On the primary antenna, the SV channel is from 0 to N-1, where N is the number of channels in the primary antenna channel configuration.
- On the secondary antenna, the SV channel count is from at N to N+(M-1), where M is the number of channels in the secondary antenna SV channel configuration.
- Even though L-Band channels cannot be configured with **ASSIGN** or **UNASSIGN**, they are included when determining N.



The **ASSIGN** and **UNASSIGN** commands are not accepted for L-Band channels. The **ASSIGNLBANDBEAM** command (see page 82) should be used for L-Band channels.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNASSIGN header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	channel	0 to N-1, where N is the number of the last GNSS channel in the current channel configuration		Channel number reset to automatic search and acquisition mode	Ulong	4	H
3	state	These return SV channel control to the automatic search engine immediately (see <i>Table 18: Channel State</i> on page 78)		Set the SV channel state (currently ignored)	Enum	4	H+4

## 2.194 UNASSIGNALL

### *Unassigns all previously assigned channels*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command cancels all previously issued **ASSIGN** or **ASSIGNALL** commands for all SV channels for a satellite system (same as **ASSIGNALL AUTO**). Tracking and control for each SV channel reverts to automatic mode.

**Message ID:** 30

#### Abbreviated ASCII Syntax:

```
UNASSIGNALL [system]
```

#### Input Example:

```
UNASSIGNALL GPS
```



Issuing the **UNASSIGNALL** command has no effect on channels that were not previously assigned using the **ASSIGN** command (see page 76) or **ASSIGNALL** command (see page 79).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNASSIGNALL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	system	See <i>Table 19: Satellite System</i> on page 80		System that will be affected by the <b>UNASSIGNALL</b> command (default = ALL)	Enum	4	H

## 2.195 UNDULATION

### *Chooses undulation*

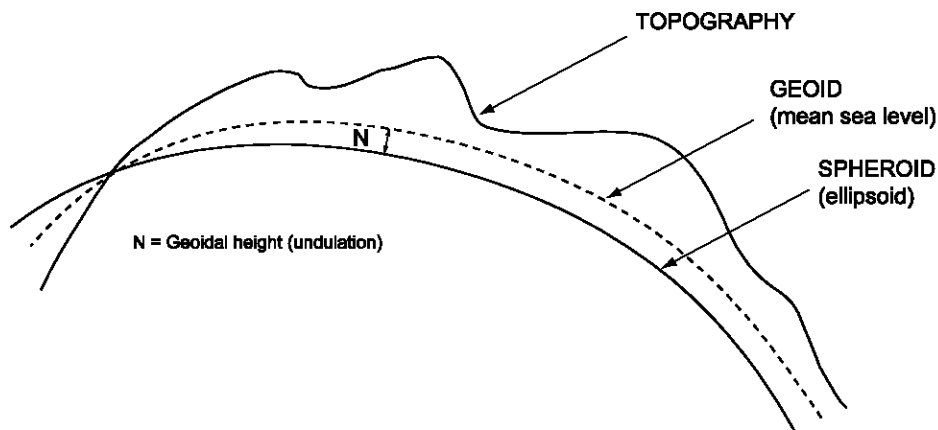
**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command permits you to enter a specific geoidal undulation value. In the option field, the EGM96 table provides ellipsoid heights at a 0.5° by 0.5° spacing while the OSU89B is implemented at a 2° by 3° spacing. In areas of rapidly changing elevation, you could be operating somewhere within the 2° by 3° grid with an erroneous height. EGM96 provides a more accurate model of the ellipsoid which results in a denser grid of heights. It is also more accurate because the accuracy of the grid points themselves has also improved from OSU89B to EGM96. For example, the default grid (EGM96) is useful where there are underwater canyons, steep drop-offs or mountains.

The undulation values reported in the position logs are in reference to the ellipsoid of the chosen datum.

Refer to the application note [APN-006: Geoid Issue](http://novatel.com/support/support-materials/application-notes), available at [novatel.com/support/support-materials/application-notes](http://novatel.com/support/support-materials/application-notes), for a description of the relationships in *Figure 11: Illustration of Undulation* below.

**Figure 11: Illustration of Undulation**



**Message ID: 214**

#### **Abbreviated ASCII Syntax:**

```
UNDULATION option [separation]
```

#### **Factory Default:**

```
UNDULATION egm96 0.0000
```

#### **ASCII Example 1:**

```
UNDULATION osu89b
```

#### **ASCII Example 2:**

```
UNDULATION USER -5.599999905
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNDULATION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	option	USER	1	Use the user specified undulation value	Enum	4	H
		OSU89B	2	Use the OSU89B undulation table			
		EGM96	3	Use global geoidal height model EGM96 table			
3	separation	±1000.0 m		The undulation value (required for the USER option) (default = 0.000)	Float	4	H+4

## 2.196 UNLOCKOUT

*Reinstates a satellite in the solution*



The **UNLOCKOUT** command has been deprecated. To allow a satellite that was previously locked out to be reinstated in the solution computation, use the **UNLOCKOUTSATELLITE** command on page 450.

**Message ID: 138**



## 2.197 UNLOCKOUTALL

*Reinstates all previously locked out satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command allows all satellites or systems which have been previously locked out (**LOCKOUT** command on page 244 or **LOCKOUTSYSTEM** command on page 246) to be reinstated in the solution computation.

**Message ID:** 139

**Abbreviated ASCII Syntax:**

UNLOCKOUTALL

**Input Example:**

UNLOCKOUTALL

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNLOCKOUTALL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

## 2.198 UNLOCKOUTSATELLITE

### *Reinstates a satellite into the solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command allows a satellite previously locked out using the **LOCKOUTSATELLITE** command (see page 245) to be added back into solution computations. To reinstate multiple satellites that have been locked out, this command must be issued multiple times.

#### **Abbreviated ASCII Syntax:**

```
UNLOCKOUTSATELLITE system prn
```

**Message ID: 2416**

#### **ASCII Example:**

```
UNLOCKOUTSATELLITE gps 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	UNLOCKOUTSATELLITE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	system	See Table 124: <i>Satellite System</i> on page 654		The satellite system for the satellite being unlocked.	Enum	4	H
3	prn	See 1.6.1 <i>PRN Numbers</i> on page 52		Unique identifier for the satellite being unlocked (added back into the solution).	Ulong	4	H+4

## 2.199 UNLOCKOUTSYSTEM

*Reinstates previously locked out system*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command allows a system which has been previously locked out (refer to the **LOCKOUTSYSTEM** command on page 246) to be reinstated in the solution computation.



If more than one system is to be reinstated, this command must be reissued for each system reinstatement.

**Message ID:** 908

**Abbreviated ASCII Syntax:**

```
UNLOCKOUTSYSTEM system
```

**Input Example:**

```
UNLOCKOUTSYSTEM glonass
```



The **UNLOCKOUTSYSTEM** command is used to reinstate a system while leaving other locked out systems unchanged.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNLOCKOUTSYSTEM header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	system	See Table 124: <i>Satellite System</i> on page 654		A single satellite system to be reinstated	Enum	4	H

## 2.200 UNLOG

### *Removes a log from logging control*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is used to remove a specific log request from the system.

**Message ID: 36**

#### **Abbreviated ASCII Syntax:**

```
UNLOG [port] message
```

#### **Input Example:**

```
UNLOG com1 bestposa
```

```
UNLOG bestposa
```



The **UNLOG** command is used to remove one or more logs while leaving other logs unchanged.

### 2.200.1 Binary

Field	Field Name	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	UNLOG (binary) header	(See Table 4: Binary Message Header Structure on page 38)	This field contains the message header	-	H	0
2	port	See Table 5: Detailed Port Identifier on page 40 (decimal port values greater than 16 may be used)	Port to which log is being sent	Enum	4	H
3	message	Any valid message ID	Message ID of log to output	Ushort	2	H+4

Field	Field Name	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit ( <i>Message Responses</i> on page 49) 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7

### 2.200.2 ASCII

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	UNLOG (ASCII) header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII or ASCII, respectively	-	H	0
2	port	See <i>Table 5: Detailed Port Identifier</i> on page 40 (decimal port values greater than 16 may be used)		Port to which log is being sent (default = THISPORT)	Enum	4	H
3	message	Message Name	N/A	Message Name of log to be disabled	Ulong	4	H+4

## 2.201 UNLOGALL

### *Removes all logs from logging control*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

If *[port]* is specified, this command disables all logs on the specified port only. All other ports are unaffected. If *[port]* is not specified this command defaults to the ALL\_PORTS setting.

**Message ID: 38**

#### Abbreviated ASCII Syntax:

```
UNLOGALL [port] [held]
```

#### Input Example:

```
UNLOGALL com2_15
UNLOGALL true
```



The **UNLOGALL** command is used to remove all log requests currently in use.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UNLOGALL header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	port	See Table 5: Detailed Port Identifier on page 40 (decimal values greater than 16 may be used)		Port to clear (default = ALL_PORTS)	Enum	4	H
3	held	FALSE	0	Does not remove logs with the HOLD parameter (default)	Bool	4	H+4
		TRUE	1	Removes previously held logs, even those with the HOLD parameter			

## 2.202 USBSTICKEJECT

### *Prepare a USB stick for removal*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7



PwrPak7M variants do not support this command.

Use this command to unmount the USB stick and prepare it for safe physical removal.

This command may fail with a *Busy* error if there is an ongoing USB stick mounting or unmounting operation.

The **FILETRANSFERSTATUS** log (see page 566) indicates the *USBSTICK UNMOUNTED* status when it is safe to physically remove the stick. This may take up to 10 seconds.

**Message ID: 2115**

#### **Abbreviated ASCII Syntax:**

```
USBSTICKEJECT
```

#### **Example:**

```
USBSTICKEJECT
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	USBSTICKEJECT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

## 2.203 USERDATUM

### *Sets user customized ellipsoid and datum transformation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command is like the **USEREXPDATUM** command but without parameter rates. See the **USEREXPDATUM** command on page 458 for details on command usage and parameter meaning.

When the **USERDATUM** command is entered, the **USEREXPDATUM** command on page 458 is issued internally with the **USERDATUM** command values. It is the **USEREXPDATUM** command on page 458 that appears in the **RXCONFIG** log (see page 876). Only a single set of **USEREXPDATUM** or **USERDATUM** entered parameters is valid at one time. Newer parameters entered by either command will overwrite previously entered parameters.



See the **DATUM** command on page 133 for important guidance and details on datum operation.

**Message ID:** 78

**Abbreviated ASCII Syntax:**

```
USERDATUM semi_major_axis flattening dx dy dz rx ry rz scale
```

**Factory Default:**

```
USERDATUM 6378137.0 298.2572235628 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

**ASCII Example:**

```
USERDATUM 6378206.400 294.97869820000 -12.0000 147.0000 192.0000 0.0000
0.0000 0.0000 0.000000000
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	USERDATUM header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	semi_major_axis	6300000.0 - 6400000.0		Datum semi-major axis (a) (metres)	Double	8	H
3	flattening	290.0 - 305.0		Reciprocal Flattening, $1/f = a/(a-b)$	Double	8	H+8



Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	dx	± 2000.0		Translation values (metres)	Double	8	H+16
5	dy	± 2000.0			Double	8	H+24
6	dz	± 2000.0			Double	8	H+32
7	rx	± 10.0 radians		Datum rotation angles A positive sign is for counter clockwise rotation and a negative sign is for clockwise rotation	Double	8	H+40
8	ry	± 10.0 radians			Double	8	H+48
9	rz	± 10.0 radians			Double	8	H+56
10	scale	± 10.0 ppm		Scale value is the difference in ppm	Double	8	H+64

## 2.204 USEREXPDATUM

### *Set custom expanded datum*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command permits entry of customized ellipsoid and datum transformation parameters. The command **DATUM USER** must be sent for the parameters set by **USERDATUM** to take effect.

Only a single set of **USEREXPDATUM** or **USERDATUM** entered parameters is valid at one time. Newer parameters entered by either command will overwrite previously entered parameters.

The transformation used by this command is the 7 parameter Helmert transformation, with the parameters provided being for the transformation from the user datum to WGS84:

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + (1 + s) \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Where:

**X, Y, Z** are the coordinates in the user datum

**X<sub>s</sub>, Y<sub>s</sub>, Z<sub>s</sub>** are the coordinates in WGS84

**T<sub>x</sub>, T<sub>y</sub>, T<sub>z</sub>** are the translation parameters corrected for translation rate (metres)

**R<sub>x</sub>, R<sub>y</sub>, R<sub>z</sub>** are the rotation parameters corrected for rotation rate (radians)

**s** is the scale difference corrected for scale rate (unitless)

A given parameter P is corrected for rate:

$$P(t) = P(t_r) + \dot{P}(t_r) (t - t_r)$$

Where:

**t<sub>r</sub>** is the reference epoch of the parameters

**t** is the current epoch

**P(t<sub>r</sub>)** is the parameter at the reference epoch

**Ḑ(t<sub>r</sub>)** is the rate of the parameter at the reference epoch

The convention used by this command differs from that used by the **DATUMTRANSFORMATION** command on page 135, and from parameter conventions seen typically in transformation sources. Care should be taken that the parameter convention matches that in the equation above.



See the **DATUM** command on page 133 for important guidance and details on datum operation.

**Message ID: 783**

#### **Abbreviated ASCII Syntax:**

```
USEREXPDATUM semi_major_axis flattening dx dy dz rx ry rz scale xvel yvel
zvel xrvel yrvel zrvel scalelevel epoch
```

**Factory Default:**

```
USEREXPDATUM 6378137.0 298.25722356280 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0
```

**ASCII Example:**

```
USEREXPDATUM 6378137.000 298.25722356280 0.000000000 0.000000000 0.000000000
0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
0.000000000 0.0000 0.000000000 0.000000000 0.000000000 0.000000000
```



Use the **USEREXPDATUM** command in a survey to fix the position with values from another known datum so that the GPS calculated positions are reported in the known datum rather than WGS84. For example, it is useful for places like Australia, where the continent is moving several centimetres a year relative to WGS84. With **USEREXPDATUM**, you can also input the velocity of the movement to account for drift over the years.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	USEREXPDATUM header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	semi_major_axis	6300000.0 - 6400000.0 m		Datum semi-major axis (a) (metres)	Double	8	H
3	flattening	290.0 - 305.0		Reciprocal Flattening, $1/f = a/(a-b)$	Double	8	H+8
4	dx	± 2000.0 m		Translation values (metres)	Double	8	H+16
5	dy	± 2000.0 m			Double	8	H+24
6	dz	± 2000.0 m			Double	8	H+32
7	rx	± 10.0 radians		Datum rotation angles A positive sign is for counter clockwise rotation and a negative sign is for clockwise rotation	Double	8	H+40
8	ry	± 10.0 radians			Double	8	H+48
9	rz	± 10.0 radians			Double	8	H+56
10	scale	± 10.0 ppm		Scale value is the difference in ppm	Double	8	H+64
11	xvel	± 2000.0 m/yr		Velocity vector along X-axis	Double	8	H+72

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
12	yvel	± 2000.0 m/yr		Velocity vector along Y-axis	Double	8	H+80
13	zvel	± 2000.0 m/yr		Velocity vector along Z-axis	Double	8	H+88
14	xrvel	± 10.0 radians/yr		Change in the rotation about X over time	Double	8	H+96
15	yrvel	± 10.0 radians/yr		Change in the rotation about Y over time	Double	8	H+104
16	zrvel	± 10.0 radians/yr		Change in the rotation about Z over time	Double	8	H+112
17	scalevel	± 10.0 ppm/yr		Change in scale from WGS84 over time	Double	8	H+120
18	epoch	0.0 year		Reference epoch of parameters (decimal year) Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	H+128

## 2.205 USERI2CBITRATE

### *Set the communication rate for the User I2C bus*

**Platform:** OEM7600, OEM7700, OEM7720

Use this command to set the communication bit rate for the User I2C bus.

**Message ID:** 2383

**Abbreviated ASCII Syntax:**

```
USERI2CBITRATE BitRate
```

**Factory Default:**

```
USERI2CBITRATE 400K
```

**ASCII Examples:**

```
USERI2CBITRATE 100K
```

```
USERI2CBITRATE 400K
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	USERI2CBITRATE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	BitRate	100K	1	Set the User I2C bus communication rate to 100 kbits/s.	Enum	4	H
		400K	2	Set the User I2C bus communication rate to 400 kbits/s.			

## 2.206 USERI2CREAD

### *Read data from devices on the I2C bus*

**Platform:** OEM7600, OEM7700, OEM7720

Use this command to read data from devices on the I2C bus.



This command only applies to OEM7 receivers that have I2C signals available on the interface connector. The compatible receivers are listed in the **Platform** section above.

The **USERI2CRESPONSE** log (see page 976) can be used to check the completion or status of the read operation. An optional user defined Transaction ID can be provided to help synchronize requests with responses in the **USERI2CRESPONSE** log (see page 976). This command is primarily intended to be used by Lua applications that need to interact with external devices.

Reading from an I2C device requires a device address, to distinguish which physical device is to be accessed, a register within the device, and the expected number of bytes to be read. Depending on the type of I2C device, register addresses can be 1 to 4 bytes in length, so the actual number of bytes for the register address must be specified.

For some I2C devices there are no registers within the device. In this case, the Register Address Length is 0 and no bytes are supplied for the Register Address.

The **USERI2CREAD** command is flexible to handle all of these situations.

**Message ID: 2232**

#### Abbreviated ASCII Syntax:

```
USERI2CREAD DeviceAddress RegisterAddressLen RegisterAddress RequestReadLen
[TransactionID]
```

#### Examples:

```
USERI2CREAD 70 1 AB 12 1234
USERI2CREAD 74 3 ABCDEF 234 5678
USERI2CREAD 74 0 234 5678
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Command header	USERI2CREAD header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	DeviceAddress	The 7 bit address of the I2C device. Valid values are 0 through 127.  For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits. There is no 0x prefix and spaces are not allowed in the string.	Uchar	1 <sup>1</sup>	H
3	RegisterAddressLen	The length of the register address that follows. Valid values are 0 through 4.	Ulong	4	H+4
4	RegisterAddress	The actual address of the register to be read. The number of bytes here must match the RegisterAddressLen. In particular, when RegisterAddressLen is 0, this field is empty (even for a binary command)  For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits for each byte in the register address. There is no 0x prefix and spaces are not allowed in the string.	Uchar Array	X <sup>1</sup>	H+8
5	RequestReadLen	The length of data expected to be retrieved from the device. Valid values are 1 through 256.	Ulong	4	H+12 <sup>2</sup>
6	TransactionID	An optional user provided ID for this transaction. Default = 0.  This transaction ID will be copied to the <b>USERI2CRESPONSE</b> log (see page 976) created for this read operation.	Ulong	4	H+16 <sup>3</sup>

<sup>1</sup>In the binary case, additional bytes of padding are added after this field to maintain 4-byte alignment for the fields that follow.

<sup>2</sup>H+8 if X=0

<sup>3</sup>H+12 if X=0

## 2.207 USERI2CWRITE

### *Write data to device on I2C bus*

**Platform:** OEM7600, OEM7700, OEM7720

Use this command to write data to devices on the I2C bus.



This command only applies to OEM7 receivers that have I2C signals available on the interface connector. The compatible receivers are listed in the **Platform** section above.

The **USERI2CRESPONSE** log (see page 976) can be used to check the completion or status of the write operation. An optional user defined Transaction ID can be provided to help synchronize requests with responses in the **USERI2CRESPONSE** log (see page 976). This command is primarily intended to be used by Lua applications that need to interact with external devices.

Writing to an I2C device requires a device address, to distinguish which physical device is to be accessed, a register within the device and the data. Depending on the type of I2C device, register addresses can be 1 to 4 bytes in length, and so the actual number of bytes for the register address must be specified.

For some I2C devices there are no registers within the device. In this case, the Register Address Length is 0, and no bytes are supplied for the Register Address.

For some other I2C devices, write operations are done in two stages:

1. The first stage sends a write command with a register address, but no data. This is a dummy write to set the register within the device for write operations that follow.
2. The second stage sends a write command with no register address, but does send a stream of data.

The **USERI2CWRITE** command is flexible to handle all of these situations.

**Message ID: 2233**

#### **Abbreviated ASCII Syntax:**

```
USERI2CWRITE DeviceAddress RegisterAddressLen RegisterAddress
WriteDataLength WriteData [TransactionID]
```

#### **Examples:**

```
USERI2CWRITE 70 1 AB 12 3132333435363738393A3B3C 1234
USERI2CWRITE 74 3 ABCDED 5 1234567890 1234
USERI2CWRITE 40 0 5 1234567890 1234
USERI2CWRITE 40 2 AABB 0 1234 (a dummy write)
```



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Command header	USERI2CWRITE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	DeviceAddress	The 7 bit address of the I2C device. Valid values 0 through 127.  For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits. There is no 0x prefix and spaces are not allowed in the string.	Uchar	1 <sup>1</sup>	H
3	RegisterAddressLen	The length of the register address that follows. Valid values are 0 through 4.	Ulong	4	H+4
4	RegisterAddress	The actual address of the register to be written. The number of bytes here must match the RegisterAddressLen. In particular, when RegisterAddressLen is 0, this field is empty (even for a binary command)  For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits for each byte in the register address. There is no 0x prefix and spaces are not allowed in the string.	Uchar Array	X <sup>1</sup>	H+8
5	WriteDataLength	The length of data to be written in bytes. Valid values are 0 through 256.	Ulong	4	H+12 <sup>2</sup>

<sup>1</sup>In the binary case, additional bytes of padding are added after this field to maintain 4-byte alignment for the fields that follow.

<sup>2</sup>H+8 if X=0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	WriteData	<p>The data to be written. The number of bytes in this data block must match the WriteDataLength. In particular, when WriteDataLength is 0, this field is empty.</p> <p>For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits for each byte in the data block. There is no 0x prefix and spaces are not allowed in the string.</p> <p>Data is streamed to the device as a series of bytes in the order provided.</p>	Uchar Array	Y <sup>1</sup>	H+16 <sup>2</sup>
7	TransactionID	<p>An optional user provided ID for this transaction. Default = 0.</p> <p>This transaction ID will be copied to the <b>USERI2CRESPONSE</b> log (see page 976) created for this write operation.</p>	Ulong	4	H+16+4*INT((Y+3)/4) <sup>3</sup>

<sup>1</sup>In the binary case, additional bytes of padding are added after this field to maintain 4-byte alignment for the fields that follow.

<sup>2</sup>H+12 if X=0

<sup>3</sup>H+12+4\*INT((Y+3)/4) if X=0

## 2.208 UTMZONE

### Sets UTM parameters

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This command sets the UTM persistence, zone number or meridian. Refer to [https://en.wikipedia.org/wiki/Universal\\_Transverse\\_Mercator\\_coordinate\\_system](https://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system).



1. The latitude limits of the UTM System are 80°S to 84°N, so if your position is outside this range, the **BESTUTM** log (see page 532) outputs a northing, easting and height of 0.0, along with a zone letter of "\*" and a zone number of 0, so that it is obvious that the data in the log is dummy data.
2. If the latitude band is X, then the Zone number should not be set to 32, 34 or 36. These zones were incorporated into other zone numbers and do not exist.

**Message ID:** 749

#### Abbreviated ASCII Syntax:

```
UTMZONE command [parameter]
```

#### Factory Default:

```
UTMZONE auto 0
```

#### ASCII Example 1:

```
UTMZONE SET 10
```

#### ASCII Example 2:

```
UTMZONE CURRENT
```



The UTM grid system is displayed on all National Topographic Series (NTS) of Canada maps and United States Geological Survey (USGS) maps. On USGS 7.5-minute quadrangle maps (1:24,000 scale), 15-minute quadrangle maps (1:50,000, 1:62,500, and standard-edition 1:63,360 scales) and Canadian 1:50,000 maps the UTM grid lines are drawn at intervals of 1,000 metres and are shown either with blue ticks at the edge of the map or by full blue grid lines. On USGS maps at 1:100,000 and 1:250,000 scale and Canadian 1:250,000 scale maps a full UTM grid is shown at intervals of 10,000 metres.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	UTMZONE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	command	See <i>Table 81: UTM Zone Commands</i> below			Enum	4	H
3	parameter	See <i>Table 81: UTM Zone Commands</i> below			Long	4	H+4

Table 81: UTM Zone Commands

Binary	ASCII	Description
0	AUTO	UTM zone default that automatically sets the central meridian and does not switch zones until it overlaps by the set persistence. This a spherical approximation to the earth unless you are at the equator (default = 0) (m)
1	CURRENT	Same as UTMZONE AUTO with infinite persistence of the current zone. The parameter field is not used
2	SET	Sets the central meridian based on the specified UTM zone. A zone includes its western boundary, but not its eastern boundary, Meridian. For example, zone 12 includes (108°W, 114°W) where 108° < longitude < 114°
3	MERIDIAN	Sets the central meridian as specified in the parameter field. In the <b>BESTUTM</b> log (see page 532), the zone number is output as 61 to indicate the manual setting (zones are set by pre-defined central meridians not user-set ones)

## 2.209 WIFIALIGNAUTOMATION

### Configure the ALIGN plug-and-play feature over Wi-Fi

Platform: PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to simplify the configuration of a pair of receivers; one acting as an ALIGN Master and the other acting as an ALIGN Rover.

Similar to the **ALIGNAUTOMATION** command (see page 68) the ALIGN Rover, acting as a Wi-Fi Client, connects to the specified Wi-Fi Access Point and configures it as an ALIGN Master.



The IP address of the ALIGN Rover and ALIGN Master must be different. Since all NovAtel receivers are delivered with the same default IP address, it is highly recommended to use the **WIFIPIPCONFIG** command (see page 473) to change the IP address of the ALIGN Rover to a non-default value such as 192.168.20.1 prior to entering the **WIFIALIGNAUTOMATION** command. Failure to do so may result in loss of wireless communication to the ALIGN Rover.

Message ID: 2214

#### Abbreviated ASCII Syntax:

```
WIFIALIGNAUTOMATION option [master_networkid] [corrections_port] [datarate]
[headingextboption] [interfacemode]
```


#### Factory Default:

```
WIFIALIGNAUTOMATION DISABLE
```

#### Example:

```
WIFIALIGNAUTOMATION enable 1 icom1 10 on novatelx
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFIALIGN AUTOMATION header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	option	ENABLE	1	Enable or disable the plug-and-play feature.	Enum	4	H
		DISABLE	0				
3	master_network	1-4		Network id of Align Master Access Point as defined in the <b>WIFINETCONFIG</b> command (see page 478).	Ulong	4	H+4
4	corrections_port	ICOM1	23	 The Master ICOM port must be configured with factory default settings. (default=ICOM1)	Enum	4	H+8
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46				
		ICOM6	47				
		ICOM7	48				
5	datarate	1, 2, 4, 5, 10 or 20		Rate (in Hz) at which heading output is required (default = ICOM1)	Ulong	4	H+12
6	headingextboption	OFF	0	Enable or disable sending HEADINGEXTB / HEADINGEXT2B back to Master (default=ON)	Enum	4	H+16
		ON	1				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
7	interfacemode	See 2.68 <i>INTERFACEMODE</i> on page 216		Interfacemode for corrections (default = NONE)  If this parameter is not specified, the ALIGN Master is not configured to generate any corrections. The assumption is that the master has been separately configured. Refer to ALIGN Over Wi-Fi Overview in the <a href="#">PwrPak7 Installation and Operation User Manual</a> or <a href="#">SMART7 Installation and Operation User Manual</a> for details on ALIGN.	Enum	4	H+20

## 2.210 WIFIAPCHANNEL

### *Set the channel for the Wi-Fi access point*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to set the operating channel for the Wi-Fi module when operating as an access point. The new channel will be used the next time the **WIFIMODE AP** or **WIFIMODE CONCURRENT** command is received.

**Message ID: 2091**

#### Abbreviated ASCII Syntax:

```
WIFIAPCHANNEL channel
```

#### Factory Default:

```
WIFIAPCHANNEL 11
```

#### Example:

```
WIFIAPCHANNEL 6
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Value	Binary Offset
1	Command header	-	-	WIFIAPCHANNEL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	channel	1-14		802.11 channel	Long	4	H



For best performance, choose one of the non-overlapping channels: 1, 6, or 11.



## 2.211 WIFIAPIPCONFIG

### Set the IP address and netmask for the Wi-Fi access point

Platform: PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to set the Wi-Fi IP address and netmask for Wi-Fi module when operating as an access point. The new network configuration takes effect the next time the **WIFIMODE AP** or **WIFIMODE CONCURRENT** command is received.

**Message ID: 2096**

#### Abbreviated ASCII Syntax:

```
WIFIAPIPCONFIG ip_address ip_netmask
```

#### Factory Default:

```
WIFIAPIPCONFIG 192.168.19.1 255.255.255.0
```

#### Example:

```
WIFIAPIPCONFIG 192.162.55.20 255.255.0.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFIAPIPCONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	ip_address	Null-terminated ASCII string		IP address, dot decimal format	String[16]	Variable	H
3	ip_netmask	Null-terminated ASCII string		IP netmask, dot decimal format (optional)  Default =255.255.255.0	String[16]	Variable	Variable

## 2.212 WIFIAPPASSKEY

### *Set Wi-Fi access point passkey*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to set the WPA2 PSK ASCII passkey for the Wi-Fi module when the receiver is operating as an Access Point.

The default passkey is printed on the receiver label.

The new passkey takes effect the next time the **WIFIMODE AP** or **WIFIMODE CONCURRENT** command is received.



The term passkey and password are the same.

**Message ID:** 2090

#### **Abbreviated ASCII Syntax:**

```
WIFIAPPASSKEY passkey
```

#### **Factory Default:**

The default passkey/password is printed on the receiver label.

#### **Example:**

```
WIFIAPPASSKEY "bysP3zE6SZmFQeyd"
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFIAPPASSKEY header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	passkey	Null-terminated ASCII string, 8 to 64 characters		WPA2 PSK ASCII passkey	String[65]	Variable	H

## 2.213 WIFIAPSSID

### Set the SSID for the Wi-Fi access point

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to explicitly set the SSID of the Access Point when the receiver is configured to operate in either AP or CONCURRENT mode (refer to the **WIFIMODE** command on the next page for details regarding the Wi-Fi configuration modes).

**Message ID: 2206**

**Abbreviated ASCII Syntax:**

```
WIFIAPSSID ssid
```

**Factory Default:**

```
<platform-dependent prefix>-<Enclosure PSN>
```

**Example:**

```
PwrPak7-NMND17190003B
```

```
SM7i-NMPX17500010L
```

**Example:**

```
WIFIAPSSID SomeSSIDName
```

```
WIFIAPSSID "SSID with spaces"
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFIAPSSID header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	SSID	Null-terminated ASCII string		SSID to be broadcast by access point	String[33]	Variable	H

## 2.214 WIFIMODE

### *Configure the receiver Wi-Fi mode*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to enable, disable or set the operating mode of Wi-Fi on the receiver. This command is also applies any Wi-Fi configuration changes specified by other Wi-Fi commands such as **WIFIAPCHANNEL**.

**Message ID: 2144**

#### **Abbreviated ASCII Syntax:**

```
WIFIMODE mode
```

#### **Factory Default:**

```
WIFIMODE AP
```

#### **Example:**

```
WIFIMODE CLIENT
```

```
WIFIMODE OFF
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFIMODE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	mode	OFF	0	Power off the Wi-Fi module	Enum	4	H
		AP	1	Configure the Wi-Fi module as an Access Point (AP)			
		CLIENT	2	Configure the Wi-Fi module as a Client/Station			
		ON	3	Supply power to the Wi-Fi module, but do not configure it.			
		CONCURRENT	4	Configure the Wi-Fi module as both an access point and Client/Station simultaneously. When configured in this mode it is possible to connect the receiver to an access point and for Clients/Stations to connect to the receiver simultaneously.			

## 2.215 WIFINETCONFIG

*Set the connection parameters of Wi-Fi to which the Wi-Fi client can connect*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this command to specify Access Points (AP) to which the Client/Station will automatically connect when the receiver's Wi-Fi Client/Station functionality is enabled (refer to the **WIFIMODE** command (see page 476) for details regarding the Wi-Fi configuration modes).



If the Wi-Fi Client/Station is already connected to an access point (defined using this command), the connection remains in effect even when a more preferred access point comes into range (preferred as indicated by `network_id`).



If Wi-Fi ALIGN automation is enabled, the Client only connects to the Access Point specified in the **WIFIALIGNAUTOMATION** command (see page 469).



The term passkey and password are the same.

**Message ID: 2213**

### Abbreviated ASCII Syntax:

```
WIFINETCONFIG network_id switch [ssid [passkey [address_mode [IP_address
]netmask [gateway [dns]]]]]]]]
```

### Factory Default:

```
WIFINETCONFIG 1 DISABLE
WIFINETCONFIG 2 DISABLE
WIFINETCONFIG 3 DISABLE
WIFINETCONFIG 4 DISABLE
```

### Example:

```
WIFINETCONFIG 1 ENABLE SomeSSID 12345678
WIFINETCONFIG 2 ENABLE AnotherSSID 1A2b3C4d STATIC 192.168.19.1
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	WIFINETCONFIG header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	network_id	1, 2, 3, 4		The Network identifier value is used to prioritize network connections when more than one network is in range of the client. The access point with the lowest network_id is chosen.	Ulong	4	H
3	switch	DISABLE	0	When set to Disable, the client will not connect to an access point with this SSID.	Enum	4	H+4
		ENABLE	1	When set to Enable, the client connects to an access point with this SSID if there are no other APs with a lower network_id in range.			
4	ssid	Null-terminated ASCII string		Access Points with non-ASCII SSIDs are not supported.  Required if switch is set to ENABLE.	String [33]	Variable	
5	passkey	Null-terminated ASCII string, 8 to 64 characters		Passkey required to connect to access point identified by "ssid" parameter. (default = "")	String [65]	Variable	

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	address_mode	DHCP	1	Use dynamic IP address (default = DHCP)	Enum	4	
		STATIC	2	Use static IP address			
7	ip_address	ddd.ddd.ddd.ddd		IP Address - decimal dot notation The IP address is mandatory when the address mode is STATIC.	String [16]	Variable	
8	netmask	ddd.ddd.ddd.ddd		Netmask – decimal dot notation (default = 255.255.255.0)	String [16]	Variable	
9	gateway	ddd.ddd.ddd.ddd		Gateway – decimal dot notation (default = 0.0.0.0)	String [16]	Variable	
10	dns	ddd.ddd.ddd.ddd		DNS server – decimal dot notation (default = 0.0.0.0)	String [16]	Variable	



## Chapter 3 Logs

Logs are the mechanism used to extract information from the receiver.

### 3.1 Log Types

See the **LOG** command on page 247, for details about requesting logs.

The receiver is capable of generating three type of logs: synchronous, asynchronous and polled. The data for synchronous logs is generated on a regular schedule. In order to output the most current data as soon as it is available, asynchronous data is generated at irregular intervals. The data in polled logs is generated on demand. The following table outlines the log types and the valid triggers to use:

**Table 82: Log Type Triggers**

Type	Recommended Trigger	Illegal Trigger
Synch	ONTIME	ONNEW, ONCHANGED
Asynch	ONCHANGED or ONCE	-
Polled	ONCE or ONTIME <sup>1</sup>	ONNEW, ONCHANGED

See *Message Time Stamps* on page 54 for information about how the message time stamp is set for each type of log.



1. The OEM7 family of receivers can handle 80 logs at a time. If an attempt is made to log more than 80 logs at a time, the receiver responds with an Insufficient Resources error.
2. Asynchronous logs, such as MATCHEDPOS, should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may result in inaccurate time tags.
3. Use the ONNEW trigger with the MARKxTIME or MARKxPOS logs.
4. Before the output of fields for logs, there is a header. See *ASCII* on page 33, *Abbreviated ASCII* on page 35 and *Binary* on page 37.

#### 3.1.1 Log Type Examples

For polled logs, the receiver only supports an offset that is:

- smaller than the logging period
- decimal values that are a multiple of the maximum logging rate defined by the receiver model. For more information see the **LOG** command on page 247.

The following are valid examples for a polled log:

```
log portstats ontime 4 2
log version once
```

For polled logs, the following examples are invalid:

---

<sup>1</sup>Polled log types do not allow fractional offsets and cannot do ontime rates faster than 1 Hz.

```
log serialconfig ontime 1 2      [offset is larger than the logging period]
```

```
log serialconfig ontime 4 1.5    [offset is not an integer]
```

For synchronous and asynchronous logs, the receiver supports any offset that is both:

- smaller than the logging period
- a multiple of the minimum logging period

For example, if the receiver supports 20 Hz logging, the minimum logging period is 1/20 Hz or 0.05 s. The following are valid examples for a synchronous or asynchronous log, on a receiver that can log at rates up to 20 Hz:

```
log bestpos ontime 1             [1 Hz]
```

```
log bestpos ontime 1 0.1
```

```
log bestpos ontime 1 0.90
```

```
log avepos ontime 1 0.95
```

```
log avepos ontime 2             [0.5 Hz]
```

```
log avepos ontime 2 1.35
```

```
log avepos ontime 2 1.75
```

For synchronous and asynchronous logs, the following examples are invalid:

```
log bestpos ontime 1 0.08      [offset is not a multiple of the minimum logging period]
```

```
log bestpos ontime 1 1.05      [offset is larger than the logging period]
```

## 3.2 ALIGNBSLNENU

### *ENU baselines using ALIGN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log outputs the RTK quality ENU baselines from ALIGN. The XYZ baselines (output in the **ALIGNBSLNXYZ** log on page 485) are rotated relative to master position (output in MASTERPOS) to compute ENU baselines.



On dual antenna receivers, the baseline vector from the primary to the secondary antenna is provided. Additionally, with an ALIGN Relative Positioning model, the baseline vector from an external base station to the primary antenna is also available. The baseline vector from an external base station to the secondary antenna is not available.

**Message ID:** 1315

**Log Type:** Asynch

**Recommended Input:**

```
log alignbslnenua onnew
```

**ASCII Example:**

```
#ALIGNBSLNENUA, COM1, 0, 29.0, FINESTEERING, 1629, 259250.000, 02040000, 100b, 39448;
SOL_COMPUTED, NARROW_INT, 4.1586, -1.9197, -0.0037, 0.0047, 0.0050, 0.0062, "0092",
"AAAA", 22, 16, 16, 16, 0, 01, 0, 33*11e1d4c0
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ALIGNBSLNENU header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	East	East Baseline (relative to master position) in metres	Double	8	H+8
5	North	North Baseline (relative to master position) in metres	Double	8	H+16
6	Up	Up Baseline (relative to master position) in metres	Double	8	H+24

Field	Field type	Description	Format	Binary Bytes	Binary Offset
7	East $\sigma$	East Baseline standard deviation in metres	Float	4	H+32
8	North $\sigma$	North Baseline standard deviation in metres	Float	4	H+36
9	Up $\sigma$	Up Baseline standard deviation in metres	Float	4	H+40
10	Rover id	Rover Receiver ID Set using the <b>SETROVERID</b> command (see page 402) on the Rover e.g., setroverid RRRR	Char[4]	4	H+44
11	Master id	Master Receiver ID Set using the <b>DGPSTXID</b> command (see page 138) on the Master Default: AAAA	Char[4]	4	H+48
12	#SVs	Number of satellites tracked	Uchar	1	H+52
13	#solnSVs	Number of satellites used in solution	Uchar	1	H+53
14	#obs	Number of satellites above elevation mask angle	Uchar	1	H+54
15	#multi	Number of satellites with multi-frequency signals above the mask angle	Uchar	1	H+55
16	Reserved		Hex	1	H+56
17	ext sol stat	Extended solution status, see <i>Table 90: Extended Solution Status</i> on page 527	Hex	1	H+57
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+58
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+59
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+60
21	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

### 3.3 ALIGNBSLNXYZ

#### XYZ baselines using ALIGN

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log outputs the RTK quality XYZ baselines from ALIGN.



On dual antenna receivers, the baseline vector from the primary to the secondary antenna is provided. Additionally, with an ALIGN Relative Positioning model, the baseline vector from an external base station to the primary antenna is also available. The baseline vector from an external base station to the secondary antenna is not available.

**Message ID:** 1314

**Log Type:** Asynch

**Recommended Input:**

```
log alignbslnxyza onnew
```

**ASCII Example:**

```
#ALIGNBSLNXYZA, COM1, 0, 29.0, FINESTEERING, 1629, 259250.000, 02040000, 9d28, 39448;
SOL_COMPUTED, NARROW_INT, 3.1901, -3.0566, 1.2079, 0.0050, 0.0054, 0.0056, "0092",
"AAAA", 22, 16, 16, 16, 0, 01, 0, 33*ac372198
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ALIGNBSLNXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	dX	X Baseline in metres	Double	8	H+8
5	dY	Y Baseline in metres	Double	8	H+16
6	dZ	Z Baseline in metres	Double	8	H+24
7	dX $\sigma$	X Baseline standard deviation in metres	Float	4	H+32
8	dY $\sigma$	Y Baseline standard deviation in metres	Float	4	H+36
9	dZ $\sigma$	Z Baseline standard deviation in metres	Float	4	H+40

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	Rover id	Rover Receiver ID Set using <b>SETROVERID</b> command (see page 402) on the Rover e.g. SETROVERID RRRR	Uchar[4]	4	H+44
11	Master id	Master Receiver Id Set using the <b>DGPSTXID</b> command (see page 138) on the Master Default: AAAA	Uchar[4]	4	H+48
12	#SVs	Number of satellites tracked	Uchar	1	H+52
13	#solnSVs	Number of satellites used in solution	Uchar	1	H+53
14	#obs	Number of satellites above elevation mask angle	Uchar	1	H+54
15	#multi	Number of satellites with multi-frequency signals above the mask angle	Uchar	1	H+55
16	Reserved		Hex	1	H+56
17	ext sol stat	Extended solution status, see <i>Table 90: Extended Solution Status</i> on page 527	Hex	1	H+57
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+58
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+59
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+60
21	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 3.4 ALIGNDOP

### Calculated DOP values

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log outputs the DOP computed using the satellites used in the heading solution. This log comes out at a default 1 Hz rate. Additional logs may be output not on the even second if the DOP changes and ALIGN is operating at greater than 1 Hz.

**Message ID:** 1332

**Log Type:** Asynch

**Recommended Input:**

```
log aligndopa onnew
```

**ASCII Example:**

```
#ALIGNDOPA, COM1, 0, 22.5, FINESTEERING, 1629, 259250.000, 02040000, de2d, 39448; 1.6160,
1.2400, 0.6900, 0.9920, 0.7130, 10.0, 16, 4, 32, 23, 10, 7, 20, 13, 30, 16, 47, 43, 46, 53, 54, 44,
45*90a72971
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ALIGNDOP header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	HTDOP	Horizontal and time DOP	Float	4	H+12
6	TDOP	GPS time DOP	Float	4	H+16
7	Elev mask	Elevation mask angle	Float	4	H+20
8	#sats	Number of satellites to follow	Ulong	4	H+24
9	sats	Satellites in use at time of calculation	Ulong	4	H+28
10	Next sat offset = H+28+(#sats * 4)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+28+ (#sats * 4)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.5 ALMANAC

### *Decoded GPS L1 C/A Almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the decoded GPS almanac parameters from L1 C/A subframes four and five, as received from the satellite, with the parity information removed and appropriate scaling applied. For more information about almanac data, refer to the GPS SPS Signal Specification.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

**Message ID:** 73

**Log Type:** Asynch

**Recommended Input:**

log almanaca onchanged

**ASCII Example:**

```
#ALMANACA,USB1,0,65.5,SATTIME,2209,494970.000,02000020,06de,16809;
31,
1,2210,61440.0,1.189947e-02,-7.49745516e-09,2.7464570e+00,9.04911221e-01,
5.43628036e-01,3.62396240e-04,-7.27595761e-12,1.45857660e-04,2.6559901e+07,
4.54861256e-02,3,0,0,TRUE,
2,2210,61440.0,2.042484e-02,-7.79461039e-09,2.6553604e+00,-1.4191382e+00,
6.98451114e-01,-6.53266907e-04,0.00000000,1.45858448e-04,2.6559805e+07,
2.35789625e-02,1,0,0,TRUE,
3,2210,61440.0,4.142284e-03,-7.64603277e-09,-2.5031848e+00,9.52756366e-01,
-5.65670396e-01,-2.45094299e-04,-1.45519152e-11,1.45862179e-04,2.6559352e+07,
3.12788270e-02,3,0,0,TRUE,
...
30,2210,61440.0,5.815029e-03,-7.89747182e-09,-3.91536612e-01,-2.6938085e+00,
-1.35149100e-01,-5.28335571e-04,0.00000000,1.45850074e-04,2.6560822e+07,
-6.21981273e-03,3,0,0,TRUE,
31,2210,61440.0,1.043320e-02,-7.71460706e-09,-3.84896228e-01,3.90039333e-01,
-5.91707997e-01,-1.77383423e-04,0.00000000,1.45852520e-04,2.6560525e+07,
1.28291128e-02,2,0,0,TRUE,
32,2210,61440.0,5.961895e-03,-8.08033658e-09,-1.4636431e+00,-2.2868654e+00,
-2.8675969e+00,-1.27792358e-04,-7.27595761e-12,1.45866076e-04,2.6558879e+07,
1.52259577e-02,3,0,0,TRUE*0eadcab6
```



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).



Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#messages	The number of satellite PRN almanac messages to follow. Set to zero until almanac data is available	Long	4	H
3	PRN	Satellite PRN number for current message (dimensionless)	Ulong	4	H+4
4	week	Almanac reference week (GPS reference week number)	Ulong	4	H+8
5	seconds	Almanac reference time (seconds into the week)	Double	8	H+12
6	ecc	Eccentricity (dimensionless)	Double	8	H+20
7	$\dot{\omega}$	Rate of right ascension (radians/second)	Double	8	H+28
8	$\omega_0$	Right ascension (radians)	Double	8	H+36
9	$\omega$	Argument of perigee (radians)	Double	8	H+44
10	$M_0$	Mean anomaly of reference time (radians)	Double	8	H+52
11	$a_{f0}$	Clock aging parameter (seconds)	Double	8	H+60
12	$a_{f1}$	Clock aging parameter (seconds/second)	Double	8	H+68
13	$N_0$	Computed mean motion (radians/second)	Double	8	H+76
14	A	Semi-major axis (metres)	Double	8	H+84
15	incl-angle	Angle of inclination relative to $0.3 \pi$ (radians)	Double	8	H+92
16	SV config	Satellite configuration	Ulong	4	H+100
17	health-prn	SV health from Page 25 of subframe 4 or 5 (6 bits)	Ulong	4	H+104
18	health-alm	SV health from almanac (8 bits)	Ulong	4	H+108
19	antispoof	Anti-spoofing on? 0 = FALSE 1 = TRUE	Bool	4	H+112
20...	Next PRN offset = H + 4 + (#messages x 112)				

---

Field	Field type	Description	Format	Binary Bytes	Binary Offset
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H + 4 + (112 x #messages)
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.6 AUTHCODES

### *List of authorization codes*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains all authorization codes (auth codes) entered into the system since the last complete firmware reload. Signature authorization codes will be maintained through a SoftLoad. The log also indicates the status of the firmware signature. For more information about firmware signatures see Upgrading Using the AUTH Command in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).



The following situations will cause an authorization code to be marked invalid:

- Authorization Code is for a different receiver
- Authorization Code has expired
- Authorization Code was entered incorrectly

If you require new authorization codes, contact NovAtel Customer Service.

**Message ID:** 1348

**Log Type:** Polled

**Recommended Input:**

```
log authcodesa once
```

**ASCII Example:**

```
#AUTHCODESA,USB1,0,67.0,FINESTEERING,2209,495534.000,02000020,2ad2,16809;VALID,1,STANDARD,TRUE,"KGHNCD,6ZD36P,GXBTC5,34M5DR,CW573K,FFNRNNCBES1"*02d3c863
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	AUTHCODES header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	AUTHCODES Signature Status	Status of the Firmware Signature 1 = NONE 2 = INVALID 3 = VALID 4 = RESERVED 5 = HIGH_SPEED	Enum	4	H

Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	Number of Auth Codes	# of Auth Codes to follow (max is 24)	Ulong	4	H+4
4	Auth code type	1=STANDARD 2=SIGNATURE 3=EMBEDDED	Enum	4	H+8
5	Valid	TRUE if the Auth Code has been verified	Bool	4	H+12
6	Auth Code String	ASCII String of the Auth Code	String [max 80]	variable <sup>1</sup>	H+16
7...	Next AuthCode = H+8+ (#AuthCodes*variable)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+ (#AuthCodes* variable)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 3.7 AVEPOS

### *Position averaging*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

When position averaging is underway, the various fields in the AVEPOS log contain the parameters being used in the position averaging process. *Table 83: Position Averaging Status* on the next page shows the possible position averaging status values seen in field #8 of the AVEPOS log table.

See the description of the **POSAVE** command on page 293. For general positioning information, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).



Asynchronous logs should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

**Message ID:** 172

**Log Type:** Asynch

**Recommended Input:**

```
log aveposa onchanged
```

**ASCII Example:**

```
#AVEPOSA,USB1,0,72.5,FINESTEERING,2210,147873.000,02100020,e3b4,16809;
51.15043181793,-114.03067226763,1097.6045,1.9300,1.9294,1.9392,COMPLETE,5,6
*83bdc6a3
```



When a GNSS position is computed, there are four unknowns being solved: latitude, longitude, height and receiver clock offset (often just called time). The solutions for each of the four unknowns are correlated to satellite positions in a complex way. Since satellites are above the antenna (none are below) there is a geometric bias. Therefore, geometric biases are present in the solutions and affect the computation of height. These biases are called DOPs (Dilution Of Precision). Smaller biases are indicated by low DOP values. VDOP (Vertical DOP) pertains to height. Most of the time, VDOP is higher than HDOP (Horizontal DOP) and TDOP (Time DOP). Therefore, of the four unknowns, height is the most difficult to solve. Many GNSS receivers output the Standard Deviations (SD) of the latitude, longitude and height. Height often has a larger value than the other two.

Accuracy is based on statistics and reliability is measured in percent. When a receiver states it can measure height to one metre, this is an accuracy. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%. In other words, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (one metre) and the result is 95% reliability (error is less than two metres 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions. See also **GPGST** log on page 624 for CEP and RMS definitions.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	AVEPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	lat	Average WGS84 latitude (degrees)	Double	8	H
3	lon	Average WGS84 longitude (degrees)	Double	8	H+8
4	hgt	Average height above sea level (m)	Double	8	H+16
5	lat $\sigma$	Estimated average standard deviation of latitude solution element (m)	Float	4	H+24
6	lon $\sigma$	Estimated average standard deviation of longitude solution element (m)	Float	4	H+28
7	hgt $\sigma$	Estimated average standard deviation of height solution element (m)	Float	4	H+32
8	posave	Position averaging status (see <i>Table 83: Position Averaging Status</i> below)	Enum	4	H+36
9	ave time	Elapsed time of averaging (s)	Ulong	4	H+40
10	#samples	Number of samples in the average	Ulong	4	H+44
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 83: Position Averaging Status

Binary	ASCII	Description
0	OFF	Receiver is not averaging
1	INPROGRESS	Averaging is in progress
2	COMPLETE	Averaging is complete

## 3.8 BDSALMANAC

### *Decoded BDS Almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the decoded BeiDou B1I almanac parameters, with the parity information removed and appropriate scaling applied. Multiple messages are transmitted, one for each SV almanac collected. For more information about almanac data, refer to the BDS Signal Specification.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

**Message ID:** 1584

**Log Type:** Asynch

**Recommended Input:**

```
log bdsalmanaca onchanged
```

### **ASCII Example:**

```
#BDSALMANACA, USB1, 58, 66.0, SATTIME, 2209, 495392.000, 02000020, 24ad, 16809;1, 853,
233472, 6493.366211, 3.9148330688e-04, 2.140348862, 1.199588705, -2.642467663,
6.62884755e-10, 1.3200623733e-02, 1.783371e-04, 4.36557e-11, 0*47ce1f59

#BDSALMANACA, USB1, 57, 66.0, SATTIME, 2209, 495422.000, 02000020, 24ad, 16809;2, 853,
233472, 6493.407715, 1.2860298157e-03, -1.063963232, -3.106212600, -2.469847262,
2.85726187e-10, 3.2794831454e-02, 4.005432e-05, -4.00178e-11, 0*8a6f2da6

#BDSALMANACA, USB1, 56, 66.0, SATTIME, 2209, 495452.000, 02000020, 24ad, 16809;3, 853,
233472, 6493.317871, 8.8787078857e-04, 0.016723986, 2.822887922, -2.736275307,
4.11445710e-10, 3.2818799903e-02, -5.550385e-04, 7.27596e-12, 0*d002a7ff

...

#BDSALMANACA, USB1, 2, 66.0, SATTIME, 2209, 494348.000, 02000020, 24ad, 16809;60, 853,
233472, 6493.458496, 4.3869018555e-04, 1.681354547, 2.010414773, 2.163612865,
2.25152236e-09, 3.6605814973e-02, -9.536743e-07, 0.00000, 58*0b0eeddc

#BDSALMANACA, USB1, 1, 66.0, SATTIME, 2209, 494378.000, 02000020, 24ad, 16809;61, 853,
233472, 6493.341309, 5.9413909912e-04, -0.385926122, -0.842142719, 1.330163202,
2.12580283e-09, 2.6886608575e-02, 0.000000, 0.00000, 58*0db11f92

#BDSALMANACA, USB1, 0, 66.0, SATTIME, 2209, 474248.000, 02000020, 24ad, 16809;62, 853,
417792, 6553.599609, 1.2499809265e-02, -1.256637361, -1.256637361, 1.256636987,
5.99202102e-07, 1.5707723583e-01, 7.810593e-04, -1.49157e-09, 100*5633ace5
```



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	satellite ID	Satellite ID/ranging code	Ulong	4	H
3	week	Week number in BeiDou time (BDT) since the BeiDou system time start epoch (January 1, 2006)	Ulong	4	H+4
4	toa	Time of almanac (s)	Ulong	4	H+8
5	RootA	Square root of semi-major axis (sqrt(m))	Double	8	H+12
6	ecc	Eccentricity (dimensionless)	Double	8	H+20
7	$\omega$	Argument of perigee (radians)	Double	8	H+28
8	$M_0$	Mean anomaly at reference time (radians)	Double	8	H+36
9	$\Omega$	Longitude of ascending node of orbital of plane computed according to reference time (radians)	Double	8	H+44
10	$\dot{\Omega}$	Rate of right ascension (radians/s)	Double	8	H+52
11	$\delta_i$	Correction of orbit reference inclination at reference time (radians)	Double	8	H+60
12	$a_0$	Constant term of clock correction polynomial (s)	Double	8	H+68
13	$a_1$	Linear term of clock correction polynomial (s/s)	Double	8	H+76
14	health	Satellite health information	Ulong	4	H+84
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+88
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.9 BDSBCNAV1EPHEMERIS

### *Decoded BeiDou B-CNAV1 ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the decoded BeiDou B-CNAV1 ephemeris for the BeiDou B1C signal.

**Message ID:** 2371

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnavlephemerisa onchanged
```

**Example log:**

```
#BDSBCNAV1EPHEMERISA, USB1, 6, 59.0, SATTIME, 2209, 496436.000, 02000020, 27d6, 16809;
25, 853, 00000000, 25, 493200, 3, 83.013672, -0.003732, 3.9006981940e-09,
1.9335870986e-14, 0.477429585, 4.3404346798e-04, 0.447233249, 1.0544811238e-01,
9.4935254055e-01, -6.84742808e-09, 3.476930542e-10, 5.401670933e-08,
-3.539025784e-08, 128.6171875, 146.8867188, 1.032091677e-05, 6.321817636e-06,
25, 493200, -7.37199793e-04, 5.41433565e-12, 0.00000000, 1.920852810e-09,
-1.711305231e-08, -2.328306437e-10, 0*eb1ad752

#BDSBCNAV1EPHEMERISA, USB1, 5, 59.0, SATTIME, 2209, 496436.000, 02000020, 27d6, 16809;
24, 853, 00000000, 25, 493200, 3, 19.027344, 0.006292, 3.9494502247e-09,
-6.8209628314e-14, 1.247259053, 4.9585988745e-04, 0.517626996, 1.3024422419e-01,
9.4798221860e-01, -6.90885921e-09, 3.485859486e-10, 2.421438694e-08,
-5.587935448e-09, 115.0976563, 154.5429688, 9.900890291e-06, 5.632638931e-06,
25, 493200, -9.63939528e-04, -4.78550533e-12, 0.00000000, 1.175794750e-08,
-1.484295353e-08, 2.910383046e-10, 0*273a26ab

...

#BDSBCNAV1EPHEMERISA, USB1, 1, 59.0, SATTIME, 2209, 496436.000, 02000020, 27d6, 16809;
39, 853, 00000000, 25, 493200, 2, 5263.894531, -0.062899, 7.7092496926e-10,
1.9584381616e-13, -2.140521199, 2.2453696001e-03, -2.871042649, -9.3801678773e-01,
9.6155928554e-01, -1.86489911e-09, -1.102724504e-09, -4.284083843e-08,
4.004687071e-08, -83.8671875, -507.8242188, 2.449937165e-05, -3.319233656e-06,
25, 493200, 3.49998591e-05, 3.10862447e-14, 0.00000000, 7.799826562e-09,
-1.426087692e-08, -3.492459655e-10, 0*890d2cfd

#BDSBCNAV1EPHEMERISA, USB1, 0, 59.0, SATTIME, 2209, 496436.000, 02000020, 27d6, 16809;
44, 853, 00000000, 25, 493200, 3, 65.369141, -0.011349, 3.8406956947e-09,
1.0343710014e-13, -0.198007475, 8.3381979493e-04, 0.582406796, 2.199813208e+00,
9.5949562234e-01, -6.96636161e-09, -4.343038048e-10, -5.215406418e-08,
-4.190951586e-08, -118.6484375, 212.3242188, 7.395632565e-06, -5.853362381e-06,
25, 493200, 6.58427074e-04, 1.22231114e-11, 0.00000000, -2.153683454e-09,
-1.199077815e-08, -2.910383046e-10, 0*aaff1aac
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV1EPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Satellite ID	ID/ranging code	Ulong	4	H
3	WN	Week number in BeiDou time (BDT). This will be offset from GPS time by 1356 weeks.	Ulong	4	H+4
4	Satellite Status	Satellite Status Information: Bits 0 to 1: Health Status (HS) Bit 2: Data Integrity Flag (DIF) Bit 3: Signal Integrity Flag (SIF) Bit 4: Accuracy Integrity Flag (AIF) Bits 5 to 8: Signal In Space Monitoring Accuracy Index (SISMAI)	Hex Ulong	4	H+8
5	IODE	Issue of data, ephemeris	Ulong	4	H+12
6	toe	Reference time of ephemeris parameters in BeiDou time (BDT). (s) This will be offset from GPS time.	Ulong	4	H+16
7	Satellite Type	2 bit orbit type: 01 indicates GEO, 10 indicates IGSO, 11 indicates MEO	Ulong	4	H+20
8	DeltaA	Semi-major axis difference at reference time (m)	Double	8	H+24
9	ADot	Change rate in Semi-major axis (m/s)	Double	8	H+32
10	DeltaN	Mean motion difference from computed value (radians/s)	Double	8	H+40
11	NDot	Rate of Mean motion difference from computed value (radians/s <sup>2</sup> )	Double	8	H+48
12	M <sub>0</sub>	Mean anomaly at reference time (radians)	Double	8	H+56
13	e	Eccentricity (unitless)	Double	8	H+64
14	Omega	Argument of perigee (radians)	Double	8	H+72
15	Omega <sub>0</sub>	Longitude of ascending node of orbital of plane at weekly epoch (radians)	Double	8	H+80

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
16	$i_0$	Inclination angle at reference time (radians)	Double	8	H+88
17	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+96
18	$i_0$ Dot	Rate of inclination angle (radians/s)	Double	8	H+104
19	$C_{is}$	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+112
20	$C_{ic}$	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+120
21	$C_{rs}$	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+128
22	$C_{rc}$	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+136
23	$C_{us}$	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+144
24	$C_{uc}$	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+152
25	IODC	Issue of data, clock	Ulong	4	H+160
26	toc	Reference time of clock parameters in BeiDou time (BDT). This will be offset from GPS time by 14 seconds. (s)	Ulong	4	H+164
27	$a_0$	Constant term of clock correction polynomial (s)	Double	8	H+168
28	$a_1$	Linear term of clock correction polynomial (s/s)	Double	8	H+176
29	$a_2$	Quadratic term of clock correction polynomial (s/s <sup>2</sup> )	Double	8	H+184
30	TGDB1Cp	Group delay differential between the B1C pilot component and the B3I signal (s)	Double	8	H+192
31	TGDB2Ap	Group delay differential between the B2a pilot component and the B3I signal (s)	Double	8	H+200
32	ISCB1Cd	Group delay differential between the B1C data and pilot components (s)	Double	8	H+208
33	Reserved		Ulong	4	H+216
34	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+220
35	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 3.10 BDSBCNAV1RAWMESSAGE

### *Framed BeiDou B-CNAV1 raw navigation message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the framed BeiDou B-CNAV1 navigation message in raw bit format.

**Message ID:** 2373

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnav1rawmessagea onnew
```

**Example log:**

```
#BDSBCNAV1RAWMESSAGEA,USB1,0,44.0,SATTIME,2209,496508.000,02000020,661c,16809;65,35,25,8edc,1aac4832339b3ffd8f4800492456a4fff4b8388a7301006a93d00a12254256670b18c4e66ba28ecf15da87ffcc009ffddd84032e24041b5fe422cd847641001322800781ffeffd00,67130e81da82fbb1ce82698141147b890a5ad77d74230493bfcc363b79e2*556d1498
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV1RAWMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Signal Channel	Signal Channel Number	Ulong	4	H
3	Satellite ID	Satellite ID / PRN	Ulong	4	H+4
4	Page ID	Page ID for Subframe 3	Ulong	4	H+8
5	Raw Subframe 1 Data	Framed raw navigation bits for Subframe 1	Hex[2]	2	H+12
6	Raw Subframe 2 Data	Framed raw navigation bits for Subframe 2 (CRC check bits removed)	Hex[72]	72	H+14
7	Raw Subframe 3 Data	Framed raw navigation bits for Subframe 3 (CRC check bits removed)	Hex[30]	30	H+86
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+116
9	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 3.11 BDSBCNAV2EPHEMERIS

### *Decoded BDS B-CNAV2 ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the Decoded BDS B-CNAV2 ephemeris for the BeiDou B2a signals.

**Message ID:** 2372

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnav2ephemerisa onchanged
```

**Example log:**

```
#BDSBCNAV2EPHEMERISA, USB1, 6, 63.5, SATTIME, 2209, 496514.000, 02000020, 21df, 16809;
34, 853, 00000000, 25, 493200, 3, 25.591797, 0.006489, 3.8598036335e-09,
-5.7266440902e-14, -1.169634521, 6.7982793553e-04, 0.749691829, 2.199470739e+00,
9.6205966277e-01, -6.88135806e-09, -3.976951370e-10, 3.259629011e-08,
-3.166496754e-08, -104.0195313, 198.4492188, 8.173286915e-06, -5.176290870e-06,
25, 493200, -4.04388702e-04, 8.52029558e-12, 0.00000000, -3.085006028e-09,
-1.635635272e-08, -2.735760063e-09, 0*ecfc685f

#BDSBCNAV2EPHEMERISA, USB1, 5, 63.5, SATTIME, 2209, 496514.000, 02000020, 21df, 16809;
39, 853, 00000000, 25, 493200, 2, 5263.894531, -0.062899, 7.7092496926e-10,
1.9584381616e-13, -2.140521199, 2.2453696001e-03, -2.871042649, -9.3801678773e-01,
9.6155928554e-01, -1.86489911e-09, -1.102724504e-09, -4.284083843e-08,
4.004687071e-08, -83.8671875, -507.8242188, 2.449937165e-05, -3.319233656e-06,
25, 493200, 3.49998591e-05, 3.10862447e-14, 0.00000000, 7.799826562e-09,
-1.426087692e-08, -2.502929419e-09, 0*4bceb44d

...

#BDSBCNAV2EPHEMERISA, USB1, 1, 63.5, SATTIME, 2209, 496514.000, 02000020, 21df, 16809;
24, 853, 00000000, 25, 493200, 3, 19.027344, 0.006292, 3.9494502247e-09,
-6.8209628314e-14, 1.247259053, 4.9585988745e-04, 0.517626996, 1.3024422419e-01,
9.4798221860e-01, -6.90885921e-09, 3.485859486e-10, 2.421438694e-08,
-5.587935448e-09, 115.0976563, 154.5429688, 9.900890291e-06, 5.632638931e-06,
25, 493200, -9.63939528e-04, -4.78550533e-12, 0.00000000, 1.175794750e-08,
-1.484295353e-08, -2.502929419e-09, 0*f6bbff08

#BDSBCNAV2EPHEMERISA, USB1, 0, 63.5, SATTIME, 2209, 496514.000, 02000020, 21df, 16809;
26, 853, 00000000, 25, 493200, 3, 107.203125, -0.005850, 3.7798003010e-09,
4.1571032661e-14, 2.051998724, 7.3472585063e-04, 0.436791966, 1.0602762178e-01,
9.4940386191e-01, -6.82564146e-09, 3.775157250e-10, -3.166496754e-08,
-3.073364496e-08, 103.6250000, 136.3750000, 1.077353954e-05, 5.120411515e-06, 25,
493200, 7.38367089e-04, -5.27311528e-12, 0.00000000, 0.00000000, -1.699663699e-08,
-2.561137080e-09, 0*bf8175af
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV2EPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Satellite ID	ID/ranging code	Ulong	4	H
3	WN	Week number in BeiDou time (BDT). This will be offset from GPS time by 1356 weeks.	Ulong	4	H+4
4	Satellite Status	Satellite Status Information: Bits 0 to 1: Health Status (HS) Bit 2: Data Integrity Flag (DIF) Bit 3: Signal Integrity Flag (SIF) Bit 4: Accuracy Integrity Flag (AIF) Bits 5 to 8: Signal In Space Monitoring Accuracy Index (SISMAI)	Hex Ulong	4	H+8
5	IODE	Issue of data, ephemeris	Ulong	4	H+12
6	toe	Reference time of ephemeris parameters in BeiDou time (BDT). This will be offset from GPS time. (s)	Ulong	4	H+16
7	Satellite Type	2 bit orbit type: 01 indicates GEO, 10 indicates IGSO, 11 indicates MEO	Ulong	4	H+20
8	DeltaA	Semi-major axis difference at reference time (m)	Double	8	H+24
9	ADot	Change rate in Semi-major axis (m/s)	Double	8	H+32
10	DeltaN	Mean motion difference from computed value (radians/s)	Double	8	H+40
11	NDot	Rate of mean motion difference from computed value (radians/s <sup>2</sup> )	Double	8	H+48
12	M <sub>0</sub>	Mean anomaly at reference time (radians)	Double	8	H+56
13	e	Eccentricity (unitless)	Double	8	H+64
14	Omega	Argument of perigee (radians)	Double	8	H+72
15	Omega <sub>0</sub>	Longitude of ascending node of orbital of plane at weekly epoch (radians)	Double	8	H+80

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
16	$i_0$	Inclination angle at reference time (radians)	Double	8	H+88
17	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+96
18	$i_0$ Dot	Rate of inclination angle (radians/s)	Double	8	H+104
19	$C_{is}$	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+112
20	$C_{ic}$	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+120
21	$C_{rs}$	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+128
22	$C_{rc}$	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+136
23	$C_{us}$	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+144
24	$C_{uc}$	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+152
25	IODC	Issue of data, clock	Ulong	4	H+160
26	toc	Reference time of clock parameters in BeiDou time (BDT). (s) This will be offset from GPS time by 14 seconds.	Ulong	4	H+164
27	$a_0$	Constant term of clock correction polynomial (s)	Double	8	H+168
28	$a_1$	Linear term of clock correction polynomial (s/s)	Double	8	H+176
29	$a_2$	Quadratic term of clock correction polynomial (s/s <sup>2</sup> )	Double	8	H+184
30	TGDB1Cp	Group delay differential between the B1C pilot component and the B3I signal (s)	Double	8	H+192
31	TGDB2Ap	Group delay differential between the B2a pilot component and the B3I signal (s)	Double	8	H+200
32	ISCB2ad	Group delay differential between the B2a data and pilot components (s)	Double	8	H+208
33	Reserved		Ulong	4	H+216
34	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+220
35	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 3.12 BDSBCNAV2RAWMESSAGE

### *Framed BeiDou B-CNAV2 raw navigation message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the framed BeiDou B-CNAV2 raw navigation message in raw bit format.

**Message ID:** 2374

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnav2rawmessagea onnew
```

**Example log:**

```
#BDSBCNAV2RAWMESSAGEA,USB1,0,44.0,SATTIME,2209,496625.000,02000020,fb5d,16809;
144,26,11,e24de868ba1a841e01147a2024d5d4871ed564210bff7bff7c019e8002218005a6001
57a*1985a1bd
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV2RAWMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Signal Channel	Signal channel number	Ulong	4	H
3	Satellite ID	Satellite ID / PRN	Ulong	4	H+4
4	Message Type	Message type contained in the data	Ulong	4	H+8
5	Raw Message Data	Framed raw navigation data, 288 bits (CRC removed)	Hex[36]	36	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
7	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–



### 3.13 BDSBCNAV3EPHEMERIS

#### *Decoded BDS B-CNAV3 ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the decoded parameters from words 10 and 30 of the BCNAV-3 data broadcast. The data includes orbital parameters, group delay, health, and other status information.

**Message ID:** 2412

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnav3ephemerisa onchanged
```

**Example log:**

```
#BDSBCNAV3EPHEMERISA, USB1, 6, 50.5, SATTIME, 2209, 496718.000, 02000020, 070d, 16809;
26, 853, 00000000, 493200, 3, 107.203125, -0.005850, 3.7798003010e-09,
4.1571032661e-14, 2.051998724, 7.3472585063e-04, 0.436791966, 1.0602762178e-01,
9.4940386191e-01, -6.82564146e-09, 3.775157250e-10, -3.166496754e-08,
-3.073364496e-08, 103.6250000, 136.3750000, 1.077353954e-05, 5.120411515e-06,
493200, 7.38367089e-04, -5.27311528e-12, 0.00000000, -1.565786079e-08, 0*25ddf0a6

#BDSBCNAV3EPHEMERISA, USB1, 5, 50.5, SATTIME, 2209, 496718.000, 02000020, 070d, 16809;
35, 853, 00000000, 493200, 3, -78.089844, 0.004464, 3.9608792722e-09,
-3.1478013195e-14, 0.693860220, 8.1312097609e-04, 0.494353216, 2.199361578e+00,
9.6212843381e-01, -6.96993318e-09, -4.284107022e-10, -1.210719347e-08,
3.632158041e-08, -136.6210938, 203.5351563, 7.832422853e-06, -6.644055247e-06,
493200, 2.72334553e-04, 1.74029680e-11, 0.00000000, -1.315493137e-08, 0*152a37f1

...

#BDSBCNAV3EPHEMERISA, USB1, 1, 50.5, SATTIME, 2209, 496719.000, 02000020, 070d, 16809;
39, 853, 00000000, 493200, 2, 5263.894531, -0.062899, 7.7092496926e-10,
1.9584381616e-13, -2.140521199, 2.2453696001e-03, -2.871042649, -9.3801678773e-01,
9.6155928554e-01, -1.86489911e-09, -1.102724504e-09, -4.284083843e-08,
4.004687071e-08, -83.8671875, -507.8242188, 2.449937165e-05, -3.319233656e-06,
493200, 3.49998591e-05, 3.10862447e-14, 0.00000000, -1.268927008e-08, 0*d3f6f534

#BDSBCNAV3EPHEMERISA, USB1, 0, 50.5, SATTIME, 2209, 496718.000, 02000020, 070d, 16809;
44, 853, 00000000, 493200, 3, 65.369141, -0.011349, 3.8406956947e-09, 1.0343710014e-13,
-0.198007475, 8.3381979493e-04, 0.582406796, 2.199813208e+00, 9.5949562234e-01,
-6.96636161e-09, -4.343038048e-10, -5.215406418e-08, -4.190951586e-08,
-118.6484375, 212.3242188, 7.395632565e-06, -5.853362381e-06, 493200,
6.58427074e-04, 1.22231114e-11, 0.00000000, -1.204898581e-08, 0*0c179456
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV3EPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Satellite ID	ID/ranging code	Ulong	4	H
3	WN	Week number in BeiDou time (BDT). This will be offset from GPS time by 1356 weeks.	Ulong	4	H+4
4	Satellite Status	Satellite Status Information: Bits 0 to 1: Health Status (HS) Bit 2: Data Integrity Flag (DIF) Bit 3: Signal Integrity Flag (SIF) Bit 4: Accuracy Integrity Flag (AIF) Bits 5 to 8: Signal In Space Monitoring Accuracy Index (SISMAI)	Hex Ulong	4	H+8
5	toe	Reference time of ephemeris parameters in BeiDou time (BDT). This will be offset from GPS time. (s)	Ulong	4	H+12
6	Satellite Type	2 bit orbit type: 01 indicates GEO, 10 indicates IGSO, 11 indicates MEO	Ulong	4	H+16
7	DeltaA	Semi-major axis difference at reference time (m)	Double	8	H+20
8	ADot	Change rate in Semi-major axis (m/s)	Double	8	H+28
9	DeltaN	Mean motion difference from computed value (radians/s)	Double	8	H+36
10	NDot	Rate of mean motion difference from computed value (radians/s <sup>2</sup> )	Double	8	H+44
11	M <sub>0</sub>	Mean anomaly at reference time (radians)	Double	8	H+52
12	e	Eccentricity (unitless)	Double	8	H+60
13	Omega	Argument of perigee (radians)	Double	8	H+68
14	Omega <sub>0</sub>	Longitude of ascending node of orbital of plane at weekly epoch (radians)	Double	8	H+76
15	i <sub>0</sub>	Inclination angle at reference time (radians)	Double	8	H+84

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
16	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+92
17	i <sub>0</sub> Dot	Rate of inclination angle (radians/s)	Double	8	H+100
18	C <sub>is</sub>	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+108
19	C <sub>ic</sub>	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+116
20	C <sub>rs</sub>	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+124
21	C <sub>rc</sub>	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+132
22	C <sub>us</sub>	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+140
23	C <sub>uc</sub>	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+148
24	toc	Reference time of clock parameters in BeiDou time (BDT). (s) This will be offset from GPS time by 14 seconds.	Ulong	4	H+156
25	a <sub>0</sub>	Constant term of clock correction polynomial (s)	Double	8	H+160
26	a <sub>1</sub>	Linear term of clock correction polynomial (s/s)	Double	8	H+168
27	a <sub>2</sub>	Quadratic term of clock correction polynomial (s/s <sup>2</sup> )	Double	8	H+176
28	TGDB2Bi	Group delay differential between the B2b1 and the B3I signal (s)	Double	8	H+184
29	Reserved		Ulong	4	H+192
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+196
31	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 3.14 BDSBCNAV3RAWMESSAGE

### *Framed BDS B-CNAV3 raw navigation message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the framed BeiDou B-CNAV3 navigation message in raw bit format.

**Message ID:** 2411

**Log Type:** Asynch

**Recommended Input:**

```
log bdsbcnav3rawmessagea onnew
```

**Example log:**

```
#BDSBCNAV3RAWMESSAGEA,USB1,0,61.0,SATTIME,2209,496760.000,02000020,5050,16809;
116,24,10,eb9060029e51aa7366001307000ce2cacc9ffcf1ccb4589080081fc9815170ba002a7
3f8e89a7f40bbda3803d0000d7ffd00398c804d4580a61c02f40000,0,0*1c4e0ec2
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSBCNAV3RAWMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Signal channel	Signal channel number	Ulong	4	H
3	Satellite ID	Satellite ID / PRN	Ulong	4	H+4
4	Message type	Message type contained in the data	Ulong	4	H+8
5	Raw message data	Framed raw navigation data, 490 bits (CRC removed)	Hex[62]	62	H+12
6	Reserved		Char	1	H+74
7	Reserved		Char	1	H+75
8	xxxx	32-bit CRC	Hex	4	H+76
9	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 3.15 BDSCLOCK

### *BeiDou time parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains time parameters transmitted by the BeiDou satellites. These parameters can be used to calculate the offset between BeiDou time (BDT) and other time frames.

**Message ID:** 1607

**Log Type:** Asynch

**Recommended Input:**

```
log bdscllocka onchanged
```

**ASCII Example:**

```
#BDSCLOCKA,USB1,0,61.5,SATTIME,2209,496388.000,02000020,3b16,16809;
-9.313225746154785e-10,-9.769962616701378e-15,4,61,6,4,
0.0000000000000000e+00,0.0000000000000000e+00,0.0000000000000000e+00,
0.0000000000000000e+00,0.0000000000000000e+00,0.0000000000000000e+00*ce955bde
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSCLOCK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	A <sub>0</sub> UTC	BDT clock bias relative to UTC (s)	Double	8	H
3	A <sub>1</sub> UTC	BDT clock rate relative to UTC (s/s)	Double	8	H+8
4	$\Delta T_{LS}$	Delta time due to leap seconds before the new leap second is effective (s)	Short	2	H+16
5	WN <sub>LSF</sub>	Week number of the new leap second (8-bit week in BeiDou time (BDT))	Ushort	2	H+18
6	DN	Day number of week of the new leap second	Ushort	2	H+20
7	$\Delta T_{LSF}$	Delta time due to leap seconds after the new leap second effective	Short	2	H+22
8	A <sub>0</sub> GPS	BDT clock bias relative to GPS time (s)	Double	8	H+24
9	A <sub>1</sub> GPS	BDT clock rate relative to GPS time (s/s)	Double	8	H+32
10	A <sub>0</sub> Gal	BDT clock bias relative to Galileo time (s)	Double	8	H+40
11	A <sub>1</sub> Gal	BDT clock rate relative to Galileo time (s/s)	Double	8	H+48
12	A <sub>0</sub> GLO	BDT clock bias relative to GLONASS time (s)	Double	8	H+56

---

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
13	A <sub>1</sub> GLO	BDT clock rate relative to GLONASS time (s/s)	Double	8	H+64
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+72
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.16 BDSEPHEMERIS

### *Decoded BDS ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains a single set of BDS B1I ephemeris parameters with appropriate scaling applied. Multiple messages are transmitted, one for each SV ephemeris collected.

**Message ID:** 1696

**Log Type:** Asynch

**Recommended Input:**

```
log bdsephemerisa onchanged
```

**ASCII Example:**

```
#BDSEPHEMERISA, USB1, 37, 62.0, SATTIME, 2209, 496784.000, 02000020, 2626, 16809; 6, 853,
2.00, 0, 8.30e-09, -1.60e-09, 0, 493200, -2.22835224e-05, 7.54090124e-11, 0.00000000,
1, 493200, 6493.813231, 2.7420885162e-03, -2.990457924, 7.3681640565e-10,
-2.306284799, -8.5916965356e-01, -1.83364781e-09, 9.4558331017e-01,
-1.049329423e-09, -5.817040801e-06, 2.914015204e-05, -660.2187500, -168.3906250,
1.862645149e-08, -1.862645149e-09*7a36ae90

#BDSEPHEMERISA, USB1, 36, 62.0, SATTIME, 2209, 446414.000, 02000020, 2626, 16809; 7, 853,
2.00, 0, 1.42e-08, 7.00e-10, 0, 446400, -1.59412972e-04, -4.91571228e-11, 0.00000000,
1, 446400, 6493.661501, 2.0398380002e-03, 3.127789902, 2.1922341725e-09,
-1.234405677, 1.121228087e+00, -3.13084470e-09, 8.7727285371e-01, 1.428630937e-11,
4.341825843e-06, -1.132534817e-05, 564.6718750, 139.0781250, 5.261972547e-08,
-2.561137080e-08*f39a85dd

...

#BDSEPHEMERISA, USB1, 1, 62.0, SATTIME, 2209, 475214.000, 02000020, 2626, 16809; 46, 853,
2.00, 0, -2.41e-08, -2.41e-08, 1, 475200, -3.22318403e-04, -2.03899120e-11, 0.00000000,
1, 475200, 5282.635483, 6.9640530273e-04, -0.109104372, 3.9308780226e-09,
2.513097275, 1.3085507441e-01, -6.78742558e-09, 9.5335228732e-01, 3.782300405e-10,
5.458947271e-06, 1.009693369e-05, 153.7500000, 108.2187500, -8.568167686e-08,
-1.001171768e-07*8c90e4b5

#BDSEPHEMERISA, USB1, 0, 62.0, SATTIME, 2208, 422804.000, 02000020, 2626, 16809; 58, 852,
2.00, 1, -9.00e-09, 0.00, 28, 295200, 2.80906679e-05, 3.90798505e-13, 0.00000000,
29, 295200, 5282.617880, 4.8023159616e-04, 0.041959073, 3.3054948300e-09,
0.355899976, -2.126947333e+00, -6.67813531e-09, 9.7631596256e-01, 2.464388366e-11,
-9.997747838e-07, 9.569805115e-06, 175.8125000, -20.7031250, -1.164153218e-08,
-9.778887033e-09*2efb9c0a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSEPHemeris header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	satellite ID	ID/ranging code	Ulong	4	H
3	Week	Week number in BeiDou time (BDT) since the BeiDou system time start epoch (January 1, 2006)	Ulong	4	H+4
4	URA	User range accuracy (m). This is the evaluated URAI/URA lookup-table value.	Double	8	H+8
5	health 1	Autonomous satellite health flag. 0 means broadcasting satellite is good and 1 means not.	Ulong	4	H+16
6	tgdl	Equipment group delay differential for the B1 signal (s)	Double	8	H+20
7	tgdl	Equipment group delay differential for the B2 signal (s)	Double	8	H+28
8	AODC	Age of data, clock	Ulong	4	H+36
9	toc	Reference time of clock parameters (s)	Ulong	4	H+40
10	$a_0$	Constant term of clock correction polynomial (s)	Double	8	H+44
11	$a_1$	Linear term of clock correction polynomial (s/s)	Double	8	H+52
12	$a_2$	Quadratic term of clock correction polynomial (s/s <sup>2</sup> )	Double	8	H+60
13	AODE	Age of data, ephemeris	Ulong	4	H+68
14	toe	Reference time of ephemeris parameters (s)	Ulong	4	H+72
15	RootA	Square root of semi-major axis (sqrt(m))	Double	8	H+76
16	ecc	Eccentricity (dimensionless)	Double	8	H+84
17	$\omega$	Argument of perigee (radians)	Double	8	H+92
18	$\Delta N$	Mean motion difference from computed value (radians/s)	Double	8	H+100
19	$M_0$	Mean anomaly at reference time (radians)	Double	8	H+108
20	$\Omega_0$	Longitude of ascending node of orbital of plane computed according to reference time (radians)	Double	8	H+116
21	$\dot{\Omega}$	Rate of right ascension (radians/s)	Double	8	H+124



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
22	$i_0$	Inclination angle at reference time (radians)	Double	8	H+132
23	IDOT	Rate of inclination angle (radians/second)	Double	8	H+140
24	$c_{uc}$	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+148
25	$c_{us}$	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+156
26	$c_{rc}$	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+164
27	$c_{rs}$	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+172
28	$c_{ic}$	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+180
29	$c_{is}$	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+188
30	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+196
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.17 BDSIONO

### *BeiDou Klobuchar ionosphere delay model*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the Klobuchar ionosphere model parameters transmitted by the BeiDou satellites.

**Message ID:** 1590

**Log Type:** Asynch

**Recommended Input:**

```
log bdsionoa onchanged
```

**ASCII Example:**

```
#BDSIONOA,USB1,9,61.0,SATTIME,2209,496844.000,02000020,1956,16809;35,
2.700835466384888e-08,1.937150955200195e-07,-1.668930053710938e-06,
2.503395080566406e-06,157696.0,-884736.0,5373952.0,-3932160.0*c84eb5f1
```

...

```
#BDSIONOA,USB1,0,61.0,SATTIME,2209,496844.000,02000020,1956,16809;6,
2.700835466384888e-08,1.937150955200195e-07,-1.668930053710938e-06,
2.503395080566406e-06,157696.0,-884736.0,5373952.0,-3932160.0*1f0a025a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSIONO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	ID	Transmitting satellite ID	Ulong	4	H
3	$\alpha_0$	Alpha parameter constant term	Double	8	H+4
4	$\alpha_1$	Alpha parameter 1st order term	Double	8	H+12
5	$\alpha_2$	Alpha parameter 2nd order term	Double	8	H+20
6	$\alpha_3$	Alpha parameter 3rd order term	Double	8	H+28
7	$\beta_0$	Beta parameter constant term	Double	8	H+36
8	$\beta_1$	Beta parameter 1st order term	Double	8	H+44
9	$\beta_2$	Beta parameter 2nd order term	Double	8	H+52
10	$\beta_3$	Beta parameter 3rd order term	Double	8	H+60
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+68
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.18 BDSRAWNAVSUBFRAME

### *Raw BeiDou subframe data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw BeiDou subframe data with parity bits removed. Only subframes that have passed the parity check are output.

**Message ID:** 1695

**Log Type:** Asynch

**Recommended Input:**

```
log bdsrawnavsubframea onchanged
```

**ASCII Example:**

```
#BDSRAWNAVSUBFRAMEA,USB1,0,61.5,SATTIME,2209,496898.000,02000020,88f3,16809;142,26,B1D1,5,e2415e53d00e945264012f45b271a033e077c1cff684b81b23204963*36497110
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BDSRAWNAVSUBFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	signal channel	Signal channel number	Ulong	4	H
3	satellite ID	Satellite ID	Ulong	4	H+4
4	data source	Source of data (refer to <i>Table 84: Data Source</i> below)	Enum	4	H+8
5	subframe ID	Subframe identifier	Ulong	4	H+12
6	raw subframe data	Framed raw navigation bits	Hex[28]	28	H+16
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 84: Data Source**

ASCII	Binary	Description
B1D1	0	Data is from a B1I/D1 signal

<b>ASCII</b>	<b>Binary</b>	<b>Description</b>
B1D2	1	Data is from a B1I/D2 signal
B2D1	65536	Data is from a B2I/D1 signal
B2D2	65537	Data is from a B2I/D2 signal

## 3.19 BESTDATUMINFO

### *Datum information for best position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **BESTPOS** log (see page 520). For differential solutions, such as RTK or PSRDIFF, where the base station datum is not known, "UNKNOWN" is reported with an EPSG code of 0. Single-point and SBAS report "ECEF" and code 0.

**Message ID:** 2305

**Log Type:** Asynch

**Recommended Input:**

```
log bestdatuminfoa onchanged
```

**ASCII Example:**

```
#BESTDATUMINFOA,USB1,0,62.5,FINESTEERING,2209,501000.000,02000020,0c2b,16809;  
"ITRF2014",1165,2022.364,GOOD*ee0ebd8b
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the <b>BESTPOS</b> log.	Char[32]	variable	H
3	epsg_code	EPSG code of the datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year). Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation status. See <i>Table 85: Transformation Status</i> on the next page.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

**Table 85: Transformation Status**

<b>Binary</b>	<b>ASCII</b>	<b>Description</b>
0	GOOD	An exact transformation was found and used.
1	ECEF_EQUIVALENCY	An exact transformation could not be found, but a similar transformation involving another Earth-fixed transformation was found and used.
2	SERVICE_DETAILS_UNKNOWN	The datum of the input solution was not known and therefore a transformation could not take place.
3	REQUESTED_TRANSFORMATION_UNAVAILABLE	No transformation could be found to transform the solution into the requested datum.

## 3.20 BESTGNSSDATUMINFO

### *Datum information for BESTGNSS position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **BESTGNSSPOS** log (see page 1060). For differential solutions, such as RTK or PSRDIFF, where the base station datum is not known, "UNKNOWN" is reported with an EPSG code of 0. Single-point and SBAS report "ECEF" and code 0.

**Message ID:** 2302

**Log Type:** Asynch

**Recommended Input:**

```
log bestgnssdatuminfoa onchanged
```

**ASCII Example:**

```
#BESTGNSSDATUMINFOA,USB1,0,62.0,FINESTEERING,2209,501000.000,02000020,1790,16809;"ITRF2014",1165,2022.364,GOOD*c889c041
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTGNSSDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of position being output by the <b>BESTGNSSPOS</b> log.	Char[32]	variable	
3	epsg_code	EPSG code of the datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year). Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation Status. See <i>Table 85: Transformation Status</i> on the previous page.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.21 BESTPOS

### *Best position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

When positioning with GNSS, there are four parameters being solved for: latitude, longitude, height and receiver clock offset from GPS time. The quality of the solution for all four parameters depends on the geometry of where the satellites are with respect to the antenna (and receiver). The strength of the positioning geometry is indicated by Dilution of Precision (DOP) values, with lower DOP numbers indicating better geometry. Because all the GNSS satellites are above terrestrial receivers, the VDOP (vertical DOP) is the largest DOP value. This is why the reported standard deviation for height is usually larger than for latitude or longitude.

Accuracy is based on statistics and reliability is measured in percentages. When a receiver states it can measure height to one metre, this is an accuracy measure. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%. In other words, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (one metre) and the result is 95% reliability (error is less than two metres 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions. See also the note in the **GPGST** log on page 624 for CEP and RMS definitions.

This log contains the best position computed by the receiver. In addition, it reports several status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used.



#### **SPAN Systems**

On systems with SPAN enabled, this log contains the best available combined GNSS and Inertial Navigation System (INS - if available) position computed by the receiver.

With the system operating in an RTK mode, BESTPOS reflects the latest low-latency solution for up to 60 seconds after reception of the last base station observation. After this 60 second period, the position reverts to the best solution available and the degradation in accuracy is reflected in the standard deviation fields. If the system is not operating in RTK mode, pseudorange differential solutions continue for the time specified in the **PSRDIFFTIMEOUT** command (see page 320). If the receiver is SPAN enabled, the GNSS+INS combined solution is also a candidate for BESTPOS output.





The RTK system in the receiver provides two kinds of position solutions. The Matched RTK position is computed with buffered observations, so there is no error due to the extrapolation of base station measurements. This provides the highest accuracy solution possible at the expense of some latency which is affected primarily by the speed of the differential data link. The **MATCHEDPOS** log (see page 704) contains the matched RTK solution and can be generated for each processed set of base station observations.

The Low-Latency RTK position is computed from the latest local observations and extrapolated base station observations. This supplies a valid RTK position with the lowest latency possible at the expense of some accuracy. The degradation in accuracy is reflected in the standard deviation and is summarized in *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss). The amount of time that the base station observations are extrapolated is in the "differential age" field of the position log. The Low-Latency RTK system extrapolates for 60 seconds. The **RTKPOS** log (see page 866) contains the Low-Latency RTK position when valid, and an "invalid" status when a Low-Latency RTK solution could not be computed. The **BESTPOS** log contains either the low-latency RTK, PPP or pseudorange-based position, whichever has the smallest standard deviation.

RTK positioning uses the carrier phase observations from the receiver. The carrier phases are precise but ambiguous: the measurement includes an unknown integer number of cycles known as the "ambiguities". Determining these ambiguities is the key to unlocking the highest-accuracy GNSS positions. This determination is known as ambiguity resolution. Before the integer ambiguities can be resolved they are first estimated as real-numbered values, "floats" in computing parlance. After some period that depends on RTK baseline length, ionosphere activity, and other observing conditions, the ambiguities can be resolved into integers, making centimetre or even sub-centimetre positioning possible.



Different positioning modes have different maximum logging rates, which are also controlled by model option. The maximum rates are: 100 Hz for RTK, 100 Hz for pseudorange based positioning, 20 Hz for GLIDE (PDP) and 20 Hz for PPP.



On SMART antennas, the position in the **BESTPOS** log may be corrected for antenna height. If so this will be indicated in the Extended Solution Status. See Terrain Compensation in the [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for details.

#### Message ID: 42


#### Log Type: Synch

#### Recommended Input:

```
log bestposa ontime 1
```

#### ASCII Example:

```
#BESTPOSA,USB1,0,58.5,FINESTEERING,2209,502061.000,02000020,cdba,16809;
SOL_COMPUTED,PPP,51.15043706870,-114.03067882331,1097.3462,-17.0001,WGS84,
0.0154,0.0139,0.0288,"TSTR",11.000,0.000,43,39,39,38,00,00,7f,37*52483ac5
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on the next page	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (metres)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">  When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84. </div>	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#solnL1SVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66

Field	Field type	Description	Format	Binary Bytes	Binary Offset
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 86: Solution Status**

Binary	ASCII	Description
0	SOL_COMPUTED	Solution computed
1	INSUFFICIENT_OBS	Insufficient observations
2	NO_CONVERGENCE	No convergence
3	SINGULARITY	Singularity at parameters matrix
4	COV_TRACE	Covariance trace exceeds maximum (trace > 1000 m)
5	TEST_DIST	Test distance exceeded (maximum of 3 rejections if distance > 10 km)
6	COLD_START	Not yet converged from cold start
7	V_H_LIMIT	Height or velocity limits exceeded (in accordance with export licensing restrictions)
8	VARIANCE	Variance exceeds limits
9	RESIDUALS	Residuals are too large
10-12	Reserved	
13	INTEGRITY_WARNING	Large residuals make position unreliable
14-17	Reserved	


Binary	ASCII	Description
18	PENDING	<p>When a <b>FIX position</b> command is entered, the receiver computes its own position and determines if the fixed position is valid</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;">  <p>PENDING implies there are not enough satellites currently tracked to verify if the FIX POSITION entered into the receiver is valid. Under normal conditions, you should only see PENDING for a few seconds on power up before the GNSS receiver has locked onto its first few satellites. If your antenna is obstructed (or not plugged in) and you have entered a FIX POSITION command, then you may see PENDING indefinitely.</p> </div>
19	INVALID_FIX	The fixed position, entered using the <b>FIX position</b> command, is not valid
20	UNAUTHORIZED	Position type is unauthorized
21	Reserved	
22	INVALID_RATE	The selected logging rate is not supported for this solution type.

Table 87: Position or Velocity Type

Binary	ASCII	Description
0	NONE	No solution
1	FIXEDPOS	Position has been fixed by the <b>FIX position</b> command or by position averaging.
2	FIXEDHEIGHT	Position has been fixed by the <b>FIX height</b> or <b>FIX auto</b> command or by position averaging
3-7	Reserved	
8	DOPPLER_VELOCITY	Velocity computed using instantaneous Doppler
9-15	Reserved	
16	SINGLE	Solution calculated using only data supplied by the GNSS satellites
17	PSRDIFF	Solution calculated using pseudorange differential (DGPS, DGNSS) corrections
18	WAAS	Solution calculated using corrections from an SBAS satellite
19	PROPAGATED	Propagated by a Kalman filter without new observations
20-31	Reserved	
32	L1_FLOAT	Single-frequency RTK solution with unresolved, float carrier phase ambiguities
33	Reserved	

Binary	ASCII	Description
34	NARROW_FLOAT	Multi-frequency RTK solution with unresolved, float carrier phase ambiguities
35-47	Reserved	
48	L1_INT	Single-frequency RTK solution with carrier phase ambiguities resolved to integers
49	WIDE_INT	Multi-frequency RTK solution with carrier phase ambiguities resolved to wide-lane integers
50	NARROW_INT	Multi-frequency RTK solution with carrier phase ambiguities resolved to narrow-lane integers
51	RTK_DIRECT_INS	RTK status where the RTK filter is directly initialized from the INS filter
52	INS_SBAS	INS position, where the last applied position update used a GNSS solution computed using corrections from an SBAS (WAAS) solution
53	INS_PSRSP	INS position, where the last applied position update used a single point GNSS (SINGLE) solution
54	INS_PSRDIFF	INS position, where the last applied position update used a pseudorange differential GNSS (PSRDIFF) solution
55	INS_RTKFLOAT	INS position, where the last applied position update used a floating ambiguity RTK (L1_FLOAT or NARROW_FLOAT) solution
56	INS_RTKFIXED	INS position, where the last applied position update used a fixed integer ambiguity RTK (L1_INT, WIDE_INT or NARROW_INT) solution
57-67	Reserved	
68	PPP_CONVERGING	Converging TerraStar-C, TerraStar-C PRO or TerraStar-X solution
69	PPP	Converged TerraStar-C, TerraStar-C PRO or TerraStar-X solution
70	OPERATIONAL	Solution accuracy is within UAL operational limit
71	WARNING	Solution accuracy is outside UAL operational limit but within warning limit
72	OUT_OF_BOUNDS	Solution accuracy is outside UAL limits
73	INS_PPP_CONVERGING	INS position, where the last applied position update used a converging TerraStar-C, TerraStar-C PRO or TerraStar-X PPP (PPP_CONVERGING) solution
74	INS_PPP	INS position, where the last applied position update used a converged TerraStar-C, TerraStar-C PRO or TerraStar-X PPP (PPP) solution

Binary	ASCII	Description
77	PPP_BASIC_CONVERGING	Converging TerraStar-L solution
78	PPP_BASIC	Converged TerraStar-L solution
79	INS_PPP_BASIC_CONVERGING	INS position, where the last applied position update used a converging TerraStar-L PPP (PPP_BASIC) solution
80	INS_PPP_BASIC	INS position, where the last applied position update used a converged TerraStar-L PPP (PPP_BASIC) solution



PPP requires access to a suitable correction stream, delivered either through L-Band or the Internet. For L-Band delivered TerraStar or Veripos service, an L-Band capable receiver and software model is required, along with a subscription to the desired service. Contact NovAtel for TerraStar and Veripos subscription details.

**Table 88: GPS and GLONASS Signal-Used Mask**

Bit	Mask	Description
0	0x01	GPS L1 used in Solution
1	0x02	GPS L2 used in Solution
2	0x04	GPS L5 used in Solution
3	0x08	Reserved
4	0x10	GLONASS L1 used in Solution
5	0x20	GLONASS L2 used in Solution
6	0x40	GLONASS L3 used in Solution
7	0x80	Reserved

**Table 89: Galileo and BeiDou Signal-Used Mask**

Bit	Mask	Description
0	0x01	Galileo E1 used in Solution
1	0x02	Galileo E5a used in Solution
2	0x04	Galileo E5b used in Solution
3	0x08	Galileo ALTBOC used in Solution
4	0x10	BeiDou B1 used in Solution (B1I and B1C)
5	0x20	BeiDou B2 used in Solution (B2I, B2a and B2b)
6	0x40	BeiDou B3 used in Solution (B3I)
7	0x80	Galileo E6 used in Solution (E6B and E6C)

**Table 90: Extended Solution Status**

Bit	Mask	Description
0	0x01	If an RTK solution: an RTK solution has been verified If a PDP solution: solution is GLIDE Otherwise: Reserved
1-3	0x0E	Pseudorange Iono Correction 0 = Unknown or default Klobuchar model 1 = Klobuchar Broadcast 2 = SBAS Broadcast 3 = Multi-frequency Computed 4 = PSRDiff Correction 5 = NovAtel Blended Iono Value
4	0x10	RTK ASSIST active
5	0x20	0 = No antenna warning 1 = Antenna information is missing See the <b>RTKANTENNA</b> command on page 332
6	0x40	Reserved
7	0x80	0 = Terrain Compensation corrections are not used 1 = Position includes Terrain Compensation corrections

**Table 91: Supplemental Position Types and NMEA Equivalents**

Value	Documented Enum Name	NMEA Equivalent
68	PPP_CONVERGING	2
69	PPP	5
70	OPERATIONAL	4
71	WARNING	5
72	OUT_OF_BOUNDS	1
77	PPP_BASIC_CONVERGING	1
78	PPP_BASIC	2

## 3.22 BESTSATS

### *Satellites used in BESTPOS*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists the used and unused satellites for the corresponding BESTPOS solution. It also describes the signals of the used satellites or reasons for exclusions.

**Message ID: 1194**

**Log Type: Synch**

**Recommended Input:**

```
log bestsatsa ontime 1
```

Abbreviated ASCII Example:

```
<BESTSATS COM1 0 62.5 FINESTEERING 2220 503251.000 02000000 be05 16807
< 36
< GPS 18 GOOD 00000007
< GPS 16 GOOD 00000003
< GPS 5 GOOD 00000003
< GPS 27 GOOD 00000007
< GPS 29 GOOD 00000003
< GPS 10 GOOD 00000007
< GPS 23 GOOD 00000007
< GPS 13 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 26 GOOD 00000007
< QZSS 194 GOOD 00000007
< GLONASS 18-3 GOOD 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 15 SUPPLEMENTARY 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 24+2 GOOD 00000003
< GLONASS 23+3 GOOD 00000001
< GLONASS 17+4 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
< GALILEO 15 GOOD 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 3 GOOD 0000000f
< GALILEO 36 GOOD 0000000f
< GALILEO 5 GOOD 0000000f
< GALILEO 21 GOOD 0000000f
< GALILEO 30 SUPPLEMENTARY 0000000f
< GALILEO 9 GOOD 0000000f
< BEIDOU 28 GOOD 0000001d
< BEIDOU 33 GOOD 0000001d
< BEIDOU 42 GOOD 0000001d
< BEIDOU 27 GOOD 0000001d
< BEIDOU 43 GOOD 0000001d
< BEIDOU 41 GOOD 0000001d
< BEIDOU 40 GOOD 0000001d
< BEIDOU 11 GOOD 00000007
< BEIDOU 14 GOOD 00000007
```



Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	See <i>Table 124: Satellite System</i> on page 654	Enum	4	H+4
4	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest-order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS, Galileo, BeiDou, NavIC and QZSS, or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems.  In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2.	Ulong	4	H+8
5	Status	Satellite status. See <i>Table 92: Observation Statuses</i> below	Enum	4	H+12
6	Signal mask	See <i>Table 93: GPS Signal Mask</i> on the next page, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H + 4 + (#entries x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4 (#entries x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 92: Observation Statuses

Value	Name	Description
0	GOOD	Observation is good
1	BADHEALTH	Satellite is flagged as bad health in ephemeris or almanac

Value	Name	Description
2	OLDEPHEMERIS	Ephemeris >3 hours old
6	ELEVATIONERROR	Satellite was below the elevation cutoff
7	MISCLOSURE	Observation was too far from predicted value
8	NODIFFCORR	No differential correction available
9	NOEPHEMERIS	No ephemeris available
10	INVALIDIODE	IODE used is invalid
11	LOCKEDOUT	Satellite has been locked out
12	LOWPOWER	Satellite has low signal power
13	OBSL2	An L2 observation not directly used in the solution
15	UNKNOWN	Observation was not used because it was of an unknown type
16	NOIONOCORR	No ionosphere delay correction was available
17	NOTUSED	Observation was not used in the solution
18	OBSL1	An L1 observation not directly used in the solution
19	OBSE1	An E1 observation not directly used in the solution
20	OBSL5	An L5 observation not directly used in the solution
21	OBSE5	An E5 observation not directly used in the solution
22	OBSB2	A B2 observation not directly used in the solution
23	OBSB1	A B1 observation not directly used in the solution
24	OBSB3	A B3 observation not directly used in the solution
25	NOSIGNALMATCH	Signal type does not match
26	SUPPLEMENTARY	Observation contributes supplemental information to the solution
99	NA	No observation available
100	BAD_INTEGRITY	Observation was an outlier and was eliminated from the solution
101	LOSSOFLOCK	Lock was broken on this signal
102	NOAMBIGUITY	No RTK ambiguity type resolved

Table 93: GPS Signal Mask

Bit	Mask	Description
0	0x01	GPS L1 used in the solution
1	0x02	GPS L2 used in the solution
2	0x04	GPS L5 used in the solution

**Table 94: GLONASS Signal Mask**

Bit	Mask	Description
0	0x01	GLONASS L1 used in the solution
1	0x02	GLONASS L2 used in the solution
2	0x04	GLONASS L3 used in the solution

**Table 95: Galileo Signal Mask**

Bit	Mask	Description
0	0x01	Galileo E1 used in the solution
1	0x02	Galileo E5A used in the solution
2	0x04	Galileo E5B used in the solution
3	0x08	Galileo ALTBOC used in the solution
4	0x10	Galileo E6 used in the solution

**Table 96: BeiDou Signal Mask**

Bit	Mask	Description
0	0x01	BeiDou B1I used in the solution
1	0x02	BeiDou B2I used in the solution
2	0x04	BeiDou B3 used in the solution
3	0x08	BeiDou B1C used in the solution
4	0x10	BeiDou B2a used in the solution
5	0x20	BeiDou B2b used in the solution

**Table 97: QZSS Signal Mask**

Bit	Mask	Description
0	0x01	QZSS L1 used in the solution
1	0x02	QZSS L2 used in the solution
2	0x04	QZSS L5 used in the solution
4	0x08	QZSS L6 used in the solution

**Table 98: NavIC Signal Mask**

Bit	Mask	Description
2	0x04	NavIC L5 used in the solution

## 3.23 BESTUTM

### *Best available UTM data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the best available position computed by the receiver in UTM coordinates.

See also the **UTMZONE** command on page 467 and the **BESTPOS** log on page 520.



The latitude limits of the UTM System are 80°S to 84°N. If your position is outside this range, the BESTUTM log outputs a northing, easting and height of 0.0, along with a zone letter of "\*" and a zone number of 0, to indicate that the data in the log is unusable.

**Message ID:** 726

**Log Type:** Synch


**Recommended Input:**

```
log bestutma ontime 1
```

**ASCII Example:**

```
#BESTUTMA,USB1,0,59.0,FINESTEERING,2209,502191.000,02000020,a40a,16809;  
SOL_COMPUTED,PPP,11,U,5670746.6432,707661.8351,1097.3653,-17.0001,WGS84,0.0152,  
0.0132,0.0286,"TSTR",11.000,0.000,44,39,39,38,0,00,7f,37*5e568127
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTUTM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	z#	Longitudinal zone number	Ulong	4	H+8
5	zletter	Latitudinal zone letter	Ulong	4	H+12
6	northing	Northing (m) where the origin is defined as the equator in the northern hemisphere and as a point 10,000,000 metres south of the equator in the southern hemisphere (that is, a 'false northing' of 10,000,000 m)	Double	8	H+16

Field	Field type	Description	Format	Binary Bytes	Binary Offset
7	easting	Easting (m) where the origin is 500,000 m west of the central meridian of each longitudinal zone (that is, a 'false easting' of 500,000 m)	Double	8	H+24
8	hgt	Height above mean sea level (m)	Double	8	H+32
9	undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum   When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.	Float	4	H+40
10	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+44
11	N $\sigma$	Northing standard deviation (m)	Float	4	H+48
12	E $\sigma$	Easting standard deviation (m)	Float	4	H+52
13	hgt $\sigma$	Height standard deviation (m)	Float	4	H+56
14	stn id	Base station ID	Char[4]	4	H+60
15	diff_age	Differential age in seconds	Float	4	H+64
16	sol_age	Solution age in seconds	Float	4	H+68
17	#SVs	Number of satellites tracked	Uchar	1	H+72
18	#solnSVs	Number of satellites used in solution	Uchar	1	H+73
19	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+74
20	#solnMultiSV	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+75
21	Reserved		Uchar	1	H+76
22	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+77
23	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+78

Field	Field type	Description	Format	Binary Bytes	Binary Offset
24	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+79
25	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+80
26	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.24 BESTVEL

### *Best available velocity data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the best available velocity information computed by the receiver. In addition, it reports a velocity status indicator, which is needed to determine whether or not the corresponding data is valid. The velocities calculated by the receiver can have a latency associated with them. When present, the velocity time of validity is the time tag in the log minus the latency value.



The velocity is typically from the same source used in the BESTPOS solution. For example, if the BESTPOS is from the pseudorange filter, then the BESTVEL velocity type is the same as for PSRVEL. However, a specific velocity source can be chosen. See the **BESTVELTYPE** command on page 107.



In a BESTVEL log, the actual speed and direction of the receiver antenna over ground is provided. The receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of motion of the GNSS antenna relative to ground.

The RTK, PDP and PPP velocities are computed from the average change in position over the time interval between consecutive solutions. As such, they are an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the BESTVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the positioning filters are updated at a rate of 2 Hz. *This average velocity translates into a velocity latency of 0.25 seconds.* To reduce the latency, increase the update rate of the positioning filter being used by requesting the BESTVEL or BESTPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds.

If the velocity in the BESTVEL log comes from the pseudorange filter, it has been computed from instantaneous Doppler measurements. You know that you have an instantaneous Doppler derived velocity solution when the velocity type is PSRDIFF, WAAS or DOPPLER\_VELOCITY. The instantaneous Doppler derived velocity has low latency and is not position change dependent. If you change your velocity quickly, you can see this in the DOPPLER\_VELOCITY solution. Under typically seen dynamics with minimal jerk, the velocity latency is zero. Under extreme, high-jerk dynamics, the latency cannot be well represented: it will still be reported as being zero, but may be as high as 0.15 seconds. Such dynamics are typically only seen in simulated trajectories.

**Message ID: 99**

**Log Type: Synch**

**Recommended Input:**

```
log bestvela ontime 1
```

**ASCII Example:**

```
#BESTVELA,USB1,0,57.5,FINESTEERING,2209,502223.000,02000020,10a2,16809;  
SOL_COMPUTED,PPP,0.250,13.000,0.0025,28.358727,0.0021,0*e9418656
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results (s)	Float	4	H+8
5	age	Differential age in seconds	Float	4	H+12
6	hor spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	vert spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-





Velocity (speed and direction) calculations are computed from either Doppler or carrier phase measurements rather than from pseudorange measurements. Typical speed accuracies are around 0.03m/s (0.07 mph, 0.06 knots).

Direction accuracy is derived as a function of the vehicle speed. A simple approach would be to assume a worst case 0.03 m/s cross-track velocity that would yield a direction error function something like:

$$d(\text{speed}) = \tan^{-1}(0.03/\text{speed})$$

For example, if you are flying in an airplane at a speed of 120 knots or 62 m/s, the approximate directional error will be:

$$\tan^{-1}(0.03/62) = 0.03 \text{ degrees}$$

Consider another example applicable to hiking at an average walking speed of 3 knots or 1.5 m/s. Using the same error function yields a direction error of about 1.15 degrees.

You can see from both examples that a faster vehicle speed allows for a more accurate heading indication. As the vehicle slows down, the velocity information becomes less and less accurate. If the vehicle is stopped, a GNSS receiver still outputs some kind of movement at speeds between 0 and 0.5 m/s in random and changing directions. This represents the noise and error of the static position.

In a navigation capacity, the velocity information provided by your GNSS receiver is as, or more, accurate than that indicated by conventional instruments as long as the vehicle is moving at a reasonable rate of speed. It is important to set the GNSS measurement rate fast enough to keep up with all major changes of the vehicle's speed and direction. It is important to keep in mind that although the velocity vector is quite accurate in terms of heading and speed, the actual track of the vehicle might be skewed or offset from the true track by plus or minus 0 to 1.8 metres as per the standard positional errors.

## 3.25 BESTXYZ

### *Best available cartesian position and velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the receiver's best available position and velocity in ECEF coordinates. The position and velocity status fields indicate whether or not the corresponding data is valid. See *Figure 12: The WGS84 ECEF Coordinate System* on page 540, for a definition of the ECEF coordinates.

See also the **BESTPOS** log on page 520 and **BESTVEL** log on page 535.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID:** 241

**Log Type:** Synch

**Recommended Input:**

```
log bestxyza ontime 1
```

**ASCII Example:**

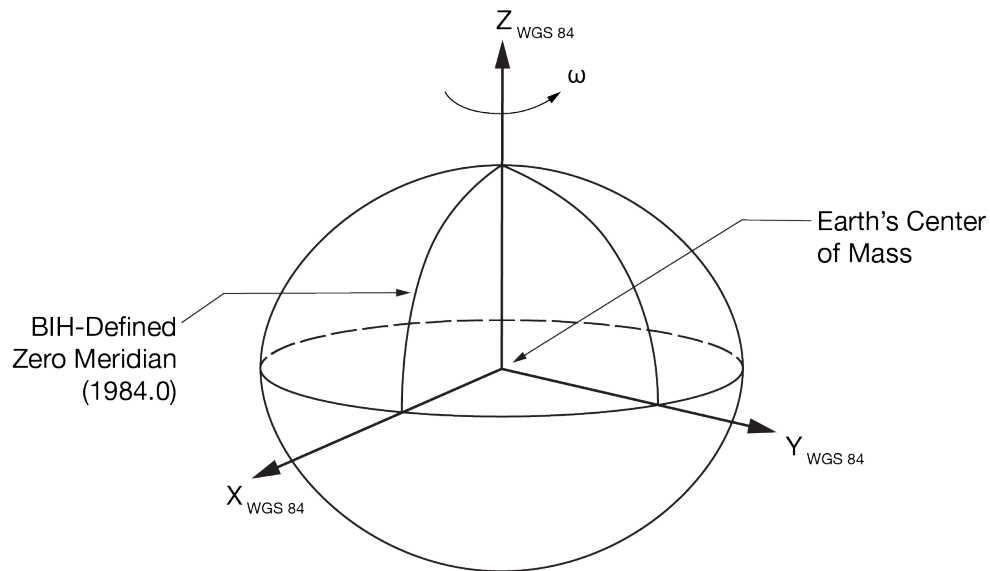
```
#BESTXYZA,USB1,0,59.0,FINESTEERING,2209,502264.000,02000020,44cf,16809;
SOL_COMPUTED,PPP,-1632848.2165,-3662158.6200,4944901.2721,0.0134,0.0197,0.0259,
SOL_COMPUTED,PPP,0.0007,0.0011,0.0009,0.0055,0.0069,0.0103,"TSTR",0.250,14.000,
0.000,44,40,40,39,0,00,7f,37*868939b6
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	P-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H+44
11	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m/s)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m/s)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m/s)	Double	8	H+68
15	V-X $\sigma$	Standard deviation of V-X (m/s)	Float	4	H+76
16	V-Y $\sigma$	Standard deviation of V-Y (m/s)	Float	4	H+80
17	V-Z $\sigma$	Standard deviation of V-Z (m/s)	Float	4	H+84
18	stn ID	Base station identification	Char[4]	4	H+88
19	V-latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellites used in solution	Uchar	1	H+105
24	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+106
25	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+107
26	Reserved		Char	1	H+108
27	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Figure 12: The WGS84 ECEF Coordinate System**

BIH - Defined CPT (1984.0)

**Table 99: Definitions**

- Origin = Earth's center of mass
- Z-Axis = Parallel to the direction of the Conventional Terrestrial Pole (CTP) for polar motion, as defined by the Bureau International de l'Heure (BIH) on the basis of the coordinates adopted for the BIH stations.
- X-Axis = Intersection of the WGS 84 Reference Meridian Plane and the plane of the CTP's Equator, the Reference Meridian being parallel to the Zero Meridian defined by the BIH on the basis of the coordinates adopted for the BIH stations.
- Y-Axis = Completes a right-handed, earth-centered, earth-fixed (ECEF) orthogonal coordinate system, measured in the plane of the CTP Equator, 90° East of the X-Axis.



These definitions are analogous to the BIH Defined Conventional Terrestrial System (CTS), or BTS, 1984.0.

## 3.26 BSLNXYZ

### *RTK XYZ baseline*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the receiver's RTK baseline in ECEF coordinates. The position status field indicates whether or not the corresponding data is valid. See *Figure 12: The WGS84 ECEF Coordinate System* on the previous page for a definition of the ECEF coordinates.

The BSLNXYZ log comes from time-matched base and rover observations such as in the **MATCHEDXYZ** log on page 709.



Asynchronous logs, such as BSLNXYZ, should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

**Message ID:** 686

**Log Type:** Asynch

**Recommended Input:**

```
log bslnxyza onchanged
```

**ASCII Example:**

```
#BSLNXYZA,USB1,0,66.0,FINESTEERING,2211,232444.000,02000020,6877,16809;  
SOL_COMPUTED,NARROW_INT,2.9912,3.1779,3.3197,0.0070,0.0074,0.0105,"K250",  
35,16,16,15,0,01,00,33*6c019b18
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	Log header	BSLNXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	bsln type	Baseline type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	B-X	X Baseline (m)	Double	8	H+8
5	B-Y	Y Baseline (m)	Double	8	H+16
6	B-Z	Z Baseline (m)	Double	8	H+24
7	B-X $\sigma$	Standard deviation of B-X (m)	Float	4	H+32

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
8	B-Y $\sigma$	Standard deviation of B-Y (m)	Float	4	H+36
9	B-Z $\sigma$	Standard deviation of B-Z (m)	Float	4	H+40
10	stn ID	Base station identification	Char[4]	4	H+44
11	#SVs	Number of satellites tracked	Uchar	1	H+48
12	#solnSVs	Number of satellites used in solution	Uchar	1	H+49
13	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+50
14	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+51
15	Reserved		Uchar	1	H+52
16	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+53
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+54
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+55
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.27 CHANCONFIGLIST

### *Channel configuration list*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the channel configuration list including the number of channels and signal types. If more than one channel configuration is available, then it can be switched using the **SELECTCHANCONFIG** command (see page 374).

**Message ID:** 1148

**Log Type:** Polled

**Recommended Input:**

```
log chanconfiglist once
```

**Abbreviated ASCII Example:**

```
<CHANCONFIGLIST USB1 0 65.0 FINESTEERING 2209 502508.560 02000020 d1c0 16809
<   3 9
<       7
<       16 GPSL1L2PL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1
<       14 GLOL1L2
<       16 GALE1E5B
<       30 BEIDOUB1B2B2B
<       5 LBAND
<       7
<       16 GPSL1L2
<       4 QZSSL1CAL2C
<       4 SBASL1
<       14 GLOL1L2
<       16 GALE1E5B
<       30 BEIDOUB1B2B
<       5 LBAND
<       7
<       16 GPSL1L2PL2CL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1
<       14 GLOL1L2PL2C
<       16 GALE1E5AE5BALTB0C
<       30 BEIDOUB1B2B2B
<       5 LBAND
<       8
<       16 GPSL1L2PL2CL5
<       4 QZSSL1CAL2CL5
<       4 SBASL1L5
<       14 GLOL1L2PL2C
<       16 GALE1E5AE5BALTB0C
<       20 BEIDOUB1B1CB2B2BB3
<       8 NAVICL5
<       5 LBAND
<       8
```

```

<      16 GPSL1L2PL2CL5L1C
<      4 QZSSL1CAL2CL5L1CL6
<      4 SBASL1L5
<      14 GLOL1L2PL2CL3
<      11 GALE1E5AE5BALTBOCE6
<      16 BEIDOU1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<      16 GPSL1L2PL2CL5
<      6 QZSSL1CAL2CL5
<      4 SBASL1L5
<      14 GLOL1L2PL2CL3
<      9 GALE1E5AE5BALTBOCE6
<      24 BEIDOU1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<      16 GPSL1L2PL2CL5
<      4 QZSSL1CAL2CL5
<      4 SBASL1L5
<      14 GLOL1L2PL2CL3
<      16 GALE1E5AE5BALTBOCE6
<      16 BEIDOU1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<      16 GPSL1L2PL2CL5
<      4 QZSSL1CAL2CL5L1CL6
<      4 SBASL1L5
<      14 GLOL1L2PL2CL3
<      11 GALE1E5AE5BALTBOCE6
<      20 BEIDOU1B1CB2B2BB3
<      8 NAVICL5
<      5 LBAND
<      8
<      16 GPSL1L2PL2CL5L1C
<      4 QZSSL1CAL2CL5L1C
<      4 SBASL1L5
<      14 GLOL1L2PL2CL3
<      16 GALE1E5AE5BALTBOC
<      20 BEIDOU1B1CB2B2B
<      8 NAVICL5
<      5 LBAND

```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CHANCONFIGLIST header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0



Field	Field type	Description	Format	Binary Bytes	Binary Offset
2	SetInUse	Current channel configuration being used. For example, if SetInUse is 2 then the second channel configuration listed in this log is the current channel configuration	Ulong	4	H
3	#chanconfigs	Number of channel configurations to follow	Ulong	4	H+4
4	#signaltypes	Total number of signal types in this channel configuration	Ulong	4	H+8
5	NumChans	Number of channels for individual signal type	Ulong	4	H+12
6	SignalType	See <i>Table 100: CHANCONFIGLIST Signal Type</i> below	Ulong	4	H+16
7	Next chanconfig offset = H + 8 + (#chanconfigs * (4 + (#signaltypes * 8)))				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 100: CHANCONFIGLIST Signal Type**

Value	Name	Description
0	GPSL1	GPS L1 C/A
1	GPSL1L2	GPS L1 C/A and L2 P(Y)
4	SBASL1	SBAS L1 C/A
5	GPSL5	GPS L5
6	GPSL1L2C	GPS L1 C/A and L2C
7	GPSL1L2AUTO	GPS L1 C/A and L2 P(Y) or L2C
8	GLOL1L2	GLONASS L1 C/A and L2 P
9	LBAND	L-Band
10	GLOL1	GLONASS L1 C/A
11	GALE1	Galileo E1
12	GALE5A	Galileo E5a
13	GALE5B	Galileo E5b
14	GALALTBOC	Galileo E5 AltBOC
15	BEIDOUB1	BeiDou B1I
16	GPSL1L2PL2C	GPS L1 C/A, L2 P(Y) and L2C

Value	Name	Description
17	GPSL1L5	GPS L1 C/A and L5
18	SBASL1L5	SBAS L1 C/A and L5
19	GPSL1L2PL2CL5	GPS L1 C/A, L2 P(Y), L2C and L5
20	GPSL1L2PL5	GPS L1 C/A, L2 P(Y) and L5
21	GALE1E5AE5B	Galileo E1, E5a and E5b
22	GALE1E5AE5BALTBOC	Galileo E1, E5a, E5b and E5 AltBOC
23	GALE1E5A	Galileo E1 and E5a
24	GLOL1L2C	GLONASS L1 C/A and L2 C/A
25	GLOL1L2PL2C	GLONASS L1 C/A, L2 P and L2 C/A
26	QZSSL1CA	QZSS L1 C/A
27	QZSSL1CAL2C	QZSS L1 C/A and L2C
28	QZSSL1CAL2CL5	QZSS L1 C/A, L2C and L5
29	QZSSL1CAL5	QZSS L1 C/A and L5
31	GALE1E5B	Galileo E1 and E5b
32	BEIDOUB1B3	BeiDou B1I and B3I
33	BEIDOUB3	BeiDou B3I
35	GALE1E5AE5BALTBOCE6	Galileo E1, E5a, E5b, E5 AltBOC and E6
36	GPSL1L2PL2CL5L1C	GPS L1 C/A, L2 P(Y), L2C, L5 and L1C
37	QZSSL1CAL2CL5L1C	QZSS L1 C/A, L2C, L5 and L1C
38	QZSSL1CAL2CL5L1CL6	QZSS L1 C/A, L2C, L5, L1C and L6
39	GLOL1L3	GLONASS L1 C/A and L3
40	GLOL3	GLONASS L3
41	GLOL1L2PL2CL3	GLONASS L1 C/A, L2 P, L2 C/A and L3
42	GPSL1L2PL2CL1C	GPS L1 C/A, L2 P(Y), L2C and L1C
43	QZSSL1CAL2CL1C	QZSS L1 C/A, L2C and L1C
44	NAVICL5	NavIC L5
45	BEIDOUB1C	BeiDou B1C
46	BEIDOUB1B1C	BeiDou B1I and B1C
49	BEIDOUB1B2IB2B	BeiDou B1I, B2I, and B2b

Value	Name	Description
50	BEIDOU B1B2B	BeiDou B1I, and B2b
51	BEIDOU B1B1CB2IB2B	BeiDou B1I, B1C, B2I and B2b
52	BEIDOU B1B1CB2IB2BB3	BeiDou B1I, B1C, B2I, B2b and B3
53	BEIDOU B1B1CB2B	BeiDou B1I, B1C and B2b
54	BEIDOU B1B2IB2BB3	BeiDou B1I, B2I, B2b and B3
55	BEIDOU B1B2B2B	BeiDou B1I, B2I/B2a and B2b
56	BEIDOU B1B1CB2B2B	BeiDou B1I, B1C, B2I/B2a and B2b
57	BEIDOU B1B1CB2B2BB3	BeiDou B1I, B1C, B2I/B2a, B2b and B3
58	BEIDOU B1B2B2BB3	BeiDou B1I, B2I/B2a, B2b and B3



Configurations with **B2I** will track the B2I signal provided by BeiDou Phase II satellites. Configurations with **B2** will track either the B2I or B2a signal corresponding to BeiDou Phase II and Phase III satellites respectively, if the frequency is enabled by the model.

## 3.28 CLOCKMODEL

### *Current clock model status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The CLOCKMODEL log contains a filtered representation of the receiver's clock bias relative to GPS system time.

**Message ID:** 16

**Log Type:** Synch

**Recommended Input:**

```
log clockmodela ontime 1
```

**ASCII Example:**

```
#CLOCKMODEL, USB1, 0, 65.0, FINESTEERING, 2209, 502563.000, 02000020, 98f9, 16809;
VALID, 0, 502563.000, 502563.000, 1.645927507e-01, -3.828977102e-02, 0.000000000,
3.705045540e-02, 8.770260153e-04, 0.000000000, 8.770260153e-04, 3.242199713e-03,
0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.227, -4.714598090e-02, FALSE
*6572f493
```



The CLOCKMODEL log can be used to monitor the clock drift of an internal oscillator once the CLOCKADJUST mode has been disabled. Watch the CLOCKMODEL log to see the drift rate and adjust the oscillator until the drift stops.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CLOCKMODEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	status	Clock model status. See <i>Table 101: Clock Model Status</i> on the next page	Enum	4	H
3	reject_count	Number of rejected instantaneous clock errors	Ulong	4	H+4
4	propagation_time	Time of last propagation	GPSec	4	H+8
5	update_time	Time of last update	GPSec	4	H+12
6	bias	Receiver clock bias (m)	Double	8	H+16
7	rate	Receiver clock bias rate (m/s)	Double	8	H+24
8	Reserved		Double	8	H+32

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	bias_variance	Receiver clock bias variance (m <sup>2</sup> )	Double	8	H+40
10	covariance	Receiver clock bias/bias rate covariance (m <sup>2</sup> /s)	Double	8	H+48
11	Reserved		Double	8	H+56
12	Reserved		Double	8	H+64
13	rate_variance	Receiver clock bias rate variance (m <sup>2</sup> /s <sup>2</sup> )	Double	8	H+72
14	Reserved		Double	8	H+80
15	Reserved		Double	8	H+88
16	Reserved		Double	8	H+96
17	Reserved		Double	8	H+104
18	instantaneous_bias	Last instantaneous receiver clock bias (m)	Double	8	H+112
19	instantaneous_rate	Last instantaneous receiver clock bias rate (m/s)	Double	8	H+120
20	Reserved		Bool	4	H+128
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+132
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 101: Clock Model Status

Clock Status (Binary)	Clock Status (ASCII)	Description
0	VALID	The clock model is valid
1	CONVERGING	The clock model is near validity
2	ITERATING	The clock model is iterating towards validity
3	INVALID	The clock model is not valid

## 3.29 CLOCKSTEERING

### *Clock steering status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The CLOCKSTEERING log is used to monitor the current state of the clock steering process. All oscillators have some inherent drift. By default the receiver attempts to steer the receiver clock to accurately match GPS reference time. If for some reason this is not desired, this behavior can be disabled using the **CLOCKADJUST** command (see page 115).



If the **CLOCKADJUST** command (see page 115) is ENABLED and the receiver is configured to use an external reference frequency (set in the **EXTERNALCLOCK** command (see page 168)), then the clock steering process takes over the VARF output pins and may conflict with a previously entered **EVENTOUTCONTROL** command (see page 165).

**Message ID:** 26

**Log Type:** Asynch

**Recommended Input:**

```
log clocksteeringa onchanged
```

**ASCII Example:**

```
#CLOCKSTEERINGA,USB1,0,64.0,FINESTEERING,2209,502647.507,02000020,0f61,16809;
INTERNAL,SECOND_ORDER,11000,6888.960937500,0.029999999,0.621999979,0.357,-0.027
*5368cd23
```



To configure the receiver to use an external reference oscillator, see the **EXTERNALCLOCK** command on page 168.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CLOCKSTEERING header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	source	Clock source, see <i>Table 102: Clock Source</i> on the next page	Enum	4	H
3	steering state	Steering state, see <i>Table 103: Steering State</i> on the next page	Enum	4	H+4
4	period	Period of the pulse width modulated signal used to control the oscillator. This value is set using the <b>CLOCKCALIBRATE</b> command (see page 117).	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	pulse width	Current pulse width of the pulse width modulated signal used to control the oscillator. The starting point for this value is set using the <b>CLOCKCALIBRATE</b> command (see page 117). The clock steering loop continuously adjusts this value in an attempt to drive the receiver clock offset and drift terms to zero.	Double	8	H+12
6	bandwidth	The current band width of the clock steering tracking loop in Hz. This value is set using the <b>CLOCKCALIBRATE</b> command (see page 117).	Double	8	H+20
7	slope	The current clock drift change in m/s/bit for a 1 LSB pulse width. This value is set using the <b>CLOCKCALIBRATE</b> command (see page 117).	Float	4	H+28
8	offset	The last valid receiver clock offset computed (m). It is the same as Field # 18 of the <b>CLOCKMODEL</b> log (see page 548).	Double	8	H+32
9	drift rate	The last valid receiver clock drift rate received (m/s). It is the same as Field # 19 of the <b>CLOCKMODEL</b> log (see page 548).	Double	8	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 102: Clock Source

Binary	ASCII	Description
0	INTERNAL	The receiver is currently steering its internal VCTCXO using an internal VARF signal
1	EXTERNAL	The receiver is currently steering an external oscillator using the external VARF signal

Table 103: Steering State

Binary	ASCII	Description
0	FIRST_ORDER	Upon start-up, the clock steering task adjusts the VARF pulse width to reduce the receiver clock drift rate to below 1 ms using a 1st order control loop. This is the normal start-up state of the clock steering loop.
1	SECOND_ORDER	Once the receiver has reduced the clock drift to below 1 m/s, it enters a second order control loop and attempts to reduce the receiver clock offset to zero. This is the normal runtime state of the clock steering process.

Binary	ASCII	Description
2	CALIBRATE_ HIGH	<p>This state corresponds to when the calibration process is measuring at the "High" pulse width setting.</p> <p>The CALIBRATE_HIGH state is only seen if you force the receiver to do a clock steering calibration using the <b>CLOCKCALIBRATE</b> command (see page 117). With the <b>CLOCKCALIBRATE</b> command (see page 117), you can force the receiver to calibrate the slope and center pulse width of the currently selected oscillator, to steer. The receiver measures the drift rate at several "High" and "Low" pulse width settings.</p>
3	CALIBRATE_ LOW	<p>This state corresponds to when the calibration process is measuring at the "Low" pulse width setting.</p> <p>The CALIBRATE_LOW state is only seen if you force the receiver to do a clock steering calibration using the <b>CLOCKCALIBRATE</b> command (see page 117). With the <b>CLOCKCALIBRATE</b> command (see page 117), you can force the receiver to calibrate the slope and center pulse width of the currently selected oscillator, to steer. The receiver measures the drift rate at several "High" and "Low" pulse width settings.</p>
4	CALIBRATE_ CENTER	<p>This state corresponds to the "Center" calibration process. Once the center has been found, the modulus pulse width, center pulse width, loop bandwidth and measured slope values are saved in NVM and are used from now on for the currently selected oscillator (INTERNAL or EXTERNAL).</p> <p>After the receiver has measured the "High" and "Low" pulse width setting, the calibration process enters a "Center calibration" process where it attempts to find the pulse width required to zero the clock drift rate.</p>



## 3.30 DATUMTRANSFORMATIONS

### *Available datum transformations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum transformations known to the receiver. This includes the default transformations included in the firmware, plus user-defined transformations entered using the **DATUMTRANSFORMATION** command (see page 135).

The entry "USER" to "WGS84" transformation in this log is the datum transformation set by the **USERDATUM** command (see page 456) or the **USEREXPDATUM** command (see page 458). The parameters in this entry will differ from those entered via the user datum commands due to differences in units and convention.

**Message ID: 2298**

**Log Type: Asynch**

**Recommended Input:**

```
log datumtransformations
```

**Abbreviated ASCII Example:**

```
<DATUMTRANSFORMATIONS USB1 0 55.0 UNKNOWN 0 0.167 02000020 f957 16809
< 12
< "ITRF2014" "ITRF2000" 2010.000 0.0007 0.0012 -0.0261 0.0000 0.0000
0.0000 2.120 0.0001 0.0001 -0.0019 0.0000 0.0000 0.0000 0.110
< "ITRF2014" "ITRF2005" 2010.000 0.0026 0.0010 -0.0023 0.0000 0.0000
0.0000 0.920 0.0003 0.0000 -0.0001 0.0000 0.0000 0.0000 0.030
< "ITRF2014" "ITRF2008" 2010.000 0.0016 0.0019 0.0024 0.0000 0.0000
0.0000 -0.020 0.0000 0.0000 -0.0001 0.0000 0.0000 0.0000 0.030
< "ITRF2014" "ETRF2000" 2010.000 0.0547 0.0522 -0.0741 1.7010 10.2900
-16.6320 2.120 0.0001 0.0001 -0.0019 0.0810 0.4900 -0.7920 0.110
< "ITRF2014" "ETRF2005" 1989.000 0.0523 0.0490 -0.0372 0.0000 0.0000
0.0000 0.290 0.0003 0.0000 -0.0001 0.0540 0.5180 -0.7810 0.030
< "ITRF2014" "ETRF2014" 2010.000 0.0000 0.0000 0.0000 1.7850 11.1510
-16.1700 0.000 0.0000 0.0000 0.0000 0.0850 0.5310 -0.7700 -0.000
< "ITRF2014" "NAD83(NSRS2011)" 1997.000 0.9950 -1.9014 -0.5228 -25.9147
-9.4264 -11.5993 1.305 0.0008 -0.0006 -0.0014 -0.0667 0.7574 0.0513 -0.072
< "ITRF2014" "NAD83(CSRS)" 1997.000 0.9950 -1.9014 -0.5228 -25.9147
-9.4264 -11.5993 1.305 0.0008 -0.0006 -0.0014 -0.0667 0.7574 0.0513 -0.072
< "ITRF2014" "NAD83(PA11)" 1997.000 0.9096 -2.0142 -0.5616 -27.7410
-13.4690 -2.7120 0.690 0.0001 0.0001 -0.0019 0.3840 -1.0070 2.1860 0.110
< "ITRF2014" "NAD83(MA11)" 1997.000 0.9096 -2.0142 -0.5616 -28.9710
-10.4200 -8.9280 0.690 0.0001 0.0001 -0.0019 0.0200 -0.1050 0.3470 0.110
< "ITRF2014" "GDA94" 1994.000 -0.0831 -0.0175 0.0360 0.4254 -2.2578
-2.4015 9.210 0.0014 0.0013 0.0008 -1.5461 -1.1820 -1.1551 0.139
< "ITRF2014" "GDA2020" 2020.000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.000 0.0000 0.0000 0.0000 -1.5038 -1.1835 -1.2072 0.000
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	DATUMTRANSFORMATIONS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	# transformations	Number of transformations to follow	Ulong	4	H
3	from	Name of from datum	Char[32]	variable	H+4
4	to	Name of to datum	Char[32]	variable	variable
5	epoch	Reference epoch of parameters (decimal year) Examples: 2011.00 = Jan 1, 2011, 2011.19 = Mar 11, 2011	Double	8	variable
6	tx	Translations at reference epoch (metres)	Float	4	variable
7	ty		Float	4	variable
8	tz		Float	4	variable
9	rx	Rotations at reference epoch (milliarcseconds)	Float	4	variable
10	ry		Float	4	variable
11	rz		Float	4	variable
12	scale	Scale difference at reference epoch (parts per billion)	Float	4	variable
13	tx_vel	Translation rates (metres/year)	Float	4	variable
14	ty_vel		Float	4	variable
15	tz_vel		Float	4	variable
16	rx_vel	Rotation rates (milliarcseconds/year)	Float	4	variable
17	ry_vel		Float	4	variable
18	rz_vel		Float	4	variable
19	scale_vel	Scale difference rate (parts per billion/year)	Float	4	variable
21...	Next transformation offset = variable				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	variable
variable	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.31 DUALANTENNAHEADING

### *Synchronous heading information for dual antenna product*

**Platform:** OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2, CPT7

The heading is the angle from True North of the primary antenna to secondary antenna vector in a clockwise direction.



You must have an ALIGN capable, dual antenna receiver to use this log.

**Message ID:** 2042

**Log Type:** Synch

**Recommended Input:**

```
log dualantennaheadinga ontime 1
```

**ASCII Example:**

```
#DUALANTENNAHEADINGA, UNKNOWN, 0, 66.5, FINESTEERING, 1949, 575614.000, 02000000, d426,
32768; SOL_COMPUTED, NARROW_INT, -1.000000000, 255.538528442, 0.006041416, 0.0,
0.043859947, 0.052394450, "J56X", 24, 18, 18, 17, 04, 01, 00, 33*1f082ec5
```

Field	Field type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	DUALANTENNAHEADING header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	length	Baseline length in metres For ALIGN Heading models, this field is -1. For ALIGN Relative Positioning models with a fixed position, this field is -1. For ALIGN Relative Positioning models, this field is the baseline length in metres, unless the position is fixed.	Float	4	H+8
5	heading	Heading in degrees (0° to 359.999°)	Float	4	H+12
6	pitch	Pitch (±90 degrees)	Float	4	H+16
7	Reserved		Float	4	H+20

Field	Field type	Description	Binary Format	Binary Bytes	Binary Offset
8	hdg std dev	Heading standard deviation in degrees	Float	4	H+24
9	ptch std dev	Pitch standard deviation in degrees	Float	4	H+28
10	stn ID	Station ID string	Char[4]	4	H+32
11	#SVs	Number of satellites tracked	Uchar	1	H+36
12	#solnSVs	Number of satellites used in solution	Uchar	1	H+37
13	#obs	Number of satellites above the elevation mask angle	Uchar	1	H+38
14	#multi	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+39
15	sol source	Solution source (see <i>Table 123: Solution Source</i> on page 648)	Hex	1	H+40
16	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+41
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+42
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+43
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.32 ETHSTATUS

### Current Ethernet status

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This log provides the current status of the Ethernet ports.

**Message ID:** 1288

**Log Type:** Polled

**Recommended Input:**

```
log ethstatusa once
```

**ASCII Example:**

```
#ETHSTATUSA,USB1,0,65.0,FINESTEERING,2209,502756.000,02000020,e89d,16809;
1,ETHA,"00-21-66-04-18-FE",NOTCONNECT*9699791a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ETHSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# of interfaces	Number of records to follow	Ulong	4	H
3	interface	Name of the Ethernet interface (e.g., ETHA)	Enum	4	H+4
4	MAC address	An identifier assigned to the network adapters or network interface card	String [18]	variable <sup>1</sup>	H+8
5	interface configuration	Current connectivity, speed and duplex settings of the Ethernet interface	Enum	4	H+26
6...	Next interface = H+4+(# of interfaces * 26)				
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(# of interfaces * 26)
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



Refer to the **ETHCONFIG** command (see page 159) for enum values.

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

### 3.33 FILELIST

#### *Display the storage media contents*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this log.

Use this log to display the root directory of the active media. A log is produced for each file and directory in the root directory.

The active media is set with the **FILEMEDIACONFIG** command on page 176.

**Message ID:** 2100

**Log Type:** Asynch

**Recommended Input:**

```
log filelista
```

**ASCII Example:**

```
#FILELISTA,USB1,16,65.5,FINESTEERING,2209,502825.000,02000020,50e7,16809;
INTERNAL_FLASH,FILE,419859,20200409,221434,
"NMNE17200009B_2020-04-09_22-14-34.LOG"*2ea4d1bf

#FILELISTA,USB1,15,65.5,FINESTEERING,2209,502825.000,02000020,50e7,16809;
INTERNAL_FLASH,FILE,14370264,20200409,223412,
"NMNE17200009B_2020-04-09_22-34-11.LOG"*36c44730

#FILELISTA,USB1,0,65.5,FINESTEERING,2209,502825.000,02000020,50e7,16809;
INTERNAL_FLASH,DIR,0,20200306,162238,"System Volume Information"*4cf7cae9
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	FILELIST header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	MassStorageDevice	Mass Storage Device See <i>Table 105: Mass Storage Device</i> on page 561	Enum	4	H
3	FileType	The type of entry for this log. See <i>Table 104: File Type</i> on the next page	Enum	4	H+4
4	FileSize	File Size (in Bytes)	Ulong	4	H+8

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	ChangeDate	Date of the last change	Ulong	4	H+12
6	ChangeTime	Time of last change	Ulong	4	H+16
7	FileName	Name of the file or directory File Name STRING Variable H + 20	String	Variable	H+20
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	Variable
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 104: File Type

Binary	ASCII	Description
0	NONE	Indicates there are no entries in the selected media
1	FILE	File
2	DIR	Directory



When there no files or directories on the specified media, a single **FILELIST** log is output with **FileType** set to *NONE* and file information set to 0 and empty strings.

## 3.34 FILESTATUS

*Displays the state of the data log file*

Platform: OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this log.

Use this log to display the current state of the data log file. Typically the **FILESTATUS** log is used to determine if the log file is open for writing or closed. However, it also shows any error that has occurred.

**Message ID: 2127**

**Log Type: Asynch**

**Recommended Input:**

```
log filestatusa
```

**ASCII Example**

```
#FILESTATUSA,USB1,0,65.5,FINESTEERING,2209,502895.032,02000020,4dbd,16809;
INTERNAL_FLASH,CLOSED,"",0,15726822,15754462,""*791f9a6e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	FILESTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	MassStorageDevice	The type of recording device See <i>Table 105: Mass Storage Device</i> on the next page.	Enum	4	H
3	FileStatus	File status See <i>Table 106: File Status</i> on the next page.	Enum	4	H+4
4	FileName	Filename of the log file	Fixed Uchar Array	MAX_FILENAME_LENGTH (MFL)	H+8



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	FileSize	File Size (bytes)	Ulong	4	H+MFL+8
6	MediaRemainingCapacity	Remaining capacity on the storage media (kb)	Ulong	4	H+MFL+12
7	MediaTotalCapacity	Total capacity of the storage media (kb)	Ulong	4	H+MFL+16
8	ErrorMsg	Error Message	String	Variable	H+MFL+20
9	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	Variable
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 105: Mass Storage Device

Binary	ASCII	Description
1	USBSTICK	USB mass storage device
3	NO_STORAGE	No mass storage
4	INTERNAL_FLASH	Internal eMMC flash

Table 106: File Status

Binary	ASCII	Description
1	OPEN	Log file is open
2	CLOSED	Log file is closed
3	BUSY	Media device is busy
4	ERROR	An error has occurred
5	COPY	Media is copying
6	PENDING	Waiting for the system to complete an operation. The file can not be opened until the operation is complete.

### 3.35 FILESYSTEMCAPACITY

*Displays storage capacity available*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this log.

Use this log to check the amount of storage capacity available in both the internal and external storage.

**Message ID: 2137**

**Log Type: Polled**

**Recommended Input:**

```
log filesystemcapacity
```

**Abbreviated ASCII Example:**

```
<FILESYSTEMCAPACITY USB1 0 66.0 FINESTEERING 2209 503457.025 02000020 fded
16809
<      2
<      USBSTICK 30984437760 27656192
<      INTERNAL_FLASH 15754461184 27639808
```

**ASCII Example:**

```
#FILESYSTEMCAPACITYA,USB1,0,65.0,FINESTEERING,2209,503492.024,02000020,fded,
16809;2,USBSTICK,30984437760,27656192,INTERNAL_FLASH,15754461184,27639808
*63cab762
```



The INTERNAL\_FLASH is only present on the PwrPak7, CPT7 and CPT7700.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	FILESYSTEMCAPACITY header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	#Dev	Number of device with data to follow	Ulong	4	H
3	MassStorageDevice	File system type (recording device) See <i>Table 105: Mass Storage Device</i> on the previous page	Enum	4	H+4
4	TotalStorage	Total storage on device in bytes	Ulong	8	H+8

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	UsedStorage	Amount of storage used on the device in bytes	Ulong	8	H+16
6	Next device offset = $H+4+(\#Dev \times 20)$				
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#Dev x 20)
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.36 FILESYSTEMSTATUS

### *Display state of recording media*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700



PwrPak7M variants do not support this log.

This log displays the current state of the recording media. It can be used to determine the state of the file system, such as any mounting errors.



When logging the FILESYSTEMSTATUS log, use the **ONNEW** or **ONCHANGED** log trigger.

**Message ID:** 2104

**Log Type:** Asynch

**Recommended Input:**

```
log filesystemstatusa onchanged
```

**Abbreviated ASCII Example:**

```
<FILESYSTEMSTATUS USB1 1 65.5 FINESTEERING 2210 166466.301 02000020 143c 16809
<   USBSTICK MOUNTED 30984440 ""
<FILESYSTEMSTATUS USB1 0 65.5 UNKNOWN 0 15.440 02000020 143c 16809
<   INTERNAL_FLASH MOUNTED 15754462 ""
```

**ASCII Example:**

```
#FILESYSTEMSTATUSA,USB1,1,65.5,FINESTEERING,2210,166466.301,02000020,143c,
16809;USBSTICK,MOUNTED,30984440,""*30c0e28c
#FILESYSTEMSTATUSA,USB1,0,65.5,UNKNOWN,0,15.440,02000020,143c,16809;
INTERNAL_FLASH,MOUNTED,15754462,""*2e6f66ef
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	FILESYSTEMSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	MassStorageDevice	Recording Device See <i>Table 105: Mass Storage Device</i> on page 561	Enum	4	H

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
3	MassStorageStatus	Media Status See <i>Table 107: Mass Storage Status</i> below	Enum	4	H+4
4	TotalCapacity	Media total capacity (in kB)	Ulong	4	H+8
5	ErrorMsg	Error Message	String	Variable	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	Variable
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

**Table 107: Mass Storage Status**

Binary	ASCII	Description
0	UNMOUNTED	Mass storage unmounted
1	MOUNTED	Mass storage mounted
2	BUSY	Mass storage busy. i.e. formatting
3	ERROR	Mounting error
4	MOUNTING	Mass storage is being mounted
5	UNMOUNTING	Mass storage is being unmounted

## 3.37 FILETRANSFERSTATUS

*Display the current state of a file transfer*

Platform: PwrPak7



PwrPak7M variants do not support this log.

This log contains the current state of a file transfer from the internal memory to a USB stick. This logs also indicates any file transfer errors that may have occurred.

**Message ID: 2101**

**Log Type: Asynch**

**Recommended Input:**

```
log filetransferstatusa onchanged
```

**ASCII Example:**

```
#FILETRANSFERSTATUSA,USB1,0,27.0,FINESTEERING,2209,503218.978,02000020,ce81,
16809;TRANSFERRING,0,27444,"NMNE17200009B_2020-04-09_22-14-34.LOG",""*4fa2a7a3

#FILETRANSFERSTATUSA,USB1,0,55.5,FINESTEERING,2209,503236.640,02000020,ce81,
16809;FINISHED,27444,27444,"NMNE17200009B_2022-05-05_23-03-19.LOG",""*2d48645b
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	FILETRANSFERSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	File Transfer Status	The status of the file transfer. See <i>Table 108: File Transfer Status</i> on the next page.	Enum	4	H
3	Total Transferred	Total amount of data transferred.(kbytes)	Ulong	4	H+4
4	Total Transfer Size	Total size of the data to transfer.(kbytes)	Ulong	4	H+8
5	Filename	Name of the file that is currently transferring.	String	Variable	H+12
6	Error Msg	Error message (if an error occurred)	String	Variable	Variable
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	Variable
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 108: File Transfer Status**

<b>Binary Value</b>	<b>ASCII Value</b>	<b>Description</b>
1	NONE	There is no file transfer in progress
2	TRANSFERRING	There is an active file transfer
3	FINISHED	The transfer has been successfully completed
4	ERROR	An error occurred during the transfer
5	CANCELLED	A user cancelled the active file transfer

## 3.38 GALALMANAC

### *Decoded Galileo Almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the decoded Galileo almanac parameters from Galileo navigation messages. Multiple messages are transmitted, one for each satellite ID with data. The Galileo almanac can come from any of the E1, E5a or E5b signals.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

**Message ID:** 1120

**Log Type:** Asynch

**Recommended Input:**

```
log galalmanaca onchanged
```

**ASCII Example:**

```
#GALALMANACA, USB1, 22, 65.5, SATTIME, 2209, 503250.000, 02000020, 131f, 16809;
2, TRUE, TRUE, 0, 0, 0, 0, 5, 1185, 502200.000, 3.052e-04, -5.852e-09, 6.4734e-01,
3.4994e-01, 2.017e+00, 2.9945e-04, 3.638e-12, 1.367e-02, -3.260e-03*d4dcde5e

#GALALMANACA, USB1, 21, 65.5, SATTIME, 2209, 503250.000, 02000020, 131f, 16809;
3, TRUE, TRUE, 0, 0, 0, 0, 5, 1185, 502200.000, 3.052e-04, -5.852e-09, -1.452e+00,
-4.6096e-01, -1.392e+00, -5.5885e-04, -3.638e-12, 2.344e-02, -2.148e-02*8b956b99

...

#GALALMANACA, USB1, 1, 65.5, SATTIME, 2209, 502900.000, 02000020, 131f, 16809;
34, TRUE, TRUE, 0, 0, 0, 0, 4, 1185, 501600.000, 2.899e-04, -5.120e-09, 2.728e+00,
1.629e+00, -1.433e+00, -5.1689e-04, -3.638e-12, 2.344e-02, 2.052e-02*6235e5c8

#GALALMANACA, USB1, 0, 65.5, SATTIME, 2209, 502950.000, 02000020, 131f, 16809;
36, TRUE, TRUE, 0, 0, 0, 0, 4, 1185, 501600.000, 2.594e-04, -5.120e-09, 2.730e+00,
2.2166e-01, 7.6191e-01, -3.7384e-04, -7.276e-12, 2.539e-02, 2.052e-02*7aa7acd3
```



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	SatId	Satellite ID	Ulong	4	H
3	FNAVReceived	Indicates F/NAV almanac data received	Bool	4	H+4
4	INAVReceived	Indicates I/NAV almanac data received	Bool	4	H+8
5	E1BHealth	E1B health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+12
6	E5aHealth	E5a health status bits (only valid if FNAVReceived is TRUE)	Uchar	1	H+13
7	E5bHealth	E5b health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+14
8	Reserved		Uchar	1	H+15
9	IODa	Almanac issue of data	Ulong	4	H+16
10	Weeks	Almanac reference week since the Galileo system time start epoch (August 22nd 1999)	Ulong	4	H+20
11	Seconds	Almanac reference time of week (s for ASCII, ms for binary)	GPSec	4	H+24
12	Ecc	Eccentricity (dimensionless)	Double	8	H+28
13	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+36
14	Omega0	Right ascension (radians)	Double	8	H+44
15	Omega	Argument of perigee (radians)	Double	8	H+52
16	M0	Mean anomaly at ref time (radians)	Double	8	H+60
17	Af0	Satellite clock correction bias (s)	Double	8	H+68
18	Af1	Satellite clock correction linear (s/s)	Double	8	H+76
19	DeltaRootA	Difference with respect to the square root of the nominal semi-major axis ( $\sqrt{m}$ ) <sup>1</sup>	Double	8	H+84
20	DeltaI	Difference with respect to the nominal inclination at reference time (radians) <sup>2</sup>	Double	8	H+92
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+100
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>Nominal value is 29,600,000 m.

<sup>2</sup>Nominal value is 56 degrees.

## 3.39 GALCLOCK

### *Galileo clock information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the Galileo time information.



This log is populated from both the I/NAV and F/NAV messages from any of the E1, E5a or E5b signals. Depending on the data source, it is possible that the time in the header of the log is earlier than the time in a previous log. This is expected behavior.

**Message ID:** 1121

**Log Type:** Asynch

**Recommended Input:**

```
log galclocka onchanged
```

**ASCII Example:**

```
#GALCLOCKA,USB1,0,66.5,SATTIME,2209,503614.000,02000020,c6cf,16809;
9.313225746e-10,0.0000000,18,432000,161,137,7,18,2.5902e-09,1.199e-14,518400,33
*802b6b93
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALCLOCK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	A0	Constant term of polynomial (s)	Double	8	H
3	A1	1st order term of polynomial (s/s)	Double	8	H+8
4	DeltaTIs	Leap second count before leap second adjustment	Long	4	H+16
5	Tot	UTC data reference time of week (s)	Ulong	4	H+20
6	WNt	UTC data reference week number (0 – 255)	Ulong	4	H+24
7	WNIsf	Week number of leap second adjustment (0 – 255)	Ulong	4	H+28
8	DN	Day number at the end of which a leap second adjustment becomes effective	Ulong	4	H+32
9	DeltaTIsf	Leap second count after leap second adjustment	Long	4	H+36
10	A0g	Constant term of the polynomial describing the difference between Galileo and GPS time (s)	Double	8	H+40

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	A1g	Rate of change of offset the offset between Galileo and GPS time (s/s)	Double	8	H+48
12	T0g	Reference time for GGTO data (s)	Ulong	4	H+56
13	WN0g	Week number of GGTO reference (0 – 63)	Ulong	4	H+60
14	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+64
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.40 GALCNAVRAWPAGE

### *Galileo raw CNAV page*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides Galileo raw C/NAV page data from Galileo E6 signals.



The **GALCNAVRAWPAGE** log is not output by default. To receive this log, data decoding for E6B/E6C must be enabled using the **DATADECODESIGNAL** command (see page 129) the specific signal.

**Message ID:** 2239

**Log Type:** Asynch

**Recommended Input:**

```
log galcnavrpagea onnew
```

**ASCII Example:**

```
#GALCNAVRAWPAGEA,COM1,0,61.0,SATTIME,2227,252250.000,02000120,ec4c,32768;
214,30,7,185,fffc13c6e6243942790079d6c8a83e7c1ee5dbcefe2b05e637d7b161a643559c44
a7e148041d9a4511720d7ead579003d92588c7ec76880035e6*cf1b4560
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALCNAVRAWPAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	signal channel	Signal channel providing the bits	Ulong	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	Message ID	ID of the message	Ushort	2	H+8
5	Page ID	ID of the transmitted page of the encoded message	Ushort	2	H+10
6	data	Raw C/NAV page data (462 bits). Does not include CRC or Tail bits.	HEX[58]	58	H+12
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	h+70
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.41 GALFNAVEPHEMERIS

### *Decoded Galileo FNAV Ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALFNAVEPHEMERIS log contains the Galileo F/NAV ephemeris information. Multiple messages are transmitted, one for each satellite ID. The F/NAV messages are only transmitted on E5a.

**Message ID:** 1310

**Log Type:** Asynch

**Recommended Input:**

```
log galfnavephemerisa onchanged
```

**ASCII Example:**

```
#GALFNAVEPHEMERISA, USB1, 21, 35.5, SATTIME, 2209, 504150.000, 02000020, 02cd, 16809;
2, 0, 0, 0, 0, 68, 107, 0, 501600, 501600, 1.94195529e+00, 3.0676e-09, 3.055053530e-04,
5.44060316e+03, 9.740321573e-01, 5.6931e-10, 6.473214246e-01, 3.503504077e-01,
-5.8141708e-09, 7.6238e-06, 4.7404e-06, 2.409e+02, 1.610e+02, -5.0291e-08, 2.9802e-08
, 2.989536151e-04, 2.359002e-12, 0.0, -6.985e-10*e65b02c5

#GALFNAVEPHEMERISA, USB1, 20, 35.5, SATTIME, 2209, 492700.000, 02000020, 02cd, 16809;
3, 0, 0, 0, 0, 52, 107, 0, 492000, 492000, -2.64850798e+00, 3.4862e-09, 3.018563148e-04,
5.44061768e+03, 9.558740377e-01, -4.0323e-10, -1.45198786e+00, -4.689335290e-01,
-5.9263183e-09, -3.3025e-06, 5.0087e-06, 2.308e+02, -7.150e+01, 1.8626e-09,
3.5390e-08, -5.578687415e-04, -3.907985e-12, 0.0, 3.260e-09*b13e7513

...

#GALFNAVEPHEMERISA, USB1, 1, 35.5, SATTIME, 2209, 504150.000, 02000020, 02cd, 16809;
34, 0, 0, 0, 0, 69, 107, 0, 502200, 502200, -1.35918661e+00, 2.4115e-09, 2.855090424e-04,
5.44061144e+03, 9.978304990e-01, -5.8931e-11, 2.72810277e+00, 1.63000994e+00,
-5.1241420e-09, -4.7926e-06, 1.3456e-05, 6.991e+01, -1.037e+02, -3.9116e-08,
-3.5390e-08, -5.168793723e-04, -1.961098e-12, 0.0, 4.657e-09*3e6bf310

#GALFNAVEPHEMERISA, USB1, 0, 35.5, SATTIME, 2209, 504150.000, 02000020, 02cd, 16809;
36, 0, 0, 0, 0, 70, 107, 0, 502800, 502800, 9.062661404e-01, 2.4169e-09, 2.578824060e-04,
5.44061532e+03, 9.979634387e-01, -6.7860e-12, 2.72986363e+00, 2.260476454e-01,
-5.1745013e-09, -4.6510e-06, 1.4180e-05, 5.316e+01, -1.043e+02, 8.0094e-08,
3.1665e-08, -3.731751349e-04, -5.456968e-12, 0.0, 6.286e-09*cfcd6ad68
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALFNAVEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	SatId	Satellite identifier	Ulong	4	H

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
3	E5aHealth	E5a health status bits	Uchar	1	H+4
4	E5aDVS	E5a data validity status	Uchar	1	H+5
5	Reserved		Uchar	1	H+6
6	Reserved		Uchar	1	H+7
7	IODnav	Issue of data ephemeris	Ushort	2	H+8
8	SISA Index	Signal in space accuracy (unitless)	Uchar	1	H+10
9	Reserved		Uchar	1	H+11
10	T0e	Ephemeris reference time (s)	Ulong	4	H+12
11	T0c	Clock correction data reference time of week from the F/NAV message (s)	Ulong	4	H+16
12	M0	Mean anomaly at ref time (radians)	Double	8	H+20
13	DeltaN	Mean motion difference (radians/s)	Double	8	H+28
14	Ecc	Eccentricity (unitless)	Double	8	H+36
15	RootA	Square root of semi-major axis	Double	8	H+44
16	I0	Inclination angle at ref time (radians)	Double	8	H+52
17	IDot	Rate of inclination angle (radians/s)	Double	8	H+60
18	Omega0	Longitude of ascending node of orbital plane at weekly epoch (radians)	Double	8	H+68
19	Omega	Argument of perigee (radians)	Double	8	H+76
20	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+84
21	Cuc	Amplitude of the cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+92
22	Cus	Amplitude of the sine harmonic correction term to the argument of latitude (radians)	Double	8	H+100
23	Crc	Amplitude of the cosine harmonic correction term to the orbit radius (m)	Double	8	H+108
24	Crs	Amplitude of the sine harmonic correction term to the orbit radius (m)	Double	8	H+116
25	Cic	Amplitude of the cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+124
26	Cis	Amplitude of the sine harmonic correction term to the angle of inclination (radians)	Double	8	H+132

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
27	Af0	SV clock bias correction coefficient from the F/NAV message (s)	Double	8	H+140
28	Af1	SV clock drift correction coefficient from the F/NAV message (s/s)	Double	8	H+148
29	Af2	SV clock drift rate correction coefficient from the F/NAV message (s/s <sup>2</sup> )	Double	8	H+156
30	E1E5aBGD	E1, E5a broadcast group delay	Double	8	H+164
31	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+172
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.42 GALFNAVRAWALMANAC

### *Galileo FNAV raw almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALFNAVRAWALMANAC log contains the Galileo F/NAV raw almanac information. This log contains the raw bits from page types 5 and 6. The F/NAV messages are only transmitted on E5a.

**Message ID:** 1123

**Log Type:** Asynch

**Recommended Input:**

```
log galfnavravalmanaca onchanged
```

**ASCII Example:**

```
#GALFNAVRAWALMANACA,USB1,22,54.5,SATTIME,2209,503250.000,02000020,5a64,16809;5,
1,502200,0800e050390bf78d307f0522b009d00080000000*7f669413

#GALFNAVRAWALMANACA,USB1,21,54.5,SATTIME,2209,503250.000,02000020,5a64,16809;5,
1,502200,0c018053b4e3c8626bff0c749fedbfff80000000*68aedae2

...

#GALFNAVRAWALMANACA,USB1,1,54.5,SATTIME,2209,503700.000,02000020,5a64,16809;6,1
,502800,8801804d09cc35b793ff2cb9efef1fff80000000*cceb28d7

#GALFNAVRAWALMANACA,USB1,0,54.5,SATTIME,2209,503750.000,02000020,5a64,16809;6,1
,502800,9001c04424d835b79d7f224edff3cfff00000000*84d3670a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALFNAVRAWALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	IODa	Almanac issue of data	Ulong	4	H
3	WNa	Almanac reference Week Number	Ulong	4	H+4
4	T0a	Almanac reference time	Ulong	4	H+8
5	Raw Data	F/NAV raw almanac data	Hex[20]	20	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+32
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–



### 3.43 GALFNAVRAWEPHEMERIS

#### *Galileo FNAV raw ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALFNAVEPHEMERIS log contains the Galileo F/NAV raw ephemeris information. This log contains raw ephemeris data from page type 1 to page type 4. The F/NAV messages are only transmitted on E5a.

**Message ID:** 1124

**Log Type:** Asynch

**Recommended Input:**

```
log galfnavrawarephemerisa onchanged
```

**ASCII Example:**

```
#GALFNAVRAWEPHEMERISA,USB1,21,31.0,SATTIME,2209,504350.000,02000020,4c19,16809;
2,2209,504350.080,4,0421120a8009cbce0000a601ace083204e41fe8942f643caaaaaa9,0,08
444f1f5411ffc06900280b0faa04d3431a5fceb018e9285ec8a0,0,0c4427af86200e46498d218d
0ffd09f11e1e141f82a1285ec8c955,0,1044ffe50010000000010000001278a189e252000b2037
0bd91e2b,0*29c25908
```

```
#GALFNAVRAWEPHEMERISA,USB1,20,31.0,SATTIME,2209,492700.000,02000020,4c19,16809;
3,2209,492700.093,4,0430d2008fedb8421ffeed01ace083204e40070942f0938aaaaaaa,0,08
3494170f49ffbf2f0027909eaa04f100c4d7379fee5d285e1298,0,0c3426f220bcece4d88b2621
f9130a811cdaf7108021285e12c155,0,103400010013000000010000001278a189e252000b2037
0bc25d28,0*582c6a94
```

```
#GALFNAVRAWEPHEMERISA,USB1,19,31.0,SATTIME,2209,474100.000,02000020,4c19,16809;
4,2209,474100.093,4,044055ed2fde5928dffdb801ace083204e41f98942e77e8aaaaaab,0,08
15bd2316e5ffbf80002e7ae8aa04f353c4d94c9cf26d285ceffb,0,0c1526f2a4fdc6630b882658
f8610b291c45f62c7b49285cf02155,0,1015ffe80007000000010000001278a189e24f0004e027
0b9e092a,0*e2d0c13d
```

...

```
#GALFNAVRAWEPHEMERISA,USB1,2,31.0,SATTIME,2209,489700.000,02000020,4c19,16809;3
3,2209,489700.095,4,0610bdfd6ff1092960001c01ace083204e41fa8942ef1c8aaaaaa9,0,08
2f44c396a3ffc8190021253caa04fd856f39e443fe01285de3b9,0,0c2f28a90fba03eb32641844
f6001fd60415f2f27f59285de3e154,0,102ffff20019000000010000001278a189e252000b2037
0bbc8129,0*8294e367
```

```
#GALFNAVRAWEPHEMERISA,USB1,1,31.0,SATTIME,2209,504350.000,02000020,4c19,16809;3
4,2209,504350.081,4,0621160b2fef101a1fff7601ace083204e400a0942f643caaaaaa8,0,08
45c89f2b5dfffc7f500256c18aa04e43b6f2724e3fd6d285ec8a3,0,0c4528a7bfe842699ecf1a60
f5f31c3808bdf30982c9285ec8c954,0,1045ffebffed000000010000001278a189e252000b2037
0bd91e2a,0*d478a330
```

```
#GALFNAVRAWEPHEMERISA,USB1,0,31.0,SATTIME,2209,504350.000,02000020,4c19,16809;3
6,2209,504350.079,4,06411a0bcff3c5949ffe8001ace083204e400d8942f643caaaaaa8,0,08
4624ecb2e0ffc7680021cd19aa04ec2f6f3982b0ffb5285ec8a0,0,0c4628a922e10935c3311a6f
f63f1dbd06a5f2f882f1285ec8c954,0,1046002b0011000000010000001278a189e252000b2037
0bd91e2b,0*3dd4e753
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALFNAVRAWEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	satID	Satellite ID	Ulong	4	H
3	weeks	GPS reference week, in weeks	Ulong	4	H+4
4	time	GPS reference time (ms)	Ulong	4	H+8
5	#recs	Number of records to follow	Ulong	4	H+12
6	Raw Data	F/NAV raw ephemeris data	Hex[27]	27	H+16
7	Reserved		UChar	1	H+43
8	Next record offset = H + 12 + (#recs x 28)				
9	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#recs x 28)
10	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.44 GALFNAVRAWPAGE

### *Raw Galileo FNAV page data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw Galileo F/NAV page data. The F/NAV messages are only transmitted on E5a.

**Log Type:** Asynch

**Recommended Input:**

```
log galfnavrpagea onchanged
```

**ASCII Example:**

```
#GALFNAVRAWPAGEA,USB1,0,53.5,SATTIME,2209,504490.000,02000020,d4fb,16809;323,11,19eaaaaaaaaaaaaaaaaab780240648d73f08d36ff0234efef6ffc0a*71133086
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALFNAVRAWPAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	signal channel	Signal channel providing the data	Ulong	4	H
3	SVID	SVID of transmitting satellite	Ulong	4	H+4
4	raw frame data	Raw F/NAV page (214 bits). Does not include CRC or Tail bits.	Hex[27]	27	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+35
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.45 GALINAVEPHEMERIS

### *Decoded Galileo INAV Ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALINAVEPHEMERIS log contains the Galileo I/NAV ephemeris information. Multiple messages are transmitted, one for each SVID with date. The I/NAV messages are only transmitted on E1 and E5b.

**Message ID:** 1309

**Log Type:** Asynch

**Recommended Input:**

```
log galinavephemerisa onchanged
```

**ASCII Example:**

```
#GALINAVEPHEMERISA, USB1, 21, 54.0, SATTIME, 2209, 504514.000, 02000020, dbe9, 16809;
2, 0, 0, 0, 0, 0, 0, 0, 0, 68, 107, 3, 501600, 501600, 1.94195529e+00, 3.0676e-09,
3.055053530e-04, 5.44060316e+03, 9.740321573e-01, 5.6931e-10, 6.473214246e-01,
3.503504077e-01, -5.8141708e-09, 7.6238e-06, 4.7404e-06, 2.409e+02, 1.610e+02,
-5.0291e-08, 2.9802e-08, 2.989528002e-04, 2.359002e-12, 0.0, -6.985e-10, -1.630e-09
*96bd12ad
```

```
#GALINAVEPHEMERISA, USB1, 20, 54.0, SATTIME, 2209, 492664.000, 02000020, dbe9, 16809;
3, 0, 0, 0, 0, 0, 0, 0, 0, 52, 107, 3, 492000, 492000, -2.64850798e+00, 3.4862e-09,
3.018563148e-04, 5.44061768e+03, 9.558740377e-01, -4.0323e-10, -1.45198786e+00,
-4.689335290e-01, -5.9263183e-09, -3.3025e-06, 5.0087e-06, 2.308e+02, -7.150e+01,
1.8626e-09, 3.5390e-08, -5.578683922e-04, -3.907985e-12, 0.0, 3.260e-09, 3.725e-09
*b4c17a47
```

...

```
#GALINAVEPHEMERISA, USB1, 1, 54.0, SATTIME, 2209, 504514.000, 02000020, dbe9, 16809;
34, 0, 0, 0, 0, 0, 0, 0, 0, 69, 107, 3, 502200, 502200, -1.35918661e+00, 2.4115e-09,
2.855090424e-04, 5.44061144e+03, 9.978304990e-01, -5.8931e-11, 2.72810277e+00,
1.63000994e+00, -5.1241420e-09, -4.7926e-06, 1.3456e-05, 6.991e+01, -1.037e+02,
-3.9116e-08, -3.5390e-08, -5.168783828e-04, -1.961098e-12, 0.0, 4.657e-09, 5.588e-09
*3526c71f
```

```
#GALINAVEPHEMERISA, USB1, 0, 54.0, SATTIME, 2209, 504514.000, 02000020, dbe9, 16809;
36, 0, 0, 0, 0, 0, 0, 0, 0, 70, 107, 3, 502800, 502800, 9.062661404e-01, 2.4169e-09,
2.578824060e-04, 5.44061532e+03, 9.979634387e-01, -6.7860e-12, 2.72986363e+00,
2.260476454e-01, -5.1745013e-09, -4.6510e-06, 1.4180e-05, 5.316e+01, -1.043e+02,
8.0094e-08, 3.1665e-08, -3.731743782e-04, -5.456968e-12, 0.0, 6.286e-09, 7.218e-09
*3ad372dc
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALINAVEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	SatId	Satellite identifier	Ulong	4	H
3	E5bHealth	E5b health status bits	Uchar	1	H+4
4	E5bDVS	E5b data validity status	Uchar	1	H+5
5	Reserved		Uchar	1	H+6
6	Reserved		Uchar	1	H+7
7	E1bHealth	E1b health status bits	Uchar	1	H+8
8	E1bDVS	E1b data validity status	Uchar	1	H+9
9	Reserved		Uchar	1	H+10
10	Reserved		Uchar	1	H+11
11	IODnav	Issue of data ephemeris	Ushort	2	H+12
12	SISA Index	Signal in space accuracy (unitless)	Uchar	1	H+14
13	INAV Source	Identifies the source signal: 0 = Unknown 1 = E1b 2 = E5b 3 = E1b and E5b	Uchar	1	H+15
14	T0e	Ephemeris reference time (s)	Ulong	4	H+16
15	T0c	Clock correction data reference time of week from the I/NAV message (s)	Ulong	4	H+20
16	M0	Mean anomaly at ref time (radians)	Double	8	H+24
17	DeltaN	Mean motion difference (radians/s)	Double	8	H+32
18	Ecc	Eccentricity (unitless)	Double	8	H+40
19	RootA	Square root of semi-major axis	Double	8	H+48
20	I0	Inclination angle at ref time (radians)	Double	8	H+56
21	IDot	Rate of inclination angle (radians/s)	Double	8	H+64
22	Omega0	Longitude of ascending node of orbital plane at weekly epoch (radians)	Double	8	H+72

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
23	Omega	Argument of perigee (radians)	Double	8	H+80
24	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+88
25	Cuc	Amplitude of the cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+96
26	Cus	Amplitude of the sine harmonic correction term to the argument of latitude (radians)	Double	8	H+104
27	Crc	Amplitude of the cosine harmonic correction term to the orbit radius (m)	Double	8	H+112
28	Crs	Amplitude of the sine harmonic correction term to the orbit radius (m)	Double	8	H+120
29	Cic	Amplitude of the cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+128
30	Cis	Amplitude of the sine harmonic correction term to the angle of inclination (radians)	Double	8	H+136
31	Af0	SV clock bias correction coefficient from the I/NAV message (s)	Double	8	H+144
32	Af1	SV clock drift correction coefficient from the I/NAV message (s/s)	Double	8	H+152
33	Af2	SV clock drift rate correction coefficient from the I/NAV message (s/s <sup>2</sup> )	Double	8	H+160
34	E1E5aBGD	E1, E5a broadcast group delay	Double	8	H+168
35	E1E5bBGD	E1, E5b broadcast group delay	Double	8	H+176
36	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+184
37	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.46 GALINAVRAWALMANAC

### Galileo INAV raw almanac

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALINAVRAWALMANAC log contains the Galileo I/NAV raw almanac information. This log contains the I/NAV raw data from word types of 7 to 10. The I/NAV messages are only transmitted on E1 and E5b.

**Message ID:** 1125

**Log Type:** Asynch

**Recommended Input:**

```
log galinavrawalmanaca onchanged
```

**ASCII Example:**

```
#GALINAVRAWALMANACA,USB1,22,53.0,SATTIME,2209,504399.000,02000020,53f2,16809;6,
1,502800,0800e05038fff78d307f05537009d00080000000*89e6eee3

#GALINAVRAWALMANACA,USB1,21,53.0,SATTIME,2209,504401.000,02000020,53f2,16809;6,
1,502800,0c018053b4efc8626bff0ca4ffedbf80000000*069e52d2

#GALINAVRAWALMANACA,USB1,20,53.0,SATTIME,2209,504431.000,02000020,53f2,16809;6,
1,502800,1001a05f199fc8626bff051a1fde5fff00000000*7b988d8c

...

#GALINAVRAWALMANACA,USB1,2,53.0,SATTIME,2209,504640.000,02000020,53f2,16809;7,1
,503400,840240440e4035b79cff28ddbff1100000000000*57fdc359

#GALINAVRAWALMANACA,USB1,1,53.0,SATTIME,2209,504311.000,02000020,53f2,16809;6,1
,502800,8801804d09cc35b793ff2cb9efef1fff80000000*bd7dfce2

#GALINAVRAWALMANACA,USB1,0,53.0,SATTIME,2209,504341.000,02000020,53f2,16809;6,1
,502800,9001c04424d835b79d7f224edff3cfff00000000*f545b33f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALINAVRAWALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	IODa	Almanac issue of data	Ulong	4	H
3	WNa	Almanac reference week number	Ulong	4	H+4
4	T0a	Almanac reference time	Ulong	4	H+8
5	Raw Data	I/NAV raw almanac data	Hex[20]	20	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+32
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.47 GALINAVRAWEPHEMERIS

### *Galileo FNAV raw ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GALINAVEPHEMERIS log contains the Galileo I/NAV raw ephemeris information. This log contains the ephemeris raw data from word types of 1 to 6. The I/NAV messages are only transmitted on E1 and E5b.

**Message ID:** 1126

**Log Type:** Asynch

**Recommended Input:**

```
log galinavrawephemerisa onchanged
```

**ASCII Example:**

```
#GALINAVRAWEPHEMERISA,USB1,21,46.5,SATTIME,2209,504780.000,02000020,bdd7,16809;
2,2209,504780.081,6,044482a13c7d504400a02c3ea8134d0f,08441a5fceb027af86200e4649
8d18e9,0c44ffc069218d0ffd09f11e1e141f6b,10440bff9400420a8009cbcc4000a601,14e083
204e41feff20250bd9e82aaaaa,180000000400000049e28627893d9e8a*7c51efc8
```

...

```
#GALINAVRAWEPHEMERISA,USB1,0,46.5,SATTIME,2209,504780.000,02000020,bdd7,16809;3
6,2209,504780.079,6,044682f093b2cb8000873466a813b0bf,08466f3982b028a922e10935c3
31ffb5,0c46ffc7681a6fff63f1dbd06a5f2f86b,10469000ac00460bcff3c5963ffe8001,14e083
204e400d83e0250bd9e82aaaaa,180000000400000049e28627893d9e8a*a5fcb10e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALINAVRAWEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	satID	Satellite ID	Ulong	4	H
3	weeks	GPS reference week, in weeks	Ulong	4	H+4
4	time	GPS reference time (ms)	Ulong	4	H+8
5	#recs	Number of records to follow	Ulong	4	H+12
6	Raw Data	I/NAV raw ephemeris data	Hex[16]	16	H+16
7	Next record offset = H + 12 + (#recs x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#recs x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	–	–	–





Value (Binary)	Signal (ASCII)	Description
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
2662	GLOL3	GLONASS L3
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10466	GALE5A	Galileo E5A
10499	GALE5B	Galileo E5B
10532	GALALTBOC	Galileo ALT-BOC
10565	GALE6C	Galileo E6C
10572	GALE6B	Galileo E6B
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2I with D1 navigation data
12804	BDSB2D2	BeiDou B2I with D2 navigation data
12877	BDSB3D1	BeiDou B3 with D1 navigation data
12880	BDSB3D2	BeiDou B3 with D2 navigation data
12979	BDSB1C	BeiDou B1C
13012	BDSB2A	BeiDou B2a
13077	BDSB2BI	BeiDou B2b (I)
14753	QZSSL1CA	QZSS L1 C/A-code
14760	QZSSL1CP	QZSS L1C P-code
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5
14891	QZSSL6P	QZSS L6P
19073	NAVICL5SPS	NavIC L5 SPS

## 3.49 GALIONO

### *Decoded Galileo ionospheric corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the decoded Galileo ionospheric corrections.

**Message ID:** 1127

**Log Type:** Asynch

**Recommended Input:**

```
log galionoa onchanged
```

**ASCII Example:**

```
#GALIONOA,USB1,0,53.5,SATTIME,2209,504850.000,02000020,d22e,16809;
1.12e+02,1.953e-01,9.5520e-03,0,0,0,0,0*38bfa05e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GALIONO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Ai0	Effective ionization level 1st order parameter (sfu)	Double	8	H
3	Ai1	Effective ionization level 2st order parameter (sfu/degree)	Double	8	H+8
4	Ai2	Effective ionization level 3st order parameter (sfu/degree <sup>2</sup> )	Double	8	H+16
5	SF1	Ionospheric disturbance flag for region 1	Uchar	1	H+24
6	SF2	Ionospheric disturbance flag for region 2	Uchar	1	H+25
7	SF3	Ionospheric disturbance flag for region 3	Uchar	1	H+26
8	SF4	Ionospheric disturbance for flag region 4	Uchar	1	H+27
9	SF5	Ionospheric disturbance for flag region 5	Uchar	1	H+28
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+29
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.50 GEODETICDATUMS

### Available datums

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log shows a list of available datums. User-defined datums can be added using the **GEODETICDATUM** command (see page 203).

The "USER" entry in the this log is the datum set by the **USERDATUM** command (see page 456) or the **USEREXPDATUM** command (see page 458).

**Message ID:** 2296

**Log Type:** Asynch

**Recommended Input:**

```
log geodeticdatums
```

**Abbreviated ASCII Example:**

```
<GEODETICDATUMS USB1 0 53.0 UNKNOWN 0 0.199 02000020 e7f2 16809
< 14
< "WGS84" 6326 EARTH_FIXED 6378137.00 298.2572235630
< "ITRF2000" 6656 EARTH_FIXED 6378137.00 298.2572221010
< "ITRF2005" 6896 EARTH_FIXED 6378137.00 298.2572221010
< "ITRF2008" 1061 EARTH_FIXED 6378137.00 298.2572221010
< "ITRF2014" 1165 EARTH_FIXED 6378137.00 298.2572221010
< "ETRF2000" 1186 PLATE_FIXED 6378137.00 298.2572221010
< "ETRF2005" 1204 PLATE_FIXED 6378137.00 298.2572221010
< "ETRF2014" 1206 PLATE_FIXED 6378137.00 298.2572221010
< "NAD83(NSRS2011)" 1116 PLATE_FIXED 6378137.00 298.2572221010
< "NAD83(CSRS)" 6140 PLATE_FIXED 6378137.00 298.2572221010
< "NAD83(PA11)" 1117 PLATE_FIXED 6378137.00 298.2572221010
< "NAD83(MA11)" 1118 PLATE_FIXED 6378137.00 298.2572221010
< "GDA94" 6283 PLATE_FIXED 6378137.00 298.2572221010
< "GDA2020" 1168 PLATE_FIXED 6378137.00 298.2572221010
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GEODETICDATUMS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# datums	Number of datums to follow	Ulong	4	H
3	name	Name of datum	Char[32]	variable	H+4
4	epsg_code	EPSG code of datum	Ulong	4	variable

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	anchor	The datum's anchor. See <i>Table 44: Datum Anchors</i> on page 204.	Enum	4	variable
6	semimajor_axis	Semi-major axis of datum's ellipsoid (m)	Double	8	variable
7	flattening	Inverse flattening of datum's ellipsoid (unitless)	Double	8	variable
8...	Next datum offset = variable				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	variable
variable	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.51 GLMLA

### *NMEA GLONASS L1 C/A Almanac data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log outputs L1 C/A almanac data for GLONASS satellites. Multiple sentences are transmitted, one for each satellite.



The following relationships enable translation between the NMEA GLONASS satellite IDs, the NovAtel GLONASS PRN IDs, and the GLONASS slot numbers:

NMEA GLONASS satellite ID	= GLONASS slot number + 64
NovAtel GLONASS PRN ID	= GLONASS slot number + 37
	= NMEA GLONASS satellite ID - 27

**Message ID: 859**

**Log Type: Asynch**

**Recommended Input:**

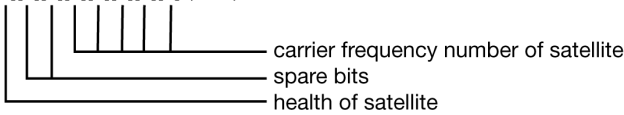
```
log glmla onchanged
```

**ASCII Example:**

```
$GLMLA,23,01,65,0864,81,01f4,1e,96d9,0000,34bf42,129a60,0c37e8,001e0d,000,202*60
$GLMLA,23,02,66,0864,9c,0890,24,d8e6,0000,34c033,019183,16aac8,00285e,000,288*6E
$GLMLA,23,03,67,0864,85,0868,24,d5c3,0000,34c034,040255,188fa3,0024a2,000,20f*33
$GLMLA,23,04,68,0864,86,04e1,21,c31c,0000,34c047,066866,1a5cae,002bbc,000,225*30
$GLMLA,23,05,69,0864,81,02e0,22,d6dc,0000,34c05d,08cdb4,1c2bda,002c12,000,218*6F
$GLMLA,23,06,70,0864,9c,0175,20,7033,0000,34c01b,0b9ef4,1e7b43,001dac,000,00d*6F
$GLMLA,23,07,71,0864,85,074e,20,d48f,0000,34c021,0e0302,0fd5ff,0024ce,000,209*35
$GLMLA,23,08,72,0864,86,092c,1e,c79b,0000,34c03b,1051a4,0e14e5,0024f8,000,011*61
$GLMLA,23,09,73,0864,9e,06bf,56,f118,0000,34bfb2,120342,18a752,001077,000,20e*39
$GLMLA,23,10,74,0864,99,0675,57,077a,0000,34c013,0077c3,04b7a2,002068,000,016*36
$GLMLA,23,11,75,0864,80,01c7,56,d960,0000,34c012,02ffd6,02ca03,002685,000,20a*35
$GLMLA,23,12,76,0864,9f,03ec,55,ae43,0000,34bfb3,059c9f,00da9f,00216d,000,205*69
$GLMLA,23,13,77,0864,9e,023b,55,3cdc,0000,34c021,07f162,10f94a,002037,000,006*3E
$GLMLA,23,14,78,0864,99,020b,55,dc08,0000,34bfba,0a9208,12ef43,001ae9,000,207*65
$GLMLA,23,15,79,0864,80,02b6,56,c127,0000,34bfbf,0d03c1,14c440,001fcd,000,219*62
$GLMLA,23,16,81,0864,84,02b6,41,fbcb,0000,34bfb5,112438,02cf59,003ad4,000,28a*64
$GLMLA,23,17,82,0864,9d,0523,41,89a7,0000,34bfcc,137a8d,011221,0040c1,000,21f*62
$GLMLA,23,18,83,0864,83,00f3,48,3129,0000,34c009,01dd22,0e9ada,004642,000,031*67
$GLMLA,23,19,84,0864,82,0430,46,8255,0000,34bfc7,04b2bb,0c7533,004691,000,012*36
$GLMLA,23,20,85,0864,84,01da,43,6b28,0000,34bf81,0773a7,0a3509,003f61,000,04f*33
$GLMLA,23,21,86,0864,9d,095f,43,0cbb,0000,34bff6,09b474,0889a2,0043a5,000,030*37
$GLMLA,23,22,87,0864,83,021f,43,30bb,0000,34c0f6,0c8f05,065ccf,004357,000,00b*37
$GLMLA,23,23,88,0864,82,019d,43,cfaf,0000,34bfd2,0eb0d3,04bb13,002bc3,000,218*36
```



Refer to the GLONASS section of *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

Field	Structure	Description	Symbol	Example
1	\$GLMLA	Log header		\$GLMLA
2	#alm	Number of NMEA almanac messages in the set	x.x	16
3	alm#	Current message number	x.x	13
4	slot	Slot number for satellite (65-96) The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.	xx	85
5	N	Calendar day count within the four year period from the last leap year	x.x	1176
6	hlth & freq	Health and frequency for satellite Health and carrier frequency numbers are represented in this 2-character Hex field as: hh = [8][7][6][5][4][3][2][1] (LSB) 	hh	88
7	ecc	Eccentricity <sup>1</sup>	hhhh	01a6
8	$\Delta T_{dot}$	Rate of change of orbital period (s/orbital period <sup>2</sup> ) <sup>1</sup>	hh	0d
9	w	Argument of perigee (PZ-90.02) (radians) <sup>1</sup>	hhhh	9dc9
10	$t_{16MSB}$	Clock offset, in seconds (s) <sup>1</sup>	hhhh	8000
11	$\Delta T$	Correction to the mean value of the Draconian period (s/orbital period) <sup>1</sup>	hhhhhh	34bff8
12	$t_{\pi}$	GLONASS Time of ascending node equator crossing (s) <sup>1</sup>	hhhhhhh	031887
13	l	Longitude of ascending node equator crossing (PZ-90.02) (radians) <sup>1</sup>	hhhhhhh	02da1e
14	$\Delta i$	Correction to nominal inclination (radians) <sup>1</sup>	hhhhhhh	002838
15	$t_{12LSB}$	Clock offset (s) <sup>1</sup>	hhh	099

<sup>1</sup>The LSB of the Hex data field corresponds to the LSB of the word indicated in the Table 4.3 of the GLONASS Interface Control Document, 1995. If the number of available bits in the Hex field is greater than the word, the MSB (upper bits) are unused and filled with zeroes.

---

Field	Structure	Description	Symbol	Example
16	t	Coarse value of the time scale shift <sup>1</sup>	hhh	242
17	xxxx	32-bit CRC (ASCII and Binary only)	Hex	*6D
18	[CR][LF]	Sentence terminator (ASCII only)	-	[CR][LF]



## 3.52 GLOALMANAC

### *Decoded GLONASS L1 C/A Almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GLONASS L1 C/A almanac reference time and week are in GPS reference time coordinates. GLONASS ephemeris information is available through the **GLOEPHEMERIS** log (see page 597).

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

**Message ID:** 718

**Log Type:** Asynch

**Recommended Input:**

```
log gloalmanaca onchanged
```

**ASCII Example:**

```
#GLOALMANACA,USB1,0,53.0,SATTIME,2209,504978.000,02000020,ba83,16809;
24,
2209,459317.000,1,1,1,0,38099.000000000,2.399074047,0.023048661,0.000476837,
-0.560765852,-2655.628906250,0.001831055,-0.000007629,
2209,424430.093,2,-4,1,0,3212.093750000,-1.309084823,0.030961245,0.002090454,
-2.181895923,-2656.099609375,0.002197266,-0.000518799,
...
2209,446938.156,23,3,1,0,25720.156250000,1.249280545,0.051649013,0.000517845,
1.196025646,-2656.480468750,-0.000183105,0.000041962,
2209,451304.593,24,2,1,0,30086.593750000,0.928882292,0.033564818,0.000393867,
-1.955729631,-2655.910156250,-0.000183105,-0.000091553*87b68608
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLOALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#recs	The number of GLONASS almanac records to follow. Set to zero until almanac data is available	Long	4	H
3	week	GPS reference week, in weeks	Ulong	4	H+4

Field	Field type	Description	Format	Binary Bytes	Binary Offset
4	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data) <b>Note:</b> The week and time are based on the GPS scale and are computed using the $N_4$ , $N^A$ , and $t_{\lambda n}^A$ values.	GPSTime	4	H+8
5	slot	Slot number for satellite, ordinal	Uchar	1	H+12
6	frequency	Frequency for satellite, ordinal (frequency channels are in the range -7 to +6)	Char	1	H+13
7	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (M type) 2 = GLO_SAT_K (K type)	Uchar	1	H+14
8	health	Satellite status where 0 = OPERATIONAL 1 = MALFUNCTION	Uchar	1	H+15
9	TlambdaN	GLONASS Time of ascending node equator crossing (s)	Double	8	H+16
10	lambdaN	Longitude of ascending node equator crossing (PZ-90.02) (radians)	Double	8	H+24
11	deltai	Correction to nominal inclination (radians) <sup>1</sup>	Double	8	H+32
12	ecc	Eccentricity	Double	8	H+40
13	ArgPerig	Argument of perigee (PZ-90.02) (radians)	Double	8	H+48
14	deltaT	Correction to the mean value of the Draconian period (s/orbital period) <sup>2</sup>	Double	8	H+56
15	deltaTD	Rate of change of orbital period (s/orbital period <sup>2</sup> )	Double	8	H+64
16	tau	Clock offset (s)	Double	8	H+72
17	Next message offset = H + 4 + (#recs x 76)				
18	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (76 x #recs)
19	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>The nominal inclination is 63 degrees.

<sup>2</sup>The nominal mean value is 43200.

## 3.53 GLOCLOCK

### *GLONASS L1 C/A clock information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the time difference information between GPS and GLONASS time as well as status flags. The status flags are used to indicate the type of time processing used in the least squares adjustment. GPS and GLONASS time are both based on the Universal Time Coordinated (UTC) time scale with some adjustments. GPS reference time is continuous and does not include any of the leap second adjustments to UTC applied since 1980. The result is that GPS reference time currently leads UTC time by 15 seconds.

GLONASS time applies leap seconds but is also three hours ahead to represent Moscow time. The nominal offset between GPS and GLONASS time is therefore due to the three hour offset minus the leap second offset. As well as the nominal offset, there is a residual offset on the order of nanoseconds which must be estimated in the least squares adjustment. The GLONASS-M satellites broadcasts this difference in the navigation message.

This log also contains information from the GLONASS navigation data relating GLONASS time to UTC.

**Message ID:** 719

**Log Type:** Asynch

**Recommended Input:**

```
log gloclocka onchanged
```

**ASCII Example:**

```
#GLOCLOCKA,USB1,0,53.0,SATTIME,2209,504978.000,02000020,1d44,16809;
0,0.000000000,0.000000000,1,7,4.749745131e-08,864,4.656612873e-10,
-0.095703125,-0.000656128,0*668057a7
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLOCLOCK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Reserved		Ulong	4	H
3			Double	8	H+4
4			Double	8	H+12
5	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (M type) 2 = GLO_SAT_K (K type)	Uchar	1	H+20

Field	Field type	Description	Format	Binary Bytes	Binary Offset
6	N <sup>4</sup>	Four-year interval number starting from 1996	Uchar	1 <sup>1</sup>	H+21
7	T <sub>GPS</sub>	Correction to GPS time relative to GLONASS time	Double	8	H+24
8	N <sup>A</sup>	GLONASS calendar day number within a four year period beginning since the leap year, in days	Ushort	2 <sup>1</sup>	H+32
9	T <sub>C</sub>	GLONASS time scale correction to UTC(SU) given at beginning of day N <sup>4</sup> (s)	Double	8	H+36
10	b1	Beta parameter 1st order term	Double	8	H+44
11	b2	Beta parameter 2nd order term	Double	8	H+52
12	Kp	Kp provides notification of the next expected leap second. For more information, see <i>Table 110: Kp UTC Leap Second Descriptions</i> below	Uchar	1	H+60
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+61
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 110: Kp UTC Leap Second Descriptions**

Kp	Information on UTC Leap Second <sup>2</sup>
00	No UTC update for this quarter
01	UTC update of plus 1 second at the end of current quarter
11	UTC update of minus 1 second at end of current quarter

<sup>1</sup>In the binary log case, additional bytes of padding are added to maintain 4-byte alignment.

<sup>2</sup>Based on GLONASS ICD version 5.1, 2008.

## 3.54 GLOEPHEMERIS

### *Decoded GLONASS L1 C/A ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains GLONASS L1 C/A ephemeris information. GLONASS ephemerides are referenced to the PZ90.02 geodetic datum. No adjustment between the GPS and GLONASS reference frames are made for positioning. Multiple messages are transmitted, one for each SVID with data.

**Message ID:** 723

**Log Type:** Asynch

**Recommended Input:**

```
log gloephemerisa onchanged
```

**Example:**

```
#GLOEPHEMERISA,USB1,11,54.0,SATTIME,2209,505128.000,02000020,8d29,16809;
51,0,1,1,2209,504918000,10782,864,0,0,93,0,-7.5577602539062500e+06,
-2.3962225585937500e+07,-4.3375678710937500e+06,1.0131835937500000e+02,
6.0211277008056641e+02,-3.4957332611083984e+03,-3.72529029846191400e-06,
-0.000000000000000000,1.86264514923095700e-06,-2.57240608334541320e-05,
5.587935448e-09,-0.000000000000000000,83910,3,2,0,28*50758dd7
```

```
#GLOEPHEMERISA,USB1,10,54.0,SATTIME,2209,498978.000,02000020,8d29,16809;
43,3,1,255,2209,499518000,10782,864,0,0,87,0,-7.5618266601562500e+06,
1.5146657226562500e+07,1.9069082519531250e+07,-2.2061262130737305e+03,
1.4006929397583008e+03,-1.9885673522949219e+03,1.86264514923095700e-06,
-0.000000000000000000,-2.79396772384643550e-06,4.82378527522087090e-05,
2.793967724e-09,-1.81898940354585640e-12,77760,3,3,0,28*a8ca7563
```

...

```
#GLOEPHEMERISA,USB1,1,54.0,SATTIME,2209,502428.000,02000020,8d29,16809;
44,12,1,255,2209,503118000,10782,864,0,0,91,0,1.8421777343750000e+05,
1.3962225585937500e+07,2.1379985351562500e+07,-2.0501766204833984e+03,
2.0764226913452148e+03,-1.3450326919555664e+03,9.31322574615478510e-07,
-0.000000000000000000,-2.79396772384643550e-06,-3.50531190633773800e-05,
1.862645149e-09,9.09494701772928230e-13,81210,3,1,0,12*30d2e4f1
```

```
#GLOEPHEMERISA,USB1,0,54.0,SATTIME,2209,505128.000,02000020,8d29,16809;
45,13,1,1,2209,504918000,10782,864,0,0,93,0,9.6398046875000000e+06,
5.4337807617187500e+06,2.3045452148437500e+07,-1.9557723999023438e+03,
2.4541549682617188e+03,2.3801517486572266e+02,9.31322574615478510e-07,
0.000000000000000000,-2.79396772384643550e-06,6.50892034173011770e-05,
-3.725290298e-09,0.000000000000000000,83910,3,2,0,28*9489ad94
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLOEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). This is also called SLOTO in Connect	Ushort	2	H
3	freqo	Frequency channel offset for satellite in the range 0 to 20	Ushort	2	H+2
4	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (M type) 2 = GLO_SAT_K (K type)	Uchar	1	H+4
5	Reserved			1	H+5
6	e week	Reference week of ephemeris (GPS reference time)	Ushort	2	H+6
7	e time	Reference time of ephemeris (GPS reference time) (ms)	Ulong	4	H+8
8	t offset	Integer seconds between GPS and GLONASS time. A positive value implies GLONASS is ahead of GPS reference time.	Ulong	4	H+12
9	Nt	Calendar number of day within 4 year interval starting at Jan 1 of a leap year	Ushort	2	H+16
10	Reserved			1	H+18
11				1	H+19
12	issue	15 minute interval number corresponding to ephemeris reference time	Ulong	4	H+20
13	health <sup>1</sup>	Ephemeris health where 0-3 = GOOD 4-15 = BAD	Ulong	4	H+24
14	pos x	X coordinate for satellite at reference time (PZ-90.02) (m)	Double	8	H+28

<sup>1</sup>The last four bits of this field are used to describe the health.

Bit 0-2: Bn

Bit 3: In

All other bits are reserved and set to 0.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
15	pos y	Y coordinate for satellite at reference time (PZ-90.02) (m)	Double	8	H+36
16	pos z	Z coordinate for satellite at reference time (PZ-90.02) (m)	Double	8	H+44
17	vel x	X coordinate for satellite velocity at reference time (PZ-90.02) (m/s)	Double	8	H+52
18	vel y	Y coordinate for satellite velocity at reference time (PZ-90.02) (m/s)	Double	8	H+60
19	vel z	Z coordinate for satellite velocity at reference time (PZ-90.02), (m/s)	Double	8	H+68
20	LS acc x	X coordinate for lunisolar acceleration at reference time (PZ-90.02), (m/s/s)	Double	8	H+76
21	LS acc y	Y coordinate for lunisolar acceleration at reference time (PZ-90.02) (m/s/s)	Double	8	H+84
22	LS acc z	Z coordinate for lunisolar acceleration at reference time (PZ-90.02) (m/s/s)	Double	8	H+92
23	tau_n	Correction to the nth satellite time t_n relative to GLONASS time t_c (s)	Double	8	H+100
24	delta_tau_n	Time difference between navigation RF signal transmitted in L2 sub-band and navigation RF signal transmitted in L1 sub-band by nth satellite (s)	Double	8	H+108
25	gamma	Frequency correction (s/s)	Double	8	H+116
26	Tk	Time of frame start (since start of GLONASS day) (s)	Ulong	4	H+124
27	P	Technological parameter	Ulong	4	H+128
28	Ft	User range	Ulong	4	H+132
29	age	Age of data (days)	Ulong	4	H+136
30	Flags	Information flags, see <i>Table 111: GLONASS Ephemeris Flags Coding</i> on the next page	Ulong	4	H+140
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+144
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 111: GLONASS Ephemeris Flags Coding

Nibble Number	Bit	Description	Range Values	Hex Value
N0	0	P1 Flag - Time interval between adjacent values of (tb) parameter in minutes	See Table 112: P1 Flag Range Values below	00000001
	1			00000002
	2	P2 Flag - Flag of oddness or evenness of the value of (tb) (for intervals of 30 or 60 minutes)	0 = even 1 = odd	00000004
	3	P3 Flag - Number of satellites with almanac information within current subframe	0 = four 1 = five	00000008
N1	4	P4 Flag - Flag to show that ephemeris parameters are present. "1" indicates that updated ephemeris or frequency/time parameters have been uploaded by the control	0 = no updated ephemeris 1 = updated ephemeris present	00000010
	5 ... 7	Reserved		
N2 through N7	8 ... 31	Reserved		

Table 112: P1 Flag Range Values

State	Description
00	0 minutes
01	30 minutes
10	45 minutes
11	60 minutes



## 3.55 GLORAWALM

### Raw GLONASS L1 C/A Almanac data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw L1 C/A almanac subframes as received from the GLONASS satellite.

**Message ID:** 720

**Log Type:** Asynch

**Recommended Input:**

```
log glorawalma onchanged
```

**Example:**

```
#GLORAWALMA,USB1,0,54.0,SATTIME,2209,505128.000,02000020,77bb,16809;
2209,505128.066,54,
056c00000000238000663a,0,06a180986fd00f0681f413,0,0796d994d30697e847824f,0,
08a2a22d5590142f089043,0,09d8e60c8c1e98066938c3,0,0aa383f11f4612510868b3,0,
0bd5c32012ae9806890ad7,0,0ca48974b95c15de04e1c8,0,0dc31c3343369808e84c87,0,
0ea5863857b4160902e010,0,0fd6dc466da6980ba882b9,0,056c00000000238000663a,0,
06a6037cf6860ed60175e7,0,0770335cf7a6980368387b,0,08a7825fabfe1267074eb7,0,
09d48f7018169804280a92,0,0aa8045c29ca127c092c13,0,0bc79b828d269807678caf,0,
0ca983b14ea4083b86bfbfd,0,0df118901a1697f655bca0,0,0eaa05896f441034067531,0,
0f077a03be1e980275f2a5,0,056c00000000238000663a,0,06ab82859406134281c7fe,0,
07d96017feb69802558034,0,08ac8141b53e10b683ec0b,0,09ae432ce4fe97f6757e56,0,
0aad01a1f294101b823be3,0,0b3cdc3f8b169804357c99,0,0cae81e5de860d74820b2c,0,
0ddc0854904697f75572fd,0,0eaf866988800fe682b6b0,0,0fc127681e0e97f7f58012,0,
056c00000000238000663a,0,0630002df3360b8e0c2628,0,070b067dcf9697ea95be30,0,
08b1a2859eb21d6a02b624,0,09fbc58921c697f6b0482d,0,0ab287c22442206085237d,0,
0b89a79bd46e97f9907a1c,0,0cb30c5d35b4232100f3ce,0,0d31290ee91698013206a7,0,
0eb40498ea662348843025,0,0f82552595de97f8f184e8,0,056c00000000238000663a,0,
06b513d46a121fb081da5d,0,076b283b9d3e97f030c88f,0,08b60c11134421d2895f81,0,
090cbb4da3a697fed0fac8,0,0ab702ccb99e21ab821f75,0,0b30bb64782e981ed0c62b,0,
0cb88609762615e1819ddc,0,0dcfaf75869e97fa50c4ac,0,0e8c51592b47c979637118,0
*21202c84
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLORAWALM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	week	GPS reference week, in weeks	Ulong	4	H
3	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+4

Field	Field type	Description	Format	Binary Bytes	Binary Offset
4	#recs	Number of records to follow	Ulong	4	H+8
5	string	GLONASS data string	String[11]	11	H+12
6	Reserved		Uchar	1	H+23
7	Next record offset = H+8+(#recs x 12)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12+ (#recsx12)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.56 GLORAWEPHEM

### Raw GLONASS L1 C/A Ephemeris data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw L1 C/A ephemeris frame data as received from the GLONASS satellite.

**Message ID:** 792

**Log Type:** Asynch

**Recommended Input:**

```
log glorawephema onchanged
```

**Example:**

```
#GLORAWEPHEMA,USB1,11,52.0,SATTIME,2209,505278.000,02000020,332d,16809;
51,0,273,2209,504918.000,4,010baa019f00a4ec2e1503,0,021ba009a24186ecd1cefe,0,
03c006b7ee8614878c8b27,0,0481af94c0000245b03989,0*67abf4ec

#GLORAWEPHEMA,USB1,10,52.0,SATTIME,2209,498978.000,02000020,332d,16809;
43,3,282,2209,499518.000,4,010ac8a34c4b14ec4e9db9,0,021ae016693d81d9554278,0,
03c0269fd12c9a53e8a941,0,0403294c60000265b0195a,0*3fa54da7

#GLORAWEPHEMA,USB1,9,52.0,SATTIME,2209,505278.000,02000020,332d,16809;
55,4,289,2209,504918.000,4,010baa17f2b8a697036183,0,021ba0072c4d9d092b7cc2,0,
03801632860c8160b6df21,0,0487a65cc0000245b049c7,0*c970c8c9

...

#GLORAWEPHEMA,USB1,2,52.0,SATTIME,2209,505278.000,02000020,332d,16809;
54,11,32,2209,504918.000,4,010baa2f8f579ccd09da20,0,021ba00859ba8d5aca447f,0,
0380361254fb92b0184451,0,04a2751800000065b04531,0*91a30fc5

#GLORAWEPHEMA,USB1,1,52.0,SATTIME,2209,502428.000,02000020,332d,16809;
44,12,297,2209,503118.000,4,010b43a0cd860805c1be0b,0,021b6021390781b451ced7,0,
0380169585419a9c1fe299,0,04824c1840000025b01dad,0*79a48a72

#GLORAWEPHEMA,USB1,0,52.0,SATTIME,2209,505278.000,02000020,332d,16809;
45,13,40,2209,504918.000,4,010baa9f4ad8092d3e708f,0,021ba027443800a9ce3fea,0,
03800603cee99ad02b9e73,0,0404440680000245b021bb,0*ba344f1d
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLORAWEPHEM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). Ephemeris relates to this slot and is also called SLOTO in NovAtel Connect	Ushort	2	H

Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	freqo	Frequency channel offset in the range 0 to 20	Ushort	2	H+2
4	sigchan	Signal channel number	Ulong	4	H+4
5	week	GPS reference week, in weeks	Ulong	4	H+8
6	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+12
7	#recs	Number of records to follow	Ulong	4	H+16
8	string	GLONASS data string	String[11]	11	H+20
9	Reserved		Uchar	1	H+31
10	Next record offset = H+20+(#recs x 12)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+20+ (#recsx12)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.57 GLORAWFRAME

### Raw GLONASS L1 C/A frame data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw GLONASS L1 C/A frame data as received from the GLONASS satellite. Multiple messages are transmitted, one for each SVID with data.

**Message ID:** 721

**Log Type:** Asynch

**Recommended Input:**

```
log glorawframea onchanged
```

**Example:**

```
#GLORAWFRAMEA,USB1,0,54.0,SATTIME,2209,505308.000,02000020,8792,16809;4,38,8,22
09,505308.075,305,305,15,010bab96ab4e821602d6a7,0,021ba01b6b7e1530c4a7e3,0,0380
0628618591fb3adfd1,0,0480917120000045b0051b,0,056c00000000238000683e,0,0630002d
f3360b8e0c2628,0,070b067dcf9697ea95be30,0,08b1a2459eb21d6a02b627,0,09fbc58921c6
97f6b0482d,0,0ab2878224422060852340,0,0b89a79bd46e97f9907a1c,0,0cb30c5d35b42321
00f3ce,0,0d31290ee91698013206a7,0,0eb40498ea662348843025,0,0f82552595de97f8f184
e8,0*08409ad3
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLORAWFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	frame#	Frame number	Ulong	4	H
3	sloto	Slot information offset - PRN identification (Slot + 37). Ephemeris relates to this slot and is also called SLOTO in NovAtel Connect.	Ushort	2	H+4
4	freqo	Frequency channel offset in the range 0 to 20	Ushort	2	H+6
5	week	GPS Week, in weeks	Ulong	4	H+8
6	time	GPS Time, in milliseconds (binary data) or seconds (ASCII data)	GPSTime	4	H+12
7	frame decode	Frame decoder number	Ulong	4	H+16
8	sigchan	Signal channel number	Ulong	4	H+20
9	#recs	Number of records to follow	Ulong	4	H+24

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	string	GLONASS data string	String[11]	11	H+28
11	Reserved		Uchar	1	H+39
12	Next record offset = H+28+ (#recs x 12)				
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H +28+ (#recs x 12)
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.58 GLORAWSTRING

### Raw GLONASS L1 C/A string

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw L1 C/A string data as received from the GLONASS satellite.

**Message ID:** 722

**Log Type:** Asynch

**Recommended Input:**

```
log glorawstringa onchanged
```

**Example:**

```
#GLORAWSTRINGA, USB1, 0, 54.0, SATTIME, 2209, 505410.000, 02000020, 50ac, 16809;
1, 1, 0770335cf7a6980368387b, 0*520359be
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GLORAWSTRING header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	slot	Slot identification	Uchar	1	H
3	freq	Frequency channel (frequency channels are in the range -7 to +13)	Char	1	H+1
4	string	GLONASS data string	Hex[11]	11	H+2
5	Reserved		Uchar	1	H+13
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+14
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.59 GPALM

### *Almanac data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log outputs raw almanac data for each GPS satellite PRN contained in the broadcast message. A separate record is logged for each PRN, up to a maximum of 32 records. GPALM outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. UTC time is then set to VALID. It takes a minimum of 12.5 minutes to collect a complete almanac following receiver boot-up. In the case of a GLONASS capable receiver, the UTC offset can be determined once the GLONASS ephemeris is decoded, which takes about 50 seconds. If an almanac was stored in NVM, the stored values are reported in the GPALM log once time is set on the receiver.



To obtain copies of ICD-GPS-200, refer to <https://www.gps.gov/technical/icwg/>.

**Message ID:** 217

**Log Type:** Asynch

**Recommended Input:**

```
log gpalm onchanged
```

**Example:**

```
$GPALM,30,01,01,2210,00,617b,0f,1da7,fd70,a10d0a,24de91,6fe696,16263f,17c,ffe*7B
$GPALM,30,02,02,2210,00,a752,0f,0f5f,fd56,a10cf7,c62dda,6c306a,1c751c,d53,000*7F
$GPALM,30,03,03,2210,00,21ef,0f,1464,fd63,a10c9d,26d19c,9a02d6,e8f3d8,eff,ffc*20
$GPALM,30,04,04,2210,00,0ed5,0f,0bfc,fd40,a10c67,871978,c614e7,5cff7f,f51,001*26
$GPALM,30,05,05,2210,00,30d6,0f,0bf0,fd56,a10d59,2996db,98511a,8266e6,faa,000*2B
$GPALM,30,06,06,2210,00,150b,0f,1d37,fd6d,a10d81,dd08a4,6f9046,1f9de5,13f,004*79
$GPALM,30,07,07,2210,00,83b7,0f,059e,fd5c,a10cf2,a38e0f,ef9113,faf04e,152,000*7D
$GPALM,30,08,08,2210,00,3d03,0f,0da3,fd22,a10da6,06d382,43d119,6f02ae,fb9,000*76
$GPALM,30,09,09,2210,00,1113,0f,0770,fd39,a10bfc,4c6129,c3e4b9,844e81,e9f,001*7C
$GPALM,30,10,10,2210,00,3e48,0f,1443,fd65,a10d0e,9a5e76,99e60d,c36848,e4c,ffc*2A
...
$GPALM,30,21,22,2210,00,6d60,0f,0c0f,fd41,a10cff,b484ba,c80d8d,69e9f1,110,002*75
$GPALM,30,22,23,2210,00,118d,0f,1111,fd62,a10d46,7a73b2,98de21,f78a52,fe9,fff*20
$GPALM,30,23,24,2210,00,662b,0f,fad0,fd4c,a10d14,212fb6,ec22c7,22a913,0fc,ffe*24
$GPALM,30,24,25,2210,00,5625,0f,0983,fd44,a10d05,28daa2,19bb67,d5b628,155,001*71
$GPALM,30,25,26,2210,00,38d6,0f,fd0b,fd36,a10d99,0fd6f2,17bfe3,ba7af2,0d8,001*7E
$GPALM,30,26,27,2210,00,5671,0f,1355,fd2a,a10cd2,1bbd7f,44a393,70647e,0d5,002*75
$GPALM,30,27,29,2210,00,119e,0f,18e8,fd35,a10d41,5beea1,47bfe9,7518fe,de5,fff*2A
$GPALM,30,28,30,2210,00,2fa3,0f,fbf2,fd4d,a10dc1,923e8f,f00c20,fa7e58,dd6,000*2E
$GPALM,30,29,31,2210,00,5578,0f,085d,fd5d,a10d86,0fe442,f05163,e7e443,f46,000*2F
$GPALM,30,30,32,2210,00,30d7,0f,09ed,fd3d,a10c3f,a2d321,c45da6,8b29e1,f7a,ffe*70
```





See the *The NMEA (National Marine Electronics Association) has defined standards that specify how electronic equipment for marine users communicate. GNSS receivers are part of this standard and the NMEA has defined the format for several GNSS data logs otherwise known as 'sentences'*, on page 611 that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPALM	Log header		\$GPALM
2	# msg	Total number of messages logged. Set to zero until almanac data is available	x.x	17
3	msg #	Current message number <sup>1</sup>	x.x	17
4	PRN	Satellite PRN number: GPS = 1 to 32	xx	28
5	GPS wk	GPS reference week number	x.x	653
6	SV hlth	SV health, bits 17-24 of each almanac page <sup>2</sup>	hh	00
7	ecc	e, eccentricity <sup>3</sup>	hhhh	3EAF
8	alm ref time	to a almanac reference time <sup>3</sup>	hh	87
9	incl angle	(sigma)i, inclination angle <sup>3</sup>	hhhh	OD68
10	omegadot	OMEGADOT, rate of right ascension <sup>3</sup>	hhhh	FD30
11	rt axis	(A) <sup>1/2</sup> , root of semi-major axis <sup>3</sup>	hhhhhh	A10CAB
12	omega	omega, argument of perigee <sup>3</sup>	hhhhhh	6EE732
13	long asc node	(OMEGA)°, longitude of ascension node <sup>3</sup>	hhhhhh	525880
14	Mo	Mo, mean anomaly <sup>3</sup>	hhhhhh	6DC5A8
15	af0	af0, clock parameter <sup>3</sup>	hhh	009
16	af1	af1, clock parameter <sup>3</sup>	hhh	005
17	*xx	Check sum	*hh	*37
18	[CR][LF]	Sentence terminator		[CR][LF]

<sup>1</sup>Variable length integer, 4-digits maximum from (2) most significant binary bits of Subframe 1, Word 3 reference Table 20-I, IS-GPS-200L, and (8) least significant bits from subframe 5, page 25, word 3 reference Table 20-I.

<sup>2</sup>Reference paragraph 20.3.3.5.1.3, Table 20-VII and Table 20-VIII, IS-GPS-200L.

<sup>3</sup>Reference Table 20-VI, IS-GPS-200L for scaling factors and units.

## 3.60 GPGGA

### *GPS fix data and undulation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains time, position and fix related data of the GNSS receiver. See also *Table 115: Position Precision of NMEA Logs* on page 616.

The GPGGA log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



The GPGGA log can be customized using the **NMEAFORMAT** command (see page 273).



The position for the **GPGGA** log is reported in the user selected datum. See the **DATUM** command on page 133 for more details.



When the solution type is INS (SPAN models), the DOPs in the **GPGGA** log are calculated at a 1 Hz rate.  
To determine if the receiver is providing an INS solution, check the **pos type** field in the **BESTPOS** log (see page 520).

**Message ID: 218**

**Log Type: Synch**

**Recommended Input:**

```
log gpgga ontime 1
```

**Example:**

```
$GPGGA,202530.00,5109.0262,N,11401.8407,W,5,40,0.5,1097.36,M,-17.00,M,18,TSTR*61
```



The NMEA (National Marine Electronics Association) has defined standards that specify how electronic equipment for marine users communicate. GNSS receivers are part of this standard and the NMEA has defined the format for several GNSS data logs otherwise known as 'sentences'.

Each NMEA sentence begins with a '\$' followed by a two-letter prefix identifying the type of sending device (for example 'GP', 'GL' or 'GN'), followed by a sequence of letters that define the type of information contained in the sentence. Data contained within the sentence is separated by commas and the sentence is terminated with a two digit checksum followed by a carriage return/line feed. Here is an example of a NMEA sentence describing time, position and fix related data:

```
$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,
-16.27,M,08,AAAA*60
```

The GPGGA sentence shown above and other NMEA logs are output the same no matter what GNSS receiver is used, providing a standard way to communicate and process GNSS information. For more information about NMEA, see the **NMEATALKER** command on page 276.

Field	Structure	Description	Symbol	Example
1	\$GPGGA	Log header		\$GPGGA
2	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	202134.00
3	lat	Latitude (DDmm.mm)	IIII.II	5106.9847
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	quality	refer to <i>Table 113: GPS Quality Indicators</i> on the next page	x	1
8	# sats	Number of satellites in use. May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below mean sea level	x.x	1062.22
11	a-units	Units of antenna altitude (M = metres)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = metres)	M	M
14	age	Age of correction data (in seconds) The maximum age reported here is limited to 99 seconds.	xx	(empty when no differential data is present)

Field	Structure	Description	Symbol	Example
15	stn ID	Differential base station ID	xxxx	(empty when no differential data is present)
16	*xx	Check sum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

Table 113: GPS Quality Indicators

Indicator	Description
0	Fix not available or invalid
1	Single point
	Converging PPP (TerraStar-L)
2	Pseudorange differential
	Converged PPP (TerraStar-L)
	Converging PPP (TerraStar-C, TerraStar-C PRO, TerraStar-X)
4	RTK fixed ambiguity solution
5	RTK floating ambiguity solution
	Converged PPP (TerraStar-C, TerraStar-C PRO, TerraStar-X)
6	Dead reckoning mode
7	Manual input mode (fixed position)
8	Simulator mode
9	WAAS (SBAS) <sup>1</sup>



Refer to the **BESTPOS** log (see page 520) and *Table 91: Supplemental Position Types and NMEA Equivalents* on page 527.

<sup>1</sup>An indicator of 9 has been temporarily set for SBAS (NMEA standard for SBAS not decided yet). This indicator can be customized using the GGAQUALITY command.

## 3.61 GPGGALONG

### *Fix data, extra precision and undulation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains, time, position, undulation and fix related data of the GNSS receiver. This is output as a GPGGGA log but the GPGGALONG log differs from the normal GPGGGA log by its extra precision. See also *Table 115: Position Precision of NMEA Logs* on page 616.

The GPGGALONG log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



The GPGGALONG log can be customized using the **NMEAFORMAT** command (see page 273).



When the solution type is INS (SPAN models), the DOPs in the **GPGGALONG** log are calculated at a 1 Hz rate.  
To determine if the receiver is providing an INS solution, check the **pos type** field in the **BESTPOS** log (see page 520).

**Message ID: 521**

**Log Type: Synch**

**Recommended Input:**

```
log gpggalong ontime 1
```

**Example:**

```
$GPGGGA,202601.00,5109.0262339,N,11401.8407317,W,5,40,0.5,1097.350,M,-17.00,M,09,TSTR*5F
```



See the Note in the **GPGGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPGGALONG	Log header		\$GPGGGA
2	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	202126.00
3	lat	Latitude (DDmm.mm)	lll.ll	5106.9847029
4	lat dir	Latitude direction (N = North, S = South)	a	N

Field	Structure	Description	Symbol	Example
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986286
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	GPS qual	Refer to <i>Table 113: GPS Quality Indicators</i> on page 612	x	1
8	# sats	Number of satellites in use (00-12). May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below msl	x.x	1062.376
11	units	Units of antenna altitude (M = metres)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = metres)	M	M
14	age	Age of Differential GPS data (in seconds) The maximum age reported here is limited to 99 seconds.	xx	10 (empty when no differential data is present)
15	stn ID	Differential base station ID, 0000-1023	xxxx	AAAA (empty when no differential data is present)
16	*xx	Check sum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

Table 114: Position Precision of NMEA Logs

NMEA Log	Latitude (# of decimal places)	Longitude (# of decimal places)	Altitude (# of decimal places)
GPGGA	4	4	2
GPGGALONG	7	7	3
GPGLL	7	7	N/A
GPRMC	7	7	N/A



Refer to the **BESTPOS** log (see page 520) and *Table 91: Supplemental Position Types and NMEA Equivalents* on page 527.

## 3.62 GPGLL

### *Geographic position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains latitude and longitude of present vessel position, time of position fix and status.

*Table 115: Position Precision of NMEA Logs* on the next page compares the position precision of selected NMEA logs.

The GPGLL log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).



The position for the **GPGLL** log is reported in the user selected datum. See the **DATUM** command on page 133 for more details.

**Message ID: 219**

**Log Type: Synch**

**Recommended Input:**

```
log gppll ontime 1
```

**Example 1 (GPS only):**

```
$GPGLL,5109.0262317,N,11401.8407304,W,202725.00,A,D*79
```

**Example 2 (Multi-constellation):**

```
$GNGLL,5109.0262321,N,11401.8407167,W,174738.00,A,D*6B
```



See the Note in the **GPGLL** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Example
1	\$GPGLL	Log header	\$GPGLL
2	lat	Latitude (DDmm.mm)	5106.7198674
3	lat dir	Latitude direction (N = North, S = South)	N
4	lon	Longitude (DDDmm.mm)	11402.3587526

Field	Structure	Description	Example
5	lon dir	Longitude direction (E = East, W = West)	W
6	utc	UTC time status of position (hours/minutes/seconds/decimal seconds)	220152.50
7	data status	Data status: A = Data valid, V = Data invalid	A
8	mode ind	Positioning system mode indicator, see <i>Table 116: NMEA Positioning System Mode Indicator</i> below	A
9	*xx	Check sum	*1B
10	[CR][LF]	Sentence terminator	[CR][LF]

Table 115: Position Precision of NMEA Logs

NMEA Log	Latitude (# of decimal places)	Longitude (# of decimal places)	Altitude (# of decimal places)
GPGGA	4	4	2
GPGGALONG	7	7	3
GPGLL	7	7	N/A
GPRMC	7	7	N/A

Table 116: NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid



## 3.63 GPGRS

### *GPS range residuals for each satellite*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Range residuals can be computed in two ways, and this log reports those residuals. Under mode 0, residuals output in this log are used to update the position solution output in the GPGGA message. Under mode 1, the residuals are recomputed after the position solution in the GPGGA message is computed. The receiver computes range residuals in mode 1. An integrity process using GPGRS would also require GPGGA (for position fix data), GPGSA (for DOP figures) and GPGSV (for PRN numbers) for comparative purposes.

The GPGRS log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



1. If the range residual exceeds  $\pm 99.9$ , then the decimal part is dropped. Maximum value for this field is  $\pm 999$ . The sign of the range residual is determined by the order of parameters used in the calculation as follows:

$$\text{range residual} = \text{calculated range} - \text{measured range}$$

2. If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GB/BD (BDS satellites only), GQ (QZSS satellites only), GI (NavIC satellites only) or GN (combined system satellites).



The position for the **GPRMC** log is reported in the user selected datum. See the **DATUM** command on page 133 for more details.

**Message ID: 220**

**Log Type: Synch**

**Recommended Input:**

```
log gpgrs ontime 1
```

**Example 1 (NMEATALKER set to GP; GPS only):**

```
$GPGRS,202812.00,1,0.5,0.1,-0.0,-0.0,-0.2,0.8,0.4,-0.3,-0.6,-0.0,,*4A
```

**Example 2 (NMEATALKER set to AUTO; Multi-constellation):**

```
$GNGRS,174837.00,1,-0.1,0.7,-0.2,0.1,0.3,0.5,-0.7,-0.5,-0.3,0.3,,*72
```

```
$GNGRS,174837.00,1,2.5,-1.3,-0.4,-0.5,2.8,-2.1,-3.3,,,,*50
```

```
$GNGRS,174837.00,1,-0.3,-0.1,-0.3,-0.5,0.8,-0.3,0.4,0.5,,,,*7D
```

```
$GNGRS,174837.00,1,0.2,-0.3,0.1,0.1,-1.3,0.1,0.0,0.3,,,,*5F
```



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPGRS	Log header		\$GPGRS
2	utc	UTC time status of position (hours/minutes/seconds/decimal seconds)	hhmmss.ss	192911.0
3	mode	Mode 0= residuals were used to calculate the position given in the matching GGA line (apriori) (not used by OEM7 receivers) Mode 1= residuals were recomputed after the GGA position was computed (preferred mode)	x	1
4 - 15	res	Range residuals for satellites used in the navigation solution. Order matches order of PRN numbers in GPGSA	x.x,x.x,.....	-13.8,- 1.9,11.4,- 33.6,0.9, 6.9,- 12.6,0.3,0.6, - 22.3
16	system ID	GNSS system ID. See <i>Table 117: System and Signal IDs</i> on the next page. This field is only output if the NMEAVERSION is 4.11 (see the <b>NMEAVERSION</b> command on page 278).	h	1
17	signal ID	ID of the ranging signal. See <i>Table 117: System and Signal IDs</i> on the next page	h	1
18	*xx	Check sum	*hh	*65
19	[CR][LF]	Sentence terminator		[CR][LF]

Table 117: System and Signal IDs

GNSS System	System ID	Signal ID	Signal Channel
GPS	1 (GP)	0	All signals
		1	L1 C/A
		2	L1 P(Y)
		3	L1 M
		4	L2 P(Y)
		5	L2C-M
		6	L2C-L
		7	L5-I
		8	L5-Q
		9 - F	Reserved
GLONASS	2 (GL)	0	All signals
		1	L1 C/A
		2	L1 P
		3	L2 C/A
		4	L2 P
		5 - F	Reserved
Galileo	3 (GA)	0	All signals
		1	E5a
		2	E5b
		3	E5a+b
		4	E6-A
		5	E6-BC
		6	L1-A
		7	L1-BC
		8 - F	Reserved

<b>GNSS System</b>	<b>System ID</b>	<b>Signal ID</b>	<b>Signal Channel</b>
BDS (BeiDou System)	4 (GB/BD)	0	All signals
		1	B1I
		2	B1Q
		3	B1C
		4	B1A
		5	B2-a
		6	B2-b
		7	B2 a+b
		8	B3I
		9	B3Q
		A	B3A
		B	B2I
		C	B2Q
		D - F	Reserved
QZSS	5 (GQ)	0	All signals
		1	L1 C/A
		2	L1C (D)
		3	L1C (P)
		4	LIS
		5	L2C-M
		6	L2C-L
		7	L5-I
		8	L5-Q
		9	L6D
		A	L6E
		B - F	Reserved

<b>GNSS System</b>	<b>System ID</b>	<b>Signal ID</b>	<b>Signal Channel</b>
NavIC	6 (GI)	0	All signals
		1	L5-SPS
		2	S-SPS
		3	L5-RS
		4	S-RS
		5	L1-SPS
		6 - F	Reserved
Reserved	7 - F		

## 3.64 GPGSA

### *GPS DOP and active satellites*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains GNSS receiver operating mode, satellites used for navigation and DOP values.

The GPGSA log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), or GN (satellites from multiple systems) or GA (Galileo satellites only).



When the solution type is INS (SPAN models), the DOPs in the **GPGGA** log are calculated at a 1 Hz rate. To determine if the receiver is providing an INS solution, check the **pos type** field in the **BESTPOS** log (see page 520).

**Message ID: 221**

**Log Type: Synch**

**Recommended Input:**

```
log gpgsa ontime 1
```

**Example 1 (GPS only):**

```
$GPGSA,M,3,05,02,31,06,19,29,20,12,24,25,,,0.9,0.5,0.7*35
```

**Example 2 (Multi-constellation):**

```
$GNGSA,M,3,03,14,17,06,12,19,02,01,24,32,,,0.8,0.5,0.6*22
```

```
$GNGSA,M,3,66,85,75,67,73,84,83,,,,,0.8,0.5,0.6*26
```

```
$GNGSA,M,3,12,11,33,31,03,24,25,08,,,,,0.8,0.5,0.6*20
```

```
$GNGSA,M,3,28,46,36,27,39,23,43,37,,,,,0.8,0.5,0.6*2B
```



The DOPs provide a simple characterization of the user satellite geometry. DOP is related to the volume formed by the intersection points of the user satellite vectors, with the unit sphere centered on the user. Larger volumes give smaller DOPs. Lower DOP values generally represent better position accuracy. The role of DOP in GNSS positioning is often misunderstood. A lower DOP value does not automatically mean a low position error. The quality of a GNSS derived position estimate depends upon both the measurement geometry as represented by DOP values and range errors caused by signal strength, ionospheric effects, multipath and so on.



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPGSA	Log header		\$GPGSA
2	mode MA	A = Automatic 2D/3D M = Manual, forced to operate in 2D or 3D	M	M
3	mode 123	Mode: 1 = Fix not available; 2 = 2D; 3 = 3D	x	3
4 - 15	prn	PRN numbers of satellites used in solution (null for unused fields), total of 12 fields GPS = 1 to 32 SBAS = 33 to 64 (add 87 for PRN number) GLO = 65 to 96 <sup>1</sup>	xx,xx,.....	18,03,13, 25,16, 24,12, 20,,,,
16	pdop	Position dilution of precision	x.x	1.5
17	hdop	Horizontal dilution of precision	x.x	0.9
18	vdop	Vertical dilution of precision	x.x	1.2
19	system ID	GNSS system ID. See <i>Table 117: System and Signal IDs</i> on page 619. This field is only output if the NMEAVERSION is 4.11 (see the <b>NMEAVERSION</b> command on page 278).		
20	*xx	Check sum	*hh	*3F
21	[CR][LF]	Sentence terminator		[CR][LF]

<sup>1</sup>The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

## 3.65 GPGST

### *Estimated error in position solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains pseudorange measurement noise statistics are translated in the position domain in order to give statistical measures of the quality of the position solution.

This log reflects the accuracy of the solution type used in the **BESTPOS** log (see page 520) and **GPGGA** log (see page 610), except for the RMS field. The RMS field, since it specifically relates to pseudorange inputs, does not represent carrier-phase based positions. Instead, it reflects the accuracy of the pseudorange position which is given in the **PSRPOS** log (see page 774).

The GPGST log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from multiple systems) or GA (Galileo satellites only).

**Message ID: 222**

**Log Type: Synch**

**Recommended Input:**

```
log gpgst ontime 1
```

**Example 1 (GPS only):**

```
$GPGST,203017.00,1.25,0.02,0.01,-16.7566,0.02,0.01,0.03*7D
```

**Example 2 (Multi-constellation):**

```
$GNGST,205246.00,1.19,0.02,0.01,-2.4501,0.02,0.01,0.03*5B
```



1. See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.
2. Accuracy is based on statistics, reliability is measured in percent. When a receiver can measure height to one metre, this is an accuracy. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%, that is, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (1 m) and the result is 95% reliability (error is less than 2 m 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions.

As examples of statistics, the GPGST message and NovAtel performance specifications use Root Mean Square (RMS). Specifications may be quoted in CEP:

- RMS - root mean square (a probability level of 68%)
- CEP - circular error probable (the radius of a circle such that 50% of a set of events occur inside the boundary)



Field	Structure	Description	Symbol	Example
1	\$GPGST	Log header		\$GPGST
2	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	173653.00
3	rms	RMS value of the standard deviation of the range inputs to the navigation process. Range inputs include pseudoranges and DGPS corrections	x.x	2.73
4	smjr std	Standard deviation of semi-major axis of error ellipse (m)	x.x	2.55
5	smnr std	Standard deviation of semi-minor axis of error ellipse (m)	x.x	1.88
6	orient	Orientation of semi-major axis of error ellipse (degrees from true north)	x.x	15.2525
7	lat std	Standard deviation of latitude error (m)	x.x	2.51
8	lon std	Standard deviation of longitude error (m)	x.x	1.94
9	alt std	Standard deviation of altitude error (m)	x.x	4.30
10	*xx	Check sum	*hh	*6E
11	[CR][LF]	Sentence terminator		[CR][LF]

## 3.66 GPGSV

### *GPS satellites in view*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the number of GPS SVs in view, PRN numbers, elevation, azimuth and SNR value. Each message includes up to four satellites. If there are more than four satellites in view, additional messages are transmitted. The total number of messages and the message number are included in each message.

The GPGSV log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



1. Satellite information may require the transmission of multiple messages. The first field specifies the total number of messages, minimum value 1. The second field identifies the order of this message (message number), minimum value 1.
2. If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems). Each system is output in a separate message.
3. The ID setting in the **NMEATALKER** command (see page 276) controls the satellites reported in this log. If the NMEATALKER ID is set to GP, only GPS satellites are reported in this log. If the NMEATALKER ID is set to AUTO, all satellites in view are reported.
4. A variable number of 'PRN-Elevation-Azimuth-SNR' sets are allowed up to a maximum of four sets per message. Null fields are not required for unused sets when less than four sets are transmitted.

**Message ID: 223**

**Log Type: Synch**

**Recommended Input:**

```
log gpgsv ontime 1
```

**Example (GPS only):**

```
$GPGSV,4,1,16,02,82,150,53,11,78,139,,12,72,191,53,25,50,296,51*77
$GPGSV,4,2,16,06,43,056,49,20,33,149,45,29,20,275,44,19,16,087,46*73
$GPGSV,4,3,16,31,11,329,42,05,10,169,42,24,07,212,44,04,01,033,*7C
$GPGSV,4,4,16,44,32,184,47,51,31,171,48,48,31,194,47,46,30,199,47*7E
```

**Example (Multi-constellation):**

```
$GLGSV,3,1,10,81,63,034,51,82,53,272,51,80,52,292,,79,38,200,49*6A
$GLGSV,3,2,10,65,24,046,42,66,16,105,47,73,15,334,46,88,14,062,46*61
$GLGSV,3,3,10,83,11,253,,72,07,001,44*68
$GPGSV,4,1,16,02,80,085,53,11,76,091,,12,61,180,51,25,58,284,52*79
```

```

$GPGSV,4,2,16,20,42,142,50,06,33,056,47,29,28,280,47,05,19,166,47*7F
$GPGSV,4,3,16,31,15,321,46,19,08,093,46,04,04,025,,09,03,055,*78
$GPGSV,4,4,16,44,32,184,48,51,31,171,48,48,31,194,47,46,30,199,48*7E
$GAGSV,3,1,09,34,72,231,53,30,65,251,53,36,51,059,51,02,36,170,49*62
$GAGSV,3,2,09,27,25,314,47,15,19,236,47,04,08,037,46,09,04,085,*65
$GAGSV,3,3,09,11,03,057,*50
$GQGSV,1,1,01,02,08,309,37*4D
$BDGSV,5,1,18,34,85,015,53,11,67,274,51,12,55,069,49,43,39,265,50*61
$BDGSV,5,2,18,23,37,289,50,25,36,225,48,44,31,078,49,22,25,064,46*69
$BDGSV,5,3,18,21,20,119,45,16,11,320,42,06,10,325,40,09,10,340,38*6D
$BDGSV,5,4,18,39,08,310,40,37,07,331,45,59,06,288,,19,05,017,*64
$BDGSV,5,5,18,31,03,333,,24,02,179,*68
$GIGSV,1,1,00,,,,,*60

```



The GPGSV log can be used to determine which GPS satellites are currently available to the receiver. Comparing the information from this log to that in the GPGSA log shows if the receiver is tracking all available satellites.



See also the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPGSV	Log header		\$GPGSV
2	# msgs	Total number of messages (1-9)	x	3
3	msg #	Message number (1-9)	x	1
4	# sats	Total number of satellites in view. May be different than the number of satellites in use (see also the <b>GPGGA</b> log on page 610)	xx	09

Field	Structure	Description	Symbol	Example
5	prn	Satellite PRN number GPS = 1 to 32 Galileo = 1 to 36 BeiDou = 1 to 63 NavIC = 1 to 14 QZSS = 1 to 10 SBAS = 33 to 64 (add 87 for PRN#s) GLONASS = 65 to 96 <sup>1</sup>	xx	03
6	elev	Elevation, degrees, 90 maximum	xx	51
7	azimuth	Azimuth, degrees True, 000 to 359	xxx	140
8	SNR	SNR (C/No) 00-99 dB, null when not tracking	xx	42
...	...	Next satellite PRN number, elev, azimuth, SNR,		
...	...	...		
...	...	Last satellite PRN number, elev, azimuth, SNR,		
variable	system ID	GNSS system ID. See <i>Table 117: System and Signal IDs</i> on page 619. This field is only output if the NMEAVERSION is 4.11 (see the <b>NMEAVERSION</b> command on page 278).		
variable	*xx	Check sum	*hh	*72
variable	[CR][LF]	Sentence terminator		[CR][LF]

<sup>1</sup>The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

## 3.67 GPHDT

### *NMEA heading log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains actual vessel heading in degrees True (from True North). See also a description of heading in the **HEADING2** log on page 646. You can also set a standard deviation threshold for this log, see the **HDTOUTTHRESHOLD** command on page 210.



You must have an ALIGN capable receiver to use this log.



The GPHDT log can only be logged using the ONCHANGED trigger. Other triggers, such as ONTIME are not accepted.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).

**Message ID: 1045**

**Log Type: Asynch**

**Recommended Input:**

```
log gphdt onchanged
```

**Example 1 (GPS only):**

```
$GPHDT,75.5664,T*36
```

**Example 2 (Combined GPS and GLONASS):**

```
$GNHDT,75.5554,T*45
```

Field	Structure	Description	Symbol	Example
1	\$GPHDT	Log header		\$GPHDT
2	heading	Heading in degrees	x.x	75.5554
3	True	Degrees True	T	T
4	*xx	Check sum	*hh	*36
5	[CR][LF]	Sentence terminator		[CR][LF]

## 3.68 GPHDTDUALANTENNA

### *Synchronous NMEA heading log*

**Platform:** OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2, CPT7

This log contains actual vessel heading in degrees True (from True North). It provides the same information as the **GPHDT** log (see page 629), but with synchronous output.



You must have an ALIGN capable, dual antenna receiver to use this log.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).

**Message ID:** 2045

**Log Type:** Synch

**Recommended Input:**

```
log gphdtdualantenna ontime 1
```

**Example 1 (GPS only):**

```
$GPHDT,75.5664,T*36
```

**Example 2 (Combined GPS and GLONASS):**

```
$GNHDT,75.5554,T*45
```

Field	Structure	Description	Symbol	Example
1	\$GPHDT	Log header		\$GPHDT
2	heading	Heading in degrees	x.x	75.5554
3	True	Degrees True	T	T
4	*xx	Check sum	*hh	*36
5	[CR][LF]	Sentence terminator		[CR][LF]

## 3.69 GPRMB

### *Navigation information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains navigation data from present position to a destination waypoint. The destination is set active by the receiver **SETNAV** command (see page 399).

The GPRMB log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

**Message ID:** 224

**Log Type:** Synch

**Recommended Input:**

```
log gprmb ontime 1
```

**Example 1 (GPS only):**

```
$GPRMB,A,4.32,L,FROM,TO,5109.7578000,N,11409.0960000,W,4.6,279.2,0.0,V,D*4A
```

**Example 2 (Multi-constellation):**

```
$GNRMB,A,4.32,L,FROM,TO,5109.7578000,N,11409.0960000,W,4.6,279.2,0.0,V,D*54
```




If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Field Description	Symbol	Example
1	\$GPRMB	Log header		\$GPRMB
2	data status	Data status: A = data valid; V = navigation receiver warning	A	A

Field	Structure	Field Description	Symbol	Example
3	xtrack	<p>Cross track error</p> <p>Represents the track error from the intended course</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">  If the cross track error exceeds 9.99 NM, displays 9.99.         </div> <div style="border: 1px solid black; padding: 5px;">  One nautical mile (NM) = 1,852 metres.         </div>	x.x	5.14
4	dir	<p>Direction to steer to get back on track (L/R)</p> <p>Direction to steer is based on the sign of the crosstrack error, that is,            L = xtrack error (+)            R = xtrack error (-)</p>	a	L
5	origin ID	Origin waypoint ID	c--c	FROM
6	dest ID	Destination waypoint ID	c--c	TO
7	dest lat	Destination waypoint latitude (DDmm.mm)	IIII.II	5109.7578000
8	lat dir	Latitude direction (N = North, S = South)	a	N
9	dest lon	Destination waypoint longitude (DDDmm.mm)	yyyyy.yy	11409.0960000
10	lon dir	Longitude direction (E = East, W = West)	a	W
11	range	<p>Range to destination, nautical miles</p> <div style="border: 1px solid black; padding: 5px;">  If the range to destination exceeds 999.9 NM, displays 999.9.         </div>	x.x	5.1
12	bearing	Bearing to destination, degrees True	x.x	303.0
13	vel	Destination closing velocity, knots	x.x	-0.0
14	arr status	<p>Arrival status:</p> <p>A = perpendicular passed</p> <p>V = destination not reached or passed</p>	A	V
15	mode ind	Positioning system mode indicator, see <i>Table 118: NMEA Positioning System Mode Indicator</i> on the next page	a	A
16	*xx	Check sum	*hh	*6F
17	[CR][LF]	Sentence terminator		[CR][LF]





The fields *origin ID*, *dest ID*, *dest lat*, *lat dir*, *dest lon* and *lon dir* are tagged from the **SETNAV** command (see page 399).

**Table 118: NMEA Positioning System Mode Indicator**

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

## 3.70 GPRMC

### *GPS specific information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains time, date, position, track made good and speed data provided by the GPS navigation receiver. RMC and RMB are the recommended minimum navigation data to be provided by a GNSS receiver.

A comparison of the position precision between this log and other selected NMEA logs can be seen in *Table 115: Position Precision of NMEA Logs* on page 616.

The GPRMC log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).



The position for the **GPRMC** log is reported in the user selected datum. See the **DATUM** command on page 133 for more details.

**Message ID: 225**

**Log Type: Synch**

**Recommended Input:**

```
log gprmc ontime 1
```

**Example 1 (GPS):**

```
$GPRMC,203522.00,A,5109.0262308,N,11401.8407342,W,0.004,133.4,130522,0.0,E,D*2B
```

**Example 2 (Multi-constellation):**

```
$GNRMC,204520.00,A,5109.0262239,N,11401.8407338,W,0.004,102.3,130522,0.0,E,D*3B
```



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Field Description	Symbol	Example
1	\$GPRMC	Log header		\$GPRMC
2	utc	UTC of position	hhmmss.ss	144326.00
3	pos status	Position status (A = data valid, V = data invalid)	A	A

Field	Structure	Field Description	Symbol	Example
4	lat	Latitude (DDmm.mm)	IIII.II	5107.0017737
5	lat dir	Latitude direction: (N = North, S = South)	a	N
6	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3291611
7	lon dir	Longitude direction: (E = East, W = West)	a	W
8	speed Kn	Speed over ground, knots	x.x	0.080
9	track true	Track made good, degrees True	x.x	323.3
10	date	Date: dd/mm/yy	xxxxxx	210307
11	mag var	Magnetic variation, degrees Note that this field is the actual magnetic variation and will always be positive. The direction of the magnetic variation is always positive.	x.x	0.0
12	var dir	Magnetic variation direction E/W Easterly variation (E) subtracts from True course. Westerly variation (W) adds to True course.	a	E
13	mode ind	Positioning system mode indicator, see <i>Table 119: NMEA Positioning System Mode Indicator</i> below	a	A
14	*xx	Check sum	*hh	*20
15	[CR][LF]	Sentence terminator		[CR][LF]

**Table 119: NMEA Positioning System Mode Indicator**

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

## 3.71 GPSCNAVRAWMESSAGE

### GPS CNAV Raw Message

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the raw CNAV message from signals which contain the CNAV message (L2C, L5). It also indicates whether the raw message is generated from an L2C signal or L5 signal.

The **GPSCNAVRAWMESSAGE** log is not output by default. To receive this log, data decoding for L2C or L5 must be enabled using the **DATADECODESIGNAL** command (see page 129) for the specific signal.

**Message ID:** 2262

**Log Type:** Asynch

**Recommended Input:**

```
log gpscnavrawmessagea onnew
```

**ASCII Example:**

```
#GPSCNAVRAWMESSAGEA,USB1,0,46.5,SATTIME,2209,506256.000,02000020,06c3,16809;
269,25,GPSL2C,10,8b64aa4cd1142707f75100a5187ffd0e6341b800a9731a4316f02b1af1c8a3
6c97f854792390*7e700506
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GPSCNAVRAWMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	signal channel	Signal channel providing the bits	Ulong	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	signal type	Signal type (L2C or L5) See <i>Table 120: Signal Type</i> on the next page	Enum	4	H+8
5	message ID	Message ID	Ulong	4	H+12
6	data	Raw message data	Hex[38]	38	H+16
7	xxxx	32-bit CRC (ASCII and binary only)	Hex	4	H+54
8	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

Table 120: Signal Type

Value (Binary)	Signal (ASCII)	Description
33	GPSL1CA	GPS L1 C/A-code
47	GPSL1CP	GPS L1C P-code
68	GPSL2Y	GPS L2 P(Y)-code
69	GPSL2C	GPS L2 C/A-code
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
2662	GLOL3	GLONASS L3
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10466	GALE5A	Galileo E5A
10499	GALE5B	Galileo E5B
10532	GALALTBOC	Galileo ALT-BOC
10565	GALE6C	Galileo E6C
10572	GALE6B	Galileo E6B
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2I with D1 navigation data
12804	BDSB2D2	BeiDou B2I with D2 navigation data
12877	BDSB3D1	BeiDou B3 with D1 navigation data
12880	BDSB3D2	BeiDou B3 with D2 navigation data
12979	BDSB1C	BeiDou B1C
13012	BDSB2A	BeiDou B2a
13077	BDSB2BI	BeiDou B2b (I)
14753	QZSSL1CA	QZSS L1 C/A-code
14760	QZSSL1CP	QZSS L1C P-code

<b>Value (Binary)</b>	<b>Signal (ASCII)</b>	<b>Description</b>
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5
14891	QZSSL6P	QZSS L6P
19073	NAVICL5SPS	NavIC L5 SPS

## 3.72 GPSEPHEM

### *Decoded GPS L1 C/A ephemerides*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains a single set of GPS ephemeris parameters.

**Message ID:** 7

**Log Type:** Asynch

**Recommended Input:**

log gpsephema onchanged

**ASCII Example:**

```
#GPSEPHEMA, USB1, 30, 52.5, SATTIME, 2209, 496800.000, 02000020, 1214, 16809;
1, 496800.0, 0, 33, 33, 2209, 2209, 504000.0, 2.656034748e+07, 3.711940331e-09,
2.013999822e+00, 1.1881368468e-02, 9.0331062162e-01, -2.292916179e-06,
1.384504139e-05, 1.27000000e+02, -4.17500000e+01, -2.440065145e-07,
-1.601874828e-07, 9.8797254649e-01, 1.546492989e-10, 2.868107379e+00,
-7.51745599e-09, 33, 504000.0, 5.122274160e-09, 3.63708e-04, -8.29914e-12,
0.00000, TRUE, 1.458576941e-04, 4.00000000e+00*b892b91a

#GPSEPHEMA, USB1, 29, 52.5, SATTIME, 2209, 506310.000, 02000020, 1214, 16809;
2, 506310.0, 0, 59, 59, 2209, 2209, 511200.0, 2.656030527e+07, 4.114457098e-09,
-3.065876369e+00, 2.0427562180e-02, -1.419168291e+00, -3.142282367e-06,
1.297891140e-05, 1.24593750e+02, -6.54062500e+01, 3.781169653e-07, 1.844018698e-07,
9.6608612461e-01, 9.000374902e-11, 2.777002903e+00, -7.50781273e-09, 59, 511200.0,
-1.769512892e-08, -6.53255e-04, 1.13687e-13, 0.00000, TRUE, 1.458584443e-04,
4.00000000e+00*613e9e7e

...

#GPSEPHEMA, USB1, 1, 52.5, SATTIME, 2209, 506310.000, 02000020, 1214, 16809;
31, 506310.0, 0, 73, 73, 2209, 2209, 511200.0, 2.656107835e+07, 4.222318734e-09,
1.928386048e+00, 1.0431749630e-02, 3.8970372621e-01, 2.447515726e-06,
1.280941069e-05, 1.34125000e+02, 4.61250000e+01, -1.508742571e-07, 2.980232239e-08,
9.5538846447e-01, -2.110802209e-10, -2.6326210009e-01, -7.85925594e-09,
73, 511200.0, -1.350417733e-08, -1.77489e-04, -1.59162e-12, 0.00000, TRUE,
1.458521844e-04, 4.00000000e+00*085ac00a

#GPSEPHEMA, USB1, 0, 52.5, SATTIME, 2209, 497010.000, 02000020, 1214, 16809;
32, 497010.0, 0, 49, 49, 2209, 2209, 504000.0, 2.655950009e+07, 4.804485840e-09,
-1.401126814e+00, 5.9581462992e-03, -2.285950321e+00, -2.717599273e-06,
6.832182407e-06, 2.48281250e+02, -5.08437500e+01, 1.899898052e-07, 1.490116119e-08,
9.5774991123e-01, -5.596661695e-10, -1.341873629e+00, -8.19927010e-09, 49, 504000.0,
4.656612873e-10, -1.26368e-04, -8.64020e-12, 0.00000, TRUE, 1.458657670e-04,
4.00000000e+00*82ba7dc8
```



The GPSEPHEM log can be used to monitor changes in the orbits of GPS satellites.



To obtain copies of IS-GPS-200, refer to the GPS website (<https://www.gps.gov/technical/icwg/>).

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GPSEPHHEM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	toe	Time stamp of subframe 1 (s)	Double	8	H+4
4	health	Health status <ul style="list-style-type: none"> <li>bits 0-5: 6-bit health code as defined in IS-GPS-200,</li> <li>bit 6: Alert flag</li> </ul>	Ulong	4	H+12
5	IODE1	Issue of ephemeris data 1	Ulong	4	H+16
6	IODE2	Issue of ephemeris data 2	Ulong	4	H+20
7	week	toe week number computed from Z count week (GPS reference week number)	Ulong	4	H+24
8	z week	Z count week number. This is the week number from subframe 1 of the ephemeris. The 'toe week' (field #7) is derived from this to account for rollover	Ulong	4	H+28
9	toe	Reference time for ephemeris (s)	Double	8	H+32
10	A	Semi-major axis (m)	Double	8	H+40
11	$\Delta N$	Mean motion difference (radians/s)	Double	8	H+48
12	$M_0$	Mean anomaly of reference time (radians)	Double	8	H+56
13	ecc	Eccentricity, dimensionless	Double	8	H+64
14	$\omega$	Argument of perigee (radians)	Double	8	H+72
15	cuc	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+80
16	cus	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+88
17	crc	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+96
18	crs	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+104




Field	Field type	Description	Format	Binary Bytes	Binary Offset
19	cic	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+112
20	cis	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+120
21	$l_0$	Inclination angle at reference time (radians)	Double	8	H+128
22	$l^0$	Rate of inclination angle (radians/s)	Double	8	H+136
23	$\Omega_0$	Longitude of ascending node of orbit plane (radians)	Double	8	H+144
24	$\dot{\omega}$	Rate of right ascension (radians/s)	Double	8	H+152
25	iodc	Issue of data clock	Ulong	4	H+160
26	toc	SV clock correction term (s)	Double	8	H+164
27	tgdc	Estimated group delay difference (s)	Double	8	H+172
28	$a_{f0}$	Clock aging parameter (s)	Double	8	H+180
29	$a_{f1}$	Clock aging parameter (s/s)	Double	8	H+188
30	$a_{f2}$	Clock aging parameter (s/s/s)	Double	8	H+196
31	AS	Anti-spoofing on: 0 = FALSE 1 = TRUE	Bool	4	H+204
32	N	Corrected mean motion (radians/s)  This field is computed by the receiver.	Double	8	H+208
33	URA	User Range Accuracy variance (m) <sup>2</sup> The ICD specifies that the URA index transmitted in the ephemerides can be converted to a nominal standard deviation value using an algorithm listed there. We publish the square of the nominal value (variance). The correspondence between the original URA index and the value output is shown in <i>Table 121: URA Variance</i> on the next page	Double	8	H+216
34	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+224
35	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 121: URA Variance

Index Value (m)	A: Standard Deviations (m)	Variance: A <sup>2</sup> (m <sup>2</sup> )
0	2.0	4
1	2.8	7.84
2	4.0	16
3	5.7	32.49
4	8	64
5	11.3	127.69
6	16.0	256
7	32.0	1024
8	64.0	4096
9	128.0	16384
10	256.0	65536
11	512.0	262144
12	1024.0	1048576
13	2048.0	4194304
14	4096.0	16777216
15	8192.0	67108864

## 3.73 GPVTG

### *Track made good and ground speed*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the track made good and speed relative to the ground.

The GPVTG log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status (see the **TIME** log on page 965) is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

**Message ID:** 226

**Log Type:** Synch

**Recommended Input:**

```
log gpvtg ontime 1
```

**Example 1 (GPS only):**

```
$GPVTG,224.592,T,224.592,M,0.003,N,0.005,K,D*20
```

**Example 2 (Multi-constellation):**

```
$GNVTG,139.969,T,139.969,M,0.007,N,0.013,K,D*3D
```



If the **NMEATALKER** command (see page 276) is set to AUTO, the talker (the first 2 characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), GA (Galileo satellites only), GQ (for QZSS satellites only), GB/BD (for BeiDou satellites only) or GN (satellites from multiple systems).



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPVTG	Log header		\$GPVTG
2	track true	Track made good, degrees True	x.x	24.168
3	T	True track indicator	T	T
4	track mag	Track made good, degrees Magnetic; Track mag = Track true + (MAGVAR correction) See the <b>MAGVAR</b> command on page 258	x.x	24.168
5	M	Magnetic track indicator	M	M
6	speed Kn	Speed over ground, knots	x.x	0.4220347

Field	Structure	Description	Symbol	Example
7	N	Nautical speed indicator (N = Knots)	N	N
8	speed Km	Speed, kilometres/hour	x.x	0.781608
9	K	Speed indicator (K = km/hr)	K	K
10	mode ind	Positioning system mode indicator, see <i>Table 122: NMEA Positioning System Mode Indicator</i> below	a	A
11	*xx	Check sum	*hh	*7A
12	[CR][LF]	Sentence terminator		[CR][LF]

**Table 122: NMEA Positioning System Mode Indicator**

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

## 3.74 GPZDA

### *UTC time and date*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The GPZDA log outputs the UTC date and time. If no valid almanac is stored in the receiver, a default UTC offset is used to generate the time until a new almanac is downloaded. If the offset is not up-to-date, this initial UTC time may be incorrect until the new almanac is present.

**Message ID:** 227

**Log Type:** Synch

**Recommended Input:**

```
log gpzda ontime 1
```

**Example:**

```
$GPZDA,204007.00,13,05,2022,,*62
```



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.

Field	Structure	Description	Symbol	Example
1	\$GPZDA	Log header		\$GPZDA
2	utc	UTC time status	hhmmss.ss	220238.00
3	day	Day, 01 to 31	xx	15
4	month	Month, 01 to 12	xx	07
5	year	Year	xxxx	1992
6	null	Local zone description—not available <div data-bbox="462 1333 998 1501" style="border: 1px solid black; padding: 5px; margin: 5px 0;">  Local time zones are not supported by OEM7 family receivers. Fields 6 and 7 are always null. </div>	xx	(empty when no data is present)
7	null	Local zone minutes description—not available	xx	(empty when no data is present)
8	*xx	Check sum	*hh	*6F
9	[CR][LF]	Sentence terminator		[CR][LF]

## 3.75 HEADING2

### *Heading information with multiple rovers*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

The heading is the angle from True North of the base to rover vector in a clockwise direction. This log can be output at both Master and Rover ends.



An ALIGN capable receiver is required to use this log.



Asynchronous logs, such as HEADING2, should only be logged ONCHANGED or ONNEW otherwise the most current data is not available or included in the output. An example of this occurrence is in the ONTIME trigger. If this trigger is not logged ONNEW or ONCHANGED, it may cause inaccurate time tags.

The HEADING2 log is dictated by the output frequency of the master receiver sending out RTCAOBS2, RTCAOBS3 or NOVATELXOBS messages. HEADING2 supports 20 Hz output rate. Ensure sufficient radio bandwidth is available between the ALIGN Master and the ALIGN Rover.

**Message ID: 1335**

**Log Type: Asynch**

**Recommended Input:**

```
log heading2a onnew
```

**ASCII Example:**

```
#HEADING2A, COM1, 0, 39.5, FINESTEERING, 1622, 422892.200, 02040000, f9bf, 6521;
SOL_COMPUTED, NARROW_INT, 0.927607417, 178.347869873, -1.3037414550, 0, 0.261901051,
0.391376048, "R222", "AAAA", 18, 17, 17, 16, 0, 01, 0, 33*8c48d77c
```

Field	Field type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	HEADING2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4

Field	Field type	Description	Binary Format	Binary Bytes	Binary Offset
4	length	<p>Baseline length in metres</p> <p>For ALIGN Heading models with position access, this field is -1.</p> <p>For ALIGN Heading models without position access, this field is only the decimal portion of the baseline in metres.</p> <p>For ALIGN Relative Positioning models receiving corrections from a master with a fixed position, this field is -1.</p> <p>For ALIGN Relative Positioning models receiving corrections from a master in moving baseline mode, this field is the complete baseline length in metres.</p>	Float	4	H+8
5	heading	Heading in degrees (0° to 359.999°)	Float	4	H+12
6	pitch	Pitch (±90 degrees)	Float	4	H+16
7	Reserved		Float	4	H+20
8	hdg std dev	Heading standard deviation in degrees	Float	4	H+24
9	ptch std dev	Pitch standard deviation in degrees	Float	4	H+28
10	rover stn ID	<p>Rover Receiver ID</p> <p>Set using the <b>SETROVERID</b> command (see page 402) on the Rover</p> <p>e.g. setroverid RRRR</p>	Char[4]	4	H+32
11	Master stn ID	<p>Master Receiver ID</p> <p>Set using the <b>DGPSTXID</b> command (see page 138) on the Master</p> <p>Default: AAAA</p>	Char[4]	4	H+36
12	#SVs	Number of satellites tracked	Uchar	1	H+40
13	#solnSVs	Number of satellites used in solution	Uchar	1	H+41
14	#obs	Number of satellites above the elevation mask angle	Uchar	1	H+42
15	#multi	Number of satellites with multi-frequency signals above the mask angle	Uchar	1	H+43
16	sol source	Solution source (see <i>Table 123: Solution Source</i> on the next page)	Hex	1	H+44

Field	Field type	Description	Binary Format	Binary Bytes	Binary Offset
17	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Uchar	1	H+45
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+46
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+47
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 123: Solution Source

Bit	Mask	Description
0-1	0x03	Reserved
2-3	0x0C	Source antenna 0 = Primary antenna 1 = Secondary antenna
4-7	0xF0	Reserved



## 3.76 HEADINGEXT

### *Extended heading log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

HEADINGEXT is a proprietary message sent from the rover receiver to the master receiver in an ALIGN or Relative INS configuration. This message provides the position information from the rover receiver that the master receiver requires to produce an ALIGN or Relative INS solution.

**Message ID:** 1132

**Log Type:** Asynch

**Recommended Input:**

```
log com2 headingextb onnew
```

## 3.77 HEADINGEXT2

### *Extended heading log in variable array form*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

HEADINGEXT2 is a proprietary message sent from the rover receiver to the master receiver in an ALIGN or Relative INS configuration. This message provides the position information from the rover receiver that the master receiver requires to produce an ALIGN or Relative INS solution.

**Message ID: 1661**

**Log Type: Asynch**

**Recommended Input:**

```
log com2 headingext2b onnew
```

## 3.78 HEADINGRATE

### *Heading rate information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log provides rate of change for the heading parameters. The heading is the angle from True North of the base to rover vector in a clockwise direction.



You must have an ALIGN capable receiver to use this log.

**Message ID:** 1698

**Log Type:** Asynch

**Recommended Input:**

```
log headingratea onchanged
```

**ASCII Example:**

```
#HEADINGRATEA, UNKNOWN, 0, 60.0, FINESTEERING, 1873, 411044.700, 02040008, c53a, 32768;  
SOL_COMPUTED, NARROW_INT, 0.025000000, 0.000000000, -0.308837891, 0.575313330,  
0.000000000, 1.264251590, 1.663657904, 0.0, "748M", "725U", 00, 0, 0, 0*66f97b96
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	HEADINGRATE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+8
5	length rate	Rate of change of the baseline length in m/s. For Z ALIGN rovers, this field outputs the decimal portion of the baseline rate.	Float	4	H+12
6	heading rate	Rate of change of the heading in degrees/s	Float	4	H+16
7	pitch rate	Rate of change of the pitch in degrees/s	Float	4	H+20

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	length rate std dev	Baseline rate standard deviation in m/s	Float	4	H+24
9	heading rate std dev	Heading rate standard deviation in degrees/s	Float	4	H+28
10	pitch rate std dev	Pitch rate standard deviation in degrees/s	Float	4	H+32
11	Reserved		Float	4	H+36
12	rover stn ID	Rover Receiver ID Set using the <b>SETROVERID</b> command (see page 402) on the Rover receiver. For example, setroverid RRRR.	Uchar	4	H+40
13	master stn ID	Master Receiver ID Set using the <b>DGPSTXID</b> command (see page 138) on the Master receiver. Default: AAAA	Uchar	4	H+44
14	sol source	Solution source (see <i>Table 123: Solution Source</i> on page 648)	Hex	1	H+48
15	Reserved		Uchar	1	H+49
16	Reserved		Uchar	1	H+50
17	Reserved		Uchar	1	H+51
18	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+52
19	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.79 HEADINGSATS

### *Satellite used in heading solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log provides information on the satellites that are used in a heading solution.



The HEADINGSATS log can only be used from the ALIGN rover.

**Message ID: 1316**

**Log Type: Asynch**

**Recommended Input:**

```
log headingsatsa onnew
```

**ASCII Example:**

```
<HEADINGSATS COM1 0 86.5 FINESTEERING 2075 411685.000 02000020 f5b0 32768
< 17
< GPS 10 GOOD 00000003
< GPS 24 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 27 GOOD 00000003
< GPS 20 GOOD 00000003
< GPS 8 GOOD 00000003
< GPS 21 GOOD 00000003
< GPS 16 GOOD 00000003
< GPS 13 GOOD 00000003
< GLONASS 14-7 NODIFFCORR 00000000
< GLONASS 6-4 GOOD 00000001
< GLONASS 22-3 GOOD 00000003
< GLONASS 15 GOOD 00000003
< GLONASS 23+3 GOOD 00000003
< GLONASS 21+4 GOOD 00000003
< GLONASS 7+5 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	HEADINGSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	Refer to <i>Table 124: Satellite System</i> on the next page.	Enum	4	H+4

Field	Field type	Description	Format	Binary Bytes	Binary Offset
4	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest-order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS, or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems.  In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2	Ulong	4	H+8
5	Status	See <i>Table 92: Observation Statuses</i> on page 529	Enum	4	H+12
6	Signal Mask	See <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>QZSS Signal Mask</i> on page 531 and <i>NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H + 4 + (#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#satx16)
9	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Table 124: Satellite System

Binary Value	ASCII Mode Name
0	GPS
1	GLONASS
2	SBAS
5	Galileo
6	BeiDou
7	QZSS
9	NavIC

## 3.80 HWMONITOR

### Monitor hardware levels

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log allows the user to monitor temperature, antenna current and voltages.

**Message ID:** 963

**Log Type:** Polled

**Recommended Input:**

```
log hwmonitora ontime 10
```

**ASCII Example:**

```
#HWMONITORA,USB1,0,51.5,FINESTEERING,2209,507561.000,02000020,52db,16809;10,
59.218563080,100,0.000122100,200,3.302808285,600,5.165323734,700,1.189255238,
800,3.301587343,f00,1.835775375,1100,5.161416531,1500,1.554945111,1700,
59.218563080,1600*38415eae
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	HWMONITOR header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	# measurements	Number of measurements to follow	Ulong	4	H
3	reading	Temperature, antenna current or voltage reading Units: <ul style="list-style-type: none"> <li>Degree Celsius for Temperature</li> <li>Amps for Antenna Current</li> <li>Volts for Voltage</li> </ul>	Float	4	H+4
4	status	See <i>Table 125: HWMONITOR Status Table</i> on the next page	HexUlong	4	H+8
5...	Next reading offset = H + 4 + (# measurements x 8)				
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(# measurements x 8)
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Table 125: HWMONITOR Status Table

Bits	Description	Applicable Platforms
0-7	Boundary Limit Status (Hex): 0x00 = Value falls within acceptable bounds 0x01 = Value is under the lower warning limit 0x02 = Value is under the lower error limit 0x03 = Value is over the upper warning limit 0x04 = Value is over the upper error limit	
8-15	Reading Type (Hex):	
	0x00 = Reserved	
	0x01 = Temperature A temperature sensor is located on the receiver and provides the approximate temperature of the PCB surface near critical components (for example, CPU, TCXO) (degrees Celsius)	All
	0x02 = Antenna Current The amount of current being drawn by the active antenna (A)	All except OEM7500, OEM7600, SMART2
	0x06 = Digital Core 3V3 Voltage Internal regulator output voltage supplying a key component on the receivers (Volts)	All except OEM7720
	0x06 = 3.3V Supply Voltage (Volts)	OEM7720
	0x07 = Antenna Voltage	All except OEM7500, SMART2
	0x08 = Digital 1V2 Core Voltage Internal regulator output voltage supplying a key component on the receiver (Volts)	All
	0x0F = Regulated Supply Voltage Internal regulator output voltage supplying a key component on the receiver (Volts)	All except OEM7720
	0x0F = Supply Voltage Voltage applied to Pins 1 and 2 of the main connector	OEM7720
0x11 = 1V8	All	
0x15 = 5V Voltage (Volts) 5V voltage output supplying peripherals such as USB and Ethernet	All except OEM7500, OEM7600, SMART2	



Bits	Description	Applicable Platforms
	0x16 = Secondary Temperature A second temperature sensor is located on the receiver PCB (degrees Celsius)	All except OEM7500, SMART2
	0x17 = Peripheral Core Voltage	All except OEM719, OEM7500, SMART2
	0x18 = Secondary Antenna Current	OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2, CPT7
	0x19 = Secondary Antenna Voltage	OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2, CPT7

## 3.81 IONUTC

### *Ionospheric and UTC data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC).

**Message ID:** 8

**Log Type:** Asynch

**Recommended Input:**

```
log ionutca onchanged
```

**ASCII Example:**

```
#IONUTCA,USB1,0,52.5,FINESTEERING,2209,507000.090,02000020,ec21,16809;
1.769512891769409e-08,2.235174179077148e-08,-1.192092895507813e-07,
-1.192092895507813e-07,1.167360000000000e+05,1.310720000000000e+05,
-1.310720000000000e+05,-2.621440000000000e+05,2210,61440,
-5.5879354476928711e-09,-1.243449788e-14,2185,7,18,18,0*5b90fa00
```



The Receiver-Independent Exchange (RINEX<sup>1</sup>) format is a broadly accepted, receiver independent format for storing GPS data. It features a non-proprietary ASCII file format that can be used to combine or process data generated by receivers made by different manufacturers.

Use the NovAtel's Convert utility to produce RINEX files from NovAtel receiver data files.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IONUTC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	a0	Alpha parameter constant term	Double	8	H
3	a1	Alpha parameter 1st order term	Double	8	H+8
4	a2	Alpha parameter 2nd order term	Double	8	H+16
5	a3	Alpha parameter 3rd order term	Double	8	H+24
6	b0	Beta parameter constant term	Double	8	H+32
7	b1	Beta parameter 1st order term	Double	8	H+40

<sup>1</sup>Refer to the U.S. National Geodetic Survey website at: [www.ngs.noaa.gov/CORS/data.shtml](http://www.ngs.noaa.gov/CORS/data.shtml).

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	b2	Beta parameter 2nd order term	Double	8	H+48
9	b3	Beta parameter 3rd order term	Double	8	H+56
10	utc wn	UTC reference week number	Ulong	4	H+64
11	tot	Reference time of UTC parameters (s)	Ulong	4	H+68
12	A0	UTC constant term of polynomial (s)	Double	8	H+72
13	A1	UTC 1st order term of polynomial (s)	Double	8	H+80
14	wn Isf	Future week number	Ulong	4	H+88
15	dn	Day number (the range is 1 to 7 where Sunday = 1 and Saturday = 7)	Ulong	4	H+92
16	deltat Is	Delta time due to leap seconds	Long	4	H+96
17	deltat Isf	Future delta time due to leap seconds	Long	4	H+100
18	Reserved			4	H+104
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+108
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.82 IPSTATS

### *IP statistics*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This log contains the current IP interface statistics.

**Message ID:** 1669

**Log Type:** Polled

**Recommended Input:**

```
log ipstatsa
```

**ASCII Example:**

```
#IPSTATSA,USB1,0,52.5,FINESTEERING,2209,507647.000,02000020,0d94,16809;
4,ETHA,0,0,61,WIFI,0,0,0,WIFI_CLIENT,0,0,0,CELL,0,0,0*c139922f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IPSTATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#Interface	Number of records to follow.	Ulong	4	H
3	Physical Interface	IP Interface Type 1 = ALL 2 = ETHA	Enum	4	H+4
4	Reserved		Ulong	4	H+8
5	Receive Bytes	Total number of bytes received	Ulong	4	H+12
6	Transmit Bytes	Total number of bytes transmitted	Ulong	4	H+16
7	Next reading offset = H+4+(#Interface * 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#Interface * 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.83 IPSTATUS

### Current network configuration status

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-I, SMART7-SI

This log provides the configuration of IP address, netmask, gateway and a list of DNS servers currently in use.

**Message ID:** 1289

**Log Type:** Polled

**Recommended Input:**

```
log ipstatusa once
```

**ASCII Example:**

```
#IPSTATUSA,USB1,0,52.5,FINESTEERING,2209,507677.000,02000020,7fe2,16809;3,ETHA,
"", "", "0.0.0.0",WIFI,"192.168.19.1","255.255.255.0","",WIFI_CLIENT,"","", "",1,
"0.0.0.0"*661b638f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IPSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	#IPrec	Number of records to follow	Ulong	4	H
3	interface	Name of the network interface 2 = ETHA	Enum	4	H+4
4	IP address	IP Address-decimal dot notation	String[16]	variable <sup>1</sup>	H+8
5	netmask	Netmask-decimal dot notation	String[16]	variable <sup>1</sup>	H+24
6	gateway	Gateway-decimal dot notation This is the default gateway that is currently in use by the receiver.	String[16]	variable <sup>1</sup>	H+40
7...	Next reading offset = H+4+(#IPrec * 52)				
8	#dnserver	Number of DNS Servers to follow	Ulong	4	H+4+(#IPrec x 52)

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
9	server IP address	IP address-decimal dot notation	String[16]	variable <sup>1</sup>	H+4+(#IPrec x 52)+4
10...	Next reading offset = H+4+(#IPrec * 52)+4+(#dnsserver * 16)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(#IPrec x 52)+4+ (#dnsserver x 16)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.84 ITBANDPASSBANK

### *Allowable band pass filter configurations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The ITBANDPASSBANK log provides information on the allowable configurations for each frequency when applying a bandpass filter. The current filters in use can be seen with the **ITFILTABLE** log on page 667.



To view the allowable configurations for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITBANDPASSBANK\_1**.

**Message ID:** 2022

**Log Type:** Asynch

**Recommended Input:**

```
log itbandpassbanka once
```

**Abbreviated ASCII Example:**

```
<ITBANDPASSBANK USB1 0 54.0 UNKNOWN 0 2.392 02000020 fb2e 16809
< 9
< GPSL5 1164.3750 1173.1250 1178.1250 1186.8750 0.05
< GALILEOE5A 1164.3750 1173.1250 1178.1250 1186.8750 0.05
< GALILEOE5B 1195.6250 1204.3750 1209.3750 1218.1250 0.05
< BEIDOUB1 1549.6875 1558.4375 1563.4375 1572.1875 0.05
< BEIDOUB2 1195.6250 1204.3750 1209.3750 1218.1250 0.05
< QZSSL5 1164.3750 1173.1250 1178.1250 1186.8750 0.05
< NAVICL5 1164.3750 1173.1250 1178.1250 1186.8750 0.05
< BEIDOUB2A 1164.3750 1173.1250 1178.1250 1186.8750 0.05
< BEIDOUB2B 1195.6250 1204.3750 1209.3750 1218.1250 0.05
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITBANDPASSBANK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# of entries	Number of entries to follow	Ulong	4	H
3	frequency	See <i>Table 54: Frequency Types</i> on page 238	Enum	4	H+4
4	min lower frequency cutoff	The minimum frequency cutoff at the lower end (MHz)	Float	4	H+8

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	max lower frequency cutoff	The maximum frequency cutoff at the lower end (MHz)	Float	4	H+12
6	min upper frequency cutoff	The minimum frequency cutoff at the upper end (MHz)	Float	4	H+16
7	max upper frequency cutoff	The maximum frequency cutoff at the upper end (MHz)	Float	4	H+20
8	frequency step	The minimum cut off frequency resolution (MHz)	Float	4	H+24
9	Next entry offset = $H + 4 + (\#entries * 24)$				
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#entries * 24)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.85 ITDETECTSTATUS

### *Interference detection status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists all of the detected interferences from all active paths where interference detection is enabled.



This log should be used with the **onchanged** trigger only.



To view the detected interferences from all active paths on the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITDETECTSTATUS\_1**.

**Message ID:** 2065

**Log Type:** Asynch

**Recommended Input:**

```
log itdetectstatusa onchanged
```

#### ASCII Example

```
#ITDETECTSTATUSA,USB2,0,74.0,FINESTEERING,1982,430605.267,0200c000,7fdb,32768;
3,
L1,STATISTICANALYSIS,-0.718,29.167,0.126,12.797,00000000,00000000,00000000,
L2,SPECTRUMANALYSIS,1249.961,71.191,-56.769,-132.907,00000000,00000000,
00000000,
L2,SPECTRUMANALYSIS,1289.512,1.978,-75.967,-138.493,00000000,00000000,
00000000*5e83b175
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITDETECTSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# of entries	Number of interferences to follow	Ulong	4	H
3	RF Path	RF path for this entry. 2 = L1 3 = L2 5 = L5	Enum	4	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	Interference detection type	Interference detection type for this entry. 0 = SPECTRUMANALYSIS 1 = STATISTICALANALYSIS	Enum	4	H+8
5	Parameter 1	The first parameter of the interference. For SPECTRUMANALYSIS type, this is the center frequency in MHz. For STATISTICALANALYSIS type, this is reserved.	Float	4	H+12
6	Parameter 2	The second parameter of the interference. For SPECTRUMANALYSIS type, this is the bandwidth in MHz. For STATISTICALANALYSIS type, this is reserved.	Float	4	H+16
7	Parameter 3	The third parameter of the interference. For SPECTRUMANALYSIS type, this is the estimated power in dBm of the interference. For STATISTICALANALYSIS type, this is reserved.	Float	4	H+20
8	Parameter 4	The fourth parameter of the interference. For SPECTRUMANALYSIS type, this is the highest estimated power spectrum density in dBmHz of the interference. For STATISTICALANALYSIS type this is reserved.	Float	4	H+24
9	Reserved 1	Reserved	Ulong	4	H+28
10	Reserved 2	Reserved	Ulong	4	H+32
11	Reserved 3	Reserved	Ulong	4	H+36
12	Next interference signal offset = H + 4 + (#entries * 36)				
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H=4+ (#entries * 36)
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.86 ITFILTTABLE

### *Filter configuration for each frequency*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The ITFILTTABLE log contains the filter configuration summary for each frequency. It lists which bandpass or notch filters are enabled and how each is configured.



To view the filter configuration summary for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITFILTTABLE\_1**.

**Message ID: 1991**

**Log Type: Asynch**

**Recommended Input:**

```
log itfilttablea once
```

**ASCII Example:**

```
#ITFILTTABLEA,USB1,0,52.5,FINESTEERING,2209,507953.506,12000020,35d0,16809;
23,
GPSL1,8,CIC3,00000000,DISABLE,0.0000,0.0000,1,ENABLE,PF0,NOTCHFILTER,1569.0000,
1581.0000,0.150,
GPSL2,4,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GLONASSL1,9,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GLONASSL2,5,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GPSL5,0,CIC3,00000000,DISABLE,0.0000,0.0000,0,
LBAND,11,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GALILEOE1,8,CIC3,00000000,DISABLE,0.0000,0.0000,1,ENABLE,PF0,NOTCHFILTER,1569.0
000,1581.0000,0.150,
GALILEOE5A,0,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GALILEOE5B,1,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GALILEOALTB0C,10,CIC3,00000000,DISABLE,0.0000,0.0000,0,
BEID0UB1,2,CIC3,00000000,DISABLE,0.0000,0.0000,0,
BEID0UB2,1,CIC3,00000000,DISABLE,0.0000,0.0000,0,
QZSSL1,8,CIC3,00000000,DISABLE,0.0000,0.0000,1,ENABLE,PF0,NOTCHFILTER,1569.0000
,1581.0000,0.150,
QZSSL2,4,CIC3,00000000,DISABLE,0.0000,0.0000,0,
QZSSL5,0,CIC3,00000000,DISABLE,0.0000,0.0000,0,
QZSSL6,6,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GALILEOE6,6,CIC3,00000000,DISABLE,0.0000,0.0000,0,
BEID0UB3,7,CIC3,00000000,DISABLE,0.0000,0.0000,0,
GLONASSL3,10,CIC3,00000000,DISABLE,0.0000,0.0000,0,
NAVICL5,0,CIC3,00000000,DISABLE,0.0000,0.0000,0,
BEID0UB1C,8,CIC3,00000000,DISABLE,0.0000,0.0000,1,ENABLE,PF0,NOTCHFILTER,1569.0
000,1581.0000,0.150,
BEID0UB2A,0,CIC3,00000000,DISABLE,0.0000,0.0000,0,
BEID0UB2B,1,CIC3,00000000,DISABLE,0.0000,0.0000,0*cfb2b0d4
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITFILTABLE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# entries	Number of records with information to follow	Ulong	4	H
3	frequency	The frequency at which the filter is applied. See <i>Table 54: Frequency Types</i> on page 238	Enum	4	H+4
4	Encoder ID	ID of the digital path used by this frequency	Ulong	4	H+8
5	DDC filter type	The DDC filter type (see <i>Table 126: DDC Filter Type</i> on the next page)	Enum	4	H+12
6	status word	Filter warning limit status. Raise a warning flag if the filter is placed too close to the center frequency of the GNSS signal (see <i>Table 127: ITFILTable Status Word</i> on the next page)	Ulong	4	H+16
7	switch	Filter is enabled or disabled (see <i>Table 128: Filter Switches</i> on page 670)	Enum	4	H+20
8	lower cut off frequency	Cut off frequency at the lower end (MHz)	Float	4	H+24
9	upper cut off frequency	Cut off frequency at the upper end (MHz)	Float	4	H+28
10	# prog filters	Number of programmable filters applied	Ulong	4	H+32
11	switch	Filter is enabled or disabled (see <i>Table 128: Filter Switches</i> on page 670)	Enum	4	H+36
12	prog filter ID	The programmable filter ID (see <i>Table 51: Programmable Filter ID</i> on page 234)	Enum	4	H+40
13	mode	Programmable filter mode (notch filter or bandpass) (see <i>Table 52: Programmable Filter Mode</i> on page 235)	Enum	4	H+44
14	lower cut off frequency	Cut off frequency at the lower end (MHz)	Float	4	H+48
15	upper cut off frequency	Cut off frequency at the upper end (MHz)	Float	4	H+52
16	notch width	Width of notch filter (MHz)	Float	4	H+56
17	Next programmable filter – variable binary offset				
18	Next frequency – variable binary offset				

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	Variable
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 126: DDC Filter Type

Binary	ASCII
0	PASSTHROUGH
1	CIC1
2	CIC2
3	CIC3
4	HALFBAND

Table 127: ITFILTable Status Word

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	First enabled filter	0 = Within acceptable limit 1 = Warning
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010		
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	Second enabled filter	0 = Within acceptable limit 1 = Warning
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	Third enabled filter	0 = Within acceptable limit 1 = Warning
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Fourth enabled filter	0 = Within acceptable limit 1 = Warning
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 128: Filter Switches

Binary Value	ASCII Value	Description
0	DISABLE	Filter disabled
1	ENABLE	Filter enabled

## 3.87 ITPROGFILTBANK

### *Allowable filter configurations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The ITPROGFILTBANK log provides information on the allowable configurations for the programmable filter for each frequency when applying either a notch filter or bandpass filter. The current filters in use can be seen with the ITFILTABLE log on page 667.



To view the allowable configurations for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITPROGFILTBANK\_1**.

**Message ID:** 2023

**Log Type:** Asynch

**Recommended Input:**

```
log itprogfiltbanka once
```

**Abbreviated ASCII Example:**

```
<ITPROGFILTBANK USB1 0 53.5 UNKNOWN 0 2.392 12000020 3696 16809
< 21
< GPSL1 5
< NOTCHFILTR 1563.0000 1574.0000 1576.0000 1587.0000 0.05 0.15
< NOTCHFILTR 1563.7500 1573.6000 1576.4000 1586.2500 0.05 0.50
< NOTCHFILTR 1564.0500 1573.3000 1576.7000 1585.9500 0.05 1.00
< NOTCHFILTR 1565.7500 1571.7000 1578.3000 1584.2500 0.05 2.50
< BANDPASSFILTR 1563.7500 1572.5000 1577.5000 1586.2500 0.05 0.00
< GPSL2 5
< NOTCHFILTR 1215.5000 1226.5000 1228.5000 1239.5000 0.05 0.15
< NOTCHFILTR 1216.2500 1226.1000 1228.9000 1238.7500 0.05 0.50
< NOTCHFILTR 1216.5500 1225.8000 1229.2000 1238.4500 0.05 1.00
< NOTCHFILTR 1218.2500 1224.2000 1230.8000 1236.7500 0.05 2.50
< BANDPASSFILTR 1216.2500 1225.0000 1230.0000 1238.7500 0.05 0.00
...
< BEIDOU2A 5
< NOTCHFILTR 1163.6250 1174.6250 1176.6250 1187.6250 0.05 0.15
< NOTCHFILTR 1164.3750 1174.2250 1177.0250 1186.8750 0.05 0.50
< NOTCHFILTR 1164.6750 1173.9250 1177.3250 1186.5750 0.05 1.00
< NOTCHFILTR 1166.3750 1172.3250 1178.9250 1184.8750 0.05 2.50
< BANDPASSFILTR 1164.3750 1173.1250 1178.1250 1186.8750 0.05 0.00
< BEIDOU2B 5
< NOTCHFILTR 1194.8750 1205.8750 1207.8750 1218.8750 0.05 0.15
< NOTCHFILTR 1195.6250 1205.4750 1208.2750 1218.1250 0.05 0.50
< NOTCHFILTR 1195.9250 1205.1750 1208.5750 1217.8250 0.05 1.00
< NOTCHFILTR 1197.6250 1203.5750 1210.1750 1216.1250 0.05 2.50
< BANDPASSFILTR 1195.6250 1204.3750 1209.3750 1218.1250 0.05 0.00
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITPROGFILTBANK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# entries	Number of entries to follow	Ulong	4	H
3	frequency	See <i>Table 54: Frequency Types</i> on page 238	Enum	4	H+4
4	# prog filters	Number of programmable filters applied with information to follow	Ulong	4	H+8
5	mode	Programmable filter mode (notch filter or bandpass) (see <i>Table 52: Programmable Filter Mode</i> on page 235)	Enum	4	H+12
6	min lower frequency cutoff	The minimum frequency cutoff at the lower end (MHz)	Float	4	H+16
7	max lower frequency cutoff	The maximum frequency cutoff at the lower end (MHz)	Float	4	H+20
8	min upper frequency cutoff	The minimum frequency cutoff at the upper end (MHz)	Float	4	H+24
9	max upper frequency cutoff	The maximum frequency cutoff at the upper end (MHz)	Float	4	H+28
10	frequency step	The minimum cut off frequency resolution (MHz)	Float	4	H+32
11	notch width	Width of notch filter (MHz)	Float	4	H+36
12	Next programmable filter – variable binary offset				
13	Next frequency – variable binary offset				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	variable
15	[CR][LF]	Sentence terminator (ASCII only)			



## 3.88 ITPSDDETECT

### *Power spectral density computed by the interference detection algorithm*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The **ITPSDDETECT** log contains the power spectral density samples for the interference detection algorithm for all available RF paths, excluding the L-Band path. The log will iterate through all available RF paths. Generally, the **ITPSDDETECT** log is output slightly slower than every 1 second. For dual antenna receivers, only the RF paths from the primary antenna are considered. Note that one log is output for every 1024 expected samples for a maximum of 32 logs and can be grouped together through the sequence number in the log header.



To view the power spectral density samples for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITPSDDETECT\_1**.

### 3.88.1 PSD Samples

PSD samples are compressed into 2 byte samples to reduce log sizes. The range of values that can be displayed is -200 dBm to +56 dBm with a 1/256 resolution. The following steps should be performed on the PSD samples in this log to convert them back into dBm units for display purposes:

1. Divide the sample by 256.0
2. Subtract 200

**Message ID: 2063**

**Log Type: Asynch**

**Recommended Input:**

```
log itpsddetect onnew
```

#### ASCII Example

```
#ITPSDDETECTA,USB1,1,52.5,FINESTEERING,2209,508190.633,12000020,32d1,16809;
04021b22,1531.287,73242.188,1024,23036,20816,21178,20917,21252,21069,21186,
21501,20228,21326,20561,21254,21157,20442,21682,20875,20916,20918,21560,20287,
20694,21350,21409,20872,20725,20847,20715,21251,20716,21121,20794,21088,21757,
21782,21421,20433,19660,21395,21592,21157,20945,21297,20162,21313,20726,21544,
21468,20795,21080,21124,20490,20914,21649,20835,21548,21097,20551,20178,21341,
20793,21013,20290,21337,21242,21183,21769,21520,21416,20519,21761,21211,21026,
21408,21539,21257,20919,20828,20916,21124,20947,21278,20213,21124,21432,21284,
21336,20645,21817,21091,21413,21390,20698,21843,20842,20867,20598,21750,20852,
...
27699,28604,28277,28894,28715,28949,29529,28176,28121,28923,28753,29411,28256,
29559,27903,29107,29586,28285,29234,28915,28149,29504,28818,29505,28672,28325,
28951,27607,28919,28644,27515,28498,28440,29568,28695,28305,28256,27916,29967,
27756,28453,27556,28395,29259,28183,29241,29288,29205,28347,29147,29471,29428,
28958,28990,29356,29099,28455,29793,28800,28017,28291,28704,27458,29274,29320,
29374,28700,27100,28272,27938,29056,29151,28173,29381,29479,29597,29123,28960,
28689,28853,28614,28530,28571,28328*fd9113f5
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITPSDDetect header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Status word	Status word containing information about the detection configuration (see <i>Table 129: Interference Detection Status Word</i> below)	Ulong	4	H
3	Frequency start	Frequency represented by first data sample in MHz	Float	4	H+4
4	Resolution bandwidth	The resolution bandwidth (RBW) in Hz	Float	4	H+8
5	# of samples	Number of spectral density samples	Ulong	4	H+12
6	Sample array	Array of power spectral density samples	Ushort	2*NumOf Samples	H+16
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16+ 2*NumOf Samples
8	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

Table 129: Interference Detection Status Word

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Frequency	See <i>Table 130: RF Frequency Path</i> on the next pageTable 2
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	Data Source	Predecimation
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	Reserved	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		

Nibble	Bit	Mask	Description	Range Value
N3	12	0x00001000	Reserved	
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000	Reserved	
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Reserved	
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000	Reserved	
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 130: RF Frequency Path

Binary	ASCII Value	Description
2	L1	L1 RF Path
3	L2	L2 RF Path
5	L5	L5 RF Path

## 3.89 ITPSDFINAL

### *Processed power spectral density*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The ITPSDFINAL log contains the samples for the spectral analysis. The rate and size is set by the update period and the FFT size respectively when issuing the **ITSPECTRALANALYSIS** command (see page 236).

PSD samples are compressed into 2 byte samples to reduce log sizes. The range of values that can be displayed is -200 dBm to +56 dBm with a 1/256 resolution. The following steps should be performed on the PSD samples in this log to convert them back into dBm units for display purposes:

1. Divide the sample by 256.0
2. Subtract 200

The number of samples are calculated according to the following table. The maximum number of samples in one ITPSDFINAL log is 1024. That means if the number of samples is less than 1024, one log is enough to output them. However, if the number of samples is larger than 1024, more than one ITPSDFINAL log is needed. For example, in postdecimation mode with the FFT size of 8K and subcarrier integration of 5, there is one log with 1024 samples and another log with 614 samples. The output logs can be grouped together through the sequence number of the log header.

Data Source	Number of Samples
PREDECIMATION	$\text{FFTsize}/(2*\text{subcarrier\_integration})$
POSTDECIMATION	$\text{FFTsize}/\text{subcarrier\_integration}$
POSTFILTER	$\text{FFTsize}/\text{subcarrier\_integration}$



As the data rate for the ITPSDFINAL log is dictated by the updateperiod parameter in the **ITSPECTRALANALYSIS** command (see page 236), do not use ONTIME or ONCE to log this message. Instead use ONNEW to log ITPSDFINAL.



The pre-decimation spectrum shows the absolute power in dBm which is proportional to the resolution bandwidth (RBW). The post-decimation and post-filter spectrum shows the signal magnitude in relative power (dB).



To view the samples for the spectral analysis for the secondary antenna of dual antenna receivers (OEM7720, PwrPak7D, PwrPak7D-E1, PwrPak7D-E2 and CPT7), log **ITPSDFINAL\_1**.

**Message ID: 1968**

**Log Type: Asynch**

**Recommended Input:**

```
log itpsdfinala onnew
```

**ASCII Example**

```
#ITPSDFINALA,USB1,0,52.5,FINESTEERING,2209,507969.564,12000020,b79a,16809;26142
860,1562.561,122070.313,204,29796,29396,29720,29568,30272,29991,30321,29749,294
82,29315,29754,29329,29859,29839,29366,29818,30178,30317,29421,30146,29896,3014
4,30760,30315,29564,30139,30164,30667,30943,29849,30659,30339,30895,30267,30462
,29965,30541,30240,30712,30883,30523,30596,30775,30638,30753,30839,29860,31314,
31508,30487,30353,28644,26580,26452,27765,29511,30887,31557,31096,31549,31135,3
1567,31193,31756,31568,31470,31732,31118,31946,31416,31526,31193,31882,31454,31
379,31558,31555,31972,32125,31970,31384,31204,32437,31611,31551,32050,31730,319
69,31871,31232,31584,32252,31373,32220,32198,32189,33447,32370,33055,33660,3318
2,33203,33523,33306,32746,33764,33909,33424,33019,33953,32423,33388,33055,32483
,33109,32435,32411,32205,32338,32310,32051,32228,31851,32049,31698,31584,31816,
31821,31399,32122,32096,32253,31889,32050,31787,31575,31775,31363,31556,31593,3
1216,30940,31538,31570,31529,31265,31295,31518,31361,29133,27947,25797,26746,29
320,30342,30694,31279,31063,31655,30911,30762,30625,30845,30991,30900,30946,308
47,30441,30702,31092,31097,30364,31093,30439,31202,30641,30051,29842,30629,3067
6,30082,30466,30499,30363,30245,30515,30441,30365,30217,30359,29207,30538,30327
,30162,30518,29656,29978,30273,30084,30297,30136,30172,29755,29455*40f7bd2d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ITPSDFINAL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	status word	Status word containing information about the configuration of the spectral analysis (see <i>Table 131: Spectral Analysis Status Word</i> on the next page)	Ulong	4	H
3	frequency start	Frequency represented by first data sample (MHz)	Float	4	H+4
4	resolution bandwidth	The resolution bandwidth (Hz)	Float	4	H+8
5	# samples	Number of spectral density samples	Ulong	4	H+12
6	sample	Power spectral density sample	Ushort	2	H+16
7	Next sample = H+16+(2*#samples)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16+ (2*#samples)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 131: Spectral Analysis Status Word

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Frequency	0 – 20 See Table 54: Frequency Types on page 238
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	Data Source	0 – 3 See Table 53: Data Sources for PSD Samples on page 237
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	FFT Size	0 – 6 See Table 55: FFT Sizes on page 239
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000	Integration Window	1 – 1024 samples
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Time Average Window	0 – 100 seconds
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000	Reserved	
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Reserved	
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		

Nibble	Bit	Mask	Description	Range Value
N7	28	0x10000000	Reserved	
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

## 3.90 J1939STATUS

### Status of CAN J1939 Node

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This logs reports the status of J1939 node, specifically J1939 Address Claim function (initiated using the **J1939CONFIG** command (see page 241)).

This log displays the status only for nodes that have been set.

**Message ID: 1907**

**Log Type: Asynch**

**Recommended Input:**

```
LOG J1939STATUSA ONCHANGED
```

**ASCII Examples:**

```
#J1939STATUSA,USB1,1,53.0,UNKNOWN,0,0.204,12000020,e9ce,16809;NODE1,DISABLED,0,FE*6980fd0f
```

```
#J1939STATUSA,USB1,0,53.0,UNKNOWN,0,0.204,12000020,e9ce,16809;NODE2,DISABLED,0,FE*cee67706
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	J1939STATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	node	J1939 Node. The node can be either NODE1 or NODE2.	Enum	4	H
3	status	Node status. See <i>Table 132: Node Status</i> on the next page	Enum	4	H+4
4	count	Number of attempts that were made to claim address. This will be 1 when the preferred address is used and may be more if the alternate range is used.	Ulong	4	H+8
5	address	Claimed CAN Address. 0xFE (NULL address) if the address could not be negotiated.	Uchar	1	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+13
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-



**Table 132: Node Status**

<b>Value</b>	<b>ASCII</b>	<b>Description</b>
1	DISABLED	Address claim activity is not taking place. The node does not have J1939 enabled.
2	CLAIMING	Address claim procedure is in progress.
3	CLAIMED	Address claimed successfully. Ready for data transfer.
4	FAILED	Address claim was not successful. No further activity is taking place.

## 3.91 LBANDBEAMTABLE

### List of L-Band beams

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists the TerraStar and Veripos L-Band beams known to the receiver.

**Message ID:** 1718

**Log Type:** Asynch

**Recommended Input:**

```
log lbandbeamtablea onchanged
```

**Abbreviated ASCII Example:**

```
<LBANDBEAMTABLE USB1 0 52.5 FINESTEERING 2209 504117.070 12000020 19b8 16809
<      8
<      "AORE" "A" 1545855000 1200 -15.50 1
<      "AORW" "B" 1545845000 1200 -54.00 1
<      "IOR" "C" 1545865000 1200 64.50 1
<      "POR" "D" 1545905000 1200 178.00 1
<      "25E" "E" 1545825000 1200 25.00 1
<      "143.5E" "F" 1545835000 1200 143.50 1
<      "98W" "G" 1545865000 1200 -98.00 1
<      "SP98W1" "SP98W1" 1545885000 1200 -98.00 1
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LBANDBEAMTABLE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	Name	Beam/transmitting satellite name	Char[8]	8	H+4
4	Reserved		Char[8]	8	H+12
5	Frequency	Frequency (Hz)	Ulong	4	H+20
6	Baud	Baud rate (bps)	Ulong	4	H+24
7	Longitude	Transmitting satellite longitude (degrees)	Float	4	H+28
8	Access	Beam service availability flag 0 = Denied 1 = Granted	Ulong	4	H+32
9	Next beam offset = H + 4 + (#entries * 32)				

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#entries * 32)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.92 LBANDTRACKSTAT

### *L-Band Beams status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the L-Band tracking and Viterbi decoding status for the tracked L-Band beams.



The maximum logging rate for LBANDTRACKSTAT is 2 Hz.

**Message ID:** 1201

**Log Type:** Synch

**Recommended Input:**

```
log lbandtrackstata ontime 1
```

**ASCII Example:**

```
#LBANDTRACKSTATA, USB1, 0, 52.5, FINESTEERING, 2209, 508418.000, 12000020, 29e3, 16809;
5,
"98W", 1545865000, 1200, 974c, 00c2, 0, -334.215, 40.513, 3.0445, 4338.944, 82624, 36, 35,
10575872, 4081, 0.0004,
"AORW", 1545845000, 1200, 974c, 00c2, 0, 48.172, 44.840, 3.8044, 4346.541, 82752, 0, 0,
10592256, 0, 0.0000,
"POR", 1545905000, 1200, 974c, 0000, 0, -440.550, 0.000, 0.0000, 0.000, 0, 0, 0, 0, 0, 0.0000,
"", 0, 0, 0000, 0003, 0, 0.000, 0.000, 0.0000, 0.000, 0, 0, 0, 0, 0, 0.0000,
"", 0, 0, 0000, 0003, 0, 0.000, 0.000, 0.0000, 0.000, 0, 0, 0, 0, 0, 0.0000*5b097814
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LBANDTRACKSTAT header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	Name	Beam/transmitting satellite name	Char[8]	8	H+4
4	Frequency	Frequency assigned to this L-Band beam (Hz)	Ulong	4	H+12
5	Baud rate	Baud rate of assigned beam	Ushort	2	H+16
6	ID	Service ID of the assigned beam	Ushort	2	H+18
7	Status	Tracking status word. See <i>Table 133: L-Band Signal Tracking Status</i> on the next page	Ushort	2	H+20
8	Reserved	Reserved	Ushort	2	H+22

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	Doppler	Signal Doppler (Hz)	Float	4	H+24
10	C/No	Carrier to noise density ratio (dB-Hz)	Float	4	H+28
11	Phase std. dev.	Phase error standard deviation (cycles)	Float	4	H+32
12	Lock time	Lock time (s)	Float	4	H+36
13	Unique word bits	Total unique word bits	Ulong	4	H+40
14	Bad unique word bits	Bad unique word bits	Ulong	4	H+44
15	Bad unique words	Bad unique words	Ulong	4	H+48
16	Viterbi symbols	Total Viterbi symbols	Ulong	4	H+52
17	Corrected Viterbi	Corrected Viterbi symbols	Ulong	4	H+56
18	Bit error rate	Estimated pre-Viterbi Bit Error Rate (BER)	Float	4	H+60
19	Next entry offset = H + 4 + (#entries x 60)				
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#entries x 60)
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 133: L-Band Signal Tracking Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x0001	Tracking State	0 = Searching, 1 = Pull-in, 2 = Tracking, 3 = Idle
	1	0x0002		
	2	0x0004	Reserved	
	3	0x0008		

Nibble	Bit	Mask	Description	Range Value
N1	4	0x0010		
	5	0x0020		
	6	0x0040	Bit Timing Lock	0 = Not Locked, 1 = Locked
	7	0x0080	Phase Locked	0 = Not Locked, 1 = Locked
N2	8	0x0100	DC Offset Unlocked	0 = Good, 1 = Warning
	9	0x0200	AGC Unlocked	0 = Good, 1 = Warning
	10	0x0400	Reserved	
	11	0x0800		
N3	12	0x1000		
	13	0x2000		
	14	0x4000		
	15	0x8000	Error	0 = Good, 1 = Error

## 3.93 LOGLIST

### *List of system logs*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists which messages are currently being logged to each port and when. The following tables show the binary and ASCII output. See also the **RXCONFIG** log on page 876 for a list of current command settings.

**Message ID: 5**

**Log Type: Polled**

**Recommended Input:**

```
log loglista once
```

**Abbreviated ASCII Example:**

```
<LOGLIST USB1 0 53.0 FINESTEERING 2209 508498.000 12000020 c00c 16809
< 32
<
< COM1 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< COM2 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< COM3 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< FILE RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< USB1 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< USB1 ITDETECTSTATUSA ONCHANGED 0.000000 0.000000 NOHOLD
< USB1_30 RXCONFIGB ONCHANGED 0.000000 0.000000 HOLD
< USB1_30 SATVIS2B ONCHANGED 0.000000 0.000000 HOLD
< USB1_30 AVEPOSB ONCHANGED 0.000000 0.000000 HOLD
< USB1_30 RXSTATUSB ONTIME 1.000000 0.000000 HOLD
< USB1_30 INTERFACEMODEB ONCHANGED 0.000000 0.000000 HOLD
< USB1_30 ITDETECTSTATUSB ONTIME 1.000000 0.000000 HOLD
< USB1_30 FILESTATUSB ONTIME 1.000000 0.000000 HOLD
< USB1_30 PORTSTATSB ONTIME 1.000000 0.000000 HOLD
< USB1_30 INSPVAXB ONTIME 1.000000 0.000000 HOLD
< USB1_30 RXSTATUSEVENTB ONCHANGED 0.000000 0.000000 HOLD
< USB1_30 TIMEB ONTIME 1.000000 0.000000 HOLD
< USB1_30 BESTPOSB ONTIME 1.000000 0.000000 HOLD
< USB1_30 HWMONITORB ONTIME 1.000000 0.000000 HOLD
< USB1_30 TRACKSTATB ONTIME 1.000000 0.000000 HOLD
< USB1_30 LBANDTRACKSTATB ONTIME 1.000000 0.000000 HOLD
< USB1_30 INSPOXB ONTIME 1.000000 0.000000 HOLD
< USB1_30 MODELFEATURESB ONCHANGED 0.000000 0.000000 HOLD
< USB2 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< USB3 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM1 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM2 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM3 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM4 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM5 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM6 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
< ICOM7 RXSTATUSEVENTA ONNEW 0.000000 0.000000 HOLD
```



Do not use undocumented logs or commands. Doing so may produce errors and void your warranty.

### 3.93.1 Binary

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LOGLIST (binary) header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#logs	Number of messages to follow, maximum = 80	Ulong	4	H
3	port	Output port, see <i>Table 5: Detailed Port Identifier</i> on page 40	Enum	4	H+4
4	message	Message ID of the log	Ushort	2	H+8
5	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit (see <i>Message Responses</i> on page 49) 0 = Original Message 1 = Response Message	Char	1	H+10
6	Reserved		Char	1	H+11
7	trigger	0 = ONNEW 1 = ONCHANGED 2 = ONTIME 3 = ONNEXT 4 = ONCE 5 = ONMARK	Enum	4	H+12
8	period	Log period for ONTIME	Double	8	H+16
9	offset	Offset for period (ONTIME trigger)	Double	8	H+24
10	hold	0 = NOHOLD 1 = HOLD	Enum	4	H+32
11...	Next log offset = H + 4 + (#logs x 32)				
variable	xxxx	32-bit CRC	Hex	4	H+4+ (#logs x 32)



**3.93.2 ASCII**

Field	Field type	Description	Format
1	Log header	LOGLIST (ASCII) header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	
2	#logs	Number of messages to follow, maximum = 80	Long
3	port	Output port, see <i>Table 5: Detailed Port Identifier</i> on page 40	Enum
4	message	Message name of log with no suffix for abbreviated ASCII, an A suffix for ASCII and a B suffix for binary	Char [ ]
5	trigger	ONNEW ONCHANGED ONTIME ONNEXT ONCE ONMARK	Enum
6	period	Log period for ONTIME	Double
7	offset	Offset for period (ONTIME trigger)	Double
8	hold	NOHOLD HOLD	Enum
9...	Next log		
variable	xxxx	32-bit CRC	Hex
variable	[CR][LF]	Sentence terminator	-

## 3.94 LUAFILELIST

### List available Lua scripts

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This sequenced log informs the user of the available scripts, obtained from the ISO loaded onto the receiver. The size of the file, last change date in `yyyymmdd` format, last change time in `hhmmss` format, and path to the files are printed as well.

**Message ID:** 2151

**Log Type:** Polled

**Recommended Input:**

```
LOG LUAFILELIST
```

### Abbreviated ASCII Example:

```
[COM1]<LUAFILELIST COM1 6 89.5 UNKNOWN 0 4.000 02444020 b447 14635
< 0 20180202 151403 "/lua/uppercase.lua"
<LUAFILELIST COM1 5 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 2706 20180129 152042 "/lua/debugloop.lua"
<LUAFILELIST COM1 4 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 4692 20180202 110107 "/lua/parsetime.lua"
<LUAFILELIST COM1 3 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 4764 20180205 105415 "/lua/scom_rx.lua"
<LUAFILELIST COM1 2 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 3728 20180202 104830 "/lua/scomtunnel.lua"
<LUAFILELIST COM1 1 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 3044 20180201 144849 "/lua/scriptargs.lua"
<LUAFILELIST COM1 0 90.5 UNKNOWN 0 4.000 02444020 b447 14635
< 2337 20180129 155140 "/lua/sendtocom2.lua"
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LUAFILELIST header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Size	File size (in Bytes)	Ulong	4	H
3	Date	Last change date When viewed as a string, the date is of the form <code>YYYYMMDD</code> . So, numerically, the date is $(\text{Year} * 10000) + (\text{Month} * 100) + (\text{Day})$ .	Ulong	4	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	Time	Last change time When viewed as a string, the time is HHMMSS. So, numerically, the time is (Hour * 10000) + (Minute * 100) + (Second).	Ulong	4	H+8
5	Path	The path to the Lua script The maximum length of this string is 256 bytes.	String	Variable	H+12

## 3.95 LUAFILESYSTEMSTATUS

### *Query mount status of Lua scripts*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this log to query the mount status of the ISO image that contains the Lua scripts loaded on to the receiver.

**Message ID:** 2150

**Log Type:** Asynch

**Recommended Input:**

```
LOG LUAFILESYSTEMSTATUS
```

**Abbreviated ASCII Example:**

```
<LUAFILESYSTEMSTATUS COM1 0 90.0 UNKNOWN 0 0.204 02444020 b8f8 14635
< MOUNTED ""
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LUAFILESYSTEMSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Status	The status of the file system. See <i>Table 134: File System Status</i> below.	Enum	4	H
3	Error	String that indicates the error message if mounting fails The maximum length of this string is 52 bytes.	String	Variable	H+4

**Table 134: File System Status**

Value	Description
1	UNMOUNTED
2	MOUNTED
3	BUSY
4	ERROR
5	UNMOUNTING
6	MOUNTING

## 3.96 LUAOUTPUT

### *Output stderr and stdout from the Lua interpreter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this log to output `stderr` and `stdout` messages from the Lua interpreter.



The **LUAOUTPUT** log uses a short header. See *Description of ASCII and Binary Logs with Short Headers* on page 48 for more information.

**Message ID:** 2240

**Log Type:** Asynch

**Recommended Input:**

```
LOG LUAOUTPUT ONNEW
```

**Abbreviated ASCII Example:**

```
<LUAOUTPUT 0 346044.929
< 1 0 STDOUT "Lua 5.3.4 Copyright (C) 1994-2017 Lua.org, PUC-Rio"
<LUAOUTPUT 0 346044.987
< 2 0 STDOUT "> "
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	LUAOUTPUT header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Sequence Number	Running number of each LUAOUTPUT log produced by the system	Ulong	4	H
3	Executor Number	Lua Executor Number that produced the data	Ulong	4	H+4
4	Data Source	See <i>Table 135: Lua Data Source</i> on the next page	Enum	4	H+8
5	Data	NULL-terminated string containing a single line of data from <code>stderr</code> or <code>stdout</code> . This string is not terminated with a carriage return or line feed. This string contains only printable characters. The maximum length of this string is 128 bytes.	String	Variable	H+12

**Table 135: Lua Data Source**

Binary	ASCII	Description
0	STDOUT	Data is from stdout
1	STDERR	Data is from stderr

## 3.97 LUASTATUS

### *Display status of Lua scripts*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

Use this log to determine which scripts are running on the receiver and whether the scripts have exited or encountered errors.

**Message ID:** 2181

**Log Type:** Collection

**Recommended Input:**

LOG LUASTATUS

**Abbreviated ASCII Example:**

```
[COM1]<LUASTATUS COM1 1 84.5 FINESTEERING 1963 402110.866 02400000 2e18 32768
< 0 "icom_rx.lua 127.0.0.1 3001" EXECUTING
<LUASTATUS COM1 0 84.5 FINESTEERING 1963 402110.866 02400000 2e18 32768
< 1 "" NOT_STARTED
```



The example above is for the projected log output for two executors.

Field	Field Type	Description	Format	Binary Bytes	Binary Format
1	Log header	LUASTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Number	Executor number	Ulong	4	H
3	Script	Script and arguments	String [256]	Variable	H+4
4	Status	Script status. See <i>Table 136: Script Status</i> below.	Enum	4	Variable

**Table 136: Script Status**

Binary	ASCII	Description
0	NOT_STARTED	There is no script running on the executor
1	EXECUTING	The script is running
2	COMPLETED	The script completed successfully
3	SCRIPT_ERROR	The script exited with an error
4	EXECUTOR_ERROR	The script executor encountered an error while attempting to run the script

## 3.98 MARKPOS, MARK2POS, MARK3POS and MARK4POS

### *Position at time of mark input event*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART2

This log contains the estimated position of the antenna when a pulse is detected at a mark input.

- MARKPOS is generated when a pulse occurs on the MK1I (EVENT\_IN1) input.
- MARK2POS is generated when a pulse occurs on the MK2I (EVENT\_IN2) input.
- MARK3POS is generated when a pulse occurs on the MK3I (EVENT\_IN3) input (OEM7600, OEM7700, OEM7720, PwrPak7, CPT7 and CPT7700 only).
- MARK4POS is generated when a pulse occurs on the MK4I (EVENT\_IN4) input (OEM7600, OEM7700 and OEM7720 only).

Refer to the *Technical Specifications* appendices in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for mark input pulse specifications and the location of the mark input pins.

The position at the mark input pulse is extrapolated using the last valid position and velocities. The latched time of mark impulse is in GPS reference weeks and seconds into the week. The resolution of the latched time is 10 ns. See also the notes on MARKPOS in the **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** log on page 699.



The position for the **MARKPOS**, **MARK2POS**, **MARK3POS** and **MARK4POS** logs are reported in the user selected datum. See the **DATUM** command on page 133 for more details.

**Message ID:** 181 (MARKPOS)  
615 (MARK2POS)  
1738 (MARK3POS)  
1739 (MARK4POS)

**Log Type:** Asynch

**Recommended Input:**

```
log markposa onnew
```



1. Use the ONNEW trigger with the MARKxTIME or MARKxPOS logs.
2. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARK1TIME, MARK2TIME, MARKPOS or MARK2POS logs.
3. Once the 1PPS signal has hit a rising edge, for both MARKxPOS and MARKxTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the **MARK1TIME**, **MARK2TIME**, **MARK3TIME** and **MARK4TIME** log on page 699.

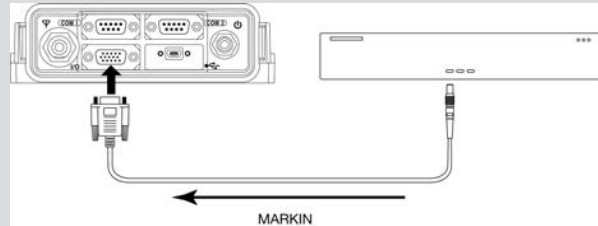


**Abbreviated ASCII Example:**

```
<MARKPOS COM1 0 89.0 FINESTEERING 1670 413138.000 02000020 c223 42770
SOL_COMPUTED SINGLE 51.11289233689 -114.02932170726 1018.9653 1049.4915 BUKIT
1.9372 1.1981 4.0909 "" 0.000 0.000 19 18 18 18 0 06 0 33
```



Consider the case where you have a user point device such as video equipment. Connect the device to the receiver's I/O port using a cable that is compatible to both the receiver and the device. Refer to your device's documentation for information about connectors and cables. The arrow along the cable in the figure below indicates a MARKIN pulse, from the user device on the right to the receiver I/O port.



Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MARKPOS / MARK2POS / MARK3POS / MARK4POS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.</p> </div>	Float	4	H+32

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.99 MARK1TIME, MARK2TIME, MARK3TIME and MARK4TIME

### *Time of mark input event*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART2

This log contains the time of the leading edge of the detected mark input pulse.

- MARK1TIME is generated when a pulse occurs on a MK1I (EVENT\_IN1) input
- MARK2TIME is generated when a pulse occurs on a MK2I (EVENT\_IN2) input
- MARK3TIME is generated when a pulse occurs on a MK3I (EVENT\_IN3) input (OEM7600, OEM7700, OEM7720, PwrPak7, CPT7 and CPT7700 only)
- MARK4TIME is generated when a pulse occurs on a MK4I (EVENT\_IN4) input (OEM7600, OEM7700 and OEM7720 only)



The MARKTIME log (message ID 231) is deprecated. Use the MARK1TIME log as a replacement.

Refer to the *Technical Specifications* appendix in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#) for mark input pulse specifications and the location of the mark input pins. The resolution of this measurement is 10 ns.



1. Use the ONNEW trigger with the MARKxTIME or the MARKxPOS logs.
2. Only the MARKxPOS logs, MARKxTIME logs and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.
3. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARK1TIME, MARK2TIME, MARKPOS or MARK2POS logs.
4. Once the 1PPS signal has hit a rising edge, for both MARKxPOS and MARKxTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the **MARKPOS, MARK2POS, MARK3POS and MARK4POS** log on page 696.

**Message ID:** 1130 (MARK1TIME)  
616 (MARK2TIME)  
1075 (MARK3TIME)  
1076 (MARK4TIME)

**Log Type:** Asynch

**Recommended Input:**

```
log mark1timea onnew
```

**ASCII Example:**

```
#MARK1TIMEA,COM1,0,77.5,FINESTEERING,1358,422621.000,02000000,292e,2214;1358,42
2621.000000500,-1.398163614e-08,7.812745577e-08,-14.000000002,VALID*d8502226
```




These logs allow you to measure the time when events are occurring in other devices (such as a video recorder). See also the **MARKCONTROL** command on page 261.



GPS reference time is the receiver's estimate of the true GPS system time. GPS reference time can be found in the header of the log. The relationship between GPS reference time and true GPS system time is:

GPS system time = GPS reference time - offset

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MARK1TIME / MARK2TIME / MARK3TIME / MARK4TIME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	week	GPS reference week number	Long	4	H
3	seconds	Seconds into the week as measured from the receiver clock, coincident with the time of electrical closure on the Mark Input port	Double	8	H+4
4	offset	Receiver clock offset, in seconds. A positive offset implies that the receiver clock is ahead of GPS system time. To derive GPS system time, use the following formula: GPS system time = GPS reference time - (offset) Where GPS reference time can be obtained from the log header	Double	8	H+12
5	offset std	Standard deviation of receiver clock offset (s)	Double	8	H+20

Field	Field type	Description	Format	Binary Bytes	Binary Offset
6	utc offset	<p>This field represents the offset of GPS system time from UTC time (s), computed using almanac parameters. UTC time is GPS reference time plus the current UTC offset minus the receiver clock offset.</p> <p>UTC time = GPS reference time - offset + UTC offset</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  0 indicates that UTC time is unknown because there is no almanac available in order to acquire the UTC offset. </div>	Double	8	H+28
7	status	Clock model status, see <i>Table 101: Clock Model Status</i> on page 549	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.100 MASTERPOS

### *Master Position using ALIGN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

ALIGN generates distance and bearing information between a Master and Rover receiver. This log outputs the position information of the master when using the ALIGN feature. This log can be output from both Y and Z ALIGN models and can be output at both Master and Rover ends.



You must have an ALIGN capable receiver to use this log.



1. ALIGN is useful for obtaining the relative directional heading of a vessel/body, separation heading between two vessels/bodies or heading information with moving base and pointing applications.
2. The log can be output at both Y and Z model Rover if it is receiving the RTCAREFEXT or NovAtelXRef message from the Master. The log can be output at any Master if the Master is receiving HEADINGEXTB or HEADINGEXT2B from the Rover.
3. MASTERPOS logging is dictated by the output frequency of the RTCAREFEXT or NovAtelXRef output frequency.

**Message ID:** 1051

**Log Type:** Asynch

**Recommended Input:**

```
log masterposa onchanged
```

**ASCII Example:**

```
#MASTERPOSA,COM1,0,21.5,FINESTEERING,1544,340322.000,02000008,5009,4655;  
SOL_COMPUTED,NARROW_INT,51.11604599076,-114.03855412002,1055.7756,16.9000,  
WGS84,0.0090,0.0086,0.0143,"AAAA",0.0,0.0,13,13,13,12,0,0,0,0*a72e8d3f
```



Asynchronous logs, such as MASTERPOS, should only be logged ONCHANGED or ONNEW otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MASTERPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	sol stat	Solution Status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position Type see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	lat	Master WGS84 Latitude in degrees	Double	8	H+8
5	long	Master WGS84 Longitude in degrees	Double	8	H+16
6	hgt	Master MSL Height in metres	Double	8	H+24
7	undulation	Undulation in metres	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER (default = WGS84)	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation in metres	Float	4	H+40
10	long $\sigma$	Longitude standard deviation in metres	Float	4	H+44
11	hgt $\sigma$	Height standard deviation in metres	Float	4	H+48
12	stn id	Receiver ID can be set using the <b>DGPSTXID</b> command (see page 138)	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14			Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#obs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#multi	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	sol source	Solution source (see <i>Table 123: Solution Source</i> on page 648)	Hex	1	H+68
20	Reserved		Uchar	1	H+69
21			Uchar	1	H+70
22			Uchar	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	HEX	1	H+72
24	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 3.101 MATCHEDPOS

### *Matched RTK position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log represents positions that have been computed from time matched base and rover observations. There is no base extrapolation error on these positions because they are based on buffered measurements, that is, they lag real time by some amount depending on the latency of the data link. If the rover receiver has not been enabled to accept RTK differential data or is not actually receiving data leading to a valid solution, this is shown in fields #2 (*sol status*) and #3 (*pos type*).

This log provides the best accuracy in static operation. For lower latency in kinematic operation, see the **RTKPOS** log (see page 866) or **BESTPOS** log (see page 520). The data in the logs changes only when a base observation (RTCMv3) changes.

A good message trigger for this log is onchanged. Then, only positions related to unique base station messages are produced and the existence of this log indicates a successful link to the base.



Asynchronous logs, such as MATCHEDPOS, should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.



The RTK system in the receiver provides two kinds of position solutions. The Matched RTK position is computed with buffered observations, so there is no error due to the extrapolation of base station measurements. This provides the highest accuracy solution possible at the expense of some latency which is affected primarily by the speed of the differential data link. The **MATCHEDPOS** log contains the matched RTK solution and can be generated for each processed set of base station observations.

The Low-Latency RTK position is computed from the latest local observations and extrapolated base station observations. This supplies a valid RTK position with the lowest latency possible at the expense of some accuracy. The degradation in accuracy is reflected in the standard deviation and is summarized in *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss). The amount of time that the base station observations are extrapolated is in the "differential age" field of the position log. The Low-Latency RTK system extrapolates for 60 seconds. The **RTKPOS** log (see page 866) contains the Low-Latency RTK position when valid, and an "invalid" status when a Low-Latency RTK solution could not be computed. The **BESTPOS** log (see page 520) contains either the low-latency RTK, PPP or pseudorange-based position, whichever has the smallest standard deviation.

**Message ID:** 96

**Log Type:** Asynch

**Recommended Input:**

log matchedposa onchanged

**ASCII Example:**

```
#MATCHEDPOSA,USB1,0,66.5,FINESTEERING,2211,232751.000,02000020,8b46,16809;
SOL_COMPUTED,NARROW_INT,51.15043862995,-114.03067350591,1098.9166,-17.0000,
WGS84,0.0070,0.0063,0.0114,"K250",0.000,0.000,37,16,16,15,00,01,00,33*755bb833
```





Measurement precision is different from the position computation precision. Measurement precision is a value that shows how accurately the actual code or carrier phase is measured by the GNSS receiver. Position precision is a value that shows the accuracy of the position computation made from the code and/or carrier phase measurements. The P-code L2 measurement precision is not as good as the C/A measurement precision because the NovAtel GNSS receiver is a civilian grade GPS device and does not have direct access to the decrypted military L2 P(Y) code. This means that NovAtel's semi-codeless P-code L2 measurements are noisier than the civilian band C/A code measurements. Refer to the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#) or [SMART7 Installation and Operation User Manual](#) for the technical specification of the receiver.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MATCHEDPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m)  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">  When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84. </div>	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48

Field	Field type	Description	Format	Binary Bytes	Binary Offset
12	stn id	Base station ID	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14			Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.102 MATCHEDSATS

### *Satellites used in MATCHEDPOS solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log lists the used and unused satellites for the corresponding MATCHEDPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

**Message ID:** 1176

**Log Type:** Asynch

**Recommended Input:**

```
log matchedsatsa onchanged
```

**Abbreviated ASCII Example:**

```
<MATCHEDSATS COM1 0 62.5 FINESTEERING 2220 503122.000 02000000 b555 16807
< 36
< GPS 18 GOOD 00000007
< GPS 16 GOOD 00000003
< GPS 5 GOOD 00000003
< GPS 27 GOOD 00000007
< GPS 29 GOOD 00000003
< GPS 10 GOOD 00000007
< GPS 23 GOOD 00000007
< GPS 13 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 26 GOOD 00000007
< QZSS 194 GOOD 00000007
< GLONASS 18-3 GOOD 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 15 GOOD 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 24+2 GOOD 00000003
< GLONASS 23+3 GOOD 00000001
< GLONASS 17+4 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
< GALILEO 15 GOOD 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 3 GOOD 0000000f
< GALILEO 36 GOOD 0000000f
< GALILEO 5 GOOD 0000000f
< GALILEO 21 GOOD 0000000f
< GALILEO 30 GOOD 0000000f
< GALILEO 9 GOOD 0000000f
< BEIDOU 28 GOOD 0000001d
< BEIDOU 33 GOOD 0000001d
< BEIDOU 42 GOOD 0000001d
< BEIDOU 27 GOOD 0000001d
< BEIDOU 43 GOOD 0000001d
< BEIDOU 41 GOOD 0000001d
< BEIDOU 40 GOOD 0000001d
< BEIDOU 11 GOOD 00000005
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MATCHEDSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See <i>Table 124: Satellite System</i> on page 654	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status ( <i>Table 92: Observation Statuses</i> on page 529)	Enum	4	H+12
6	Signal mask	See <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H + 4 + (#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	1	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### 3.103 MATCHEDXYZ

#### *Matched RTK Cartesian position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the receiver's matched position in ECEF coordinates. It represents positions that have been computed from time matched base and rover observations. There is no base station extrapolation error on these positions because they are based on buffered measurements, that is, they lag real time, by some amount, depending on the latency of the data link. If the rover receiver has not been enabled to accept RTK differential data or is not actually receiving data leading to a valid solution, this is reflected by the code shown in field #2 (solution status) and #3 (position type). See *Figure 12: The WGS84 ECEF Coordinate System* on page 540 for a definition of the ECEF coordinates.

This log provides the best accuracy in static operation. For lower latency in kinematic operation, see the **BESTXYZ** log (see page 538) or **RTKXYZ** log (see page 873). The data in the logs changes only when a base observation (RTCMv3) changes.

The time stamp in the header is the time of the matched observations that the computed position is based on and not the current time.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID:** 242

**Log Type:** Asynch

**Recommended Input:**

```
log matchedxyza onchanged
```



Asynchronous logs, such as MATCHEDXYZ, should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

**ASCII Example:**

```
#MATCHEDXYZA,USB1,0,66.5,FINESTEERING,2211,233514.000,02000020,1d7b,16809;  
SOL_COMPUTED,NARROW_INT,-1632848.2280,-3662159.5450,4944902.5909,0.0070,  
0.0077,0.0104,"K250",36,16,16,15,0,01,00,33*e2f7457b
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MATCHEDXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0

Field	Field type	Description	Format	Binary Bytes	Binary Offset
2	P-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40
10	stn ID	Base station ID	Char[4]	4	H+44
11	#SVs	Number of satellites tracked	Uchar	1	H+48
12	#solnSVs	Number of satellites used in solution	Uchar	1	H+49
13	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+50
14	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+51
15	Reserved		Char	1	H+52
16	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+53
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+54
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+55
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.104 MODELFEATURES

### *States features available for current loaded model*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The MODELFEATURES log states the features available for the current loaded model.

To see which satellite systems are available for the current model, use the **CHANCONFIGLIST** log (see page 543).

Most features have a boolean state: authorized or unauthorized. However, some have more complex licensed states with varying degrees of capability.



This log is best viewed in Abbreviated ASCII.

**Message ID: 1329**

**Log Type: Polled**

**Recommended Input:**

```
log modelfeatures once
```

**Abbreviated ASCII Example:**

```
<MODELFEATURES USB1 0 53.5 FINESTEERING 2209 508953.938 12000020 141a 16809
< 26
< 100HZ MAX_MSR_RATE
< 100HZ MAX_POS_RATE
< SINGLE ANTENNA
< COMMERCIAL_MEMS IMU
< STANDARD INS
< AUTHORIZED MEAS_OUTPUT
< AUTHORIZED DGPS_TX
< AUTHORIZED RTK_TX
< AUTHORIZED RTK_FLOAT
< AUTHORIZED RTK_FIXED
< AUTHORIZED PPP
< AUTHORIZED LOW_END_POSITIONING
< AUTHORIZED RAIM
< AUTHORIZED INTERFERENCE_MITIGATION
< AUTHORIZED NTRIP
< AUTHORIZED INS_PLUS_PROFILES
< AUTHORIZED HEAVE
< AUTHORIZED DEBUG
< UNAUTHORIZED ALIGN_HEADING
< UNAUTHORIZED ALIGN_RELATIVE_POS
< UNAUTHORIZED API
< UNAUTHORIZED SPRINKLER
< UNAUTHORIZED RTKASSIST
< UNAUTHORIZED SCINTILLATION
< UNAUTHORIZED GENERIC_IMU
< UNAUTHORIZED RELATIVE_INS
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MODELFEATURES header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# Feature	Number of features in list	Ulong	4	H
3	Feature Status	Licensing status of feature See <i>Table 137: Feature Status</i> below	Enum	4	H+4
4	Feature Type	Type of feature See <i>Table 138: Feature Type</i> on the next page	Enum	4	H+8
5...	Next feature = H+4+(# Feature x 8)				
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+(# Feature x 8)
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 137: Feature Status

Value	Name	Description
0	AUTHORIZED	The feature is authorized
1	UNAUTHORIZED	The feature is unauthorized
2	0Hz	Disables output of POS logs
6	20Hz	Maximum logging rate for POS or MSR logs is 20 Hz
8	100Hz	Maximum logging rate for POS or MSR logs is 100 Hz
9	RATE_INVALID	Option bits don't correspond to a valid rate
15	STANDARD	SPAN Standard Model
20	COMMERCIAL_MEMS	IMU Grade-Commercial MEMS
21	TACTICAL	IMU Grade-Tactical
22	HIGH_GRADE_TACTICAL	IMU Grade-High Grade Tactical
23	NAVIGATION	IMU Grade-Navigation
25	SINGLE	Single antenna
26	DUAL	Dual antenna
30	LITE	SPAN Lite Model
33	CONSUMER_MEMS	IMU Grade-Consumer MEMS



Value	Name	Description
37	RADIO_TX	External Radio Control
38	GLIDE	GLIDE
39	TILT	Tilt

Table 138: Feature Type

Value	Name	Description
0	MAX_MSR_RATE	Maximum measurement logging rate
1	MAX_POS_RATE	Maximum position logging rate
3	MEAS_OUTPUT	Output of raw measurements (phase and pseudorange)
4	DGPS_TX	Transmission of DGPS (non RTK) corrections
5	RTK_TX	Transmission of RTK corrections
6	RTK_FLOAT	RTK float positioning
7	RTK_FIXED	RTK fixed positioning
8	RAIM	Extended RAIM
9	LOW_END_POSITIONING	GLIDE and TerraStar-L positioning
10	ALIGN_HEADING	Heading
11	ALIGN_RELATIVE_POS	Heading and Relative Positioning
12	API	Lua Scripted User Interface (formerly User Application API)
15	NTRIP	NTRIP Server/Client
19	PPP	TerraStar-C, TerraStar-C PRO, or TerraStar-X positioning
20	SCINTILLATION	Scintillation
22	INS	Inertial (SPAN)
23	IMU	IMU Grade
26	INTERFERENCE_MITIGATION	Interference Mitigation
28	ANTENNA	Number of antenna enabled on the receiver
29	GENERIC_IMU	SPAN Generic IMU Interface
30	INS_PLUS_PROFILES	SPAN Plus Profiles
31	HEAVE	SPAN Heave Option

Value	Name	Description
32	RELATIVE_INS	SPAN Relative INS
33	PRODUCT_FEATURE	Additional product feature has been authorized. See the <b>Feature Status</b> field for details.
34	SPRINKLER	Sprinkler
999	MODEL_INVALID	If a bad model is loaded, MODELFEATURES will contain one entry: MODEL_INVALID STATUS_INVALID

## 3.105 NAVICALMANAC

### *Decoded NavIC Almanac*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the decoded NavIC almanac parameters from NavIC navigation messages. Multiple messages are transmitted, one for each satellite ID with data.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

**Message ID:** 2122

**Log Type:** Asynch

**Recommended Input:**

```
log navicalmanaca onchanged
```

#### **ASCII Example:**

```
#NAVICALMANACA, COM1, 4, 69.5, SATTIME, 1943, 158160.000, 02000020, fb6e, 32768; 919,
0.001982212, 86400, 0.075264303, 8.457495146e-10, 6493.383789062, 1.327344662,
2.996060720, 2.542881375, -0.000580788, 7.275957614e-12, 6, 0, 0, 5*05cfbc62

#NAVICALMANACA, COM1, 3, 69.5, SATTIME, 1943, 156276.000, 02000020, fb6e, 32768; 919,
0.001962662, 0, 0.509411950, 2.742971399e-10, 6493.538574219, 1.844826864,
3.107479183, -3.001633760, -0.000161171, -5.093170330e-11, 4, 0, 0, 7*8fbd9e3a

#NAVICALMANACA, COM1, 2, 69.5, SATTIME, 1943, 158148.000, 02000020, fb6e, 32768; 919,
0.001979351, 86400, 0.499982612, 2.400099974e-10, 6493.359375000, -1.300198895,
-3.061969089, 0.047002130, 0.000025749, -3.637978807e-12, 5, 0, 0, 5*be12ffa2

#NAVICALMANACA, COM1, 1, 69.5, SATTIME, 1943, 157620.000, 02000020, fb6e, 32768; 919,
0.001854897, 86400, 0.509561753, 1.371485699e-10, 6493.388671875, 1.842267109,
3.032190537, 2.385950946, 0.000114441, -5.456968211e-11, 2, 0, 0, 5*b64cf69c

#NAVICALMANACA, COM1, 0, 69.5, SATTIME, 1943, 156804.000, 02000020, fb6e, 32768; 919,
0.000161171, 86400, 0.076541746, 1.142904749e-09, 6493.613281250, 1.349937548,
0.783248119, 0.142653098, 0.000204086, -8.003553376e-11, 7, 0, 0, 7*495808b9
```



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVICALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	WNa	Week number for the almanac since the NavIC system time start epoch (August 22nd 1999)	Ulong	4	H
3	Ecc	Eccentricity (dimensionless)	Double	8	H+4
4	Toa	Time of Almanac (s)	Ulong	4	H+12
5	I0	Inclination angle (radians)	Double	8	H+16
6	OmegaDot	Rate of RAAN (radians/s)	Double	8	H+24
7	RootA	Square root of semi-major axis (sqrt(m))	Double	8	H+32
8	Omega0	Longitude of ascending node (radians)	Double	8	H+40
9	Omega	Argument of perigee (radians)	Double	8	H+48
10	M0	Mean Anomaly (radians)	Double	8	H+56
11	Af0	Clock bias A0 (s)	Double	8	H+64
12	Af1	Clock Drift A1 (s/s)	Double	8	H+72
13	AlmSVID	PRN ID for Almanac	Ulong	4	H+80
14	InterSigCorr	Inter Signal Correction (s)	Ulong	4	H+84
15	Spare		Ulong	4	H+88
16	PRN	Transmitting Satellite Identifier	Ulong	4	H+92
17	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+96
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.106 NAVICEPHEMERIS

### *Decoded NavIC Ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains NavIC ephemeris parameters with the appropriate scaling applied. Multiple messages are transmitted, one for each SV ephemeris collected.

**Message ID:** 2123

**Log Type:** Asynch

**Recommended Input:**

```
log navicephemerisa onchanged
```

**ASCII Example:**

```
#NAVICEPHEMERISA, COM1, 5, 74.0, SATTIME, 1943, 255984.000, 02000020, 01fa, 32768; 2, 919,
1.05838757e-04, -5.63886715e-11, 0.00000000, 0, 252000, -1.86e-09, 3.2829938927e-09,
11, 0, 0, 0, 1.922249794e-06, 1.032650471e-05, 2.011656761e-07, 4.097819328e-08,
-230.9375000, 66.1250000, -5.239503961e-10, 0, 1.900019163, 252000, 1.8492219970e-03,
6493.385761, 1.842761896e+00, 3.027013584, -2.94012247e-09, 5.0965660552e-01,
0, 0, 0*d2f4c9a5
```

```
#NAVICEPHEMERISA, COM1, 4, 74.0, SATTIME, 1943, 255984.000, 02000020, 01fa, 32768; 6, 919,
-5.79587650e-04, 1.02318154e-11, 0.00000000, 1, 252000, -1.86e-09, 8.5817860373e-09,
11, 0, 0, 0, -1.282989979e-05, 2.417713404e-06, 1.974403858e-07, 2.644956112e-07,
-83.3125000, -395.3125000, -5.535944880e-10, 0, 2.050709297, 252000,
1.9699299010e-03, 6493.408867, 1.328589850e+00, 2.996532035, -7.66746224e-09,
7.5298187077e-02, 0, 0, 0*50cdb388
```

...

```
#NAVICEPHEMERISA, COM1, 0, 74.0, SATTIME, 1943, 255984.000, 02000020, 01fa, 32768; 7, 919,
1.90386083e-04, -8.28777047e-11, 0.00000000, 1, 255024, -1.40e-09, 6.3988379659e-09,
252, 0, 0, 0, -8.992850780e-06, -1.732259989e-06, -9.313225746e-08, -2.235174179e-08,
60.1250000, -266.1875000, -3.928735076e-10, 0, -0.445949980, 255024,
2.4348858278e-04, 6493.269802, 1.351327715e+00, 1.099632488, -5.54308803e-09,
7.6573741924e-02, 0, 0, 0*01bf330e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVICEPHEMERIS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	PRN	Satellite Identifier	Ulong	4	H
3	WN	Week number since the NavIC system time start epoch (August 22nd 1999)	Ulong	4	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	Af0	Clock bias (s)	Double	8	H+8
5	Af1	Clock drift (s/s)	Double	8	H+16
6	Af2	Clock drift rate (s/s <sup>2</sup> )	Double	8	H+24
7	URA	SV Accuracy	Ulong	4	H+32
8	toc	Reference time for the satellite clock corrections (s)	Ulong	4	H+36
9	TGD	Total group delay (s)	Double	8	H+40
10	DeltaN	Mean motion difference (radian/s)	Double	8	H+48
11	IODEC	Issue of data ephemeris and clock	Ulong	4	H+56
12	Reserved		Ulong	4	H+60
13	L5 Health	Health status of navigation data on L5 SPS signal 0=OK; 1=bad	Ulong	4	H+64
14	S Health	Health status of navigation data on S SPS signal 0=OK; 1=bad	Ulong	4	H+68
15	Cuc	Amplitude of the cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+72
16	Cus	Amplitude of the sine harmonic correction term to the argument of latitude (radians)	Double	8	H+80
17	Cic	Amplitude of the cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+88
18	Cis	Amplitude of the sine harmonic correction term to the angle of inclination (radians)	Double	8	H+96
19	Crc	Amplitude of the cosine harmonic correction term to the orbit radius (m)	Double	8	H+104
20	Crs	Amplitude of the sine harmonic correction term to the orbit radius (m)	Double	8	H+112
21	IDOT	Rate of inclination angle (radians/s)	Double	8	H+120
22	Spare		Ulong	4	H+128
23	M0	Mean anomaly (radians)	Double	8	H+132
24	toe	Time of ephemeris (s)	Ulong	4	H+140
25	Ecc	Eccentricity (dimensionless)	Double	8	H+144

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
26	RootA	Square root of semi-major axis (sqrt(m))	Double	8	H+152
27	Omega0	Longitude of ascending node (radians)	Double	8	H+160
28	Omega	Argument of perigee (radians)	Double	8	H+168
29	OmegaDot	Rate of RAAN (radians/s)	Double	8	H+176
30	I0	Inclination angle (radians)	Double	8	H+184
31	Spare		Ulong	4	H+192
32	Alert flag	The utilization of navigation data shall be at the users' own risk. 1=Alert; 0=OK	Ulong	4	H+196
33	AutoNav flag	When set to 1, satellite is in AutoNav mode. Satellite broadcasts primary navigation parameters from AutoNav data sets with no uplink from ground for maximum of 7 days	Ulong	4	H+200
34	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+204
35	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.107 NAVICIONO

### *NavIC ionospheric coefficients parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains NavIC ionospheric coefficients parameters. Ionospheric error correction for single frequency (L5) users of the NavIC are provided through a set of eight coefficients. The correction coefficients are:

- 4 Alpha Coefficients ( $\alpha_n$ ;  $n=0,1,2,3$ )

These are the coefficients of the cubic polynomial representing the amplitude of the positive cosine curve in the cosine model approximation of ionospheric delay.

- 4 Beta Coefficients ( $\beta_n$ ;  $n=0,1,2,3$ )

These are the coefficients of the cubic polynomial representing the period of the positive cosine curve in the cosine model approximation of ionospheric delay.

**Message ID:** 2124

**Log Type:** Asynch

**Recommended Input:**

```
log navicionoa onchanged
```

**ASCII Example:**

```
#NAVICIONOA,COM1,0,92.5,SATTIME,1944,166272.000,02000020,56c0,32768;
5,2.980232238769531e-08,3.874301910400390e-07,-2.562999725341796e-06,
-7.510185241699216e-06,558.0,168.0,-2286.0,2286.0,0*2b250bbd
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVICIONO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	PRN	Satellite Identifier of the transmitting NavIC SV	Ulong	4	H
3	Alpha 0	Alpha parameter constant term	Double	8	H+4
4	Alpha 1	Alpha parameter 1st order term	Double	8	H+12
5	Alpha 2	Alpha parameter 2nd order term	Double	8	H+20
6	Alpha 3	Alpha parameter 3rd order term	Double	8	H+28
7	Beta 0	Beta parameter constant term	Double	8	H+36
8	Beta 1	Beta parameter 1st order term	Double	8	H+44
9	Beta 2	Beta parameter 2nd order term	Double	8	H+52
10	Beta 3	Beta parameter 3rd order term	Double	8	H+60



---

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	Spare		Ulong	4	H+68
12	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.108 NAVICRAWSUBFRAME

### Raw NavIC subframe data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the raw NavIC subframe data with parity bits removed. Only subframes that have passed the parity check are output.

**Message ID:** 2105

**Log Type:** Asynch

**Recommended Input:**

```
log navicrawsubframea onchanged
```

### ASCII Example:

```
#NAVICRAWSUBFRAMEA, COM1, 0, 72.5, SATTIME, 1943, 159168.000, 02000020, 76af, 32768;182,
7, 1, 8b19e883971a005bf4880009ab3f400eac0af84f7541befff78018e6d7e1dfacd1*88c2ba19

#NAVICRAWSUBFRAMEA, COM1, 0, 72.5, SATTIME, 1943, 159168.000, 02000020, 76af, 32768;177,
2, 1, 8b19e883970e8fc3f8500009ab3f00087f0af8415e4232800f7fd9eb8650b7b630*c7e27e82

#NAVICRAWSUBFRAMEA, COM1, 0, 72.5, SATTIME, 1943, 159168.000, 02000020, 76af, 32768;181,
6, 1, 8b19e88397b3e73401600009ab3f0012370af84f550327c032800ad1d9da339260*0bb7b256

#NAVICRAWSUBFRAMEA, COM1, 0, 72.5, SATTIME, 1943, 159168.000, 02000020, 76af, 32768;180,
5, 1, 8b19e88397036703ff1c0049ab3fc009b10af84fe7e3773ffd7fd6d8f5fddc4181*f42f59ab
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVICRAWSUBFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Signal channel	Signal channel providing the data	Ulong	4	H
3	PRN	Satellite Identifier of transmitting NavIC SV	Ulong	4	H+4
4	Subframe Id	Subframe ID	Ulong	4	H+8
5	Raw subframe data	Raw subframe data (262 bits). Does not include CRC or Tail bits	Hex[33]	33	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+45
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.109 NAVICSYSCLOCK

### *NavIC clock parameters*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log provides the NavIC system time offset with respect to UTC, UTC (NPLI) and other GNSS times such as GPS, GALILEO, GLONASS.

**Message ID:** 2125

**Log Type:** Asynch

**Recommended Input:**

```
log navicsysclocka onchanged
```

**ASCII Example:**

```
#NAVICSYSCLOCKA,COM1,0,93.0,SATTIME,1944,166320.000,02000020,3dfd,32768;7,
-7.625203579664230e-09,-1.598721155460225e-14,0.000000000000000e+00,18,32508,
920,905,7,18,2,2.149608917534351e-07,-5.151434834260726e-14,
-1.998997755520149e-19,32508,920,0*f6617e67
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVICSYSCLOCK header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite Identifier of the transmitting NavIC SV	Ulong	4	H
3	A0 <sub>utc</sub>	Bias coefficient of the NavIC time scale relative to the UTC time scale (s)	Double	8	H+4
4	A1 <sub>utc</sub>	Drift coefficient of the NavIC time scale relative to the UTC time scale (s/s)	Double	8	H+12
5	A2 <sub>utc</sub>	Drift rate coefficient of the NavIC time scale relative to the UTC time scale (s/s <sup>2</sup> )	Double	8	H+20
6	$\Delta t_{LS}$	Current or past leap second count (s)	Long	4	H+28
7	T <sub>outc</sub>	Time data reference time of week (s)	Ulong	4	H+32
8	WN <sub>outc</sub>	Time data reference week number since the NavIC system time start epoch (August 22, 1999)	Ulong	4	H+36
9	WN <sub>LSF</sub>	Leap second reference week number since the NavIC system time start epoch (August 22, 1999)	Ulong	4	H+40
10	DN	Leap second reference day number	Ulong	4	H+44

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	$\Delta t_{LSF}$	Current or future leap second count (s)	Long	4	H+48
12	GNSSID	Timescale for the time offsets with respect to NavIC (Table 139: GNSS Time Scales below)	Ulong	4	H+52
13	A0	Bias coefficient of the NavIC time scale relative to the GNSS time scale (s)	Double	8	H+56
14	A1	Drift coefficient of the NavIC time scale relative to the GNSS time scale (s/s)	Double	8	H+64
15	A2	Drift rate correction coefficient of the NavIC time scale relative to the GNSS time scale (s/s <sup>2</sup> )	Double	8	H+72
16	Tot	Time data reference time of week (s)	Ulong	4	H+80
17	WNot	Time data reference week number since the NavIC system time start epoch (August 22, 1999)	Ulong	4	H+84
18	Spare		Ulong	4	H+88
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+92
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 139: GNSS Time Scales**

GNSS ID	Time Scale
0	GPS
1	GALILEO
2	GLONASS
3-6	Reserved
7	UTC (NPLI)

## 3.110 NAVIGATE

### *User navigation data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

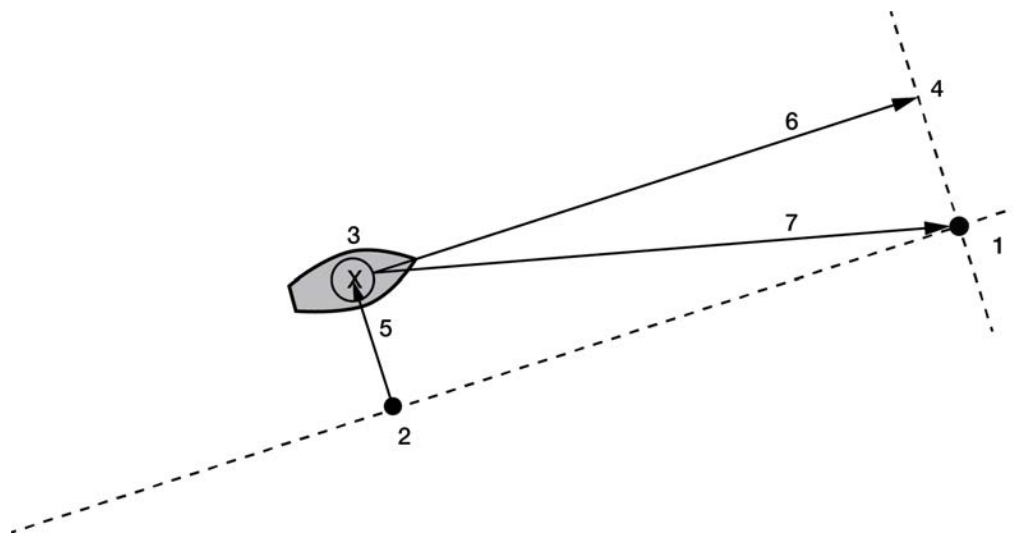
This log reports the status of the waypoint navigation progress. It is used in conjunction with the **SETNAV** command (see page 399).

See the figure below for an illustration of navigation parameters.



The **SETNAV** command (see page 399) must be enabled before valid data will be reported from this log.

**Figure 13: Navigation Parameters**



Reference	Description
1	TO lat-lon
2	X-Track perpendicular reference point
3	Current GPS position
4	A-Track perpendicular reference point
5	X-Track (cross track)
6	A-Track (along track)
7	Distance and bearing from 3 to 1

**Message ID:** 161

**Log Type:** Sync

**Recommended Input:**

```
log navigatea ontime 1
```

**ASCII Example:**

```
#NAVIGATEA,USB1,0,48.0,FINESTEERING,2209,509171.000,12000020,aece,16809;
SOL_COMPUTED,PPP,SOL_COMPUTED,NOVELOCITY,8567.8503,279.156559,-3362.2613,
8005.7921,9999,0.000*1c71003c
```



Use the **NAVIGATE** log in conjunction with the **SETNAV** command (see page 399) to tell you where you currently are with relation to known To and From points. You can find a specific latitude, longitude or height knowing from where you started. For example, a backpacker could use these two commands to program a user supplied graphical display, on a digital GPS compass, to show their progress as they follow a defined route.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	NAVIGATE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	vel status	Velocity status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H+8
5	nav type	Navigation data type (see <i>Table 140: Navigation Data Type</i> on the next page)	Enum	4	H+12
6	distance	Straight line horizontal distance from current position to the destination waypoint, in metres (see <i>Figure 13: Navigation Parameters</i> on the previous page). This value is positive when approaching the waypoint and becomes negative on passing the waypoint.	Double	8	H+16
7	bearing	Direction from the current position to the destination waypoint, in degrees, with respect to True North (or magnetic if corrected for magnetic variation by the <b>MAGVAR</b> command on page 258).	Double	8	H+24
8	along track	Horizontal track distance from the current position to the closest point on the waypoint arrival perpendicular; expressed in metres. This value is positive when approaching the waypoint and becomes negative on passing the waypoint.	Double	8	H+32

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
9	xtrack	The horizontal distance (perpendicular track error), from the vessel's present position to the closest point on the great circle line, that joins the FROM and TO waypoints. If a "track offset" has been entered in the <b>SETNAV</b> command (see page 399), xtrack is the perpendicular error from the "offset track". Xtrack is expressed in metres. Positive values indicate the current position is right of the Track, while negative offset values indicate left.	Double	8	H+40
10	eta week	Estimated GPS reference week number at time of arrival at the "TO" waypoint, along track arrival perpendicular based on current position and speed, in units of GPS reference weeks. If the receiving antenna is moving at a speed of less than 0.1 m/s, in the direction of the destination, the value in this field is "9999".	Ulong	4	H+48
11	eta secs	Estimated GPS seconds into week at time of arrival at destination waypoint along track arrival perpendicular, based on current position and speed, in units of GPS seconds into the week. If the receiving antenna is moving at a speed of less than 0.1 m/s in the direction of the destination, the value in this field is "0.000".	Double	8	H+52
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+60
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 140: Navigation Data Type

Binary	ASCII	Description
0	GOOD	Navigation is good
1	NOVELOCITY	Navigation has no velocity
2	BADNAV	Navigation calculation failed for an unknown reason
3	FROM_TO_SAME	"From" is too close to "To" for computation
4	TOO_CLOSE_TO_TO	Position is too close to "To" for computation
5	ANTIPODAL_WAYPTS	Waypoints are antipodal on surface

### 3.111 NMEA Standard Logs

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains NMEA logs.

<b>GLMLA</b>	<b>GLONASS Almanac Data</b>
<b>GPALM</b>	<b>Almanac Data</b>
<b>GPGGA</b>	<b>Global Position System Fix Data and Undulation</b>
<b>GPGGALONG</b>	<b>GPS Fix Data, Extra Precision and Undulation</b>
<b>GPGLL</b>	<b>Geographic Position</b>
<b>GPGRS</b>	<b>GPS Range Residuals for Each Satellite</b>
<b>GPGSA</b>	<b>GPS DOP on Active Satellites</b>
<b>GPGST</b>	<b>Estimated Error in Position Solution</b>
<b>GPGSV</b>	<b>GPS Satellites in View</b>
<b>GPHDT</b>	<b>NMEA Heading Log (ALIGN)</b>
<b>GPRMB</b>	<b>Navigation Information</b>
<b>GPRMC</b>	<b>GPS Specific Information</b>
<b>GPVTG</b>	<b>Track Made Good and Ground Speed</b>
<b>GPZDA</b>	<b>UTC Time and Date</b>

The NMEA log structures follow format standards as adopted by the National Marine Electronics Association. The reference document used is "Standard For Interfacing Marine Electronic Devices NMEA 0183 Version 3.01". For further information, refer to [www.nmea.org](http://www.nmea.org). The following table contains excerpts from Table 6 of the NMEA Standard which defines the variables for the NMEA logs. The actual format for each parameter is indicated after the description.



See the Note in the **GPGGA** log (see page 610) that applies to all NMEA logs.



1. Spaces may only be used in variable text fields.
2. A negative sign "-" (HEX 2D) is the first character in a Field if the value is negative. The sign is omitted if the value is positive.
3. All data fields are delimited by a comma (,).
4. Null fields are indicated by no data between two commas (,,). Null fields indicate invalid data or no data available.
5. The NMEA Standard requires that message length be limited to 82 characters.

Field Type	Symbol	Definition
<b>Special Format Fields</b>		



Field Type	Symbol	Definition
Status	A	Single character field: A = Yes, Data Valid, Warning Flag Clear V = No, Data Invalid, Warning Flag Set
Latitude	llll.ll	Fixed/Variable length field: degrees minutes.decimal - 2 fixed digits of degrees, 2 fixed digits of mins and a <i>variable</i> number of digits for decimal-fraction of minutes. Leading zeros always included for degrees and minutes to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required
Longitude	yyyyy.yy	Fixed/Variable length field: degrees minutes.decimal - 3 fixed digits of degrees, 2 fixed digits of minutes and a <i>variable</i> number of digits for decimal-fraction of minutes. Leading zeros always included for degrees and minutes to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required
Time	hhmmss.ss	Fixed/Variable length field: hours minutes seconds.decimal - 2 fixed digits of hours, 2 fixed digits of minutes, 2 fixed digits of seconds and <i>variable</i> number of digits for decimal-fraction of seconds. Leading zeros always included for hours, minutes and seconds to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required.
Defined field		Some fields are specified to contain predefined constants, most often alpha characters. Such a field is indicated in this standard by the presence of one or more valid characters. Excluded from the list of allowable characters are the following which are used to indicate field types within this standard: "A", "a", "c", "hh", "hhmmss.ss", "llll.ll", "x", "yyyyy.yy"
<b>Numeric Value Fields</b>		
Variable numbers	x.x	Variable length integer or floating numeric field. Optional leading and trailing zeros. The decimal point and associated decimal-fraction are optional if full resolution is not required (example: 73.10 = 73.1 = 073.1 = 73)
Fixed HEX field	hh____	Fixed length HEX numbers only, MSB on the left
<b>Information Fields</b>		
Variable text	c--c	Variable length valid character field
Fixed alpha field	aa____	Fixed length field of uppercase or lowercase alpha characters
Fixed number field	xx____	Fixed length field of numeric characters
Fixed text field	cc____	Fixed length field of valid characters

## 3.112 NOVATELXOBS

### *NovAtel proprietary RTK correction*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

NOVATELX is a combined observation and reference station message used to transmit RTK corrections. This is a proprietary message and supports sending data for all systems.

**Message ID: 1618**

**Log Type: Synch**

**Recommended Input:**

```
log com2 novatelxobs ontime 1
```

To calculate the size of the NOVATELXOBS messages, use the following formula.

$$\text{Size} = 168 + s * (6 + p * (54 + f*33))$$

where:

s = number of systems (maximum 8)

p = number of PRN per system (maximum 64)

f = number of signals data per PRN – 1 (maximum 10 signals possible)

**Example size calculations:**

- For 2 systems (GPS and GLONASS), 12 PRN per system, and 2 signals per satellite (L1CA, L2PY)
  - Size =  $168 + 2 * (6 + 12 * (54 + 33))$
  - = 2268 bits per second
  - = 284 bytes + NovAtelXHeader (8 bytes)
- For 3 systems (GPS, BEIDOU and GLONASS), 12 PRN per system, and 2 signals per satellite (L1CA, L2PY)
  - Size =  $168 + 3 * (6 + 12 * (54 + 33))$
  - = 3318 bits per second
  - = 415 bytes + NovAtelXHeader (8 bytes)
- For 3 systems (GPS, BEIDOU and GLONASS), 12 PRN per system, and 3 signals per satellite (L1CA, L2PY, L2C)
  - Size =  $168 + 3 * (6 + 12 * (54 + 2*33))$
  - = 4506 bits per second
  - = 564 bytes + NovAtelXHeader (8 bytes)

### 3.113 NOVATELXREF

*NovAtel proprietary reference station message for use in ALIGN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

NOVATELXREF is a proprietary extended reference station message for use in ALIGN configurations only. This message enables the output of the **MASTERPOS** log (see page 702), **ROVERPOS** log (see page 851) and **ALIGNBSLNENU** log (see page 483) on the rover.

**Message ID:** 1620

**Log Type:** Synch

**Recommended Input:**

```
log com2 novatelxref ontime 1
```

## 3.114 OCEANIXINFO

### *Oceanix subscription information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains details on the Oceanix subscription.

**Message ID:** 2159



**Log Type:** Asynch

**Recommended Input:**

```
log oceanixinfoa onchanged
```

**ASCII Example:**

```
#OCEANIXINFOA,USB1,0,53.0,FINESTEERING,2209,504117.071,12000020,9ce8,16809;
"QV579:5420:8616",BUBBLE,80001803,0,0,0,LOCAL_AREA*bb830405
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	OCEANIXINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PAC	Product activation code	Char[16]	16	H
3	Type	Subscription type (see <i>Table 141: Oceanix Subscription Type</i> on the next page)	Enum	4	H+16
4	Subscription permissions	<p>Services permitted by the subscription (see <i>Table 142: Oceanix Subscription Details Mask</i> on the next page)</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">  Bits in the Reserved areas of this field may be set, but the Reserved bits should be ignored. </div> <div style="border: 1px solid black; padding: 5px;">  On activation of an Oceanix Premium subscription, both Oceanix H and Oceanix Premium bits are set so the value of the non-reserved bits will be 0x00000006. </div>	Hex	4	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	Service End Day	Last day of the year for which service is available. Service expires at the end of this UTC day.  For example, if the Service End Year and Day are 2019 and 15, respectively, then the service will expire on January 15, 2019 at 24:00 UTC.	Ulong	4	H+24
6	Service End Year	Year that the subscription ends.	Ulong	4	H+28
7	Reserved		Ulong	4	H+32
8	Region restriction	For region restricted subscriptions, the type of region restriction (see <i>Table 143: Oceanix Region Restriction</i> on the next page)	Enum	4	H+36
9	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+40
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 141: Oceanix Subscription Type**

ASCII	Binary	Description
UNASSIGNED	0	Decoder has not had an assigned operating mode
TERM	1	Term subscription
MODEL	5	Reserved
BUBBLE	100	Receiver is operating in an Oceanix-permitted subscription-free bubble
INCOMPATIBLE_SUBSCRIPTION	104	Subscription is incompatible with this version of firmware

**Table 142: Oceanix Subscription Details Mask**

Bit	Mask	Description
0	0x00000001	Reserved
1	0x00000002	Oceanix H service
2	0x00000004	Oceanix Premium service
3-31	0xFFFFFFFF8	Reserved

**Table 143: Oceanix Region Restriction**

ASCII	Binary	Description
NONE	0	Oceanix operation has no region restrictions
GEOGATED	1	Oceanix operation is limited to land regions. GEOGATED is also the default value reported if there is no subscription
NEARSHORE	3	Oceanix operation is limited to land and near shore (coastal) regions

## 3.115 OCEANIXSTATUS

### *Oceanix decoder and subscription status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains status information for the Oceanix decoder and subscription.

**Message ID:** 2160

**Log Type:** Asynch

**Recommended Input:**

```
log oceanixstatusa onchanged
```

**ASCII Example:**

```
#OCEANIXSTATUSA,USB1,0,53.5,FINESTEERING,2209,509307.069,12000020,049a,16809;
ENABLE,LOCKED,IN_REGION*10589346
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	OCEANIXSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Access	Access status. ENABLE (1) if the subscription is valid. DISABLE (0) otherwise	Enum	4	H
3	Sync state	Decoder data synchronization state (see <i>Table 144: Decoder Data Synchronization State</i> below)	Enum	4	H+4
4	Region restriction status	Region restriction status (see <i>Table 145: Region Restriction Status</i> on the next page)	Enum	4	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 144: Decoder Data Synchronization State**

ASCII	Binary	Description
NO_SIGNAL	0	None of the decoders have received data in the last 30 seconds
SEARCH	1	At least one decoder is receiving data and is searching for the format
LOCKED	2	At lease one decoder has locked onto the format

**Table 145: Region Restriction Status**

<b>ASCII</b>	<b>Binary</b>	<b>Description</b>
UNKNOWN	0	Region restriction status cannot be determined
IN_REGION	1	Receiver is within the permitted region
OUT_OF_REGION	2	Receiver is outside the permitted region



## 3.116 PASSCOM, PASSAUX, PASSUSB, PASSETH1, PASSICOM, PASSNCOM

### *Redirects data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The pass-through logging feature enables the receiver to redirect any ASCII or binary data, input at a specified port, to any specified receiver port. It allows the receiver to perform bi-directional communications with other devices such as a modem, terminal or another receiver. See also the **INTERFACEMODE** command on page 216.

There are many pass through logs: PASSCOM1, PASSCOM2, PASSCOM3, PASSCOM4, PASSCOM5, PASSCOM6, PASSAUX, PASSETH1, PASSICOM1, PASSICOM2, PASSICOM3, PASSICOM4, PASSICOM5, PASSICOM6, PASSICOM7, PASSNCOM1, PASSNCOM2, PASSNCOM3 allow for redirection of data that is arriving at COM1, COM2, COM3, virtual COM1, virtual COM2 or AUX. PASSCOM4 and PASSCOM5 are only available on OEM7600, OEM7700 and OEM7720 receivers. The AUX port is available on some products. PASSUSB1, PASSUSB2, PASSUSB3 are used to redirect data from USB1, USB2 or USB3. PASSETH1 is only available on receivers supporting Ethernet and can be used to redirect data from ETH1.

A pass through log is initiated the same as any other log, that is, log [to-port] [data-type] [trigger]. However, pass-through can be more clearly specified as: log [to-port] [from-port-AB] [onchanged]. Now, the [from-port-AB] field designates the port which accepts data (that is, COM1, COM2, COM3, COM4, COM5, COM6, AUX, USB1, USB2 or USB3) as well as the format in which the data is logged by the [to-port] (A for ASCII or B for Binary).



To pass through data arriving on all ports, use the **PASSTHROUGH** log (see page 742).

When the [from-port-AB] field is suffixed with an [A], all data received by that port is redirected to the [to-port] in ASCII format and logs according to standard NovAtel ASCII format. Therefore, all incoming ASCII data is redirected and output as ASCII data. However, any binary data received is converted to a form of ASCII hexadecimal before it is logged.

When the [from-port-AB] field is suffixed with a [B], all data received by that port is redirected to the [to-port] exactly as it is received. The log header and time tag adhere to standard NovAtel Binary format followed by the pass through data as it was received (ASCII or binary).

Pass through logs are best utilized by setting the [trigger] field as onchanged or onnew.

If the data being injected is ASCII, then the data is grouped together with the following rules:

- blocks of 80 characters
- any block of characters ending in a <CR>
- any block of characters ending in a <LF>
- any block remaining in the receiver code when a timeout occurs (100 ms)

If the data being injected is binary or the port INTERFACEMODE mode is set to GENERIC, then the data is grouped as follows:

- blocks of 80 bytes
- any block remaining in the receiver code when a timeout occurs (100 ms)

If a binary value is encountered in an ASCII output, then the byte is output as a hexadecimal byte preceded by a backslash and an x. For example 0A is output as \x0A. An actual '\ ' in the data is output as \\ . The output counts as one pass through byte although it is four characters.

The first character of each pass-through record is time tagged in GPS reference weeks and seconds.

<b>PASSAUX</b>	<b>Message ID: 690</b>
<b>PASSCCOM1</b>	<b>Message ID: 1893</b>
<b>PASSCCOM2</b>	<b>Message ID: 1894</b>
<b>PASSCCOM3</b>	<b>Message ID: 1895</b>
<b>PASSCCOM4</b>	<b>Message ID: 1930</b>
<b>PASSCCOM5</b>	<b>Message ID: 1937</b>
<b>PASSCCOM6</b>	<b>Message ID: 1938</b>
<b>PASSCOM1</b>	<b>Message ID: 233</b>
<b>PASSCOM2</b>	<b>Message ID: 234</b>
<b>PASSCOM3</b>	<b>Message ID: 235</b>
<b>PASSCOM4</b>	<b>Message ID: 1384</b>
<b>PASSCOM5</b>	<b>Message ID: 1576</b>
<b>PASSCOM6</b>	<b>Message ID: 1577</b>
<b>PASSCOM7</b>	<b>Message ID: 1701</b>
<b>PASSCOM8</b>	<b>Message ID: 1702</b>
<b>PASSCOM9</b>	<b>Message ID: 1703</b>
<b>PASSCOM10</b>	<b>Message ID: 1704</b>
<b>PASSETH1</b>	<b>Message ID: 1209</b>
<b>PASSICOM1</b>	<b>Message ID: 1250</b>
<b>PASSICOM2</b>	<b>Message ID: 1251</b>
<b>PASSICOM3</b>	<b>Message ID: 1252</b>
<b>PASSICOM4</b>	<b>Message ID: 1385</b>
<b>PASSICOM5</b>	<b>Message ID: 2119</b>
<b>PASSICOM6</b>	<b>Message ID: 2120</b>
<b>PASSICOM7</b>	<b>Message ID: 2121</b>
<b>PASSNCOM1</b>	<b>Message ID: 1253</b>
<b>PASSNCOM2</b>	<b>Message ID: 1254</b>
<b>PASSNCOM3</b>	<b>Message ID: 1255</b>
<b>PASSUSB1</b>	<b>Message ID: 607</b>
<b>PASSUSB2</b>	<b>Message ID: 608</b>
<b>PASSUSB3</b>	<b>Message ID: 609</b>

**Log Type: Asynch**

**Recommended Input:**

```
log passcom1a onchanged
```



Asynchronous logs should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

**ASCII Example 1:**

```
#PASSCOM2A,COM1,0,59.5,FINESTEERING,1337,400920.135,02000000,2b46,1984;80,
#BESTPOSA,COM3,0,80.0,FINESTEERING,1337,400920.000,02000000,4ca6,1899;
SOL_COMPUT*f9dfab46
#PASSCOM2A,COM1,0,64.0,FINESTEERING,1337,400920.201,02000000,2b46,1984;80,ED,
SINGLE,51.11636326036,-114.03824210485,1062.6015,-16.2713,WGS84,1.8963,
1.0674*807fd3ca
#PASSCOM2A,COM1,0,53.5,FINESTEERING,1337,400920.856,02000000,2b46,1984;49,,
2.2862,"",0.000,0.000,9,9,0,0,0,0,0*20b24878\x0d\x0a*3eef4220
#PASSCOM1A,COM1,0,53.5,FINESTEERING,1337,400922.463,02000000,13ff,1984;17,
unlog passcom2a\x0d\x0a*ef8d2508
```

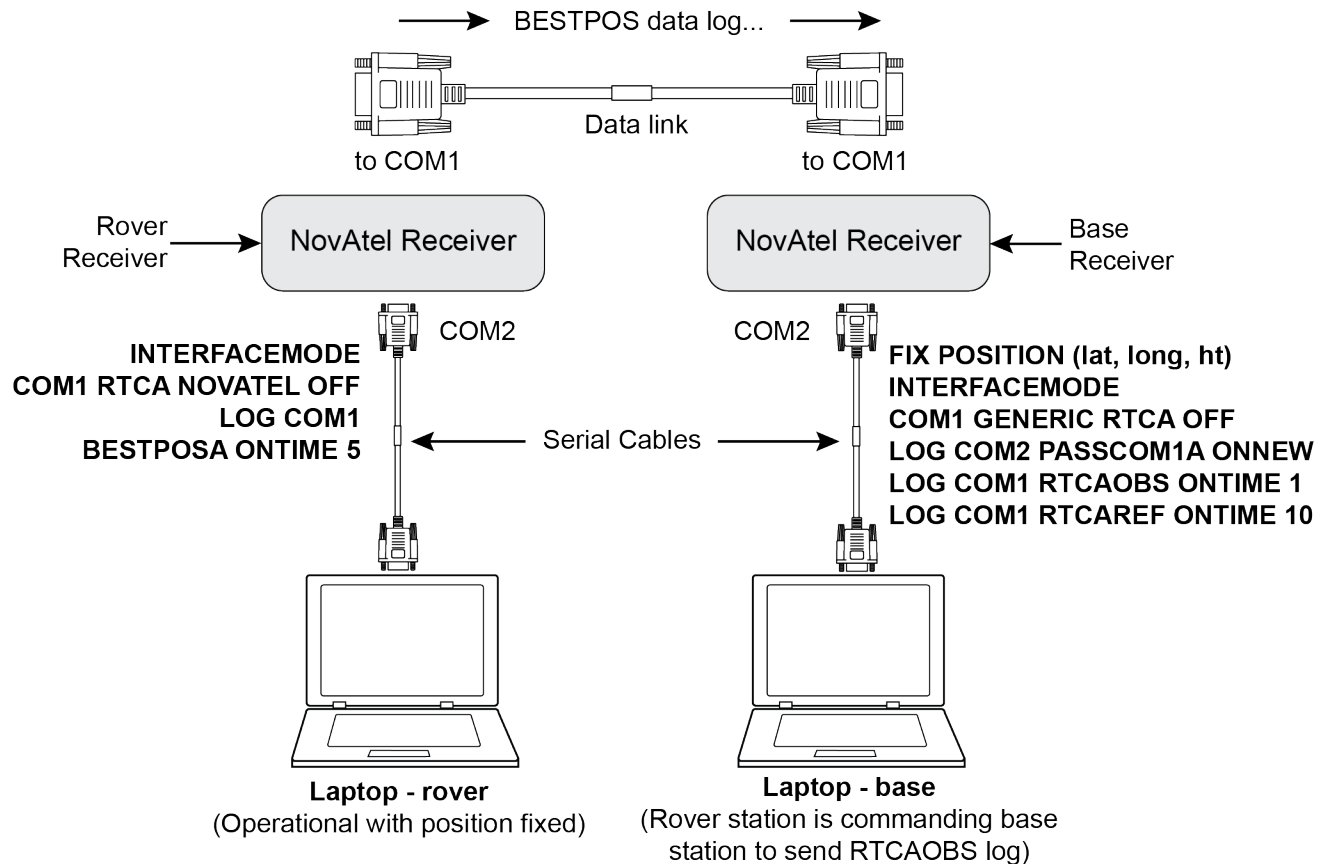
**ASCII Example 2:**

```
#PASSUSB1A,USB1,0,41.5,FINESTEERING,2209,509394.765,12000020,4491,16809;52,\xaa
D\x12\x1c\xb2\x00\x00\xde\x14\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\xde\x00\x00\x00\x06\x00\x00\x00Sendhxdt\x00\x00\x00\
x00\x19W\x10Z*69380120
```



For example, you could connect two OEM7 family receivers together via their COM1 ports such as in the *Figure 14: Pass Through Log Data* on the next page (a rover station to base station scenario). If the rover station is logging BESTPOSA data to the base station, it is possible to use the pass through logs to pass through the received BESTPOSA data to a disk file (let's call it diskfile.log) at the base station host PC hard disk.

Figure 14: Pass Through Log Data



Under default conditions, the two receivers "chatter" back and forth with the Invalid Command Option message (due to the command interpreter in each receiver not recognizing the command prompts of the other receiver). The chattering in turn causes the accepting receiver to transmit new pass through logs with the response data from the other receiver. To avoid the chattering problem, use the **INTERFACEMODE** command (see page 216) on the accepting port to disable error reporting from the receiving port command interpreter.

If the accepting port's error reporting is disabled by INTERFACEMODE, the BESTPOSA data record passes through and creates two records.

The reason that two records are logged from the accepting receiver is the first record was initiated by receipt of the BESTPOSA first terminator <CR>. The second record followed in response to the BESTPOSA second terminator <LF>.

Note the time interval between the first character received and the terminating <LF> can be calculated by differencing the two GPS reference time tags. This pass through feature is useful for time tagging the arrival of external messages. These messages can be any user related data. When using this feature for tagging external events, it is recommended that the rover receiver be disabled from interpreting commands so the receiver does not respond to the messages, using the **INTERFACEMODE** command (see page 216).

If the BESTPOSB binary log data is input to the accepting port (log com2 passcom1a onchanged), the BESTPOSB binary data at the accepting port is converted to a variation of ASCII hexadecimal before it is passed through to com2 port for logging.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PASSCOM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#bytes	Number of bytes to follow	Ulong	4	H
3	data	Message data	Char [80]	80	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#bytes)
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.117 PASSTHROUGH

### *Redirected data from all ports*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log outputs pass through data from all receiver ports. The behavior is the same as the port specific pass through logs described in *PASSCOM*, *PASSAUX*, *PASSUSB*, *PASSETH1*, *PASSICOM*, *PASSNCOM* on page 737.

**Message ID:** 1342

**Log Type:** Asynch

**Recommended Input:**

```
log passthrougha onchanged
```

**ASCII Example:**

```
#PASSTHROUGHHA,USB1,0,53.0,FINESTEERING,2209,509529.759,12000020,5fa3,16809;USB1,52,\xaaD\x12\x1c\xb2\x00\x00\xde\x14\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xde\x00\x00\x00\x06\x00\x00\x00Sendhxdt\x00\x00\x00\x00\x19W\x10Z*72b6022f
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PASSTHROUGH header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Port	See <i>Table 70: COM Port Identifiers</i> on page 385	Enum	4	H
3	#bytes	Number of bytes to follow	Ulong	4	H+4
4	data	Message data	Char[80]	80	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8+#bytes
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.118 PDPDATUMINFO

### *Datum information for the PDP position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **PDPPOS** log (see page 748).



The datum message for TerraStar Correction Services is broadcast periodically. Therefore, it can take several hours, from the start of receiving TerraStar corrections, for the **PPPDATUMINFO** log to reflect the correct datum and co-ordinate epoch for the PPP position solution. Once the PPP datum is known, it will persist through a **RESET** or a power cycle. It will not persist through a **FRESET**.

**Message ID:** 2303

**Log Type:** Asynch

**Recommended Input:**

```
log pdpdatuminfoa onchanged
```

**ASCII Example:**

```
#PDPDATUMINFOA, USB1, 0, 44.0, FINESTEERING, 2209, 510156.500, 12000020, f611, 16809;  
"ECEF", 0, 2022.364, GOOD*e30218ea
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the <b>PDPPOS</b> log.	Char[32]	variable	H
3	epsg_code	EPSG code of datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year) Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation Status. See <i>Table 85: Transformation Status</i> on page 518.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.119 PDPDOP

### *DOP values for the satellites in the PDP solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PDPDOP log contains the Dilution Of Precision (DOP) values for the satellites being used in the PDP solution. The PDP DOPs are updated every 60 seconds or whenever the satellites used in the PDP solution changes.

DOP values are a measure of the solution strength. Essentially, the DOPs reflect the geometry of the satellites used in the solution. Solutions with good counts of well-distributed satellites will have low DOPs and should be accurate and reliable. Solutions with fewer or poorly-distributed satellites will have high DOPs and be less accurate and reliable. As a rough guideline, PDOP values less than 4 imply a solution with reasonable geometry.

There can be many reasons for high DOP values. The most common reason is that there are obstructions limiting satellite visibility. Even if satellites are visible and being tracked they might still not be used in the solution if, for example, they are unhealthy or there are not corrections available for them. The **PDPSATS** log (see page 750) will inform which satellites are being tracked and explain why a tracked satellite is not used in the solution.

The DOPs do not consider that different satellites or signals will be weighted differently in the solution. Therefore, they do not completely reflect the solution quality. Ultimately, the standard deviations reported in the **PDPPOS** log (see page 748) are the best reflection of the solution accuracy.

**Message ID: 1998**

**Log Type: Asynch**

**Recommended Input:**

```
log pdpdopa onchanged
```

**ASCII Example:**

```
#PDPDOPA,USB1,0,52.5,FINESTEERING,2209,510180.000,12000020,3bf3,16809;1.1720,0.7320,0.4070,0.6100,0.4550,5.0,38,5,2,31,9,18,6,29,20,12,25,56,39,46,40,55,52,38,54,36,2,30,34,4,27,15,9,25,34,23,22,44,43,42,37,11,12,9,21*1ccc85ba
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPDOP header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	gdop	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H
3	pdop	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	hdop	Horizontal dilution of precision	Float	4	H+8
5	htdop	Horizontal position and time dilution of precision	Float	4	H+12
6	tdop	GPS time dilution of precision – assumes 3D position is known and only the GPS receiver clock offset is unknown	Float	4	H+16
7	cutoff	GPS elevation cut-off angle	Float	4	H+20
8	#PRN	Number of satellites PRNs to follow	Long	4	H+24
9	PRN	PRN of a satellite used in the position solution	Ulong	4	H+28
10	Next PRN offset = H+28+(#prn x 4)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+28+ (#prn x 4)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.120 PDPDOP2

### *DOP values for the satellites used in the PDP solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PDPDOP2 log contains the Dilution Of Precision (DOP) values for the satellites being used in the PDP solution. This log is similar to the **PDPDOP** log (see page 744) but contains the per-system TDOPs; see the **PDPDOP** log on page 744 for more information on the DOPs.

**Message ID:** 1995

**Log Type:** Asynch

**Recommended Input:**

```
log pdpdop2a onchanged
```

**ASCII Example:**

```
#PDPDOP2A,USB1,0,52.5,FINESTEERING,2209,510180.000,12000020,2488,16809;1.1720,0.7320,0.4070,0.6080,4,GPS,0.4550,GLONASS,0.4650,GALILEO,0.4810,BEIDOU,0.4270*0a9746dd
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPDOP2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	GDOP	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H
3	PDOP	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4
4	HDOP	Horizontal dilution of precision	Float	4	H+8
5	VDOP	Vertical dilution of precision	Float	4	H+12
6	#systems	Number of systems	Ulong	4	H+16
7	system	See <i>Table 146: System Used for Timing</i> on the next page	Enum	4	H+20
8	TDOP	Time dilution of precision	Float	4	H+24
9	Next system offset = H+20+(#systems x 8)				

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20+ (#systems x 8)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 146: System Used for Timing

Binary	ASCII
0	GPS <sup>1</sup>
1	GLONASS
2	GALILEO
3	BEIDOU
4	NAVIC
99	AUTO <sup>2</sup>

<sup>1</sup>GPS setting includes QZSS satellites.

<sup>2</sup>AUTO is used only as a backup system (not available for primary system field).

## 3.121 PDPPOS

### *PDP filter position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PDPPOS log contains the receiver position computed by the receiver with the PDP filter enabled. See also the **PDPFILTER** command on page 286.

**Message ID:** 469


**Log Type:** Synch

**Recommended Input:**

```
log pdpposa ontime 1
```

**ASCII Example:**

```
#PDPPOSA,USB1,0,52.5,FINESTEERING,2209,510261.000,12000020,0649,16809;
SOL_COMPUTED,WAAS,51.15043435551,-114.03068191005,1097.5002,-17.0000,WGS84,
0.7923,0.6986,1.3420,"131",7.000,0.000,44,40,0,0,00,86,7f,37*7dd56614
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m)  <div style="border: 1px solid black; padding: 5px; width: fit-content;">  When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84. </div>	Float	4	H+32

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#sats	Number of satellites tracked	Uchar	1	H+64
16	#sats soln	Number of satellites in the solution	Uchar	1	H+65
17	Reserved		Uchar	1	H+66
18			Uchar	1	H+67
19			Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.122 PDPSATS

### *Satellites used in PDPPPOS solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists the used and unused satellites for the corresponding PDPPPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

**Message ID:** 1234

**Log Type:** Synch

**Recommended Input:**

```
log pdpsatsa ontime 1
```

**Abbreviated ASCII Example:**

```
<PDPSATS USB1 0 51.5 FINESTEERING 2209 510295.000 12000020 74d6 16809
< 45
< GPS 5 GOOD 00000003
< GPS 2 GOOD 00000003
< GPS 31 GOOD 00000003
< GPS 9 GOOD 00000007
< GPS 18 GOOD 00000007
< GPS 6 GOOD 00000007
< GPS 29 GOOD 00000003
< GPS 20 GOOD 00000003
< GPS 12 GOOD 00000003
< GPS 25 GOOD 00000007
< QZSS 194 SUPPLEMENTARY 00000007
< SBAS 135 NOTUSED 00000000
< SBAS 138 NOTUSED 00000000
< SBAS 131 NOTUSED 00000000
< SBAS 133 NOTUSED 00000000
< GLONASS 19+3 GOOD 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 9-2 GOOD 00000003
< GLONASS 3+5 GOOD 00000003
< GLONASS 18-3 GOOD 00000003
< GLONASS 15 GOOD 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 17+4 GOOD 00000003
< GALILEO 36 GOOD 0000000f
< GALILEO 2 GOOD 0000000f
< GALILEO 30 GOOD 0000000f
< GALILEO 34 GOOD 0000000f
< GALILEO 4 GOOD 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 15 GOOD 0000000f
< GALILEO 9 GOOD 0000000f
< GALILEO 5 SUPPLEMENTARY 0000000f
< BEIDOU 25 GOOD 0000001d
< BEIDOU 34 GOOD 0000001d
< BEIDOU 23 GOOD 0000001d
```

```

<      BEIDOU 22 GOOD 0000001d
<      BEIDOU 44 GOOD 0000001d
<      BEIDOU 43 GOOD 0000001d
<      BEIDOU 42 GOOD 0000001d
<      BEIDOU 28 NOEPHEMERIS 00000000
<      BEIDOU 37 GOOD 0000001d
<      BEIDOU 11 GOOD 00000007
<      BEIDOU 12 GOOD 00000007
<      BEIDOU 9  GOOD 00000007
<      BEIDOU 21 GOOD 0000001d

```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See <i>Table 124: Satellite System</i> on page 654	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status (see <i>Table 92: Observation Statuses</i> on page 529)	Enum	4	H+12
6	Status mask	See <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+(#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.123 PDPVEL

### *PDP filter velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PDPVEL log contains the pseudorange velocity computed by the receiver with the PDP filter enabled. See also the **PDPFILTER** command on page 286.

**Message ID:** 470

**Log Type:** Synch

**Recommended Input:**

```
log pdpvela ontime 1
```

**ASCII Example:**

```
#PDPVELA,USB1,0,51.5,FINESTEERING,2209,510354.000,12000020,3303,16809;  
SOL_COMPUTED,WAAS,0.250,5.000,0.0036,356.812176,-0.0052,0*6e811151
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	vel type	Velocity type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+8
5	age	Differential age in seconds	Float	4	H+12
6	hor spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	height	Height in metres where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.124 PDPXYZ

### *PDP filter Cartesian position and velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PDPXYZ log contains the Cartesian position in X, Y and Z coordinates as computed by the receiver with the PDP filter enabled. See also the **PDPFILTER** command on page 286.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID:** 471

**Log Type:** Synch

**Recommended Input:**

```
log pdpxyza ontime 1
```

**ASCII Example:**

```
#PDPXYZA,USB1,0,51.5,FINESTEERING,2209,510374.000,12000020,5d37,16809;
SOL_COMPUTED,WAAS,-1632848.5654,-3662158.8160,4944901.1475,0.6048,0.7566,
1.0662,SOL_COMPUTED,WAAS,0.0003,0.0024,0.0011,0.0048,0.0069,0.0085,"131",
0.250,6.000,0.000,45,41,41,41,0,86,7f,37*ea8d8f90
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PDPXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	P-sol status	Solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	V-sol status	Solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H+44
11	vel type	Velocity type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m)	Double	8	H+68
15	V-X $\sigma$	Standard deviation of V-X (m)	Float	4	H+76
16	V-Y $\sigma$	Standard deviation of V-Y (m)	Float	4	H+80
17	V-Z $\sigma$	Standard deviation of V-Z (m)	Float	4	H+84
18	stn ID	Base station ID	Char[4]	4	H+88
19	V-latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#sats	Number of satellite vehicles tracked	Uchar	1	H+104
23	#sats soln	Number of satellite vehicles used in solution	Uchar	1	H+105
24	Reserved		Uchar	1	H+106
25			Uchar	1	H+107
26			Uchar	1	H+108
27	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.125 PORTSTATS

### Port statistics

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log conveys various status parameters of the receiver's communication ports. The receiver maintains a running count of a variety of status indicators of the data link. This log outputs a report of those indicators.

**Message ID:** 72

**Log Type:** Polled

**Recommended Input:**

```
log portstatsa once
```

**ASCII example:**

```
#PORTSTATSA,USB1,0,50.0,FINESTEERING,2209,510485.000,12000020,a872,16809;26,COM
1,0,957,0,0,142,0,0,0,0,COM2,0,957,0,0,142,0,0,0,0,COM3,0,957,0,0,142,0,0,0,0,F
ILE,0,0,0,0,0,0,0,0,0,USB1,180716,115373935,180716,0,0,0,0,0,0,USB2,0,0,0,0,0,0
,0,0,0,0,USB3,0,0,0,0,0,0,0,0,0,COM4,140,0,140,0,40,0,0,0,0,ICOM1,0,0,0,0,0,0,0,0
,0,ICOM2,0,0,0,0,0,0,0,0,0,ICOM3,0,0,0,0,0,0,0,0,0,NCOM1,0,0,0,0,0,0,0,0,0,NCOM
2,0,0,0,0,0,0,0,0,0,NCOM3,0,0,0,0,0,0,0,0,0,ICOM4,0,0,0,0,0,0,0,0,0,WCOM1,0,7,0
,0,0,0,0,0,0,COM5,0,6,0,0,4,0,0,0,0,CCOM1,0,0,0,0,0,0,0,0,0,CCOM2,0,0,0,0,0,0,0
,0,0,CCOM3,0,0,0,0,0,0,0,0,0,CCOM4,0,0,0,0,0,0,0,0,0,CCOM5,0,0,0,0,0,0,0,0,0,CC
OM6,0,0,0,0,0,0,0,0,0,ICOM5,0,0,0,0,0,0,0,0,0,ICOM6,0,0,0,0,0,0,0,0,0,ICOM7,0,0
,0,0,0,0,0,0,0*2739d038
```



Parity and framing errors occur for COM ports if poor transmission lines are encountered or if there is an incompatibility in the data protocol. If errors occur, you may need to confirm the bit rate, number of data bits, number of stop bits and parity of both the transmit and receiving ends. Characters may be dropped when the CPU is overloaded.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PORTSTATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#port	Number of ports with information to follow	Long	4	H
3	port	Refer to <i>Table 34: Communications Port Identifiers</i> on page 152	Enum	4	H+4
4	rx chars	Total number of characters received through this port	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	tx chars	Total number of characters transmitted through this port	Ulong	4	H+12
6	acc rx chars	Total number of accepted characters received through this port	Ulong	4	H+16
7	dropped rx chars	Number of software overruns in receive	Ulong	4	H+20
8	interrupts	Number of interrupts on this port	Ulong	4	H+24
9	breaks	Number of breaks (only for serial ports)	Ulong	4	H+28
10	par err	Number of parity errors (only for serial ports)	Ulong	4	H+32
11	frame err	Number of framing errors (only for serial ports)	Ulong	4	H+36
12	rx overruns	Number of hardware overruns in receive	Ulong	4	H+40
13	Next port offset = H+4+(#port x 40)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#port x 40)
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.126 PPPDATUMINFO

### *Datum information for the PPP position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **PPPPOS** log (see page 758).



The datum message for TerraStar Correction Services is broadcast periodically. Therefore, it can take several hours, from the start of receiving TerraStar corrections, for the **PPPDATUMINFO** log to reflect the correct datum and co-ordinate epoch for the PPP position solution. Once the PPP datum is known, it will persist through a **RESET** or a power cycle. It will not persist through a **FRESET**.

**Message ID:** 2293

**Log Type:** Asynch

**Recommended Input:**

```
log pppdatuminfoa onchanged
```

**ASCII Example:**

```
#PPPDATUMINFOA,USB1,0,51.5,FINESTEERING,2209,510600.000,12000020,ea8d,16809;
"ITRF2014",1165,2022.364,GOOD*62e40570
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the <b>PPPPOS</b> log.	Char[32]	variable	H
3	epsg_code	EPSG code of datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year) Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation Status. See <i>Table 85: Transformation Status</i> on page 518.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.127 PPPPOS

### *PPP filter position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the position solution computed by the PPP filter.

**Message ID:** 1538

**Log Type:** Synch

**Recommended Input:**

```
log pppposa ontime 1
```

**ASCII Example:**

```
#PPPPOSA,USB1,0,52.0,FINESTEERING,2209,510777.000,12000020,9078,16809;  
SOL_COMPUTED,PPP,51.15043722698,-114.03067878229,1097.3465,-17.0001,WGS84,  
0.0167,0.0118,0.0293,"TSTR",17.000,0.000,45,41,41,41,00,00,7f,37*e8b0f4f7
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	Type	Position type (see <i>Table 147: Position Type</i> on the next page)	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) <sup>1</sup>	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40

<sup>1</sup>When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID TSTL = TerraStar L service TSTR = TerraStar C or C-Pro service TSX = TerraStar X Regional service OCXH = Oceanix service	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Reserved		Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526 or <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 147: Position Type

ASCII	Binary	Description
NONE	0	No solution
PPP_CONVERGING	68	Converging TerraStar-C, TerraStar-C PRO, TerraStar-X solution
PPP	69	Converged TerraStar-C, TerraStar-C PRO, TerraStar-X solution
PPP_BASIC_CONVERGING	77	Converging TerraStar-L solution
PPP_BASIC	78	Converged TerraStar-L solution

## 3.128 PPPSATS

### *Satellites used in the PPPPOS solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists the used and unused satellites for the corresponding PPPPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

**Message ID:** 1541

**Log Type:** Synch

**Recommended Input:**

```
log pppsatsa ontime 1
```

**Abbreviated ASCII Example:**

```
<PPPSATS COM1 0 62.0 FINESTEERING 2220 503464.000 02000000 ce3f 16807
< 35
< GPS 18 GOOD 00000007
< GPS 16 GOOD 00000003
< GPS 5 GOOD 00000003
< GPS 27 GOOD 00000007
< GPS 29 GOOD 00000003
< GPS 10 GOOD 00000007
< GPS 23 GOOD 00000007
< GPS 13 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 26 GOOD 00000007
< QZSS 194 SUPPLEMENTARY 00000007
< GLONASS 18-3 SUPPLEMENTARY 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 24+2 GOOD 00000003
< GLONASS 23+3 SUPPLEMENTARY 00000001
< GLONASS 17+4 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
< GALILEO 15 GOOD 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 3 GOOD 0000000f
< GALILEO 36 GOOD 0000000f
< GALILEO 5 GOOD 0000000f
< GALILEO 21 GOOD 0000000f
< GALILEO 13 SUPPLEMENTARY 0000000f
< GALILEO 9 GOOD 0000000f
< BEIDOU 28 GOOD 0000001d
< BEIDOU 33 GOOD 0000001d
< BEIDOU 42 GOOD 0000001d
< BEIDOU 27 GOOD 0000001d
< BEIDOU 43 GOOD 0000001d
< BEIDOU 41 GOOD 0000001d
< BEIDOU 40 SUPPLEMENTARY 0000001d
< BEIDOU 11 SUPPLEMENTARY 00000007
< BEIDOU 14 SUPPLEMENTARY 00000007
```



Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	Satellite system (see <i>Table 124: Satellite System</i> on page 654)	Enum	4	H+4
4	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest-order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS, or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems.  In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2.	Ulong	4	H+8
5	Status	Satellite status (see <i>Table 92: Observation Statuses</i> on page 529)	Enum	4	H+12
6	Signal Mask	Signals used in the solution (see <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531)	Hex	4	H+16
7	Next satellite offset = H + 4 + (#entries x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#entries x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.129 PPPSEEDAPPLICATIONSTATUS

### *Progress or outcome of the latest PPP seeding attempt*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log is used to report the progress or outcome of the latest attempt to apply a static PPP seed. For a static seed to shorten the PPP convergence time, the receiver must remain stationary until this log indicates motion is permitted. The receiver may move during **PENDING\_MOTION\_ALLOWED** and seeding and fast convergence will be successful, but the reported accuracy will be highest if the receiver remains stationary until **APPLIED** status, which normally takes about two minutes longer. Large movements between recording the seed and the end of **PENDING** status will result in the seed being discarded and a full convergence taking place; but movements during this time of less than 10m may cause an incorrect seed to be applied and an incorrect PPP solution to be reported for a few minutes before this is discovered. This can be avoided by erasing the seed with **PPPSEED CLEAR**. A status of **APPLIED** means that a seed was successfully applied.

The **PPPSEED** command (see page 304) is used to control the application of seeds. Seeds can be manually applied by **PPPSEED SET** or **RESTORE**, or automatically by **PPPSEED AUTO**.

**PPPSEEDAPPLICATIONSTATUS** does not report the results of the **PPPDYNAMICSEED** command (see page 301).

#### Message ID 2250

#### Log type: Asynch

#### Recommended Input

```
log pppseedapplicationstatusa onchanged
```

#### ASCII Example

```
#PPPSEEDAPPLICATIONSTATUSA,USB1,0,50.5,UNKNOWN,0,0.000,12000020,4bd6,16809;
UNAVAILABLE,0.000000000000,0.000000000000,0.0000,0.0000,0.0000,0.0000*d604a4a4
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPSEEDAPPLICATIONSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Status	See <i>Table 148: PPP Seed Application Status</i> on the next page.	Enum	4	H
3	Latitude	Latitude Range: $\pm 90$ (degrees)	Double	8	H+4
4	Longitude	Longitude Range: $\pm 180$ (degrees)	Double	8	H+12

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	Height	Ellipsoidal height Range: > -2000 (m)	Double	8	H+20
6	Latitude std. dev.	Latitude standard deviation (m)	Float	4	H+28
7	Longitude std. dev.	Longitude standard deviation (m)	Float	4	H+32
8	Height std. dev.	Ellipsoidal height standard deviation (m)	Float	4	H+36
9	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+40
10	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

Table 148: PPP Seed Application Status

Binary	ASCII	Description
0	UNAVAILABLE	No seed was stored in NVM before the time of seeding or has been set, so a full convergence solution will be provided. Move as desired.
1	AVAILABLE	A seed is stored in NVM but seeding is not enabled, so a full convergence solution will be provided. Either enable seeding or move as desired.
2	PENDING	A seed is stored in NVM or has been set and seeding is enabled and the system is waiting to be able to use the seed for rapid convergence. Remain stationary.
3	APPLIED	The system has used the seed to complete rapid convergence. Move as desired.
4	PENDING_MOTION_ALLOWED	The system is currently able to use the seed for rapid convergence. The receiver may move with minor loss of accuracy or remain stationary for a minute or two to improve convergence accuracy.
10	REJECTED_MOTION_DETECTED	The seed was rejected because motion was detected or explicitly set via PPPDYNAMICS DYNAMIC (see <b>PPPDYNAMICS</b> command on page 300) before the seed could be used. A full convergence solution will be provided. Move as desired.
11	REJECTED_BAD_POSITION	The seed and seeded solution were discarded because the seed position was found to be incorrect. A full convergence solution will be provided. Move as desired.  Note the possibility that an incorrect position may have been reported as accurate before this point.
20	DISCARDED_RAPID_CONVERGENCE	A seed was not required for fast convergence of the available solution so it was not used. Move as desired.

### 3.130 PPPSEEDSTORESTATUS

#### *Status of the NVM-stored PPP seed*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the availability of the PPP seed stored in NVM. When the status is **AVAILABLE**, a seed has been stored in NVM and is available for recall.

Seed storing and recall is controlled by the **PPPSEED** command (see page 304).

#### **Message ID 2251**

#### **Log type: Asynch**

#### **Recommended Input**

```
log pppseedstorestatusa onchanged
```

#### **ASCII Example**

```
#PPPSEEDSTORESTATUSA,USB1,0,48.5,UNKNOWN,0,0.195,12000020,edad,16809;
UNAVAILABLE,0.000*e3794bc7
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPSEEDSTORESTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Status	Status of the stored PPP seed: 0 = UNAVAILABLE 1 = AVAILABLE	Enum	4	H
3	Horizontal std. dev.	Horizontal standard deviation (metres)	Float	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.131 PPPXYZ

### *PPP filter Cartesian position and velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PPPXYZ log contains the Cartesian position in X, Y and Z coordinates as computed by the receiver with the PPP filter enabled.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command on page 284.

**Message ID:** 1543

**Log Type:** Synch

**Recommended Input:**

```
log pppxyza ontime 1
```

**ASCII Example:**

```
#PPPXYZA,USB1,0,50.5,FINESTEERING,2209,510993.000,12000020,951f,16809;
SOL_COMPUTED,PPP,-1632848.2151,-3662158.6164,4944901.2646,0.0137,0.0208,0.0239,
SOL_COMPUTED,PPP,-0.0002,0.0021,-0.0039,0.0052,0.0071,0.0092,"TSTR",0.250,
13.000,0.000,47,43,43,42,0,00,7f,37*429a6672
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PPPXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	P-sol status	Position solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	V-sol status	Velocity solution status (refer to <i>Table 86: Solution Status</i> on page 523)	Enum	4	H+44
11	vel type	Velocity type (refer to <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m)	Double	8	H+68
15	V-X $\sigma$	Standard deviation of V-X (m)	Float	4	H+76
16	V-Y $\sigma$	Standard deviation of V-Y (m)	Float	4	H+80
17	V-Z $\sigma$	Standard deviation of V-Z (m)	Float	4	H+84
18	stn ID	Base station ID	Char[4]	4	H+88
19	V-latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#sats	Number of satellite vehicles tracked	Uchar	1	H+104
23	#sats soln	Number of satellite vehicles used in solution	Uchar	1	H+105
24	Reserved		Uchar	1	H+106
25			Uchar	1	H+107
26			Uchar	1	H+108
27	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.132 PROFILEINFO

### *Profile information in NVM*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log outputs a list of Profiles in the system. Refer also to the **PROFILE** command on page 315.



A list may consist of a maximum of 20 profiles.

**Message ID:** 1412

**Log Type:** Polled

**Recommended Input:**

```
log profileinfoa onchanged
```

**ASCII Examples:**

```
#PROFILEINFOA,USB1,0,39.5,FINESTEERING,2209,511199.410,12000020,ae3a,16809;
"BASE",0,0*aefd0c3e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PROFILEINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Name	Profile Name	String [Max 20]	variable <sup>1</sup>	H
3	Status Word	Refer to <i>Table 149: Status Word</i> on the next page	Ulong	4	variable
4	# of Commands	Number of commands assigned to the Profile	Ulong	4	variable
5	Command	Profile command	String [Max 150]	variable <sup>1</sup>	variable
6	Next command offset = variable				
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	variable
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

**Table 149: Status Word**

Bit #	Mask	Description
0	0x00000001	Activate Flag (0 – Deactivate (default), 1 – Activate)
1 – 3	0x0000000E	Reserved
4	0x00000010	Command 1 validation Flag (0 – Valid (default), 1 – Invalid)
5	0x00000020	Command 2 validation Flag
6	0x00000040	Command 3 validation Flag
7	0x00000080	Command 4 validation Flag
8	0x00000100	Command 5 validation Flag
9	0x00000200	Command 6 validation Flag
10	0x00000400	Command 7 validation Flag
11	0x00000800	Command 8 validation Flag
12	0x00001000	Command 9 validation Flag
13	0x00002000	Command 10 validation Flag
14	0x00004000	Command 11 validation Flag
15	0x00008000	Command 12 validation Flag
16	0x00010000	Command 13 validation Flag
17	0x00020000	Command 14 validation Flag
18	0x00040000	Command 15 validation Flag
19	0x00080000	Command 16 validation Flag
20	0x00100000	Command 17 validation Flag
21	0x00200000	Command 18 validation Flag
22	0x00400000	Command 19 validation Flag
23	0x00800000	Command 20 validation Flag
24 - 31	0xFF000000	Reserved



### 3.133 PSRDATUMINFO

#### *Datum information for the pseudorange position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **PSRPOS** log (see page 774).

**Message ID:** 2300

**Log Type:** Asynch

**Recommended Input:**

```
log psrdatuminfoa onchanged
```

**ASCII Example:**

```
#PSRDATUMINFOA,USB1,0,51.5,COARSESTEERING,2209,511539.000,02000020,f85f,16809;  
"ECEF",0,2022.364,GOOD*bbfbd59
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the <b>PSRPPPOS</b> log.	Char[32]	variable	H
3	epsg_code	EPSG code of datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year) Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation Status. See <i>Table 85: Transformation Status</i> on page 518.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.134 PSRDOP

### *DOP values for the satellites used in the PSR solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PSRDOP log contains the Dilution Of Precision (DOP) values for the satellites being used in the PSR solution. The PSR DOPs are updated every 60 seconds or whenever the satellites used in the PSR solution changes.

DOP values are a measure of the solution strength. Essentially, the DOPs reflect the geometry of the satellites used in the solution. Solutions with good counts of well-distributed satellites will have low DOPs and should be accurate and reliable. Solutions with fewer or poorly-distributed satellites will have high DOPs and be less accurate and reliable. As a rough guideline, PDOP values less than 4 imply a solution with reasonable geometry.

There can be many reasons for high DOP values. The most common reason is that there are obstructions limiting satellite visibility. Even if satellites are visible and being tracked they might still not be used in the solution if, for example, they are unhealthy or there are not corrections available for them. The **PSRSATS** log (see page 776) will inform which satellites are being tracked and explain why a tracked satellite is not used in the solution.

The DOPs do not consider that different satellites or signals will be weighted differently in the solution. Therefore, they do not completely reflect the solution quality. Ultimately, the standard deviations reported in the **PSRPOS** log (see page 774) are the best reflection of the solution accuracy.



1. If a satellite is locked out using the **LOCKOUT** command (see page 244), it will still show in the prn list but it will be significantly deweighted in the dop calculation.
2. The vertical dilution of precision can be calculated by:  

$$vdop = \sqrt{pdop^2 - hdop^2}$$
3. If the DOP is not yet calculated, a default value of 9999.0 is displayed.

**Message ID: 174**

**Log Type: Asynch**

**Recommended Input:**

```
log psrdopa onchanged
```

**ASCII Example:**

```
#PSRDOPA,USB1,0,50.0,FINESTEERING,2209,511740.000,02000020,1779,16809;
1.8150,1.6190,0.8940,1.2150,0.8220,5.0,10,26,2,18,12,5,20,29,31,9,25*982c4334
```

Field	Field type		Format	Binary Bytes	Binary Offset
1	Log header	PSRDOP header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	gdop	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H
3	pdop	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4
4	hdop	Horizontal dilution of precision.	Float	4	H+8
5	htdop	Horizontal position and time dilution of precision.	Float	4	H+12
6	tdop	GPS time dilution of precision – assumes 3D position is known and only the GPS receiver clock offset is unknown	Float	4	H+16
7	cutoff	GPS elevation cut-off angle	Float	4	H+20
8	#PRN	Number of satellites PRNs to follow	Long	4	H+24
9	PRN	PRN of SV PRN tracking, null field until position solution available	Ulong	4	H+28
10...	Next PRN offset = H+28+(#prn x 4)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+28+ (#prn x 4)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### 3.135 PSRDOP2

#### *DOP values for the satellites used in the PSR solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PSRDOP2 log contains the Dilution Of Precision (DOP) values for the satellites being used in the PSR solution. This log is similar to the **PSRDOP** log (see page 770) but contains the per-system TDOPs; see the **PSRDOP** log on page 770 for more information on the DOPs.

**Message ID:** 1163

**Log Type:** Asynch

**Recommended Input:**

```
log psrdop2a onchanged
```

**ASCII Example:**

```
#PSRDOP2A,USB1,0,18.5,FINESTEERING,2209,511740.000,02000020,0802,16809;
1.8150,1.6190,0.8940,1.3490,1,GPS,0.8220*2a32bc78
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRDOP2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	GDOP	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H
3	PDOP	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4
4	HDOP	Horizontal dilution of precision	Float	4	H+8
5	VDOP	Vertical dilution of precision	Float	4	H+12
6	#systems	Number of systems	Ulong	4	H+16
7	system	See <i>Table 150: System Used for Timing</i> on the next page	Enum	4	H+20
8	TDOP	Time dilution of precision	Float	4	H+24
9	Next satellite offset = H+20+(#systems x 8)				
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20+ (#systems x 8)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 150: System Used for Timing**

<b>Binary</b>	<b>ASCII</b>
0	GPS <sup>1</sup>
1	GLONASS
2	GALILEO
3	BEIDOU
4	NAVIC
99	AUTO <sup>2</sup>

---

<sup>1</sup>GPS setting includes QZSS satellites.

<sup>2</sup>AUTO is used only as a backup system (not available for primary system field).

## 3.136 PSRPOS

### *Pseudorange position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the position computed by the receiver, along with three status flags. In addition, it reports other status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections.

**Message ID:** 47

**Log Type:** Synch

**Recommended Input:**

```
log psrposa ontime 1
```

**ASCII Example:**

```
#PSRPOSA,USB1,0,20.0,FINESTEERING,2209,511779.000,02000020,28ff,16809;
SOL_COMPUTED,WAAS,51.15043801969,-114.03066782703,1096.7864,-17.0000,WGS84,
0.9069,0.8826,1.8779,"133",4.000,0.000,45,10,0,0,00,06,00,03*bc4f3b04
```



There are DGPS use cases in which the base receiver is not maintained or controlled by the positioning user. For example, the US Coast Guard operates a differential correction service which broadcasts GPS differential corrections over marine radio beacons. As a user, all you need is a marine beacon receiver and a GNSS receiver to achieve positioning accuracy of less than 1 metre. In this case, the Coast Guard owns and operates the base receiver at known coordinates. Other examples of users appearing to use only one GNSS receiver include FM radio station correction services, privately owned radio transmitters and corrections carried by communication satellites. Some of the radio receivers have built-in GNSS receivers and combined antennas, so they even appear to look as one self-contained unit.

The major factors degrading GPS signals which can be removed or reduced with differential methods are the atmosphere, ionosphere, satellite orbit errors, and satellite clock errors. Some errors which are not removed include receiver noise and multipath.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4

Field	Field type	Description	Format	Binary Bytes	Binary Offset
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) <sup>1</sup>	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	Reserved		Uchar	1	H+66
18			Uchar	1	H+67
19			Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.

## 3.137 PSRSATS

### *Satellites used in PSRPOS solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log lists the used and unused satellites for the corresponding PSRPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

**Message ID:** 1162

**Log Type:** Synch

**Recommended Input:**

```
log psrsatsa ontime 1
```

**Abbreviated ASCII Example:**

```
<PSRSATS COM1 0 76.5 FINESTEERING 2220 502666.000 02000008 fea4 16807
< 36
< GPS 13 GOOD 00000003
< GPS 29 GOOD 00000003
< GPS 10 GOOD 00000003
< GPS 16 GOOD 00000003
< GPS 18 GOOD 00000003
< GPS 27 GOOD 00000003
< GPS 5 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 23 GOOD 00000003
< GPS 26 GOOD 00000003
< QZSS 194 GOOD 00000003
< GLONASS 23+3 NOIONOCORR 00000000
< GLONASS 24+2 GOOD 00000003
< GLONASS 15 GOOD 00000003
< GLONASS 17+4 GOOD 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
< GALILEO 36 GOOD 00000009
< GALILEO 30 GOOD 00000009
< GALILEO 15 GOOD 00000009
< GALILEO 21 GOOD 00000009
< GALILEO 3 GOOD 00000009
< GALILEO 9 GOOD 00000009
< GALILEO 27 GOOD 00000009
< GALILEO 5 GOOD 00000009
< BEIDOU 33 GOOD 00000011
< BEIDOU 40 GOOD 00000011
< BEIDOU 28 GOOD 00000011
< BEIDOU 14 NOIONOCORR 00000000
< BEIDOU 21 GOOD 00000011
< BEIDOU 41 GOOD 00000011
< BEIDOU 27 GOOD 00000011
< BEIDOU 43 GOOD 00000011
< BEIDOU 42 GOOD 00000011
```



Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See <i>Table 124: Satellite System</i> on page 654	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status (see <i>Table 92: Observation Statuses</i> on page 529)	Enum	4	H+12
6	Signal mask	See <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.138 PSRVEL

### *Pseudorange velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

In the PSRVEL log, the actual speed and direction of the receiver antenna over ground is provided. The velocity measurements sometimes have a latency associated with them. The time of validity is the time tag in the log minus the latency value.



In a PSRVEL log, the actual speed and direction of the receiver antenna over ground is provided. The receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of motion of the GNSS antenna relative to ground.

The velocity in the PSRVEL log is determined by the pseudorange filter. Velocities from the pseudorange filter are calculated from the Doppler.

The velocity status indicates varying degrees of velocity quality. To ensure healthy velocity, the velocity sol-status must also be checked. If the sol-status is non-zero, the velocity is likely invalid. It should be noted that the receiver does not determine the direction a vessel, craft, or vehicle is pointed (heading), but rather the direction of the motion of the GPS antenna relative to the ground.

The latency of the instantaneous Doppler velocity is always 0.15 seconds. The latency represents an estimate of the delay caused by the tracking loops under acceleration of approximately 1 G. For most users, the latency can be assumed to be zero (instantaneous velocity).

**Message ID: 100**

**Log Type: Synch**

**Recommended Input:**

```
log psrvela ontime 1
```

**ASCII Example:**

```
#PSRVELA,USB1,0,51.5,FINESTEERING,2209,511827.000,02000020,0dd6,16809;  
SOL_COMPUTED,WAAS,0.000,4.000,0.0175,290.743174,0.0309,0*3d24adcc
```



Consider the case where vehicles are leaving a control center. The control center's coordinates are known but the vehicles are on the move. Using the control center's position as a reference, the vehicles are able to report where they are with PSRPOS and their speed and direction with PSRVEL at any time.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+8
5	age	Differential age in seconds	Float	4	H+12
6	hor spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	vert spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### 3.139 PSRXYZ

#### *Pseudorange Cartesian position and velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the receiver's pseudorange position and velocity in ECEF coordinates. The position and velocity status field's indicate whether or not the corresponding data is valid. See *Figure 12: The WGS84 ECEF Coordinate System* on page 540 for a definition of the ECEF coordinates.

The velocity status indicates varying degrees of velocity quality. To ensure healthy velocity, the velocity sol-status must also be checked. If the sol-status is non-zero, the velocity is likely invalid. It should be noted that the receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of the motion of the GNSS antenna relative to the ground.

The latency of the instantaneous Doppler velocity is always 0.15 seconds. The latency represents an estimate of the delay caused by the tracking loops under acceleration of approximately 1 G. For most users, the latency can be assumed to be zero (instantaneous velocity).



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID:** 243

**Log Type:** Synch

**Recommended Input:**

```
log psrxyza ontime 1
```

**ASCII Example:**

```
#PSRXYZA,USB1,0,50.5,FINESTEERING,2209,511855.000,02000020,c94c,16809;
SOL_COMPUTED,WAAS,-1632847.3796,-3662158.7826,4944901.0871,1.0126,0.9519,
1.6561,SOL_COMPUTED,WAAS,0.0025,-0.0009,0.0025,0.1210,0.1138,0.1979,"131",
0.000,5.000,0.000,46,10,0,0,0,06,00,03*be512b3f
```



The instantaneous Doppler is the measured Doppler frequency which consists of the satellite's motion relative to the receiver (Satellite Doppler + User Doppler) and the clock (local oscillator) drift.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	PSRXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	P-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H

Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H+44
11	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m/s)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m/s)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m/s)	Double	8	H+68
15	V-X $\sigma$	Standard deviation of V-X (m/s)	Float	4	H+76
16	V-Y $\sigma$	Standard deviation of V-Y (m/s)	Float	4	H+80
17	V-Z $\sigma$	Standard deviation of V-Z (m/s)	Float	4	H+84
18	stn ID	Base station ID	Char[4]	4	H+88
19	V-latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellites used in solution	Uchar	1	H+105
24	Reserved		Char	1	H+106
25			Char	1	H+107
26			Char	1	H+108
27	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+109

Field	Field type	Description	Format	Binary Bytes	Binary Offset
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.140 QZSSALMANAC

### Decoded QZSS Almanac parameters

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the decoded L1 C/A almanac parameters as received from the satellite with the parity information removed and appropriate scaling applied.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.



For more information about QZSS almanac data, refer to the Interface Specifications for QZSS at <http://qzss.go.jp/en/technical/ps-is-qzss/ps-is-qzss.html>.

**Message ID:** 1346

**Log Type:** Asynch

**Recommended Input:**

```
log qzssalmanaca onchanged
```

**ASCII Example:**

```
#QZSSALMANACA,USB1,0,52.0,SATTIME,2209,435870.000,02000020,67d2,16809;4,
194,2210,86016.0,7.608324e-02,-2.30866759e-09,5.57636471e-01,-1.5856923e+00,
-2.8557607e+00,0.00000000,0.00000000,7.29378066e-05,4.2157755e+07,
7.30630256e-01,0,0,195,2210,86016.0,7.458836e-02,-2.59439378e-09,2.2676078e+00,
-1.5598701e+00,1.6893387e+00,-1.90734863e-06,0.00000000,7.29131785e-05,
4.2167248e+07,7.11593314e-01,0,0,196,2210,86016.0,7.522970e-02,-3.50871758e-09,
-2.2752950e+00,-1.5559876e+00,-9.89514980e-02,1.02043152e-04,3.63797881e-12,
7.29238543e-05,4.2163133e+07,6.00026173e-01,0,0,199,2210,86016.0,2.670288e-04,
9.60039990e-10,-1.6991414e+00,-1.1489437e+00,-1.2297942e+00,0.00000000,
0.00000000,7.29167972e-05,4.2165853e+07,1.21040672e-03,0,0*37178436
```



The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position (**SETAPPROXTIME** and **SETAPPROXPOS**), as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	#messages	Number of satellite PRN almanac messages to follow. Set to zero until almanac data is available	Ulong	4	H
3	PRN	Satellite PRN number for current message (dimensionless)	Ulong	4	H+4
4	week	Almanac reference week (GPS reference week number)	Ulong	4	H+8
5	seconds	Almanac reference time (s)	Double	8	H+12
6	ecc	Eccentricity (dimensionless)	Double	8	H+20
7	$\dot{\omega}$	Rate of right ascension (radians/s)	Double	8	H+28
8	$\omega_0$	Right, ascension (radians)	Double	8	H+36
9	$\omega$	Argument of perigee (radians)	Double	8	H+44
10	$M_0$	Mean anomaly of reference time (radians)	Double	8	H+52
11	$a_{f0}$	Clock aging parameter (s)	Double	8	H+60
12	$a_{f1}$	Clock aging parameter (s/s)	Double	8	H+68
13	N	Corrected mean motion (radians/s)	Double	8	H+76
14	A	Semi-major axis (m)	Double	8	H+84
15	inclination angle	Angle of inclination	Double	8	H+92
16	health-prn	SV health from Page 25 of subframe 4 or 5 (6 bits)	Ulong	4	H+100
17	health-alm	SV health from almanac (8 bits)	Ulong	4	H+104
18	Next PRN offset = H+4+(#messages x 104)				
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#messages x 104)
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.141 QZSSCNAVRAWMESSAGE

### QZSS CNAV Raw Message

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the raw CNAV message from signals which contain the CNAV message (L2C, L5). It also indicates whether the raw message is generated from an L2C signal or L5 signal.

The **QZSSCNAVRAWMESSAGE** log is not output by default. To receive this log, data decoding for L2C or L5 must be enabled using the **DATADECODESIGNAL** command (see page 129)

```
DATADECODESIGNAL QZSSL2C ENABLE
```

```
DATADECODESIGNAL QZSSL5 ENABLE
```

**Message ID: 2261**

**Log Type: Asynch**

**Recommended Input:**

```
log qzsscnavrawmessagea onnew
```

**ASCII Example:**

```
#QZSSCNAVRAWMESSAGEA,USB1,0,45.5,SATTIME,2209,512022.000,02000020,65aa,16809;
173,194,QZSSL5,11,8b08ba6ad35a0dd279c8877112f86741c12c200aa038609c8d40313bdfe35
d0478e809512d50*a22c707e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSCNAVRAWMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	signal channel	Signal channel providing the bits	Ulong	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	signal type	Signal type (L2C or L5) See <i>Table 151: Signal Type</i> on the next page	Enum	4	H+8
5	message ID	Message ID	Ulong	4	H+12
6	data	Raw message data	Hex[38]	38	H+16
7	xxxx	32-bit CRC (ASCII and binary only)	Hex	4	H+54
8	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

**Table 151: Signal Type**

Value (Binary)	Signal (ASCII)	Description
33	GPSL1CA	GPS L1 C/A-code
47	GPSL1CP	GPS L1C P-code
68	GPSL2Y	GPS L2 P(Y)-code
69	GPSL2C	GPS L2 C/A-code
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
2662	GLOL3	GLONASS L3
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10466	GALE5A	Galileo E5A
10499	GALE5B	Galileo E5B
10532	GALALTBOC	Galileo ALT-BOC
10565	GALE6C	Galileo E6C
10572	GALE6B	Galileo E6B
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2I with D1 navigation data
12804	BDSB2D2	BeiDou B2I with D2 navigation data
12877	BDSB3D1	BeiDou B3 with D1 navigation data
12880	BDSB3D2	BeiDou B3 with D2 navigation data
12979	BDSB1C	BeiDou B1C
13012	BDSB2A	BeiDou B2a
13077	BDSB2BI	BeiDou B2b (I)
14753	QZSSL1CA	QZSS L1 C/A-code
14760	QZSSL1CP	QZSS L1C P-code

<b>Value (Binary)</b>	<b>Signal (ASCII)</b>	<b>Description</b>
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5
14891	QZSSL6P	QZSS L6P
19073	NAVICL5SPS	NavIC L5 SPS

## 3.142 QZSSEPHemeris

### Decoded QZSS parameters

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains a single set of QZSS L1 C/A ephemeris parameters.

**Message ID:** 1336

**Log Type:** Asynch

**Recommended Input:**

```
log qzssephemerisa onchanged
```

### ASCII Example:

```
#QZSSEPHemerisA, USB1, 2, 50.5, SATTIME, 2209, 512010.000, 02000020, 1e9d, 16809;
194, 512010.0, 0, 121, 121, 2209, 2209, 514800.0, 4.215826861e+07, 2.486532145e-09,
-3.127432312e+00, 7.6143292477e-02, -1.585891666e+00, 3.105588257e-05,
-7.426366210e-06, 4.20750000e+02, 1.02396875e+03, 4.302710295e-07,
-8.195638657e-08, 7.3058678204e-01, 3.739441477e-10, 6.7849504293e-01,
-2.86190492e-09, 889, 514800.0, 9.313225746e-10, -1.13156e-07, -1.13687e-13, 0.00000,
FALSE, 7.293896144e-05, 7.84000000e+00, 0, 0, 0, 0*42695181
```

```
#QZSSEPHemerisA, USB1, 1, 50.5, SATTIME, 2209, 460800.000, 02000020, 1e9d, 16809;
195, 460800.0, 0, 65, 65, 2209, 2209, 464400.0, 4.216786739e+07, 1.912936824e-09,
-2.252659029e+00, 7.4505718076e-02, -1.560284732e+00, -2.508051693e-05,
2.811849117e-05, -7.96906250e+02, -7.62375000e+02, 1.281499863e-06,
7.413327694e-07, 7.1167492174e-01, -6.507413917e-10, 2.388640708e+00,
-2.51189034e-09, 833, 464400.0, 0.000000000, -2.11736e-06, -2.27374e-13, 0.00000,
FALSE, 7.291348520e-05, 7.84000000e+00, 0, 0, 0, 0*f8a88fc9
```

```
#QZSSEPHemerisA, USB1, 0, 50.5, SATTIME, 2209, 482400.000, 02000020, 1e9d, 16809;
196, 482400.0, 0, 89, 89, 2209, 2209, 486000.0, 4.216483448e+07, 4.154101606e-09,
-2.467916384e+00, 7.5259242672e-02, -1.556757964e+00, 8.281320333e-06,
-1.931935549e-05, 7.30312500e+02, 2.36562500e+02, -1.218169928e-06,
-1.806765795e-07, 5.9983243978e-01, 8.503925651e-10, -2.154062884e+00,
-4.07552690e-09, 857, 486000.0, -3.725290298e-09, 1.01195e-04, 2.50111e-12, 0.00000,
FALSE, 7.292359328e-05, 7.84000000e+00, 0, 0, 0, 0*fd6bdfea
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSEPHemeris header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	tow	Time stamp of subframe 0 (s)	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	health	Health status - bits 0-5: 6-bit health code as defined in QZSS Interface Specification - bit 6: Alert flag	Ulong	4	H+12
5	IODE1	Issue of ephemeris data 1	Ulong	4	H+16
6	IODE2	Issue of ephemeris data 2	Ulong	4	H+20
7	week	GPS reference week number	Ulong	4	H+24
8	z week	Z count week number. This is the week number from subframe 1 of the ephemeris. The 'toe week' (field #7) is derived from this to account for rollover	Ulong	4	H+28
9	toe	Reference time for ephemeris (s)	Double	8	H+32
10	A	Semi-major axis (m)	Double	8	H+40
11	$\Delta N$	Mean motion difference (radians/s)	Double	8	H+48
12	$M_0$	Mean anomaly of reference time (radius)	Double	8	H+56
13	ecc	Eccentricity (dimensionless)	Double	8	H+64
14	$\omega$	Argument of perigee (radians)	Double	8	H+72
15	cuc	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+80
16	cus	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+88
17	crc	Amplitude of cosine harmonic correction term to the orbit radius (m)	Double	8	H+96
18	crs	Amplitude of sine harmonic correction term to the orbit radius (m)	Double	8	H+104
19	cic	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+112
20	cis	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+120
21	$l_0$	Inclination angle at reference time (radians)	Double	8	H+128
22	$\dot{l}$	Rate of inclination angle (radians/s)	Double	8	H+136
23	$\omega_0$	Right ascension (radians)	Double	8	H+144
24	$\dot{\omega}$	Rate of right ascension (radians/s)	Double	8	H+152
25	iodc	Issue of data clock	Ulong	4	H+160

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
26	toc	SV clock correction term (s)	Double	8	H+164
27	tgd	Estimated group delay difference (s)	Double	8	H+172
28	$a_{f0}$	Clock aging parameter (s)	Double	8	H+180
29	$a_{f1}$	Clock aging parameter (s/s)	Double	8	H+188
30	$a_{f2}$	Clock aging parameter (s/s/s)	Double	8	H+196
31	AS	Anti-spoofing on: 0= FALSE 1=TRUE	Enum	4	H+204
32	N	Corrected mean motion (radians/s)	Double	8	H+208
33	URA	User Range Accuracy variance, $m^2$ .  The ICD specifies that the URA index transmitted in the ephemerides can be converted to a nominal standard deviation value using an algorithm listed there. We publish the square of the nominal value (variance)	Double	8	H+216
34	Fit Interval	Curve fit interval: 0 = Ephemeris data are effective for 2 hours 1 = Ephemeris data are effective for more than 2 hours	Uchar	1	H+224
35	Reserved		Uchar	1	H+225
36	Reserved		Uchar	1	H+226
37	Reserved		Uchar	1	H+227
38	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+228
39	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### 3.143 QZSSIONUTC

#### *QZSS ionospheric and time information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC) for QZSS.

**Message ID:** 1347

**Log Type:** Asynch

**Recommended Input:**

```
log qzssionutca onchanged
```

**ASCII Example:**

```
#QZSSIONUTCA,USB1,0,51.5,FINESTEERING,2209,512070.265,02000020,158b,16809;
1.862645149230957e-08,7.450580596923828e-09,-3.576278686523438e-07,
-5.364418029785156e-07,1.290240000000000e+05,1.966080000000000e+05,
-6.553600000000000e+04,-6.553600000000000e+05,2210,94208,
-2.7939677238464355e-09,0.000000000,2185,7,18,18,0*d3e60828
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSIONUTC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	a0	Alpha parameter constant term	Double	8	H
3	a1	Alpha parameter 1st order term	Double	8	H+8
4	a2	Alpha parameter 2nd order term	Double	8	H+16
5	a3	Alpha parameter 3rd order term	Double	8	H+24
6	b0	Beta parameter constant term	Double	8	H+32
7	b1	Beta parameter 1st order term	Double	8	H+40
8	b2	Beta parameter 2nd order term	Double	8	H+48
9	b3	Beta parameter 3rd order term	Double	8	H+56
10	utc wn	UTC reference week number (GPS reference week number)	Ulong	4	H+64
11	tot	Reference time of UTC parameters (s)	Ulong	4	H+68
12	A0	UTC constant term of polynomial (s)	Double	8	H+72

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
13	A1	UTC 1st order term of polynomial (s/s)	Double	8	H+80
14	wn lsf	Future week number (GPS reference week number)	Ulong	4	H+88
15	dn	Day number (the range is 1 to 7 where Sunday=1 and Saturday=7)	Ulong	4	H+92
16	deltat ls	Delta time due to leap seconds	Long	4	H+96
17	deltat lsf	Future delta time due to leap seconds	Long	4	H+100
18	Reserved			4	H+104
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+108
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.144 QZSSRAWALMANAC

### Raw QZSS almanac data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the undecoded L1 C/A almanac subframes as received from the QZSS satellite.

**Message ID:** 1345

**Log Type:** Asynch

**Recommended Input:**

```
log qzssrawalmanaca onchanged
```

**ASCII Example:**

```
#QZSSRAWALMANACA, USB1, 0, 52.0, SATTIME, 2209, 512070.000, 02000020, 64c4, 16809;
4294967202, 86016.000, 14,
0, 0000000000000000000000000000000000000000000000000000000000000000,
2, 8b02aaa6a195c283c115dc4cff3600cae73416b85cbf64a18ba556000003,
3, 8b02aaa6ab96c3778215cfe3ff1d00caed0d5c6402c071f744d471ff0018,
4, 8b02aaa6b096c47cc3158728fec00caea84a34bd0c09a76fbf7e60d002e,
0, 0000000000000000000000000000000000000000000000000000000000000000,
0, 0000000000000000000000000000000000000000000000000000000000000000,
7, 8b02aaa6a694c702301500ca005400caec31bac550d13015cde4c8000003,
0, 0000000000000000000000000000000000000000000000000000000000000000,
0, 0000000000000000000000000000000000000000000000000000000000000000,
0, 0000000000000000000000000000000000000000000000000000000000000000,
51, 8b02aaa68310f315a2f8000f0be03efbe00000000000000000000000000000000,
55, 8b02aaa69795f72929616262626f6f626f6f6f0000000000000000000000000002,
56, 8b02aaa68591f81401faf73f0cfff6000000ffffffffffd17a2128907120000,
61, 8b02aaa68615fd21e506002f4aac61000000ffffffffffd17a2128907120002*7fe16ef0
```



The OEM7 family of receivers automatically saves almanacs in their Non-Volatile Memory (NVM), therefore creating an almanac boot file is not necessary.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSRAWALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	ref week	Almanac reference week number	Ulong	4	H
3	ref secs	Almanac reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+4
4	#subframes	Number of subframes to follow	Ulong	4	H+8

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	svid	SV ID (satellite vehicle ID) SV ID 1 to 10 corresponds to QZSS PRN 193 to 202. Any other values indicate the page ID. Refer to QZSS Interface Specification for more details.	Hex	2	H+12
6	data	Subframe page data	Hex	30	H+14
7	Next subframe offset = $H+12+(\#subframe \times 32)$				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#subframes x 32)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.145 QZSSRAWCNAVMESSAGE

### Raw QZSS L2C and L5 CNAV message

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides the raw QZSS L2C and L5 CNAV message.



The QZSSRAWCNAVMESSAGE log is not output by default. To receive this log, data decoding for QZSSL2C or QZSSL5 must be enabled using the **DATADECODESIGNAL** command (see page 129) for the specific signal.

**Message ID:** 1530

**Log Type:** Collection

**Recommended Input:**

```
log qzssrawcnavmessage onnew
```

**ASCII Example:**

```
#QZSSRAWCNAVMESSAGEA,USB1,0,53.0,SATTIME,2209,512148.000,02000020,20f7,16809;
173,194,30,8b09ea6b7b5a740d69fff865ffff080001a0007fbc07c0431401faf73f0cfff6a100
0e2e9710*db33f7fd
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSRAWCNAVMESSAGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	signal channel	Signal channel providing the bits	Ulong	4	H
3	PRN	QZSS satellite PRN number	Ulong	4	H+4
4	message ID	CNAV message ID	Ulong	4	H+8
5	data	CNAV raw message data	Hex[38]	38	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.146 QZSSRAWEPHEM

### QZSS Raw ephemeris information

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw binary information for subframes one, two and three from the satellite with the parity information removed. Each subframe is 240 bits long (10 words - 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 subframes). This information is preceded by the PRN number of the satellite from which it originated. This message is not generated unless all 10 words from all 3 frames have passed parity.

**Message ID:** 1331

**Log Type:** Asynch

**Recommended Input:**

```
log qzssrawephema onnew
```

**ASCII Example:**

```
#QZSSRAWEPHEMA, USB1, 2, 53.5, SATTIME, 2209, 512160.000, 02000020, 2f9e, 16809;
194, 2209, 514800, 8b02aaa6b884286103800000000000000000000000002797daf00fffffffc35, 8b
02aaa6b90a797ffff1b328093b29e412126fc40eef06dcae784ed7daf7f, 8b02aaa6b98c00e71ba4
f5c3ffd41dc44beb3498bf628cc6ffe0b379105d*44abb439
```

```
#QZSSRAWEPHEMA, USB1, 1, 53.5, SATTIME, 2209, 460800.000, 02000020, 2f9e, 16809;
195, 2209, 464400, 8b02aa9600842861038000000000000000000000000041716100fffeffb8f4, 8b
02aa96010a41a0b414eca437e9fecb6726259d0d3af8caed6ea971617e, 8b02aa96018c02b06152
6d24018e1cfff09e89c63c06da3d1ffe48741e38b*c0feac5d
```

```
#QZSSRAWEPHEMA, USB1, 0, 53.5, SATTIME, 2209, 482400.000, 02000020, 2f9e, 16809;
196, 2209, 486000, 8b02aa9d08862861038000000000000000000000000f85976a70000160d438a, 8b
02aa9d0908591d922d6f9b72b2c9115e26886124d77ccaeb906376a77f, 8b02aa9d098efd72a83c
4f23ff9f18707ab35b4ac0926ce9ffd36d592537*8f7c95d2
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSRAWEPHEM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Satellite PRN number	Ulong	4	H
3	ref week	Ephemeris reference week number	Ulong	4	H+4
4	ref secs	Ephemeris reference time (s)	Ulong	4	H+8
5	subframe1	Subframe 1 data	Hex	30	H+12
6	subframe2	Subframe 2 data	Hex	30	H+42

---

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	subframe3	Subframe 3 data	Hex	30	H+72
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+102
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.147 QZSSRAWSUBFRAME

### Raw QZSS subframe data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw QZSS subframe data.

A raw QZSS subframe is 300 bits in total, 10 words of 30 bits each. This includes the parity 6 bits at the end of each word, for a total of 60 parity bits. Note that in Field #4, the 'data' field below, the 60 parity bits are stripped out and only the raw subframe data remains, for a total of 240 bits. There are two bytes added onto the end of this 30 byte packed binary array to pad out the entire data structure to 32 bytes in order to maintain 4 byte alignment.

**Message ID:** 1330

**Log Type:** Asynch

**Recommended Input:**

```
log qzssrawsubframea onnew
```

**ASCII Example:**

```
#QZSSRAWSUBFRAMEA,USB1,0,50.5,SATTIME,2209,512202.000,02000020,e56b,16809;194,3,8b02aaa6bc0f00e71ba4f5c3ffd41dc44beb3498bf628cc6ffe0b379105d,171*f92b64e0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	QZSSRAWSUBFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	subframe ID	Subframe ID	Ulong	4	H+4
4	data	Raw subframe data	Hex [30]	32 <sup>1</sup>	H+8
5	chan	Signal channel number that the frame was decoded on	Ulong	4	H+40
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
7	[CR][LF]	Sentence terminator	-	-	-

<sup>1</sup>In the binary log case, an additional 2 bytes of padding are added to maintain 4-byte alignment.

## 3.148 RADARSTATUS

### *Emulated Radar Signal Information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides information on the velocity and Emulated Radar (ER) signal.

**Message ID:** 1877

**Log Type:** Synch

**Recommended Input:**

```
log radarstatusa ontime 1
```

**ASCII Example:**

```
#RADARSTATUSA,USB1,0,54.0,FINESTEERING,2209,512323.000,02000020,3a93,16809;
00000000,SOL_COMPUTED,PPP,0.0019,0.0000,0.000000*638616c3
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RADARSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Emulated radar status	Bit field representing current state of radar emulation <i>Table 152: Emulated Radar Status</i> on the next page	Hex	4	H
3	Solution status	Solution status as reported in <b>BESTVEL</b> log on page 535	Enum	4	H+4
4	Velocity type	Velocity type as reported in <b>BESTVEL</b> log on page 535	Enum	4	H+8
5	Hor speed	Horizontal speed over ground (m/sec)	Double	8	H+12
6	Smooth hor speed	Smoother horizontal speed over ground (m/sec)	Enum	4	H+20
7	Frequency	Output frequency (Hz)	Double	8	H+28

Table 152: Emulated Radar Status

Bit	Mask	Description
0	0x01	Feature enabled
1	0x02	Velocity type good
2	0x04	Velocity >1 kph
3-4	0x18	<p>Current response mode being applied will equal the <b>RADARCONFIG</b> command on page 323 response mode parameter unless it is 2. If response mode is set to 2 in the <b>RADARCONFIG</b> command , this will be either 500 or 1000, depending on speed.</p> <p>Mapped as follows:</p> <ul style="list-style-type: none"> <li>- 00: Response mode 1 (no smoothing)</li> <li>- 01: Response mode 500 (500 ms smoothing)</li> <li>- 10: Response mode 1000 (1000 ms smoothing)</li> <li>- 11: Response mode 2000 (2000 ms smoothing)</li> </ul>
5-31	0xFFFFE700	Reserved



## 3.149 RAIMSTATUS

### *RAIM status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides information on Receiver Autonomous Integrity Monitoring (RAIM) status (refer to the **RAIMMODE** command on page 326).

**Message ID:** 1286

**Log Type:** Synch

**Recommended Input:**

```
log raimstatusa ontime 1
```

**ASCII Example:**

```
#RAIMSTATUSA,USB1,0,54.0,FINESTEERING,2209,512358.000,02000020,bf2d,16809;  
DEFAULT,PASS,NOT_AVAILABLE,0.000,NOT_AVAILABLE,0.000,0*3f408598
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAIMSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	RAIM Mode	RAIM mode (refer to <i>Table 153: RAIM Mode Types</i> on the next page)	Enum	4	H
3	Integrity status	Integrity Status (see <i>Table 154: Integrity Status</i> on the next page)	Enum	4	H+4
4	HPL status	Horizontal protection level status (see <i>Table 155: Protection Level Status</i> on page 803)	Enum	4	H+8
5	HPL	Horizontal protection level (m)	Double	8	H+12
6	VPL status	Vertical protection level status (see <i>Table 155: Protection Level Status</i> on page 803)	Enum	4	H+20
7	VPL	Vertical protection level (m)	Double	8	H+24
8	#SVs	Number of excluded satellites	Ulong	4	H+32
9	System	Satellite system (see <i>Table 124: Satellite System</i> on page 654)	Enum	4	H+36

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
10	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest order bytes, interpreted as a USHORT, are the system identifier. For instance, the PRN for GPS or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems.  In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel - 2 is output as 13-2.	Ulong	4	H+40
11	Next offset field = H+36+(#SVs * 8)				
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36 + (#SVs * 8)
13	[CR][LF]	Sentence terminator (ASCII only)			

Table 153: RAIM Mode Types

Binary	ASCII	Description
0	DISABLE	Do not do integrity monitoring of least squares solution
1	USER	User will specify alert limits and probability of false alert
2	DEFAULT	Use NovAtel RAIM (default)
3	APPROACH	Default numbers for non-precision approach navigation modes are used - HAL = 556 m (0.3 nm), VAL = 50 m for LNAV/VNAV
4	TERMINAL	Default numbers for terminal navigation mode are used - HAL = 1 nm, no VAL requirement
5	ENROUTE	Default numbers for enroute navigation mode are used - HAL = 2 nm, no VAL requirement

Table 154: Integrity Status

Binary	ASCII	Description
0	NOT_AVAILABLE	RAIM is unavailable because either there is no solution or because the solution is unique, that is, there is no redundancy
1	PASS	RAIM succeeded. Either there were no bad observations or the bad observations were successfully removed from the solution
2	FAIL	RAIM detected a failure and was unable to isolate the bad observations

Table 155: Protection Level Status

Binary	ASCII	Description
0	NOT_AVAILABLE	When RAIM is not available. For example, immediately after issuing a <b>FRESET</b> command (see page 193) or when there are not enough satellites tracked to have the required redundant observations.
1	PASS	Current protection levels are below alert limits, meaning positioning accuracy requirements are fulfilled. HPL < HAL VPL < VAL
2	ALERT	Current protection levels are above alert limits, meaning required positioning accuracy cannot be guaranteed by RAIM algorithm. HPL ≥ HAL VPL ≥ VAL

## 3.150 RANGE

### *Satellite range information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The RANGE log contains the channel measurements for the currently tracked satellites. When using this log, please keep in mind the constraints noted along with the description.

It is important to ensure that the receiver clock has been set. This can be monitored by the bits in the Receiver Status field of the log header. Large jumps in pseudorange as well as Accumulated Doppler Range (ADR) occur as the clock is being adjusted. If the ADR measurement is being used in precise phase processing, it is important not to use the ADR if the "parity known" flag, in the ch-tr-status field, is not set as there may exist a half (1/2) cycle ambiguity on the measurement. The tracking error estimate of the pseudorange and carrier phase (ADR) is the thermal noise of the receiver tracking loops only. It does not account for possible multipath errors or atmospheric delays.

If multiple signals are being tracked for a given PRN, an entry for each signal, with the same PRN, appears in the RANGE logs. As shown in *Table 156: Channel Tracking Status* on page 806, these entries can be differentiated by bits 21-25, which indicate the signal type of the observation.



For dual antenna receivers, a RANGE\_1 log can be requested to get RANGE data from the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGEA\_1, and for a binary log enter RANGEB\_1.

**Message ID:** 43

**Log Type:** Synch

**Recommended Input:**

```
log rangea ontime 30
```

**ASCII Example:**

```
#RANGEA,USB1,0,54.0,FINESTEERING,2209,512449.000,02000020,5103,16809;154,
26,0,24101771.233,0.199,-126655684.482618,0.012,2806.247,44.4,853.017,1810dc04,
26,0,24101778.285,0.063,-98692757.083922,0.012,2186.688,44.3,852.637,11305c0b,
26,0,24101777.807,0.210,-98692761.337851,0.013,2186.688,43.9,852.897,0230dc0b,
26,0,24101777.726,0.048,-94580569.886827,0.007,2095.792,45.5,890.436,01d0dc04,
2,0,22499844.130,0.095,-118237471.207489,0.006,-2253.363,50.8,948.960,0810dc24,
2,0,22499844.354,0.042,-92133093.288779,0.006,-1755.868,47.8,905.245,01305c2b,
18,0,23410900.110,0.140,-123025123.733624,0.009,3172.589,47.4,948.419,0810dc44,
18,0,23410904.748,0.058,-95863754.520894,0.009,2472.151,45.0,905.239,01305c4b,
18,0,23410905.377,0.134,-95863750.778546,0.009,2472.152,47.8,908.359,0230dc4b,
...
11,0,21627872.790,0.045,-87086480.543534,0.004,-456.096,54.2,909.624,0034df64,
11,0,21627872.930,0.029,-91514607.194262,0.005,-479.336,49.8,908.797,0054df64,
14,0,25543595.901,0.066,-133012227.032812,0.013,3483.877,41.7,942.083,08149fc4,
14,0,25543601.865,0.118,-102853526.267150,0.008,2693.828,45.9,909.810,1034dfc4,
14,0,25543600.620,0.077,-108083356.036078,0.012,2830.744,41.4,909.024,0054dfc4,
23,0,24918187.158,0.026,-97784306.543904,0.005,-1719.866,50.8,904.036,0134dfe4,
23,0,24918183.473,0.056,-105436980.058879,0.008,-1854.495,44.1,908.614,0054dfe4
*b35a6012
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RANGE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	# obs	Number of observations with information to follow <sup>1</sup>	Ulong	4	H
3	PRN/slot	Satellite PRN number of range measurement Refer to <i>PRN Numbers</i> on page 52	Ushort	2	H+4
4	glofreq	(GLONASS Frequency + 7) (see GLONASS Slot and Frequency Numbers section of this manual)	Ushort	2	H+6
5	psr	Pseudorange measurement (m)	Double	8	H+8
6	psr $\sigma$	Pseudorange measurement standard deviation (m)	Float	4	H+16
7	adr	Carrier phase, in cycles (accumulated Doppler range)	Double	8	H+20
8	adr $\sigma$	Estimated carrier phase standard deviation (cycles)	Float	4	H+28
9	dopp	Instantaneous carrier Doppler frequency (Hz)	Float	4	H+32
10	C/No	Carrier to noise density ratio $C/No = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	Number of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see <i>Table 156: Channel Tracking Status</i> on the next page and the example in <i>Figure 15: Channel Tracking Example</i> on the next page)	Ulong	4	H+44
13...	Next PRN offset = H + 4 + (#obs x 44)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 44)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>Satellite PRNs may have multiple lines of observations, one for each signal tracked.

Figure 15: Channel Tracking Example

		N7				N6				N5				N4				N3				N2				N1				N0												
0x		0				8				1				0				9				C				0				4												
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
Binary		0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0									
Data		Channel Assignment		Lock Flag	Digital filtering on signal		Phase Measurement		Primary L1		Signal Type				Grouping		Reserved				Satellite System				Correlator Spacing				Code Locked Flag		Parity Flag		Phase Lock Flag		Channel Number				Tracking State			
Value		Automatic		Lock Out	No Digital Filter		Half Cycle		Primary		L1 C/A				Grouped		Reserved				GPS				PAC				Locked		Known		Locked		Channel 0				L1 Phase Lock Loop			

Table 156: Channel Tracking Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Tracking state	0-23, see Table 157: Tracking State on page 808
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	SV channel number	(n-1) (0 = first, n = last) n depends on the receiver
	5	0x00000020		
	6	0x00000040		
N2	7	0x00000080	Phase lock flag	0 = Not locked, 1 = Locked
	8	0x00000100		
	9	0x00000200		
	10	0x00000400		
	11	0x00000800	Parity known flag	0 = Not known, 1 = Known

Nibble	Bit	Mask	Description	Range Value	
N3	12	0x00001000	Code locked flag	0 = Not locked, 1 = Locked	
	13	0x00002000	Correlator type	0-7, see <i>Table 158: Correlator Type</i> on the next page	
	14	0x00004000			
	15	0x00008000			
N4	16	0x00010000	Satellite system	0 = GPS	4 = BeiDou
	17	0x00020000		1 = GLONASS	5 = QZSS
	18	0x00040000		2 = SBAS	6 = NavIC
	19	0x00080000	3 = Galileo	7 = Other	
N5	20	0x00100000	Grouping	0 = Not grouped, 1 = Grouped	
	21	0x00200000	Signal type (Dependent on satellite system above)	<u>GPS:</u> 0 = L1C/A 5 = L2P 9 = L2P (Y), semi-codeless 14 = L5 (Q) 16 = L1C (P) 17 = L2C (M)  <u>GLONASS:</u> 0 = L1C/A 1 = L2C/A 5 = L2P 6 = L3 (Q)  <u>BeiDou:</u> 0 = B1 (I) with D1 data 1 = B2 (I) with D1 data 2 = B3 (I) with D1 data 4 = B1 (I) with D2 data 5 = B2 (I) with D2 data 6 = B3 (I) with D2 data 7 = B1C (P) 9 = B2a (P) 11 = B2b (I)  <u>Galileo:</u> 2 = E1 (C) 6 = E6B 7 = E6C 12 = E5a (Q) 17 = E5b (Q) 20 = E5AltBOC (Q)  <u>QZSS:</u> 0 = L1C/A 14 = L5 (Q) 16 = L1C (P) 17 = L2C (M) 27 = L6P  <u>SBAS:</u> 0 = L1C/A 6 = L5 (I)  <u>NavIC:</u> 0 = L5 SPS  <u>Other:</u> 19 = L-Band	
	22	0x00400000			
	23	0x00800000			
N6	24	0x01000000	Reserved		
	25	0x02000000			
	26	0x04000000			
	27	0x08000000	Primary L1 channel	0 = Not primary, 1 = Primary	

Nibble	Bit	Mask	Description	Range Value
N7	28	0x10000000	Carrier phase measurement <sup>1</sup>	0 = Half Cycle Not Added 1 = Half Cycle Added
	29	0x20000000	Digital filtering on signal	0 = No digital filter 1 = Digital filter
	30	0x40000000	PRN lock flag <sup>2</sup>	0 = PRN Not Locked Out 1 = PRN Locked Out
	31	0x80000000	Channel assignment	0 = Automatic, 1 = Forced

Table 157: Tracking State

State	Description
0	Idle
1	Sky search
2	Wide frequency band pull-in
3	Narrow frequency band pull-in
4	Phase lock loop
6	Channel steering
7	Frequency lock loop
9	Channel alignment
10	Code search
11	Aided phase lock loop
23	Side peak detection
24	FFT sky search

Table 158: Correlator Type

State	Description
0	N/A
1	Standard correlator: spacing = 1 chip

<sup>1</sup>This bit is zero until the parity is known and the parity known flag (bit 11) is set to 1.

After a loss of lock, there is a half cycle ambiguity on the ADR (carrier phase) until enough navigation data has been decoded to determine the correct phase of the carrier. At the point this is determined, the "parity known" and "half cycle added" flags will get set. If the half cycle flag is set to 1, it indicates that a half cycle was added to the ADR to correct an inverted phase.

<sup>2</sup>A PRN can be locked out using the **LOCKOUT** command.



State	Description
2	Narrow correlator: spacing < 1 chip
3	Reserved
4	Pulse Aperture Correlator (PAC)
5	Narrow PAC
6	Multipath Estimation and Correction (MEC)

Table 159: RINEX Mappings

GNSS System	Frequency Band	Frequency	Observation Codes					RINEX Phase Shift
			Signal Type	Pseudo Range	Carrier Phase	Doppler	Signal Strength	
GPS	L1	1575.42	L1CA	C1C	L1C	D1C	S1C	0
			L1C(P)	C1L	L1L	D1L	S1L	0.25
	L2	1227.6	L2C(M)	C2S	L2S	D2S	S2S	-0.25
			L2P	L2P	C2P	D2P	S2P	0
			L2P(Y)	C2W	L2W	D2W	S2W	0
L5	1176.45	L5(Q)	C5Q	L5Q	D5Q	S5Q	-0.25	
GLONASS	G1	1598.0625-1609.3125	L1CA	C1C	L1C	D1C	S1C	0
	G2	1242.9375-1251.6875	L2CA	C2C	L2C	D2C	S2C	0
			L2P	C2P	L2P	D2P	S2P	0.25
G3	1202.025	L3(Q)	C3Q	L3Q	D3Q	S3Q	0.25	
Galileo	E1	1575.42	E1C	C1C	L1C	D1C	S1C	0.5
	E5a	1176.45	E5a(Q)	C5Q	L5Q	D5Q	S5Q	-0.25
	E5b	1207.14	E5b(Q)	C7Q	L7Q	D7Q	S7Q	-0.25
	E5 (E5a+E5b)	1191.795	E5AltBOC (Q)	C8Q	L8Q	D8Q	S8Q	-0.25
	E6	1278.75	E6B	C6B	L6B	D6B	S6B	0
E6C			C6C	L6C	D6C	S6C	-0.5	
SBAS	L1	1575.42	L1CA	C1C	L1C	D1C	S1C	
	L5	1176.45	L5(I)	C5I	L5I	D5I	S5I	

GNSS System	Frequency Band	Frequency	Observation Codes					RINEX Phase Shift
			Signal Type	Pseudo Range	Carrier Phase	Doppler	Signal Strength	
QZSS	L1	1575.42	L1CA	C1C	L1C	D1C	S1C	0
			L1C(P)	C1L	L1L	D1L	S1L	0.25
	L2	1227.6	L2C(M)	C2S	L2S	D2S	S2S	0
	L5	1176.45	L5(Q)	C5Q	L5Q	D5Q	S5Q	-0.25
	L6	1278.75	L6(P)	C6L	L6L	D6L	S6L	0
BeiDou	B1	1561.098	B1(I)	C2I	L2I	D2I	S2I	0
	B1C	1575.42	B1C(P)	C1P	L1P	D1P	S1P	0.25
	B2	1207.14	B2(I)	C7I	L7I	D7I	S7I	0
	B2a	1176.45	B2a(P)	C5P	L5P	D5P	S5P	0.25
	B2b	1207.14	B2b(I)	C7D	L7D	D7D	S7D	0
	B3	1268.52	B3(I)	C6I	L6I	D6I	S6I	0
NavIC	L5	1176.45	L5SPS	C5A	L5A	D5A	S5A	0

## 3.151 RANGECMP

### *Compressed version of the RANGE log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the RANGE data in a compressed format.



For dual antenna receivers, a RANGECMP\_1 log can be requested to get RANGECMP data from the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGECMPA\_1, and for a binary log enter RANGECMPB\_1.

**Message ID:** 140

**Log Type:** Synch

**Recommended Input:**

```
log rangecmpa ontime 10
```


**ASCII Example:**

```
#RANGECMPA, USB1, 0, 54.5, FINESTEERING, 2209, 512487.000, 02000020, 9691, 16809; 154,
04dc1018c2e70a40d6a37b0b9fb702f5531a616f20030000,
0b5c3011667f08400ea47b0bf32b559f501a546f20030000,
0bdc3002667f08f00aa47b0bb027559f631a5d6fe0020000,
04dcd001dd2408400aa47b0bc1e906de301a0e7440030000,
24dc1008122cf77f198ebc0ab3e387f221025f7bc0030000,
2b5c3001041fff96f1a8ebc0a0e1324812002e87560030000,
44dc1008fa5d0c6073fa260b0a8d9fac32124d7b80030000,
4b5c300100a309f097fa260b0cc9aaca4012e87520030000,
4bdc300200a309309dfa260bcaccaaca32124b7680030000,
44dcd001303c09e0c3fa260b1b378e8710127976e0030000,
449c1042175e0c607efa260b4b7e9fac30127a7680030000,
64dc10180ea8f00f2091680bcb3e8a81420c3e7b40030000,
6b5c3011410bf43f4991680b5b9218a9400c187600030000,
6bdc3002420bf47f4b91680b189518a9430c557640030000,
...
24de15181dcb00b02e50bd1457da6be063c25179e0020000,
649f14086ea6fd7fdfa6500a20df2dc9200b2b7be0030000,
64df3400ca2efe2fe2a6500ad00de6ce100b7476e0030000,
64df54002e17fe9fe3a6500a0b35518b200b5976c0030000,
849f1408c398f57f19a0dc0b79e06fc7400cef7920030000,
84df3410a5f4f73f29a0dc0bac969887210c6e7680030000,
84df5400f18bf74f28a0dc0bcb6ce1b9300c547640030000,
c49f14081d9a0d101c032b0c6dfa6994500e837ae0020000,
c4df341087840a504a032b0c8a2e24e0420e7a7620030000,
c4df5400900d0bf040032b0c9d7c6b90510e6176c0020000,
e49f1408a208f76f49cfe30b1bbac2c22017117880030000,
e4dff400a9f3f6ff35cfe30ba97395b04217527660030000,
e4df34013c3ef92f51cfe30bf503eeaa1017c175e0030000,
e43f74011a11f91f49cfe30b8bebfa832017547680030000,
e4df5400e2b6f86f34cfe30bbeb814b63017547600030000*5bf4bd0a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RANGECMP header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#obs	Number of satellite observations with information to follow	Ulong	4	H
3	1st range record	Compressed range log in format of <i>Table 160: Range Record Format (RANGECMP only)</i> below	Hex	24	H+4
4	Next rangecmp offset = H+4 (#obs x 24)				
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 24)
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 160: Range Record Format (RANGECMP only)

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
Channel Tracking Status	Channel tracking status word	0-31	32	see <i>Table 156: Channel Tracking Status</i> on page 806	-
Doppler Frequency	Instantaneous carrier Doppler frequency	32-59	28	1/256	Hz
Pseudorange (PSR)	Pseudorange measurement	60-95	36	1/128	m

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
ADR	<p>ADR (Accumulated Doppler Range) is calculated as follows:</p> $\text{ADR\_ROLLS} = (\text{RANGECMP\_PSR} / \text{WAVELENGTH} + \text{RANGECMP\_ADR}) / \text{MAX\_VALUE}$ <p>Round to the closest integer</p> <p>IF (ADR_ROLLS ≤ 0)            ADR_ROLLS = ADR_ROLLS - 0.5            ELSE            ADR_ROLLS = ADR_ROLLS + 0.5</p> <p>At this point integerise ADR_ROLLS            CORRECTED_ADR = RANGECMP_ADR - (MAX_VALUE * ADR_ROLLS)            where            ADR has units of cycles            WAVELENGTH = 0.1902936727984 for GPS L1            WAVELENGTH = 0.2442102134246 for GPS L2            MAX_VALUE = 8388608</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  GLONASS satellites emit L1 and L2 carrier waves at a satellite-specific frequency, refer to the GLONASS section of <i>An Introduction to GNSS</i> available on our website at <a href="http://novatel.com/an-introduction-to-gnss">novatel.com/an-introduction-to-gnss</a> </div>	96-127	32	1/256	cycles
StdDev-PSR	Pseudorange measurement standard deviation	128-131	4	See Table 161: StdDev-PSR Values on the next page	m
StdDev-ADR	ADR measurement standard deviation	132-135	4	(n+1)/512	cycles
PRN/Slot	Refer to <i>PRN Numbers</i> on page 52	136-143	8	1	-

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
Lock Time	Number of seconds of continuous tracking (no cycle slipping) This field is constrained to a maximum value of 2,097,151 which represents a lock time of 65535.96875 s (2097151/32).	144-164	21	1/32	s
C/No	Carrier to noise density ratio The C/No is constrained to a value between 20-51 dB-Hz. Thus, if it is reported that C/No = 20 dB-Hz, the actual value could be less. Likewise, if it is reported that C/No = 51, the true value could be greater.	165-169	5	(20+n)	dB-Hz
GLONASS Frequency number	GLONASS Frequency number	170-175	n+7	1	
Reserved		176-191	16		

Table 161: StdDev-PSR Values

Code	StdDev-PSR (m)
0	0.050
1	0.075
2	0.113
3	0.169
4	0.253
5	0.380
6	0.570
7	0.854
8	1.281
9	2.375
10	4.750
11	9.500
12	19.000
13	38.000

<b>Code</b>	<b>StdDev-PSR (m)</b>
14	76.000
15	152.000

## 3.152 RANGECMP2

### *Compressed version of the RANGE log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the RANGE data in a compressed format to handle more channels and different channel types than the RANGECMP log.



For dual antenna receivers, a RANGECMP2\_1 log can be requested to get RANGECMP2 data from the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGECMP2A\_1, and for a binary log enter RANGECMP2B\_1.

**Message ID:** 1273

**Log Type:** Synch

**Recommended Input:**

```
log rangecmp2a ontime 10
```

**Example:**

```
#RANGECMP2A,USB1,0,55.0,FINESTEERING,2209,512524.000,02000020,1fe3,16809;2256,0
01a00a09fbc856c0540e1ffff6dd8c80c004201d1ffe4ffff4578e87c983d82d1ffe5ffff0dd8c8
75308082d1ffe7ffff0d5a667570f102faff01020010405f8592fb2fe1ffff2dbda41eb81980160
0e4ffff055ce61e0806001600021200542192852b0650e1ffff2dbc860f50e280ffffe4ffff0579
6655c83a820000e5ffff0dbc865f4800820000e7ffff0d3f62ad3033040900efffff097cc61e009
681fcff030c00ecf8b58551f83fe1ffff6dda06508021800d00e4ffff457948a400aa800c00e5ff
...
5cdfe3fe1ffff291f8417b01900d5ffe3ffff0d7f001b285f00d6ffedffff0d3e041ef86680deff
3c0c600078ef05cafa3fe1ffff29796829181500e5ffe3ffff4d9c6449685f80f5ffedffff0d79c
647307b80f6ff3e0e6018ff1306cc0630e1ffff2977e80f203a011100e3ffff4db80669d069830b
00edffff0d954856b8c88200003f176024e7f2057efb5fe1ffff293d0466980501dffff3ffff0db
be63bf81b02d6fff4ffff0d3f226fd0e380e2fff5ffff031ca45db89d80e9ffedffff0d784636b8
3900eaff*e438ba70
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RANGECMP2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	# bytes	Number of bytes in the compressed binary range data <sup>1</sup>	Uchar	4	H

<sup>1</sup>Maximum is 2880 bytes for 120 channels; maximum 5760 for 240 channels.



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
3	RangeData	Compressed binary range data in the format of <i>Table 162: Satellite Block of the Range Record Format (RANGECMP2 only)</i> below and <i>Table 163: Signal Block of the Range Record Format (RANGECMP2 only)</i> on the next page <sup>1</sup>	Uchar	#bytes	H+4
4	xxxx	32-bit CRC (ASCII and binary only)	Hex	4	H+4+ (# bytes)
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 162: Satellite Block of the Range Record Format (RANGECMP2 only)**

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
SV Channel Number	Receiver SV channel number	0-7	8	-	-
Satellite Identifier	Satellite identifier specific to the satellite system Refer to <i>PRN Numbers</i> on page 52	8-15	8	-	-
GLONASS Frequency Identifier	GLONASS frequency channel offset by +7	16-19	4	(7+Ch#)	-
Satellite System Identifier	Defined in <i>Table 124: Satellite System</i> on page 654	20-24	5	-	Enum
Reserved		25	1	-	-
Pseudorange Base	Pseudorange base value to be combined with PSR Diff and Phasorange Diff in each following Signal Block	26-54	29	1	m
Doppler Base	Doppler base value to be combined with the Scaled Doppler Diff value in each following Signal Block	55-75	21	1	Hz
Number of Signal Blocks	The number of Signal Blocks to follow this Satellite Block. See <i>Table 163: Signal Block of the Range Record Format (RANGECMP2 only)</i> on the next page for Signal Block definition	76-79	4	-	-

<sup>1</sup>The compressed binary range data is organized into satellite blocks, one for each satellite. Each satellite block is followed by a variable number of signal blocks corresponding to the same satellite. For example, a Satellite Block for GPS PRN 17 may be followed by two Signal Blocks for the L1 C/A and L2C signals.

**Table 163: Signal Block of the Range Record Format (RANGECMP2 only)**

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
Signal Type	Defined in <i>Table 167: Signal Type (only in RANGECMP2)</i> on page 821	0-4	5	-	Enum
Phase Lock	Phase Lock: 0 = Not locked, 1 = Locked	5	1	-	Bool
Parity Known	Parity Known: 0 = Not known, 1 = Known	6	1	-	Bool
Code Lock	Code Lock: 0 = Not locked, 1 = Locked	7	1	-	Bool
Locktime	Time of continuous tracking with no cycle slips. The locktime value saturates at a maximum of 131071 ms	8-24	17	1	ms
Correlator Type	Correlator type: (see <i>Table 158: Correlator Type</i> on page 808)	25-28	4	-	Enum
Primary Signal	Primary signal: 0 = Not primary, 1 = Primary	29	1	-	Bool
Carrier Phase Measurement	Carrier phase measurement: 0 = Half cycle not added, 1 = Half cycle added	30	1	-	Bool
Reserved		31	1	-	-
C/No	Carrier to Noise density ratio	32-36	5	(20 + n)	dB-Hz
StdDev PSR	Pseudorange Standard Deviation (defined <i>Table 164: Std Dev PSR Scaling</i> on the next page)	37-40	4	Bit Field in <i>Table 164: Std Dev PSR Scaling</i> on the next page	-
StdDev ADR	Carrier-Phase Standard Deviation (defined <i>Table 165: Std Dev ADR Scaling</i> on page 820)	41-44	4	Bit Field in <i>Table 165: Std Dev ADR Scaling</i> on page 820	-
PSR Diff	Pseudorange Diff to be combined with Pseudorange base i.e., PSR = PSRBase + PSRDiff/128	45-58	14	1/128	m (unsigned)

Data	Description	Bits first to last	Length (bits)	Scale Factor	Units
Phaserange Diff	Phaserange Diff to be combined with Pseudorange Base i.e., ADR = PSRBase + PhaserangeDiff/2048	59-78	20	1/2048	m (unsigned)
Scaled Doppler Diff <sup>1</sup>	Doppler Diff to be combined with Doppler Base. Note that all Doppler values are scaled to the L1/E1 equivalent value. (refer to <i>Table 166: L1/E1/B1 Scaling</i> on the next page) i.e., Doppler = (DopplerBase + ScaledDopplerDiff/256)/L1ScaleFactor	79-95	17	1/256	Hz (signed)

Table 164: Std Dev PSR Scaling

PSR Std Dev Bit Field Value	Represented Std Dev (m)
0	0.02
1	0.03
2	0.045
3	0.066
4	0.099
5	0.148
6	0.22
7	0.329
8	0.491
9	0.732
10	1.092
11	1.629
12	2.43
13	3.625
14	5.409
15	>5.409

<sup>1</sup>The Scaled Doppler Diff field is the only field in the RANGECP2 that should be parsed as Two's Complement. The most significant byte (MSB) determines whether the number will be positive (< 0x7) or negative (> 0x7). Two's complement should be applied prior to AND, right bit shift computations.

**Table 165: Std Dev ADR Scaling**

ADR Std Dev Bit Field Value	Represented Std Dev (cycles)
0	0.00391
1	0.00521
2	0.00696
3	0.00929
4	0.01239
5	0.01654
6	0.02208
7	0.02947
8	0.03933
9	0.05249
10	0.07006
11	0.09350
12	0.12480
13	0.16656
14	0.22230
15	>0.22230

**Table 166: L1/E1/B1 Scaling**

Satellite System	Signal Type	L1/E1/B1 Scale Factor
GPS	L1CA	1.0
	L2Y	154/120
	L2C	154/120
	L5Q	154/115
GLONASS	L1CA	1.0
	L2CA	9/7
	L2P	9/7
SBAS	L1CA	1.0
	L5I	154/115

Satellite System	Signal Type	L1/E1/B1 Scale Factor
Galileo	E1	1.0
	E5A	154/115
	E5B	154/118
	AltBOC	154/116.5
	E6C	154/125
	E6B	154/125
QZSS	L1CA	1.0
	L2C	154/120
	L5Q	154/115
	L6P	154/125
LBAND	LBAND	1.0
BDS	B1	1.0
	B1C	1526/1540
	B2	1526/1180
	B2a	1526/1150
	B2b	1526/1180
	B3	1526/1240
NAVIC	L5SPS	1.0

Table 167: Signal Type (only in RANGECMP2)

Satellite System	Signal Type	Value
GPS	L1CA	1
	L2Y	4
	L2CM	5
	L5Q	7
	L1C	15
GLONASS	L1CA	1
	L2CA	3
	L2P	4
	L3Q	6

Satellite System	Signal Type	Value
SBAS	L1CA	1
	L5I	2
Galileo	E1C	1
	E5AQ	2
	E5BQ	3
	AltBOCQ	4
	E6C	5
	E6B	12
QZSS	L1CA	1
	L2CM	3
	L5Q	4
	L1C	8
	L6P	11
LBAND	LBAND	1
BDS	B1D1I	1
	B1D2I	2
	B2D1I	3
	B2D2I	4
	B3D1I	13
	B3D2I	16
	B1CP	19
	B2AP	20
	B2B_I	21
NAVIC	L5SPS	1

### 3.153 RANGECMP4

#### *Highly compressed version of the RANGE log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the RANGE data in a more heavily compressed format compared to the RANGECMP2 log.



For dual antenna receivers, a RANGECMP4\_1 log can be requested to get RANGECMP4 data from the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGECMP4A\_1, and for a binary log enter RANGECMP4B\_1.



1. Channels that have been manually assigned to a PRN with **ASSIGN** or **ASSIGNALL** are not reported in the **RANGECMP4** log.
2. L-Band channels are not reported in the **RANGECMP4** log.
3. The **RANGECMP4** log should not be logged with an offset. The RANGECMP4 implementation is dependent on getting measurements at the even second. The allowable triggers are the same as other measurement logs such as **RANGE**, **RANGECMP** and **RANGECMP2**.
4. **RANGECMP4** is a complex log. For more information about decoding the **RANGECMP4** log, refer to *Example of Bit Parsing a RANGECMP4 Log* on page 1193.

**Message ID:** 2050

**Log Type:** Synch

**Recommended Input:**

```
log rangecmp4a ontime 10
```

**Example:**

```
#RANGECMP4A,USB1,0,54.0,FINESTEERING,2209,512549.000,02000020,fb0e,16809;1843,e
70012090a5300000000b280e3bcf3c77be740d2fb1aa652a93f153ffce7b40f22ddbdc060018dc
fe090078834fb2f11b379bb20b008429d3dfbe17915a80ca1b80ff1ffd9ee45e02a02700fd3ffcb
6b712ef93b55da3b7fffc4a12eedb8f09a0c0ece1df0228716fe6370208717f0b20dfbd08d60abc
a5fe1b0298746f1317599ab27898fe574c724cb71f224d01f066bf0310e3de085605fc5cfd0f00c
4bbaf19ebf6fc563064803636d24bdccf6887808923c0fecf785f337202025080fbbf077ec41613
...
f0c0164bc97e1ded4165bc259007451d5d7dcccb74b201d9006033e8745f345cffef00019fbf047e
84342030ef010e82b4f710577970a10a5809201a3ec40d2bea90aaedff6be9967ffb3e626603604
700b440107c9235f88f9c019c00edf00114436060ec9f03e4bf87b8d1805701c00a00d8ef43709f
8488d89005203102780100000000000008d8087b9f657632597403427023097a17f38fbab2f165
09ffd030031ee67e67ce09cfe9f09f4b52782e4ff721bc0f90f*4fb4c740
```


Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RANGECMP4 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	# bytes	Number of bytes in the compressed binary Range Data.	Uchar	4	H
3	Range Data	<p>The compressed binary range data is organized into satellite system blocks which break down into measurement blocks for each active signal within each system. Refer to the following tables for more details about this format:</p> <p><i>Table 168: Header</i> on the next page (sent once)</p> <p><i>Table 169: Satellite and Signal Block</i> on the next page (sent once per satellite system bit set to 1 in the GNSS Field found in <i>Table 168: Header</i> on the next page)</p> <p><i>Table 170: Measurement Block Header</i> on page 826 (sent once for each bit set to 1 in the Satellites Field found in <i>Table 169: Satellite and Signal Block</i> on the next page)</p> <p><i>Table 171: Primary Reference Signal Measurement Block</i> on page 827 and <i>Table 172: Secondary Reference Signals Measurement Block</i> on page 828, or <i>Table 173: Primary Differential Signal Measurement Block</i> on page 829 and <i>Table 174: Secondary Differential Signals Measurement Block</i> on page 831, Measurement Block (sent for each bit set to 1 in the Included Signals Field for a given satellite found in <i>Table 169: Satellite and Signal Block</i> on the next page)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The byte data is received MSB first so each group of bytes (as defined by the number of needed bits) must be swapped prior to processing. </div>	Uchar	# bytes	H+4
4	xxxx	32-bit CRC (ASCII only)	Hex	4	H+4+(# bytes)
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



Table 168: Header


Data Name	Description	Bits	Scale Factor
GNSS	<p>Indicates which satellite system data is encoded and in what order. When the bit is set the satellite data is included. Data for each system is encoded sequentially:</p> <p>Bit 0 = GPS            Bit 1 = GLONASS            Bit 2 = SBAS            Bit 5 = Galileo            Bit 6 = BeiDou            Bit 7 = QZSS            Bit 9 = NavIC</p>	16	1
Bit Sum:		16	
 This block is sent once per message			

Table 169: Satellite and Signal Block

Data Name	Range	Description	Bits	Scale Factor
Satellites	0... 1.84467E+19	<p>Indicates which satellites are present for this system and their order in the message. Each PRN is represented by a bit. (Bit 0 = PRN 1, Bit 1 = PRN 2, ...)</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>GLONASS Satellite: This value represents the Slot ID of the satellite (range of 1 to 24 where Bit 0 = Slot ID 1). In the event the Slot ID is between 43 and 63, the actual GLONASS Slot ID has not yet been determined and has been replaced with a temporary Slot ID calculated using the GLONASS Frequency Number. See the GLONASS Frequency Number field in <i>Table 170: Measurement Block Header</i> on the next page for more details.</li> <li>SBAS Satellite PRNs 120 to 158 are offset by 120. (Bit 0 = PRN 120, Bit 1 = 121, ...)</li> <li>SBAS Satellite PRNs 183 to 191 are offset by 130</li> <li>QZSS Satellite PRNs are offset by 193</li> </ul>	64	1


Data Name	Range	Description	Bits	Scale Factor
Signals	0... 65535	Indicates which signals are present for this system and their order in the message. Each signal is represented by a bit as defined in <i>Table 175: Signal Bit Mask</i> on page 832.	16	1
Included Signals	0... mxn	A two dimensional field to tell the decoder which signals are present for each of the satellites.  <b>m</b> = The number of rows equals the number of bits set to 1 found in the Satellites field. (Maximum number of PRNs in the satellite system)  <b>n</b> = The number of columns equals the number of bits set to 1 found in the Signals field. (Maximum number of Signals in the satellite system)	mxn	
Bit Sum:			80 + mxn	
 This block is sent once for each bit set to 1 in the GNSS field found in <i>Table 168: Header</i> on the previous page.				

Table 170: Measurement Block Header

Data Name	Range	Description	Bits	Scale Factor
Data Format Flag	0... 1	Identifies what type of Measurement Block will be used: 0 = Reference ( <i>Table 171: Primary Reference Signal Measurement Block</i> on the next page and <i>Table 172: Secondary Reference Signals Measurement Block</i> on page 828) 1 = Differential ( <i>Table 173: Primary Differential Signal Measurement Block</i> on page 829 and <i>Table 174: Secondary Differential Signals Measurement Block</i> on page 831)	1	1
Ref Data Block ID	0... 7	This ID identifies to which reference data the Differential Data is linked. This value is incremented by 1 each time a new Reference Measurement Block is used.	3	1



Data Name	Range	Description	Bits	Scale Factor
GLONASS Frequency Number	0... 20 (-7 to +13)	<p>These bits are only present for GLONASS satellites in the Reference Data. This represents the GLONASS Frequency Number which identifies the frequency offset of the carrier frequency. The value will appear as a number between 0 and 20 which directly translates into a frequency offset number between -7 to +13.</p> <p>If the GLONASS Slot ID is unknown, a temporary Slot ID for this satellite will be set between 43 and 63 based on the GLONASS Frequency Number: PRN = 63 – GLONASS Frequency Number</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The GLONASS Frequency Number used in this calculation is the 0 to 20 value, not the adjusted -7 to +13 value. </div>	5	1
Bit Sum:			4 (Non-GLONASS)	9 (GLONASS)
<div style="border: 1px solid black; padding: 5px;">  This block is sent once for each bit set to 1 in the Satellites field found in <i>Table 169: Satellite and Signal Block</i> on page 825. </div>				

Table 171: Primary Reference Signal Measurement Block

Data Name	Range	Description	Bits	Scale Factor
Parity Flag	0... 1	0 = Parity Unknown 1 = Parity Known	1	1
½ Cycle Flag	0... 1	0 = Half Cycle Not Added 1 = Half Cycle Added	1	1
C/No	0... 63.95	C/No	11	0.05 dBHz
Lock Time	0... 15	The Lock Time – See <i>Table 176: Lock Time</i> on page 832	4	1
Pseudorange Std Dev	0... 15	The Pseudorange Standard Deviation (m) – See <i>Table 178: Pseudorange Std Dev</i> on page 834	4	
ADR Std Dev	0... 15	The ADR Standard Deviation (cycles) – See <i>Table 177: ADR Std Dev</i> on page 833	4	



Data Name	Range	Description	Bits	Scale Factor
Primary Pseudorange	0... 68719476.74	The Pseudo Range of the 1st signal (Signals field in <i>Table 169: Satellite and Signal Block</i> on page 825). If this value equals $(2^{37}-1) = 137438953471$ , it represents a signal that is not locked.	37	0.0005 m
PhaseRange – Primary Pseudorange	±419.4303	(2's Complement) If this value equals $-(2^{23}-1) = -4194304$ , it represents the signal is not locked.	23	0.0001 m
Primary Doppler	±3355.4431	(2's Complement) If this value equals $-(2^{26}-1) = -33554432$ , it represents an invalid Doppler.	26	0.0001 m/s
Bit Sum:			111	
<div style="border: 1px solid black; padding: 10px; margin: 10px 0;">  This block is sent once for the first bit set to 1 in the Included Signals field found in <i>Table 169: Satellite and Signal Block</i> on page 825. For any bits set to 1 after the first bit set to 1, refer to <i>Table 172: Secondary Reference Signals Measurement Block</i> below. </div> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;">  This table is for Reference blocks only, as indicated by the Data Format Flag (see <i>Table 170: Measurement Block Header</i> on page 826). </div>				

Table 172: Secondary Reference Signals Measurement Block

Data Name	Range	Description	Bits	Scale Factor
Parity Flag	0... 1	0 = Parity Unknown 1 = Parity Known	1	1
½ Cycle Flag	0... 1	0 = Half Cycle Not Added 1 = Half Cycle Added	1	1
C/No Indicator	0... 63.95	C/No	11	0.05 dBHz
Lock Time	0... 15	The Lock Time – See <i>Table 176: Lock Time</i> on page 832	4	1
Pseudorange Std Dev	0... 15	The Pseudorange Standard Deviation (m) – See <i>Table 178: Pseudorange Std Dev</i> on page 834	4	
ADR Std Dev	0... 15	The ADR Standard Deviation (cycles) – See <i>Table 177: ADR Std Dev</i> on page 833	4	



Data Name	Range	Description	Bits	Scale Factor
Pseudorange – Primary Signal Pseudorange	±262.1435	(2's Complement) If this value equals $-(2^{20}-1) = -524288$ , it indicates the signal is not locked.	20	0.0005 m
Phaserange – Pseudorange	±419.4303	(2's Complement) If this value equals $-(2^{23}-1) = -4194304$ , it indicates the signal is not locked.	23	0.0001 m
Doppler – Primary Doppler	±0.8191	(2's Complement) If this value equals $-(2^{14}-1) = -8192$ , it indicates an invalid Doppler.	14	0.0001 m/s
Bit Sum:			82	
 This block is sent once for each bit set to 1 after the first bit set to 1 in the Included Signals field found in <i>Table 169: Satellite and Signal Block</i> on page 825.				
 This table is for Reference blocks only, as indicated by the Data Format Flag (see <i>Table 170: Measurement Block Header</i> on page 826).				

Table 173: Primary Differential Signal Measurement Block

Data Name	Range	Description	Bits	Scale Factor
Parity Flag	0... 1	0 = Parity Unknown 1 = Parity Known	1	1
½ Cycle Flag	0... 1	0 = Half Cycle Not Added 1 = Half Cycle Added	1	1
C/No	0... 63.95	C/No	11	0.05 dBHz
Lock Time	0... 15	The Lock Time – See <i>Table 176: Lock Time</i> on page 832	4	1
Pseudorange Std Dev	0... 15	The Pseudorange Standard Deviation (m) – See <i>Table 178: Pseudorange Std Dev</i> on page 834	4	
ADR Std Dev	0... 15	The ADR Standard Deviation (cycles) – See <i>Table 177: ADR Std Dev</i> on page 833	4	





Data Name	Range	Description	Bits	Scale Factor
Pseudorange – Predicted Pseudorange	±131.0715	(2's Complement) If this value equals $-(2^{19}-1) = -262144$ , it indicates a signal that is not locked.  The Predicted Pseudorange = reference pseudorange plus (the reference doppler x time difference between the reference log and the differential log). The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	19	0.0005 m
Phaserange – Predicted Phaserange	±3.2767	(2's Complement) If this value equals $-(2^{16}-1) = -32768$ , it indicates the signal is not locked.  The Predicted Phaserange = reference phaserange plus (the reference doppler x time difference between the reference log and the differential log). The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	16	0.0001 m
Doppler – Reference Doppler	±13.1071	(2's Complement) If this value equals $-(2^{18}-1) = -131072$ , it indicates an invalid Doppler.  The Reference Doppler is the Doppler for that PRN and for that signal from the Reference log. The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	18	0.0001 m/s
Bit Sum:			78	
<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;">  This block is sent once for each bit set to 1 after the first bit set to 1 in the Included Signals field found in <i>Table 169: Satellite and Signal Block</i> on page 825. For any bits set to 1 after the first bit set to 1, refer to <i>Table 174: Secondary Differential Signals Measurement Block</i> on the next page. </div> <div style="border: 1px solid black; padding: 10px;">  This table is for Differential blocks only, as indicated by the Data Format Flag (see <i>Table 170: Measurement Block Header</i> on page 826). </div>				

Table 174: Secondary Differential Signals Measurement Block

Data Name	Range	Description	Bits	Scale Factor
Parity Flag	0... 1	0 = Parity Unknown 1 = Parity Known	1	1
½ Cycle Flag	0... 1	0 = Half Cycle Not Added 1 = Half Cycle Added	1	1
C/No	0... 63.95	C/No	11	0.05 dBHz
Lock Time	0... 15	The Lock Time – See <i>Table 176: Lock Time</i> on the next page	4	1
Pseudorange Std Dev	0... 15	The Pseudorange Standard Deviation (m) – See <i>Table 178: Pseudorange Std Dev</i> on page 834	4	1
ADR Std Dev	0... 15	The ADR Std Dev (cycles)– See <i>Table 177: ADR Std Dev</i> on page 833	4	1
Pseudorange – Predicted Pseudorange	±131.0715	(2's Complement) If this value equals $-(2^{19}-1) = -262144$ , it indicates the signal is not locked.  The Predicted Pseudorange = reference pseudorange plus (the reference doppler x time difference between the reference log and the differential log). The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	19	0.0005 m
Phaserange – Predicted Phaserange	±3.2767	(2's Complement) If this value equals $-(2^{16}-1) = -32768$ , it indicates the signal is not locked.  The Predicted Phaserange = reference phaserange plus (the reference doppler x time difference between the reference log and the differential log). The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	16	0.0001 m
Doppler – Reference Doppler	±13.1071	(2's Complement) If this value equals $-(2^{14}-1) = -8192$ , it indicates an invalid Doppler.  The Reference Doppler is the Doppler for that PRN and for that signal from the Reference log. The Reference log and Differential logs used must contain matching Ref Data Block ID references ( <i>Table 170: Measurement Block Header</i> on page 826).	14	0.0001 m/s

Data Name	Range	Description	Bits	Scale Factor
			Bit Sum:	74
 This block is sent once for each bit set to 1 after the first bit set to 1 in the Included Signals field found in <i>Table 169: Satellite and Signal Block</i> on page 825.				
 This table is for Differential blocks only, as indicated by the Data Format Flag (see <i>Table 170: Measurement Block Header</i> on page 826).				

**Table 175: Signal Bit Mask**

	GPS	GLONASS	SBAS	Galileo	BeiDou	QZSS	NavIC
Bit 1	L1CA	L1CA	L1CA	E1	B1I	L1CA	L5SPS
Bit 2			L5I	E5A	B1GEO		
Bit 3		L2CA		E5B	B2I	L2C	
Bit 4	L2Y	L2P		ALTBOC	B2GEO	L5Q	
Bit 5	L2C			E6C	B3I		
Bit 6	L2P	L3			B3GEO		
Bit 7	L5Q				B1CP		
Bit 8						L1C	
Bit 9					B2AP		
Bit 10							
Bit 11					B2bl	L6P	
Bit 12				E6B			
Bit 13							
Bit 14							
Bit 15	L1C						

**Table 176: Lock Time**

Indicator (i)	Minimum Lock Time (ms)	Range of Indicated Lock Times (t represents the Lock Time) (ms)
0	0	$0 \leq t < 16$



Indicator (i)	Minimum Lock Time (ms)	Range of Indicated Lock Times (t represents the Lock Time) (ms)
1	16	$16 \leq t < 32$
2	32	$32 \leq t < 64$
3	64	$64 \leq t < 128$
4	128	$128 \leq t < 256$
5	256	$256 \leq t < 512$
6	512	$512 \leq t < 1024$
7	1024	$1024 \leq t < 2048$
8	2048	$2048 \leq t < 4096$
9	4096	$4096 \leq t < 8192$
10	8192	$8192 \leq t < 16384$
11	16384	$16384 \leq t < 32768$
12	32768	$32768 \leq t < 65536$
13	65536	$65536 \leq t < 131072$
14	131072	$131072 \leq t < 262144$
15	262144	$262144 \leq t$

Table 177: ADR Std Dev

ADR Std Dev (cycles)	
0	$\leq 0.0039$
1	$\leq 0.0052$
2	$\leq 0.0070$
3	$\leq 0.0093$
4	$\leq 0.0124$
5	$\leq 0.0165$
6	$\leq 0.0221$
7	$\leq 0.0295$
8	$\leq 0.0393$
9	$\leq 0.0525$
10	$\leq 0.0701$

<b>ADR Std Dev (cycles)</b>	
11	$\leq 0.0935$
12	$\leq 0.1248$
13	$\leq 0.1666$
14	$\leq 0.2223$
15	$> 0.2223$

**Table 178: Pseudorange Std Dev**

<b>Pseudorange Std Dev (m)</b>	
0	$\leq 0.020$
1	$\leq 0.030$
2	$\leq 0.045$
3	$\leq 0.066$
4	$\leq 0.099$
5	$\leq 0.148$
6	$\leq 0.220$
7	$\leq 0.329$
8	$\leq 0.491$
9	$\leq 0.732$
10	$\leq 1.092$
11	$\leq 1.629$
12	$\leq 2.430$
13	$\leq 3.625$
14	$\leq 5.409$
15	$> 5.409$

## 3.154 RANGEGPSL1

### *L1 version of the RANGE log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log is identical to the **RANGE** log (see page 804) except that it only includes GPS L1 C/A observations.

**Message ID:** 631

**Log Type:** Synch

**Recommended Input:**

```
log rangegpsl1a ontime 30
```

**ASCII Example:**

```
#RANGEGPSL1A,USB1,0,55.0,FINESTEERING,2209,512583.000,02000020,5862,16809;
10,
26,0,24030889.100,0.199,-126283196.274792,0.012,2752.781,44.3,987.017,1810dc04,
2,0,22557616.108,0.123,-118541064.672676,0.007,-2277.646,48.6,1082.960,
0810dc24,
18,0,23330317.018,0.126,-122601657.447823,0.009,3147.617,48.4,1082.419,
0810dc44,
12,0,23997160.481,0.151,-126105903.891743,0.009,-3940.361,46.8,1081.940,
1810dc64,
5,0,20822721.607,0.083,-109424141.788836,0.005,1522.260,52.0,1082.908,1810dc84,
20,0,20953423.978,0.087,-110110984.182224,0.007,-827.676,51.6,1082.829,
0810dca4,
29,0,20734204.928,0.077,-108958983.599878,0.006,1473.416,52.6,1082.708,
1810dcc4,
31,0,24809616.479,0.239,-130375401.308516,0.014,-2112.775,42.8,1082.271,
0810dd04,
9,0,25163112.825,0.207,-132233041.316998,0.013,-1279.863,44.0,1065.218,
1810dd24,
25,0,21543476.977,0.093,-113211728.999577,0.008,-2507.055,50.9,1082.449,
0810ddc4
*51b3705b
```



Since the RANGEGPSL1 log includes only GPS L1 C/A observations, it is smaller in size than the RANGE log which contains entries for multiple systems and signals. Use the RANGEGPSL1 log when data throughput is limited and you are only interested in GPS L1 C/A range data. For GPS L1 only models, RANGE and RANGEGPSL1 logs are identical.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RANGEGPSL1 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	# obs	Number of L1 observations with information to follow	Long	4	H
3	PRN	Satellite PRN number of range measurement (1-32)	Ushort	2	H+4
4	Reserved		Ushort	2	H+6
5	psr	Pseudorange measurement (m)	Double	8	H+8
6	psr std	Pseudorange measurement standard deviation (m)	Float	4	H+16
7	adr	Carrier phase, in cycles (accumulated Doppler range)	Double	8	H+20
8	adr std	Estimated carrier phase standard deviation (cycles)	Float	4	H+28
9	dopp	Instantaneous carrier Doppler frequency (Hz)	Float	4	H+32
10	C/No	Carrier to noise density ratio $C/No = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	Number of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see <i>Table 156: Channel Tracking Status</i> on page 806)	Ulong	4	H+44
13...	Next PRN offset = $H + 4 + (\#obs \times 44)$				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 44)
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.155 RAWALM

### *Raw GPS L1 C/A Almanac data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the undecoded GPS almanac subframes as received from the satellite. For more information about Almanac data, refer to *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).

**Message ID:** 74

**Log Type:** Asynch

**Recommended Input:**

```
log rawalma onchanged
```


**ASCII Example:**

```
#RAWALMA, USB1, 0, 54.0, SATTIME, 2209, 512250.000, 02000020, cc1b, 16809;
2210, 61440.000, 36,
1, 8b0284a683b741617b0f1da7fd7000a10d0a6fe69624de9116263f2fffd1,
2, 8b0284a6863442a7520f0f5ffd5600a10cf76c306ac62dda1c751caa000e,
3, 8b0284a688b74321ef0f1464fd6300a10c9d9a02d626d19ce8f3d8dfff9d,
4, 8b0284a68b36440ed50f0bfcfd4000a10c67c614e78719785cff7fea0026,
5, 8b0284a68db74530d60f0bf0fd5600a10d5998511a2996db8266e6f5000b,
6, 8b0284a6903646150b0f1d37fd6d00a10d816f9046dd08a41f9de527009f,
7, 8b0284a692b44783b70f059efd5c00a10cf2ef9113a38e0ffaf04e2a000a,
8, 8b0284a69536483d030f0da3fd2200a10da643d11906d3826f02aef70006,
9, 8b0284a697b44911130f0770fd3900a10bfcc3e4b94c6129844e81d3003e,
10, 8b0284a69a354a3e480f1443fd6500a10d0e99e60d9a5e76c36848c9ff92,
11, 8b0284a69cb44b02f50f0d78fd5bffa10c457199187328c76e423efc003b,
12, 8b0284a69f354c46e70f11a3fd4900a10c0d1ce6f23456f0e271e5e3ffda,
13, 8b0284a6a1b44d306c0f10d1fd4800a10ce3ca2eb72584145db6b328005d,
14, 8b0284a6a4374e0f210f06d6fd3e00a10ce41b91cf7e35bbe16d55f10018,
15, 8b0284a6a6b54f73050ff7bffd2100a10daebf79842c82e548176cf80022,
16, 8b0284a6a93550685a0f1190fd4600a10cd81da8911daac8963f89beffee,
17, 8b0284a6abb7516ea50f1790fd3400a10bfff47360bc326976e454250002d,
18, 8b0284a6ae345212f30f1319fd6000a10c85702a367b999023da9d16ffab,
19, 8b0284a6b0b7534ba10f16cdfd3400a10d3049099c53ddb4cf43e7140037,
20, 8b0284a6b3365427670f0074fd4b00a10c2a939b0180ea134062d6430011,
21, 8b0284a6b5b755c6f10f0ba5fd5600a10db26c197fd946af759ce6150007,
22, 8b0284a6b836566d600f0c0ffd4100a10cffc80d8db484ba69e9f1220040,
23, 8b0284a6bab457118d0f1111fd6200a10d4698de217a73b2f78a52fdffe7,
24, 8b0284a6bd3658662b0ffad0fd4c00a10d14ec22c7212fb622a9131fffd0,
25, 8b0284a685b05956250f0983fd4400a10d0519bb6728daa2d5b6282a0034,
26, 8b0284a688315a38d60ffd0bfd3600a10d9917bfe30fd6f2ba7af21b0023,
27, 8b0284a68ab35b56710f1355fd2a00a10cd244a3931bbd7f70647e1a0055,
0, 00000000000000000000000000000000000000000000000000000000000000,
29, 8b0284a692325d119e0f18e8fd3500a10d4147bfe95beea17518fcbcff7,
30, 8b0284a694b35e2fa30ffbf2fd4d00a10dc1f00c20923e8ffa7e58ba001a,
31, 8b0284a697325f55780f085dfd5d00a10d86f051630fe442e7e443e80018,
32, 8b0284a699b16030d70f09edfd3d00a10c3fc45da6a2d3218b29e1efffc,
51, 8b0284a6bfb4730fa200000000000000fc000000000000000000155557,
55, 8b0284a6ab3177444d3039272f4c20394a2d4e503a2e4854384c4f504c03,
```

56,8b0284a6adb0781303fefe3908fefeccccff2ffffffa0fa2128907120001,  
63,8b0284a6bf327fb9bcababbbca9ca9ac9999cbbbb9abab00000fc000000\*f7f7e59e



The OEM7 family of receivers automatically saves almanacs in their Non-Volatile Memory (NVM), therefore creating an almanac boot file is not necessary.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWALM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	ref week	Almanac reference week number	Ulong	4	H
3	ref secs	Almanac reference time (ms)	GPSTime	4	H+4
4	#subframes	Number of subframes to follow	Ulong	4	H+8
5	svid	SV ID (satellite vehicle ID) A value between 1 and 32 for the SV ID indicates the PRN of the satellite. Any other values indicate the page ID.  See section 20.3.3.5.1.1, Data ID and SV ID, of IS-GPS-200L for more details. To obtain copies of IS-GPS-200, refer to the GPS website ( <a href="http://www.gps.gov/technical/icwg/">http://www.gps.gov/technical/icwg/</a> ).	Ushort	2	H+12
6	data	Subframe page data	Hex	30	H+14
7...	Next subframe offset = H+12+(#subframe x 32)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#subframes x 32)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.156 RAWCNAVFRAME

### Raw GPS CNAV frame data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides raw frame data from signals which contain the CNAV message (L2C, L5).



The RAWCNAVFRAME log is not output by default. To receive this log, data decoding for L2C or L5 must be enabled using the **DATADECODESIGNAL** command (see page 129) for the specific signal.

**Message ID:** 1066

**Log Type:** Asynch

**Recommended Input:**

```
log rawcnavframea onnew
```

**ASCII Example:**

```
#RAWCNAVFRAMEA,USB1,0,46.0,SATTIME,2209,512724.000,02000020,ee56,16809;
193,18,30,8b49ea6e7adde57d742cca59ffa72001ee0ffe82cffc7fe01303fefe3908fefca1000
628aca0*40babc24
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWCNAVFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	signal channel	Signal channel providing the bits	Ulong	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	frame ID	frame ID	Ulong	4	H+8
5	data	Raw frame data	Hex[38]	38	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.157 RAWEPHEM

### *Raw GPS L1 C/A ephemeris*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw binary information for subframes one, two and three from the GPS satellite L1 C/A signal with the parity information removed. Each subframe is 240 bits long (10 words - 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 subframes). This information is preceded by the PRN number of the satellite from which it originated. This message is not generated unless all 10 words from all 3 frames have passed parity.

Ephemeris data whose Time of Ephemeris (TOE) is older than six hours is not shown. Multiple logs are output, one for each GPS satellite with collected ephemeris information.

**Message ID:** 41

**Log Type:** Asynch

**Recommended Input:**

```
log rawephema onnew
```

**ASCII Example:**

```
#RAWEPHEMA, USB1, 30, 54.0, SATTIME, 2209, 496800.000, 02000020, 58ba, 16809;1, 2209, 5040
00, 8b0284a1b8a52850000724918b913e21a92dc6ee0b217b0c00ffb72fac04, 8b0284a1b92b21f
ac82899520ec7b7fb31061550921d09a10d62b87b0c7c, 8b0284a1b9adff7d74db71f3ffaa2840e
d6e0fe024cddf1effadc82106c4*0ea3cca1
```

```
#RAWEPHEMA, USB1, 29, 54.0, SATTIME, 2209, 512730.000, 02000020, 58ba, 16809;2, 2209, 5184
00, 8b0284a6e82728500051d2718b913e21a92dc6eeda427e90000001aa606d, 8b0284a6e8aa42f
8f82d56addef337fa840a758d8d1c42a10d56437e907e, 8b0284a6e92f00777124a07100c7275ca
76f0e76c62e1e8bffad9c42067f*5581fdb5
```

```
#RAWEPHEMA, USB1, 28, 54.0, SATTIME, 2209, 496800.000, 02000020, 58ba, 16809;3, 2209, 5040
00, 8b0284a1b8a52850000724918b913e21a92dc6ee04167b0c00ff88e02542, 8b0284a1b92b160
57c326224c51f2604ed021ebf6a0db8a10cf92a7b0c7c, 8b0284a1b9adffef9ef8203a001627ab6
c56208226d033b4ffa63a160f90*73505674
```

...

```
#RAWEPHEMA, USB1, 2, 54.0, SATTIME, 2209, 489600.000, 02000020, 58ba, 16809;30, 2209, 4968
00, 8b02849f60a52850000724918b913e21a92dc6ee0847794a00fff3bad3c7, 8b02849f612b470
46033f40bb132b0041702f88bd31993a10e34eb794a7d, 8b02849f61ad0032f5025ef6004b26264
dc7108e922777edffa6fe47025e*9763ae6b
```

```
#RAWEPHEMA, USB1, 1, 54.0, SATTIME, 2209, 512730.000, 02000020, 58ba, 16809;31, 2209, 5184
00, 8b0284a6e82728500051d2718b913e21a92dc6eee34b7e9000fff2e8bc16, 8b0284a6e6294b0
5352e16795ad6fd044f055757e01ac0a10df3d67e907e, 8b0284a6e6afffe1f5457d19ffd026ed0
e00105f0fe105f5ffab7a4bf840*7838d9ba
```

```
#RAWEPHEMA, USB1, 0, 54.0, SATTIME, 2209, 497010.000, 02000020, 58ba, 16809;32, 2209, 5040
00, 8b0284a1ca242850000724918b913e21a92dc6ee01317b0c00ffb4ef6fcd, 8b0284a1c82b31f
9a5348cc6e9b79bfa4d030cf2370e54a10cba597b0c7f, 8b0284a1c8ad0066c953c01700082705b
1a61f09a2dcac5bffa65331e786*f299f05c
```





A way to use only one receiver and achieve better than 1 metre accuracy is to use precise orbit and clock files. Three types of GPS ephemeris, clock and earth orientation solutions are compiled by an elaborate network of GNSS receivers around the world all monitoring the satellite characteristics. IGS rapid orbit data is processed to produce files that correct the satellite clock and orbit parameters. Since there is extensive processing involved, these files are available on a delayed schedule from the US National Geodetic Survey at: [www.ngs.noaa.gov/orbits](http://www.ngs.noaa.gov/orbits)

Precise ephemeris files are available today to correct GPS data which was collected a few days ago. All you need is one GNSS receiver and a computer to process on. Replace the ephemeris data with the precise ephemeris data and post-process to correct range values.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWEPHEM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	ref week	Ephemeris reference week number	Ulong	4	H+4
4	ref secs	Ephemeris reference time (s)	Ulong	4	H+8
5	subframe1	Subframe 1 data	Hex[30]	30	H+12
6	subframe2	Subframe 2 data	Hex[30]	30	H+42
7	subframe3	Subframe 3 data	Hex[30]	30	H+72
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+102
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.158 RAWGPSSUBFRAME

### Raw GPS L1 C/A subframe data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw GPS L1 C/A subframe data.

A raw GPS L1 C/A subframe is 300 bits in total. This includes the parity bits which are interspersed with the raw data ten times in six bit chunks, for a total of 60 parity bits. Note Field #5, below, has these 60 parity bits stripped out and only the raw subframe data remains, for a total of 240 bits.

**Message ID:** 25

**Log Type:** Asynch

**Recommended Input:**

```
log rawgpssubframea onnew
```

**ASCII Example:**

```
#RAWGPSSUBFRAMEA,USB1,0,47.0,SATTIME,2209,512808.000,02000020,0457,16809;2,2,4,8b0284a6eeb37a4134a0f0469296b5ee80dc7da6ce1ada076697d4296ac6,2*3f833bb6
```



The RAWGPSSUBFRAME log can be used to receive the data bits with the parity bits stripped out. Alternately, you can use the **RAWGPSWORD** log (see page 843) to receive the parity bits in addition to the data bits.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWGPSSUBFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	decode #	Frame decoder number	Long	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	subframe id	Subframe ID	Ulong	4	H+8
5	data	Raw subframe data	Hex[30]	32 <sup>1</sup>	H+12
6	signal channel	Signal channel number that the frame was decoded on	Ulong	4	H+44
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary log case, an additional 2 bytes of padding are added to maintain 4-byte alignment.

## 3.159 RAWGPSWORD

### *Raw GPS L1 C/A navigation word*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This message contains the framed GPS L1 C/A raw navigation words. Each log contains a new 30 bit navigation word (in the least significant 30 bits), plus the last 2 bits of the previous word (in the most significant 2 bits). The 30 bit navigation word contains 24 bits of data plus 6 bits of parity. The GPS reference time stamp in the log header is the time the first bit of the 30 bit navigation word was received. Only navigation data that has passed parity checking appears in this log. One log appears for each PRN being tracked every 0.6 seconds if logged ONNEW or ONCHANGED.

**Message ID:** 407

**Log Type:** Asynch

**Recommended Input:**

```
log rawgpsworda onnew
```

**ASCII Example:**

```
#RAWGPSWORDA,USB1,0,55.5,FINESTEERING,2209,512874.069,02000020,9b16,16809;
29,22c0a120*ed4326e9
```



The RAWGPSWORD log can be used to receive the parity bits in addition to the data bits. Alternately, you can use the RAWGPSSUBFRAME log which already has the parity bits stripped out

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWGPSWORD header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	nav word	Raw navigation word	Hex[4]	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.160 RAWSBASFRAME

### Raw SBAS frame data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw SBAS frame data of 226 bits (8-bit preamble, 6-bit message type and 212 bits of data but without a 24-bit CRC). Only frame data with a valid preamble and CRC are reported.

**Message ID:** 973

**Log Type:** Asynch

**Recommended Input:**

```
log rawsbasframea onnew
```

**ASCII Example:**

```
#RAWSBASFRAMEA,USB1,0,54.0,SATTIME,2209,512912.000,02000020,58e4,16809;166,131,2,c6089ffc001ffc008000000011ffc001ffdfffffffe797a15739bb95c0,166*2725a69f
```



The RAWSBASFRAME log output contains all the raw data required for an application to compute its own SBAS correction parameters.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWSBASFRAME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	decode #	Frame decoder number	Ulong	4	H
3	PRN	SBAS satellite PRN number	Ulong	4	H+4
4	SBAS frame ID	SBAS frame ID	Ulong	4	H+8
5	raw frame data	Raw SBAS frame data. There are 226 bits of data and 6 bits of padding.	Hex[29]	32 <sup>1</sup>	H+12
6	signal channel	Signal channel number that the frame was decoded on.	Ulong	4	H+44
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

## 3.161 RAWSBASFRAME2

### Raw SBAS frame data 2

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the raw SBAS frame data of 226 bits (8-bit preamble, 6-bit message type and 212 bits of data but without a 24-bit CRC). It also contains the transmitted frequency. Only frame data with a valid preamble and CRC are reported.

**Message ID:** 2185

**Log Type:** Asynch

**Recommended Input:**

```
log rawsbasframe2a onnew
```

**ASCII Example:**

```
#RAWSBASFRAME2A,USB1,0,48.5,SATTIME,2209,512947.000,02000020,b39f,16809;135,160,1,0,0,3,9a0c9ffc00ffe9ffc033ffc005ffdfc121ffc00403ba2b9b97bbb95c0*4e57856f
```



The **RAWSBASFRAME2** log output contains all the raw data required for an application to compute its own SBAS correction parameters.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWSBASFRAME2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PRN	SBAS satellite PRN number	Ulong	4	H
3	signal channel	Signal channel number that the frame was decoded on	Ulong	4	H+4
4	SBAS Signal Source	Identifies the source of the SBAS signal: 1 – SBASL1CA 2 – SBASL5I	Uchar	1	H+8
5	SBAS Preamble Type	Identifies what preamble was used when tracking the SBAS signal: 0 – SBASL1CA 8-bit Preamble 1 – SBASL5I Preamble	Uchar	1	H+9
6	Reserved		Ushort	2	H+10

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	SBAS frame ID	SBAS frame ID	Ulong	4	H+12
8	data	Raw SBAS frame data. There are 226 bits of data and 6 bits of padding.	Hex[29]	32 <sup>1</sup>	H+16
9	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

---

<sup>1</sup>In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

## 3.162 REFSTATION

### *Base station position and health*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the ECEF Cartesian position of the base station as received through the RTCMV3 message. It also features a time tag, the health status of the base station and the station ID. This information is set at the base station using the **FIX** command (see page 182) and the **DGPSTXID** command (see page 138). See *Figure 12: The WGS84 ECEF Coordinate System* on page 540 for a definition of the ECEF coordinates.

The base station health, Field #6, may be one of 8 values (0 to 7). Values 0 through 5 indicate the scale factor that is multiplied with the satellite UDRE one-sigma differential error values. Below are values 0 to 5 and their corresponding UDRE scale factors:

0: 1 (Health OK) 0.75 2: 0.5 3: 0.3 4: 0.2 5: 0.1

The base station health field only applies to RTCM base stations. A value of 6 means the base station transmission is not monitored and a value of 7 means that the base station is not working.

**Message ID:** 175

**Log Type:** Asynch

**Recommended Input:**

```
log refstationa onchanged
```

**ASCII Example:**

```
#REFSTATIONA,USB1,0,68.0,FINESTEERING,2211,233731.221,02000020,4e46,16809;00000000,-1632851.222,-3662162.724,4944899.271,0,NOVATELX,"K250"*380be9f1
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	REFSTATION header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	status	Status of the base station information (see <i>Table 179: Base Station Status</i> on the next page)	Ulong	4	H
3	x	ECEF X value (m)	Double	8	H+4
4	y	ECEF Y value (m)	Double	8	H+12
5	z	ECEF Z value (m)	Double	8	H+20
6	health	Base station health, see the description at the start of this section	Ulong	4	H+28
7	stn type	Station type (see <i>Table 180: Base Station Type</i> on the next page)	Enum	4	H+32

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	stn ID	Base station ID	Char[5]	8 <sup>1</sup>	H+36
9	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 179: Base Station Status

Bit #	Mask	Description	Bit = 0	Bit = 1
0	0x00000001	Validity of the base station	Valid	Invalid

Table 180: Base Station Type

Binary	ASCII	Description
0	NONE	Base station is not used
1 - 3	Reserved	
4	RTCMV3	Base station is RTCMV3



The REFSTATION log can be used for checking the operational status of a remotely located base station. You can verify that the base station is operating properly without traveling to it. This is especially useful for RTK work on long baselines.

<sup>1</sup>In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.



### 3.163 REFSTATIONINFO

#### *Base Station position information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This is an extended version of the REFSTATION log with latitude, longitude and ellipsoidal height of the base station in WGS84. In addition to the base station position, ARP height, antenna model name and antenna serial number are available if provided by the base station only through RTCMV3.

**Message ID:** 1325

**Log Type:** Asynch

**Recommended Input:**

```
log refstationinfoa onchanged
```

**ASCII Example:**

```
#REFSTATIONINFOA,USB1,0,65.0,FINESTEERING,2211,233851.222,02000020,d38f,16809;
51.150391053,-114.030694086,1081.916599573,WGS84,0.000,0,NOVATELX,"K250","",""
*bc1b1a21
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	REFSTATIONINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	H	0	
2	latitude	Latitude (degrees)	Double	8	H
3	longitude	Longitude (degrees)	Double	8	H+8
4	height	Ellipsoidal Height (m)	Double	8	H+16
5	datum	Datum ID number 61 = WGS84 63 = USER (default = WGS84)	Enum	4	H+24
6	ARP height	Base Antenna ARP (m)	Float	4	H+28
7	health	Base Station Health, see <i>Table 179: Base Station Status</i> on the previous page	Ulong	4	H+32
8	Ref Stn Type	Base Station Type, see ( <i>Table 180: Base Station Type</i> on the previous page)	Enum	4	H+36

---

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	stn ID	Base Station ID	Char[5]	8 <sup>1</sup>	H+40
10	Ant Model	Base Antenna Model Name	Char[32]	32	H+48
11	Ant Serial	Base Antenna Serial Number	Char[32]	32	H+80
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+112
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

---

<sup>1</sup>In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

## 3.164 ROVERPOS

### *Position using ALIGN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

ALIGN generates distance and bearing information between a master and rover receiver. This log outputs the position information of the rover when using the ALIGN feature. This log can only be output from a Y ALIGN model and can be output at both Master and Rover ends.



You must have an ALIGN capable receiver to use this log.



- ALIGN is useful for obtaining the relative directional heading of a vessel/body, separation heading between two vessels/bodies, or heading information with moving base and pointing applications.
- The log can be output at the Y model Rover only if it is receiving the RTCAREFEXT message from the Master. The log can be output at any Master if the Master is receiving HEADINGEXTB from the Rover.
- ROVERPOS is dependent on the output frequency of the RTCAREFEXT message from the master to the rover.
- On dual antenna receivers, the **ROVERPOS** log outputs the position for the secondary antenna input.

**Message ID:** 1052

**Log Type:** Asynch

**Recommended Input:**

log roverposa onchanged

**ASCII Example:**

```
#ROVERPOSA,COM1,0,21.5,FINESTEERING,1544,340322.000,02000008,7453,4655;
SOL_COMPUTED,NARROW_INT,51.11605565964,-114.03854655975,1055.8559,-16.9000,
WGS84,0.0130,0.0122,0.0206,"RRRR",0.0,0.0,13,12,12,11,0,0,0,0*635b3a1c
```



Asynchronous logs, such as ROVERPOS, should only be logged ONCHANGED or ONNEW otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	ROVERPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol stat	Solution Status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position Type see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	lat	Rover WGS84 latitude (degrees)	Double	8	H+8
5	long	Rover WGS84 longitude (degrees)	Double	8	H+16
6	hgt	Rover MSL Height (m)	Double	8	H+24
7	undulation	Undulation (m)	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER (default = WGS84)	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	long $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Rover ID (default = "RRRR")	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14	Reserved		Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#obs	Number of satellites above elevation mask angle	Uchar	1	H+66
18	#multi	Number of satellites with multi-frequency signals above elevation mask angle	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20			Uchar	1	H+69
21			Uchar	1	H+70
22			Uchar	1	H+71

---

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	1	H+72
24	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 3.165 RTCMV3 Standard Logs

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

NovAtel's RTCMv3 logs are implementations of the messages described by the RTCM SC-104 committee's "Differential GNSS (Global Navigation Satellite Systems) Services – Version 3" standard. These messages are primarily intended to support RTK operations. They are also an alternative raw data format to NovAtel's proprietary messages.

The RTCMv3 logs can be divided into several categories that are described below. An RTK base station must minimally transmit one or more observable message, together with one or more station and antenna message. The **GENERATERTKCORRECTIONS** command on page 201 illustrates an appropriate set of messages and is an easy way to configure logging.

### Example Input:

```
interfacemode com2 none RTCMV3
fix position 51.1136 -114.0435 1059.4
thisantennatype NOV702
log com2 rtkm1006 ontime 10
log com2 rtkm1230 ontime 10
log com2 rtkm1033 ontime 10 2
log com2 rtkm1004 ontime 1
log com2 rtkm1012 ontime 1
```

### Example Input using MSM4 messages:

```
interfacemode com1 none rtkmv3 off
thisantennatype nov850
fix position 51.1136 -114.0435 1059.4
log com1 rtkm1006 ontime 10
log com1 rtkm1230 ontime 10
log com1 rtkm1033 ontime 10 2
log com1 rtkm1074 ontime 1
log com1 rtkm1084 ontime 1
log com1 rtkm1094 ontime 1
log com1 rtkm1124 ontime 1
```

### 3.165.1 Legacy Observable Messages

The legacy observable messages contain GPS and GLONASS code and phase observables. The *extended* messages additionally contain the C/N0.

**Table 181: Legacy Observable Messages**

Log Name	Message ID	Description
RTCM1001	772	GPS L1-only observables, basic
RTCM1002	774	GPS L1-only observables, extended
RTCM1003	776	GPS L1/L2 basic observables, basic
RTCM1004	770	GPS L1/L2 basic observables, extended
RTCM1009	885	GLONASS L1-only observables, basic
RTCM1010	887	GLONASS L1-only observables, extended
RTCM1011	889	GLONASS L1/L2 basic observables, basic
RTCM1012	891	GLONASS L1/L2 basic observables, extended

### 3.165.2 MSM Observable Messages

The Multiple Signal Messages (MSM) are observable messages for all current GNSS systems. They provide a standardized framework for message content and are designed to support future systems and signals.



Sending legacy (1001-1004 and 1009-1012) and MSM messages in the same stream can cause problems for remote RTK users and is not recommended.

Each GNSS system has a set of seven MSM types numbered from 1 to 7. The MSM type for each GNSS system provides the same generic information. Generally, as the MSM number increases, more information is available in the messages. For example, MSM1 for each GNSS system provides the code measurements for the system, while MSM3 provides both the code and phase.

The information encoded in each MSM variant is described in *Table 182: MSM Type Descriptions* below for the descriptions of each of the seven MSM types. For RTK operations, MSM3 is minimally recommended.

**Table 182: MSM Type Descriptions**

Message	Description
MSM1	Provides the code measurements.
MSM2	Provides the phase measurements.
MSM3	Provides the data from MSM1 (code) and MSM2 (phase) in a single message.
MSM4	Provides all the data from MSM3 (code and phase) and adds the CNR measurements.
MSM5	Provides all the data from MSM4 (code, phase and CNR) and adds the doppler measurements.
MSM6	Provides the same information as MSM4, but has extended resolution on the measurements.
MSM7	Provides the same information as MSM5, but has extended resolution on the measurements.

*Table 183: MSM Log Names* on the next page lists the MSM message name and *Table 184: MSM Message IDs* on the next page lists the message IDs.

Table 183: MSM Log Names

Message	GPS	GLONASS	Galileo	QZSS	BeiDou
MSM1	RTCM1071	RTCM1081	RTCM1091	RTCM1111	RTCM1121
MSM2	RTCM1072	RTCM1082	RTCM1092	RTCM1112	RTCM1122
MSM3	RTCM1073	RTCM1083	RTCM1093	RTCM1113	RTCM1123
MSM4	RTCM1074	RTCM1084	RTCM1094	RTCM1114	RTCM1124
MSM5	RTCM1075	RTCM1085	RTCM1095	RTCM1115	RTCM1125
MSM6	RTCM1076	RTCM1086	RTCM1096	RTCM1116	RTCM1126
MSM7	RTCM1077	RTCM1087	RTCM1097	RTCM1117	RTCM1127

Table 184: MSM Message IDs

Message	GPS	GLONASS	Galileo	QZSS	BeiDou
MSM1	1472	1479	1486	1648	1592
MSM2	1473	1480	1487	1649	1593
MSM3	1474	1481	1488	1650	1594
MSM4	1475	1482	1489	1651	1595
MSM5	1476	1483	1490	1652	1596
MSM6	1477	1484	1491	1653	1597
MSM7	1478	1485	1492	1654	1598

### 3.165.3 Station and Antenna Messages

The station and antenna messages listed in *Table 185: Station and Antenna Messages* on the next page provide the base station's coordinates and hardware. Remote RTK users require this information so that they can position themselves relative to a base station.

- Message Type 1005 provides the Earth-Centered, Earth-Fixed (ECEF) coordinates of the Antenna Reference Point (ARP). The ARP is an explicit physical point on the antenna, typically the center of its base. It is related to the antenna phase center from where the measurements are emitted via the Phase Center Offsets (PCOs). The PCOs can be set using the **THISANTENNAPCO** command (see page 426) or **THISANTENNATYPE** command (see page 428). If the PCOs are not set, then the coordinates transmitted by Message types 1005 and 1006 will be those that the receiver is fixed to by the **FIX** command (see page 182).
- Message Type 1006 is the same as 1005 but additionally provides the antenna height. This value is always set to zero by the receiver firmware.
- Message Type 1007 provides the base station antenna type. Conventionally, the antenna name from the International GNSS Service (IGS) is used. The antenna name can be set using the **THISANTENNATYPE** command (see page 428).
- Message Type 1008 is the same as 1007 but additionally provides the antenna serial number. The serial number is always set to null by the receiver firmware.



- Message Type 1033, like message types 1007 and 1008, also provides the antenna information. Message type 1033 additionally provides the receiver type and firmware version. The primary use of this information is to more-easily enable RTK rovers to fix their GLONASS ambiguities. This information is filled automatically and appropriately by the receiver firmware.

For a receiver operating as an RTK base station, the recommended messages to transmit are 1006 and 1033. With these messages remote RTK users have all the information describing the base station.

**Table 185: Station and Antenna Messages**

Log Name	Message ID	RTCM Message Type	Description
RTCM1005	765	1005	Stationary RTK Base Station Antenna Reference Point (ARP)
RTCM1006	768	1006	Stationary RTK Base Station ARP with Antenna Height
RTCM1007	852	1007	Extended Antenna Descriptor and Setup Information
RTCM1008	854	1008	Extended Antenna Reference Station Description and Serial Number
RTCM1033	1097	1033	Receiver and antenna descriptors

### 3.165.4 Ephemeris Messages

The ephemeris messages listed in *Table 186: Ephemeris Messages* below provide the satellite ephemerides. For RTK operations this information is optional, as RTK rovers will be downloading their own ephemerides directly from the satellites.

There are two messages for each ephemeris type. For the messages logged ONTIME (e.g. LOG RTCM1019 ONTIME 10) a single satellite's ephemeris is output at each ONTIME interval. The ephemerides will be cycled through in numerical order. For the messages logged ONCHANGED (e.g., LOG RTCM1019ASYNC ONCHANGED), new or changed ephemerides will be output as soon as they are available.

**Table 186: Ephemeris Messages**

Log Name	Message ID	RTCM Message Type	Description
RTCM1019	893	1019	GPS Ephemerides, logged ONTIME
RTCM1019ASYNC	2088	1019	GPS Ephemerides, logged ONCHANGED
RTCM1020	895	1020	GLONASS Ephemerides, logged ONTIME
RTCM1020ASYNC	2089	1020	GLONASS Ephemerides, logged ONCHANGED
RTCM1042	2171	1042	BeiDou Ephemerides, logged ONTIME
RTCM1042ASYNC	2170	1042	BeiDou Ephemerides, logged ONCHANGED
RTCM1044	2177	1044	QZSS Ephemerides, logged ONTIME
RTCM1044ASYNC	2176	1044	QZSS Ephemerides, logged ONCHANGED

Log Name	Message ID	RTCM Message Type	Description
RTCM1045	2173	1045	Galileo F/NAV Ephemerides, logged ONTIME
RTCM1045ASYNC	2172	1045	Galileo F/NAV Ephemerides, logged ONCHANGED
RTCM1046	2175	1046	Galileo I/NAV Ephemerides, logged ONTIME
RTCM1046ASYNC	2174	1046	Galileo I/NAV Ephemerides, logged ONCHANGED

### 3.165.5 GLONASS Bias Information Message

Message 1230 communicates the GLONASS Code-Phase Bias (CPB) information. This information allows RTK rovers to compensate for differences in biases between themselves and the base station, enabling or improving the resolution of GLONASS ambiguities.

CPB values for all GLONASS FDMA signals available in the data stream of both MSM and Legacy (1081-1088 and 1009-1012) are mandatory for full GLONASS interoperability.

With some base stations, message 1033 is not sufficient for full GLONASS interoperability. The message 1033 was not originally designed to compensate for CPB and therefore message 1230 has been developed to improve GLONASS interoperability and should be used to assure full interoperability.

**Table 187: GLONASS Bias Information Message**

Log Name	Message ID	RTCM Message Type	Description
RTCM1230	1655	1230	GLONASS L1 and L2 Code-Phase Biases

## 3.166 RTKASSISTSTATUS

### *RTK ASSIST status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log provides information on the state of RTK ASSIST.

RTK ASSIST operates in two modes: coast and full assist. The RTKASSISTSTATUS log reports which mode is currently available. Coast mode is available as soon as the RTK ASSIST corrections are received from the L-Band satellite, while full assist mode requires a convergence period. In coast mode, position error growth during RTK correction outages is slightly worse than in full assist mode and RTK will not resume following a full signal outage until after RTK corrections are restored. Full assist gives the lowest position error growth during RTK correction outages, and makes it possible for RTK to resume even if there are complete GNSS signal outages during the RTK ASSIST period.

The RTK ASSIST ACTIVE state reported in the RTKASSISTSTATUS log is also reported in the RTKPOS and BESTPOS extended solution status field. See *Table 90: Extended Solution Status* on page 527.

The RTKASSISTSTATUS log reports the time remaining in the RTK ASSIST ACTIVE state. Once RTK ASSIST becomes active, the remaining time will count down from the time out set by the **RTKASSISTTIMEOUT** command (see page 335).

The corrections age reported in the RTKASSISTSTATUS log should typically be below 30 seconds. If the age exceeds this value, then L-Band tracking is likely degraded. The most likely cause of degraded L-Band tracking are obstructions between the antenna and the L-Band satellite.

**Message ID:** 2048

**Log Type:** Synch

**Recommended Input:**

```
log rtkassiststatusa ontime 5
```

**ASCII Example:**

```
#RTKASSISTSTATUSA,USB1,0,55.5,FINESTEERING,2209,513133.000,02000020,80fe,16809;  
INACTIVE,UNAVAILABLE,0.0,13.0*4966edec
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKASSISTSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	State	State: INACTIVE (0) ACTIVE (1)	Enum	4	H

Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	Mode	Mode: UNAVAILABLE (0) COAST (1) ASSIST (2)	Enum	4	H+4
4	Remaining time	Time remaining in seconds	Float	4	H+8
5	Corrections age	Age of the RTK ASSIST corrections in seconds. Maximum value of 120 seconds.	Float	4	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.167 RTKDATUMINFO

### *Datum information for the RTK position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution in the **RTKPOS** log (see page 866).

**Message ID:** 2304

**Log Type:** Asynch

**Recommended Input:**

```
log rtkdatuminfoa onchanged
```

**ASCII Example:**

```
#RTKDATUMINFOA,USB1,0,66.5,FINESTEERING,2211,234000.000,02000020,0b0c,16809;
"UNKNOWN",0,0.000,SERVICE_DETAILS_UNKNOWN*08717033
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the <b>RTKPOS</b> log.	Char[32]	variable	H
3	epsg_code	EPSG code of datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year) Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation Status. See <i>Table 85: Transformation Status</i> on page 518.	Enum	4	variable
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
7	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.168 RTKDOP

### *DOP values for the satellites used in the RTK solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

The RTKDOP log contains the Dilution Of Precision (DOP) values for the satellites being used in the RTK solution. Note that unlike the **PSRDOP** log (see page 770), the RTKDOP log is synchronous. DOP values will be calculated at the requested rate, up to a maximum rate of 1 Hz.

DOP values are a measure of the solution strength. Essentially, the DOPs reflect the geometry of the satellites used in the solution. Solutions with good counts of well-distributed satellites will have low DOPs and should be accurate and reliable. Solutions with fewer or poorly-distributed satellites will have high DOPs and be less accurate and reliable. As a rough guideline, PDOP values less than 4 imply a solution with reasonable geometry.

There can be many reasons for high DOP values. The most common reason is that there are obstructions limiting satellite visibility. Even if satellites are visible and being tracked they might still not be used in the solution if, for example, they are unhealthy or there aren't corrections available for them. The **RTKSATS** log (see page 869) will inform which satellites are being tracked and explain why a tracked satellite is not used in the solution.

The DOPs do not consider that different satellites or signals will be weighted differently in the solution. Therefore, they do not completely reflect the solution quality. Ultimately, the standard deviations reported in the **RTKPOS** log (see page 866) are the best reflection of the solution accuracy.

**Message ID: 952**

**Log Type: Synch**

**Recommended Input:**

```
log rtkdopa ontime 10
```

**ASCII Example:**

```
#RTKDOPA,USB1,0,66.5,FINESTEERING,2211,234163.000,02000020,b42b,16809;
1.3390,1.0580,0.6760,0.8610,0.5340,5.0,16,17,3,6,12,19,2,24,14,1,46,57,38,47,
45,55,56*6757f996
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKDOP header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	HTDOP	Horizontal and time DOP	Float	4	H+12

Field	Field type	Description	Format	Binary Bytes	Binary Offset
6	TDOP	GPS Time DOP	Float	4	H+16
7	elev mask	GPS elevation mask angle	Float	4	H+20
8	#sats	Number of satellites to follow	Ulong	4	H+24
9	sats	Satellites in use at time of calculation	Ulong	4	H+28
10	Next satellite offset = H+28+(#sats * 4)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+28+ (#sats * 4)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.169 RTKDOP2

### *DOP values for the satellites used in the RTK solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

The RTKDOP2 log contains the Dilution Of Precision (DOP) values for the satellites being used in the RTK solution. This log is similar to the **RTKDOP** log (see page 862) but contains the per-system TDOPs; see the **RTKDOP** log on page 862 for more information on the DOPs.

**Message ID:** 1172

**Log Type:** Synch

**Recommended Input:**

```
log rtkdop2a ontime 10
```

**ASCII Example:**

```
#RTKDOP2A,USB1,0,66.5,FINESTEERING,2211,234263.000,02000020,ab50,16809;
1.4290,1.1150,0.6970,0.8710,2,GPS,0.5660,GLONASS,0.6910*eced00f5
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKDOP2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	VDOP	Vertical DOP	Float	4	H+12
6	#systems	Number of entries to follow	Ulong	4	H+16
7	system	See <i>Table 188: System Used for Timing</i> on the next page	Enum	4	H+20
8	TDOP	Time DOP (Dilution of Precision)	Float	4	H+24
9	Next satellite offset = H+20+(#systems * 8)				
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20+ (#systems * 8)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



**Table 188: System Used for Timing**

Binary	ASCII
0	GPS <sup>1</sup>
1	GLONASS
2	GALILEO
3	BEIDOU
4	NAVIC
99	AUTO <sup>2</sup>

---

<sup>1</sup>GPS setting includes QZSS satellites.

<sup>2</sup>AUTO is used only as a backup system (not available for primary system field).

## 3.170 RTKPOS

### *RTK low latency position data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the low latency RTK position computed by the receiver, along with two status flags. In addition, it reports other status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. This log is recommended for kinematic operation. Better accuracy can be obtained in static operation with the **MATCHEDPOS** log (see page 704).

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation. The degradation in accuracy, due to differential age, is reflected in the standard deviation fields. See also the **DGPSTXID** command (see page 138).



The RTK system in the receiver provides two kinds of position solutions. The Matched RTK position is computed with buffered observations, so there is no error due to the extrapolation of base station measurements. This provides the highest accuracy solution possible at the expense of some latency which is affected primarily by the speed of the differential data link. The **MATCHEDPOS** log (see page 704) contains the matched RTK solution and can be generated for each processed set of base station observations.

The Low-Latency RTK position is computed from the latest local observations and extrapolated base station observations. This supplies a valid RTK position with the lowest latency possible at the expense of some accuracy. The degradation in accuracy is reflected in the standard deviation. The amount of time that the base station observations are extrapolated is in the "differential age" field of the position log. The Low-Latency RTK system extrapolates for 60 seconds. The **RTKPOS** log contains the Low-Latency RTK position when valid, and an "invalid" status when a Low-Latency RTK solution could not be computed. The **BESTPOS** log (see page 520) contains either the low-latency RTK, PPP or pseudorange-based position, whichever has the smallest standard deviation.

**Message ID:** 141

**Log Type:** Synch

**Recommended Input:**

```
log rtkposa ontime 1
```

**ASCII Example:**

```
#RTKPOSA,USB1,0,66.5,FINESTEERING,2211,234400.000,02000020,69d3,16809;
SOL_COMPUTED,NARROW_INT,51.15043865488,-114.03067348527,1098.9222,-17.0000,
WGS84,0.0075,0.0067,0.0124,"K250",1.000,0.000,33,30,30,29,00,01,3f,37*19a70ef3
```



Consider the case of a racing car, on a closed circuit, requiring RTK operation. In this situation, you would have to send live data to the pits using a radio link.

RTK operation enables live centimetre level position accuracy. When answers are required in the field, the base station must transmit information to the rover in real-time. For RTK operation, extra equipment such as radios are required to transmit and receive this information. The base station has a corresponding base radio and the rover station has a corresponding rover radio.

Post-processing can provide post-mission position and velocity data using raw GNSS data collected from the car. The logs necessary for post-processing include:

```
RANGECMPB ONTIME 1
RAWEPHEMB ONNEW
```

These are examples of data collection for post-processing, and real-time operation. OEM7-based output is compatible with post-processing software from the NovAtel's Waypoint Products Group or refer to our website at [novatel.com/products/waypoint-software](http://novatel.com/products/waypoint-software) for more details.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m)  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">  When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84. </div>	Float	4	H+32
8	datum id#	Datum ID number 61 = WGS84 63 = USER	Enum	4	H+36

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	lat $\sigma$	Latitude standard deviation (m)	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation (m)	Float	4	H+44
11	hgt $\sigma$	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.171 RTKSATS

### *Satellites used in RTKPOS solution*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log lists the used and unused satellites for the corresponding RTKPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

**Message ID:** 1174

**Log Type:** Synch

**Recommended Input:**

```
log rtkstatsa ontime 1
```

**Abbreviated ASCII Example:**

```
<RTKSATS COM1 0 64.0 FINESTEERING 2220 503054.000 02000000 95e7 16807
< 37
< GPS 18 GOOD 00000007
< GPS 16 GOOD 00000003
< GPS 5 GOOD 00000003
< GPS 27 GOOD 00000007
< GPS 29 GOOD 00000003
< GPS 10 GOOD 00000007
< GPS 23 GOOD 00000007
< GPS 13 GOOD 00000003
< GPS 15 GOOD 00000003
< GPS 26 GOOD 00000007
< QZSS 194 GOOD 00000007
< GLONASS 18-3 GOOD 00000003
< GLONASS 9-2 SUPPLEMENTARY 00000003
< GLONASS 2-4 GOOD 00000003
< GLONASS 15 GOOD 00000003
< GLONASS 1+1 GOOD 00000003
< GLONASS 24+2 GOOD 00000003
< GLONASS 23+3 GOOD 00000001
< GLONASS 17+4 GOOD 00000003
< GLONASS 8+6 GOOD 00000003
< GALILEO 15 GOOD 0000000f
< GALILEO 27 GOOD 0000000f
< GALILEO 3 GOOD 0000000f
< GALILEO 36 GOOD 0000000f
< GALILEO 5 GOOD 0000000f
< GALILEO 21 GOOD 0000000f
< GALILEO 30 GOOD 0000000f
< GALILEO 9 GOOD 0000000f
< BEIDOU 28 GOOD 0000001d
< BEIDOU 33 GOOD 0000001d
< BEIDOU 42 GOOD 0000001d
< BEIDOU 27 GOOD 0000001d
< BEIDOU 43 GOOD 0000001d
< BEIDOU 41 GOOD 0000001d
< BEIDOU 40 GOOD 0000001d
```

```

<      BEIDOU 11 GOOD 00000007
<      BEIDOU 14 GOOD 00000007

```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKSATS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	Satellite system (refer to <i>Table 124: Satellite System</i> on page 654)	Enum	4	H+4
4	Satellite ID	Satellite identifiers	Ulong	4	H+8
5	Status	Satellite status. See <i>Table 92: Observation Statuses</i> on page 529	Enum	4	H+12
6	Signal mask	See <i>Table 93: GPS Signal Mask</i> on page 530, <i>Table 94: GLONASS Signal Mask</i> on page 531, <i>Table 95: Galileo Signal Mask</i> on page 531, <i>Table 96: BeiDou Signal Mask</i> on page 531, <i>Table 97: QZSS Signal Mask</i> on page 531 and <i>Table 98: NavIC Signal Mask</i> on page 531	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.172 RTKVEL

### *RTK velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the RTK velocity information computed by the receiver. In addition, it reports a velocity status indicator that is useful in indicating whether or not the corresponding data is valid and differential age is useful in predicting anomalous behavior brought about by outages in differential corrections. The velocity measurements sometimes have a latency associated with them. The time of validity is the time tag in the log minus the latency value.



Velocities from the RTK filter are calculated from the delta-position. In RTKVEL, the velocity type is the same as the position type.



In an RTKVEL log, the actual speed and direction of the receiver antenna over ground is provided. The receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of motion of the GNSS antenna relative to ground.

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation.

The velocity is computed from consecutive RTK low latency updates. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the RTKVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the RTK low latency filter is updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency can be reduced by increasing the update rate of the RTK low latency filter by requesting the BESTVEL, RTKVEL, BESTPOS or RTKPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

**Message ID: 216**

**Log Type: Synch**

**Recommended Input:**

```
log rtkvela ontime 1
```

**ASCII Example:**

```
#RTKVELA,USB1,0,66.0,FINESTEERING,2211,234952.000,02000020,2e4e,16809;  
SOL_COMPUTED,NARROW_INT,0.250,1.000,0.0019,51.860326,-0.0004,0*4a6563f7
```



Consider the case of an unmanned aircraft. A base station must send differential correction data to the remote aircraft. In this type of application, the aircraft's radio may pass the differential solution, for example RTKVEL, to the positioning system so it can process it and generate precise position information for the flight controls.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results	Float	4	H+8
5	age	Differential age in seconds	Float	4	H+12
6	hor spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	vert spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.173 RTKXYZ

### *RTK Cartesian position and velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log contains the receiver's low latency position and velocity in ECEF coordinates. The position and velocity status fields indicate whether or not the corresponding data is valid. See *Figure 12: The WGS84 ECEF Coordinate System* on page 540 for a definition of the ECEF coordinates.

The velocity measurements have a latency associated with them. The time of validity is the time tag in the log minus the latency value.

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation. The degradation in accuracy due to differential age is reflected in the standard deviation fields. See also the **DGPSTXID** command (see page 138).

The velocity is computed from consecutive RTK low latency updates. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the RTKVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the RTK low latency filter is updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency can be reduced by increasing the update rate of the RTK low latency filter by requesting the **BESTXYZ** message at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

See also the **BESTXYZ** log on page 538 and **MATCHEDXYZ** log on page 709.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID:** 244

**Log Type:** Synch

**Recommended Input:**

```
log rtkxyza ontime 1
```

**ASCII Example:**

```
#RTKXYZA,USB1,0,64.5,FINESTEERING,2211,235074.000,02000020,40ee,16809;
SOL_COMPUTED,NARROW_INT,-1632848.2305,-3662159.5450,4944902.5890,0.0074,0.0088,
0.0110,SOL_COMPUTED,NARROW_INT,-0.0010,0.0011,-0.0005,0.0063,0.0067,0.0102,
"K250",0.250,1.000,0.000,37,31,31,30,0,01,3f,37*969e6a51
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RTKXYZ header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	P-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	pos type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X $\sigma$	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y $\sigma$	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z $\sigma$	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H+44
11	vel type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m)	Double	8	H+68
15	V-X $\sigma$	Standard deviation of V-X (m)	Float	4	H+76
16	V-Y $\sigma$	Standard deviation of V-Y (m)	Float	4	H+80
17	V-Z $\sigma$	Standard deviation of V-Z (m)	Float	4	H+84
18	stn ID	Base station identification	Char[4]	4	H+88
19	V-latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellites used in solution	Uchar	1	H+105

Field	Field type	Description	Format	Binary Bytes	Binary Offset
24	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+106
25	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+107
26	Reserved		Char	1	H+108
27	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.174 RXCONFIG

### *Receiver configuration*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log is used to output a list of all current command settings. When requested, an RXCONFIG log is output for each setting. See also the **LOGLIST** log on page 687 for a list of currently active logs. One log is output for each current command.

**Message ID:** 128

**Log Type:** Collection

**Recommended Input:**

```
log rxconfiga once
```

### **ASCII Example:**

```
#RXCONFIGA, COM2, 240, 82.0, UNKNOWN, 0, 0.857, 1a4c4020, f702, 16942; #ADJUST1PPSA, COM2,
240, 82.0, UNKNOWN, 0, 0.857, 1a4c4020, f702, 16942; OFF*44ebb463*c212d337

#RXCONFIGA, COM2, 239, 82.0, UNKNOWN, 0, 0.858, 1a4c4020, f702, 16942; #AGCMODEA, COM2, 239
, 82.0, UNKNOWN, 0, 0.858, 1a4c4020, f702, 16942; 0, AUTO*7cdceff7*3548e623

#RXCONFIGA, COM2, 238, 82.0, UNKNOWN, 0, 0.858, 1a4c4020, f702, 16942; #AGCMODEA, COM2, 238
, 82.0, UNKNOWN, 0, 0.858, 1a4c4020, f702, 16942; 1, AUTO*38d81341*51af2d69

#RXCONFIGA, COM2, 237, 82.0, UNKNOWN, 0, 0.860, 1a4c4020, f702, 16942; #AGCMODEA, COM2, 237
, 82.0, UNKNOWN, 0, 0.860, 1a4c4020, f702, 16942; 2, AUTO*ad664e8e*24cbe593

#RXCONFIGA, COM2, 236, 82.0, UNKNOWN, 0, 0.860, 1a4c4020, f702, 16942; #AGCMODEA, COM2, 236
, 82.0, UNKNOWN, 0, 0.860, 1a4c4020, f702, 16942; 3, AUTO*e962b238*a7f2317d

#RXCONFIGA, COM2, 235, 82.0, UNKNOWN, 0, 0.862, 1a4c4020, f702, 16942; #ANTENNAPOWERA, COM
2, 235, 82.0, UNKNOWN, 0, 0.862, 1a4c4020, f702, 16942; ON*4e212f4a*3c076320

...

#RXCONFIGA, COM2, 225, 82.0, UNKNOWN, 0, 0.877, 1a4c4020, f702, 16942; #CCOMCONFIGA, COM2,
225, 82.0, UNKNOWN, 0, 0.877, 1a4c4020, f702, 16942; CCOM5, NONE, NONE, 0, 0, 00*57e233f3*a4
e74dde

#RXCONFIGA, COM2, 224, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; #CCOMCONFIGA, COM2,
224, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; CCOM6, NONE, NONE, 0, 0, 00*2bb09198*bc
103ec9

#RXCONFIGA, COM2, 223, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; #CLOCKADJUSTA, COM2
, 223, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; ENABLE*45672fcf*a6db790f

#RXCONFIGA, COM2, 222, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; #CLOCKOFFSETA, COM2
, 222, 82.0, UNKNOWN, 0, 0.880, 1a4c4020, f702, 16942; 0*efd0112a*75ccf8c7

#RXCONFIGA, COM2, 221, 82.0, UNKNOWN, 0, 0.882, 1a4c4020, f702, 16942; #CLOCKSTEERINGBIAS
TOLERANCEA, COM2, 221, 82.0, UNKNOWN, 0, 0.882, 1a4c4020, f702, 16942; 100.0*92e89c27*0c5
8c011

#RXCONFIGA, COM2, 220, 82.0, UNKNOWN, 0, 0.882, 1a4c4020, f702, 16942; #CLOCKSTEERINGSOR
CEA, COM2, 220, 82.0, UNKNOWN, 0, 0.882, 1a4c4020, f702, 16942; SINGLE*1c34a54a*9e3a5e6d
```



The embedded CRCs are inverted to prevent RXCONFIG output being treated as input by mistake. To make the embedded messages recognizable to the receiver, invert the CRC. For example, consider the first embedded message above.

```
#RXCONFIGA, COM2, 240, 82.0, UNKNOWN, 0, 0.857, 1a4c4020, f702, 16942;
#ADJUST1PPSA, COM2, 240, 82.0, UNKNOWN, 0, 0.857, 1a4c4020, f702, 16942;
OFF*44ebb463*c212d337
```

The embedded CRC for the ADJUST1PPS command is **44ebb463**

This can be inverted with the "Bitwise exclusive OR operator" (XOR) in the following algorithm:

$$b = 0x44ebb463 \wedge 0xFFFFFFFF$$

$$b = 0xbb144b9c$$

The command becomes:

```
#ADJUST1PPSA, COM2, 240, 82.0, UNKNOWN, 0, 0.857, 1a4c4020, f702, 16942;
OFF*bb144b9c
```



Do not use undocumented commands or logs! Doing so may produce errors and void your warranty.



The RXCONFIG log can be used to ensure your receiver is correctly setup for your application.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RXCONFIG header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	e header	Embedded header	-	h	H
3	e msg	Embedded message	Varied	a	H+h
4	e xxxx	Embedded (inverted) 32-bit CRC (ASCII and Binary only). The embedded CRC is inverted so that the receiver does not recognize the embedded messages as messages to be output but continues with the RXCONFIG message. If you wish to use the messages output from the RXCONFIG log, simply flip the embedded CRC around for individual messages	Ulong	4	H+h+a
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+h+a+4
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.175 RXSTATUS

### *Receiver status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log conveys various status parameters of the GNSS receiver system. These include the Receiver Status and Error words which contain several flags specifying status and error conditions. If an error occurs (shown in the Receiver Error word), the receiver idles all channels, turns off the antenna and disables the RF hardware as these conditions are considered to be fatal errors. The log contains a variable number of status words to allow for maximum flexibility and future expansion. For information about resolving an error, refer to Examining the RXSTATUS Log in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

The receiver gives the user the ability to determine the importance of the status bits. In the case of the Receiver Status, setting a bit in the priority mask causes the condition to trigger an error. This causes the receiver to idle all channels, turn off the antenna and disable the RF hardware, the same as if a bit in the Receiver Error word is set. Setting a bit in an Auxiliary Status priority mask causes that condition to set the bit in the Receiver Status word corresponding to that Auxiliary Status. See also the **STATUSCONFIG** command on page 415.



Field #4, the receiver status word as represented in *Table 190: Receiver Status* on page 882, is also in Field #8 of the header. See the ASCII Example below and *Table 190: Receiver Status* on page 882 for clarification.

Refer also to Built-In Status Tests in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

**Message ID: 93**

**Log Type: Asynch**

**Recommended Input:**

```
log rxstatusa onchanged
```

**Abbreviated ASCII Example:**

```
<RXSTATUS USB1 0 66.5 FINESTEERING 2210 333374.033 02000020 2ae1 16809
< 00000000 5
< 02000020 00000000 00030000 00020000 (Receiver Status)
< 00040000 00001008 00000000 00000000 (Aux1 Status)
< 00000000 00000000 80000000 00000000 (Aux2 Status)
< 82000000 00000000 00000000 00000000 (Aux3 Status)
< 0030c000 00000000 ffffffff 00000000 (Aux4 Status)
```



Receiver errors automatically generate event messages. These event messages are output in RXSTATUSEVENT logs. It is also possible to have status conditions trigger event messages to be generated by the receiver. This is done by setting/clearing the appropriate bits in the event set/clear masks. The set mask tells the receiver to generate an event message when the bit becomes set. Likewise, the clear mask causes messages to be generated when a bit is cleared. See the **STATUSCONFIG** command on page 415 for details.

If you wish to disable all these messages without changing the bits, simply UNLOG the **RXSTATUSEVENT** logs on the appropriate ports. See the **UNLOG** command on page 452.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RXSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	error	Receiver error (see <i>Table 189: Receiver Error</i> on page 881). A value of zero indicates no errors	Ulong	4	H
3	# stats	Number of status codes (including Receiver Status). Each status code consists of 4 fields, the status, priority mask, event set mask and event clear mask. Each set is repeated for each status type.  Note that for clarity, the Receiver Status, Auxiliary 1 Status, Auxiliary 2 Status, Auxiliary 3 Status and Auxiliary 4 status are listed separately in this message	Ulong	4	H+4
4	rxstat	Receiver status word (see <i>Table 190: Receiver Status</i> on page 882)	Ulong	4	H+8
5	rxstat pri	Receiver status priority mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+12
6	rxstat set	Receiver status event set mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+16
7	rxstat clear	Receiver status event clear mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+20
8	aux1stat	Auxiliary 1 status word (see <i>Table 192: Auxiliary 1 Status</i> on page 884)	Ulong	4	H+24
9	aux1stat pri	Auxiliary 1 status priority mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+28
10	aux1stat set	Auxiliary 1 status event set mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+32

Field	Field type	Description	Format	Binary Bytes	Binary Offset
11	aux1stat clear	Auxiliary 1 status event clear mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+36
12	aux2stat	Auxiliary 2 status word (see <i>Table 193: Auxiliary 2 Status</i> on page 886)	Ulong	4	H+40
13	aux2stat pri	Auxiliary 2 status priority mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+44
14	aux2stat set	Auxiliary 2 status event set mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+48
15	aux2stat clear	Auxiliary 2 status event clear mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+52
16	aux3stat	Auxiliary 3 status word (see <i>Table 194: Auxiliary 3 Status</i> on page 887)	Ulong	4	H+56
17	aux3stat pri	Auxiliary 3 status priority mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+60
18	aux3stat set	Auxiliary 3 status event set mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+64
19	aux3stat clear	Auxiliary 3 status event clear mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+68
20	aux4stat	Auxiliary 4 status word (see <i>Table 196: Auxiliary 4 Status</i> on page 889)	Ulong	4	H+72
21	aux4stat pri	Auxiliary 4 status priority mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+76
22	aux4stat set	Auxiliary 4 status event set mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+80
23	aux4stat clear	Auxiliary 4 status event clear mask, which can be set using the <b>STATUSCONFIG</b> command on page 415	Ulong	4	H+84
24	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+88
25	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



Table 189: Receiver Error

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Dynamic Random Access Memory (DRAM) status RAM failure on an OEM7 card may also be indicated by a flashing red LED.	OK	Error
	1	0x00000002	Invalid firmware	OK	Error
	2	0x00000004	ROM status	OK	Error
	3	0x00000008	Reserved		
N1	4	0x00000010	Electronic Serial Number (ESN) access status	OK	Error
	5	0x00000020	Authorization code status	OK	Error
	6	0x00000040	Reserved		
	7	0x00000080	Supply voltage status	OK	Error
N2	8	0x00000100	Reserved		
	9	0x00000200	Temperature status (as compared against acceptable limits)	OK	Error
	10	0x00000400	MINOS status	OK	Error
	11	0x00000800	PLL RF status. Error with an RF PLL. See AUX2 status bits ( <i>Table 193: Auxiliary 2 Status</i> on page 886) for individual PLL status	OK	Error
N3	12	0x00001000	Reserved		
	13	0x00002000			
	14	0x00004000			
	15	0x00008000	NVM status	OK	Error
N4	16	0x00010000	Software resource limit exceeded	OK	Error
	17	0x00020000	Model invalid for this receiver	OK	Error
	18	0x00040000	Reserved		
	19	0x00080000			
N5	20	0x00100000	Remote loading has begun	No	Yes
	21	0x00200000	Export restriction	OK	Error
	22	0x00400000	Safe Mode	OK	Error

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1		
N6	23	0x00800000	Reserved				
	24	0x01000000					
	25	0x02000000					
	26	0x04000000					
	27	0x08000000					
N7	28	0x10000000					
	29	0x20000000					
	30	0x40000000					
	31	0x80000000				Component hardware failure	OK

Table 190: Receiver Status

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Error flag, see <i>Table 189: Receiver Error</i> on the previous page	No error	Error
	1	0x00000002	Temperature status	Within specifications	Warning
	2	0x00000004	Voltage supply status	OK	Warning
	3	0x00000008	Primary antenna power status See the <b>ANTENNAPOWER</b> command on page 70	Powered	Not powered
N1	4	0x00000010	LNA Failure	OK	Failure
	5	0x00000020	Primary antenna open circuit flag This flag is only available on certain products.	OK	Open, antenna disconnected
	6	0x00000040	Primary antenna short circuit flag This flag is only available on certain products.	OK	Short circuit detected
	7	0x00000080	CPU overload flag This flag is only available on certain products.	No overload	Overload

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N2	8	0x00000100	COM port transmit buffer overrun. See AUX2 status bits ( <i>Table 193: Auxiliary 2 Status</i> on page 886) for individual COM port status	OK	COM buffer overrun
	9	0x00000200	Spoofing Detection Status	Not Detected	Detected
	10	0x00000400	Reserved		
	11	0x00000800	Link overrun flag This flag indicates if any of the USB, ICOM, CCOM, NCOM or File ports are overrun. See AUX1, AUX2 and AUX3 status bits ( <i>Table 192: Auxiliary 1 Status</i> on the next page, <i>Table 193: Auxiliary 2 Status</i> on page 886 or <i>Table 194: Auxiliary 3 Status</i> on page 887) for the specific port for which the buffer is overrun.	No overrun	Overrun
N3	12	0x00001000	Input overrun flag This flag is set if any of the receiver ports (COM, USB, ICOM or NCOM) experience an input overrun.	No overrun	Overrun
	13	0x00002000	Aux transmit overrun flag	No overrun	Overrun
	14	0x00004000	Antenna gain state See the AUX3 status bits ( <i>Table 194: Auxiliary 3 Status</i> on page 887) for the antenna gain status.	OK	Out of range
	15	0x00008000	Jammer Detected See the AUX1 status bits ( <i>Table 192: Auxiliary 1 Status</i> on the next page) for individual RF status	OK	Jammer Detected
N4	16	0x00010000	INS reset flag	No INS reset	INS reset
	17	0x00020000	IMU communication failure	No error	No IMU communication
	18	0x00040000	GPS almanac flag/UTC known	Valid	Invalid
	19	0x00080000	Position solution flag	Valid	Invalid

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N5	20	0x00100000	Position fixed flag, see the <b>FIX</b> command on page 182	Not fixed	Fixed
	21	0x00200000	Clock steering status	Enabled	Disabled
	22	0x00400000	Clock model flag	Valid	Invalid
	23	0x00800000	External oscillator locked flag	Unlocked	Locked
N6	24	0x01000000	Software resource	OK	Warning
	25	0x06000000	Version bit 0	See Table 191: Version Bits below	
	26		Version bit 1	See Table 191: Version Bits below	
	27	0x08000000	Tracking mode	Normal tracking	HDR tracking
N7	28	0x10000000	Digital Filtering Enabled	Disabled	Enabled
	29	0x20000000	Auxiliary 3 status event flag	No event	Event
	30	0x40000000	Auxiliary 2 status event flag	No event	Event
	31	0x80000000	Auxiliary 1 status event flag	No event	Event

Table 191: Version Bits

Bit 26	Bit 25	Description
0	0	Interpret Status/Error Bits as OEM6 or earlier format
0	1	Interpret Status/Error Bits as OEM7 format
1	0	Reserved for a future version
1	1	Reserved for a future version

Table 192: Auxiliary 1 Status

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Jammer detected on RF1	OK	Jammer detected
	1	0x00000002	Jammer detected on RF2	OK	Jammer detected
	2	0x00000004	Jammer detected on RF3	OK	Jammer detected
	3	0x00000008	Position averaging	Off	On

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N1	4	0x00000010	Jammer detected on RF4	OK	Jammer detected
	5	0x00000020	Jammer detected on RF5	OK	Jammer detected
	6	0x00000040	Jammer detected on RF6	OK	Jammer detected
	7	0x00000080	USB connection status	Connected	Not connected
N2	8	0x00000100	USB1 buffer overrun flag	No overrun	Overrun
	9	0x00000200	USB2 buffer overrun flag	No overrun	Overrun
	10	0x00000400	USB3 buffer overrun flag	No overrun	Overrun
	11	0x00000800	Reserved		
N3	12	0x00001000	Profile Activation Bit	OK	Error
	13	0x00002000	Throttled Ethernet Reception	OK	Throttled
	14	0x00004000	Reserved		
	15	0x00008000	Reserved		
N4	16	0x00010000	Reserved		
	17	0x00020000	Reserved		
	18	0x00040000	Ethernet not connected	Connected	Not connected
	19	0x00080000	ICOM1 buffer overrun flag	No overrun	Overrun
N5	20	0x00100000	ICOM2 buffer overrun flag	No overrun	Overrun
	21	0x00200000	ICOM3 buffer overrun flag	No overrun	Overrun
	22	0x00400000	NCOM1 buffer overrun flag	No overrun	Overrun
	23	0x00800000	NCOM2 buffer overrun flag	No overrun	Overrun
N6	24	0x01000000	NCOM3 buffer overrun flag	No overrun	Overrun
	25	0x02000000	Reserved		
	26	0x04000000	Reserved		
	27	0x08000000	Reserved		

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N7	28	0x10000000	Reserved		
	29	0x20000000	Reserved		
	30	0x40000000	Status error reported by the IMU. May be treated as a notice unless the issue persists.	OK	Status Error Reported
	31	0x80000000	IMU measurement outlier detected. Indicates when the SPAN system has detected an outlier in the IMU performance. May be treated as a notice unless the issue persists.	OK	Outlier detected

Table 193: Auxiliary 2 Status

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	SPI Communication Failure	OK	Error
	1	0x00000002	I <sup>2</sup> C Communication Failure	OK	Error
	2	0x00000004	COM4 buffer overrun flag	No overrun	Buffer Overrun
	3	0x00000008	COM5 buffer overrun flag	No overrun	Buffer Overrun
N1	4	0x00000010	Reserved		
	5	0x00000020	Reserved		
	6	0x00000040	Reserved		
	7	0x00000080	Reserved		
N2	8	0x00000100	Reserved		
	9	0x00000200	COM1 buffer overrun flag	OK	Buffer Overrun
	10	0x00000400	COM2 buffer overrun flag	OK	Buffer Overrun
	11	0x00000800	COM3 buffer overrun flag	OK	Buffer Overrun
N3	12	0x00001000	PLL RF1 unlock flag	OK	PLL Unlock
	13	0x00002000	PLL RF2 unlock flag	OK	PLL Unlock
	14	0x00004000	PLL RF3 unlock flag	OK	PLL Unlock
	15	0x00008000	PLL RF4 unlock flag	OK	PLL Unlock

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N4	16	0x00010000	PLL RF5 unlock flag	OK	PLL Unlock
	17	0x00020000	PLL RF6 unlock flag	OK	PLL Unlock
	18	0x00040000	CCOM1 buffer overrun	OK	Buffer Overrun
	19	0x00080000	CCOM2 buffer overrun	OK	Buffer Overrun
N5	20	0x00100000	CCOM3 buffer overrun	OK	Buffer Overrun
	21	0x00200000	CCOM4 buffer overrun	OK	Buffer Overrun
	22	0x00400000	CCOM5 buffer overrun	OK	Buffer Overrun
	23	0x00800000	CCOM6 buffer overrun	OK	Buffer Overrun
N6	24	0x01000000	ICOM4 buffer overrun	OK	Buffer Overrun
	25	0x02000000	ICOM5 buffer overrun	OK	Buffer Overrun
	26	0x04000000	ICOM6 buffer overrun	OK	Buffer Overrun
	27	0x08000000	ICOM7 buffer overrun	OK	Buffer Overrun
N7	28	0x10000000	Secondary antenna power status See the <b>ANTENNAPOWER</b> command (see page 70)	Powered	Not Powered
	29	0x20000000	Secondary antenna open circuit This flag is only available on certain products	OK	Open, antenna disconnected
	30	0x40000000	Secondary antenna short circuit This flag is only available on certain products	OK	Short circuit detected
	31	0x80000000	Reset loop detected	OK	Reset Detected

Table 194: Auxiliary 3 Status

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	SCOM buffer overrun flag. This flag is set if any of the SCOM ports (SCOM1 – SCOM4) experience overrun.	No overrun	Overrun
	1	0x00000002	WCOM1 buffer overrun flag	No overrun	Overrun
	2	0x00000004	FILE buffer overrun flag	No overrun	Overrun
	3	0x00000008	Reserved		

Nibble	Bit	Mask	Description	Bit = 0	Bit = 1
N1	4	0x00000010	Antenna 1 Gain State	<i>Table 195: Antenna Gain State on the next page</i>	
	5	0x00000020			
	6	0x00000040	Antenna 2 Gain State	<i>Table 195: Antenna Gain State on the next page</i>	
	7	0x00000080			
N2	8	0x00000100	GPS reference time is incorrect. Reset the receiver to correct it.	OK	Time incorrect
	9	0x00000200	Reserved		
	10	0x00000400			
	11	0x00000800			
N3	12	0x00001000	Reserved		
	13	0x00002000			
	14	0x00004000			
	15	0x00008000			
N4	16	0x00010000	Reserved		
	17	0x00020000			
	18	0x00040000			
	19	0x00080000			
N5	20	0x00100000	Reserved		
	21	0x00200000			
	22	0x00400000			
	23	0x00800000			
N6	24	0x01000000	Spoofing Calibration Status	Successful	Failed
	25	0x02000000	Spoofing Calibration Required	Not required	Required
	26	0x04000000	Reserved		
	27	0x08000000			
N7	28	0x10000000	Reserved		
	29	0x20000000	Web content is corrupt or does not exist	Content is OK	Error with content
	30	0x40000000	RF Calibration Data is present and in error	Data is OK	Data has an error
	31	0x80000000	RF Calibration Data is present	No data found	Data exists and has no errors



Table 195: Antenna Gain State

Bits 4-5 or Bits 6-7	Description
00	Antenna Gain in range
01	Antenna Gain Low This state indicates that the input signal is very weak (under -160 dBm/Hz). It can indicate the antenna is not operating correctly, the antenna is not suitable for NovAtel receivers, or there is no antenna connected.
10	Antenna Gain High. This state indicates that the input signal is very strong (above -120 dBm/Hz). This can be caused by a strong in-band interference or by too much signal gain or too many LNAs cascaded in the path.
11	Antenna Gain Anomaly. This state indicates that an anomaly has been detected for the input signal. It can be caused by strong in-band or out-of-band interference, or by the antenna being disconnected/changed during operation.

Table 196: Auxiliary 4 Status

Nibble	Bit	Mask	Description	Bit = 1
N0	0	0x00000001	GNSS Tracked Status	<60% of available satellites are tracked well
	1	0x00000002		<15% of available satellites are tracked well
	2	0x00000004	Reserved	
	3	0x00000008		
N1	4	0x00000010	Reserved	
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	Reserved	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		

Nibble	Bit	Mask	Description	Bit = 1
N3	12	0x00001000	Clock freewheeling due to bad position integrity	Clock freewheeling
	13	0x00002000	Reserved	
	14	0x00004000	Usable RTK Corrections Percentage of expected measurements which have timely RTK corrections (latency <20 seconds)	<60% of expected corrections available
	15	0x00008000		<15% of expected corrections available
N4	16	0x00010000	Bad RTK Geometry	PDOP >5.0
	17	0x00020000	Reserved	
	18	0x00040000		
	19	0x00080000	Long RTK Baseline	Baseline >50 km
N5	20	0x00100000	Poor RTK COM Link (poor correction quality)	Corrections quality ≤60%
	21	0x00200000	Poor ALIGN COM Link (poor correction quality)	Corrections quality ≤60%
	22	0x00400000	GLIDE Not Active	GLIDE not active
	23	0x00800000	Bad PDP Geometry	PDOP >5.0
N6	24	0x01000000	No TerraStar Subscription	No subscription
	25	0x02000000	Reserved	
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000	Bad PPP Geometry	PDOP >5.0
	29	0x20000000	Reserved	
	30	0x40000000	No INS Alignment	No alignment
	31	0x80000000	INS not converged	Not converged



Only GPS and GLONASS are considered in the Auxiliary 4 status word states.



For bits relating to RTK, ALIGN or INS, the bits will only be set if the receiver has that type of positioning is enabled via Auth Code.

## 3.176 RXSTATUSEVENT

### *Status event indicator*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log is used to output event messages as indicated in the **RXSTATUS** log (see page 878). An event message is automatically generated for all receiver errors, which are indicated in the receiver error word. In addition, event messages can be generated when other conditions, which are indicated in the receiver status and auxiliary status words, are met. Whether or not an event message is generated under these conditions is specified using the **STATUSCONFIG** command (see page 415).

On start-up, the receiver is set to log **RXSTATUSEVENTA ONNEW HOLD** on all ports. You can remove this message using the **UNLOG** command (see page 452). To remove this log using an **UNLOGALL** command (see page 454), you must use the **True** option. Logging **RXSTATUSEVENT** on all ports is a factory default setting. If it is unlogged, the **RXSTATUSEVENT** log will not be collected until the next start-up. After a start-up, logging **RXSTATUSEVENT** on all ports will start again.



See also Built-In Status Tests in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

**Message ID:** 94

**Log Type:** Asynch

**Recommended Input:**

```
log rxstatuseventa onchanged
```

**ASCII Example 1:**

```
#RXSTATUSEVENTA,USB1,0,41.5,FINESTEERING,2209,513459.504,02080020,b967,16809;
AUX4,28,SET,"High PPP PDOP"*0b568a59
```

**ASCII Example 2:**

```
#RXSTATUSEVENTA,USB1,0,50.5,FINESTEERING,2209,513460.010,02080020,b967,16809;
AUX4,0,SET,"<60% available SVs tracked well"*a6aeeadd
```



When a fatal event occurs (for example, in the event of a receiver hardware failure), a bit is set in the receiver error word, part of the **RXSTATUS** log (see page 878) to indicate the cause of the problem. Bit 0 is set in the receiver status word to show that an error occurred, the error strobe is driven high and the LED flashes red and yellow showing an error code. An **RXSTATUSEVENT** log is generated on all ports to show the cause of the error. Receiver tracking is disabled at this point but command and log processing continues to allow you to diagnose the error. Even if the source of the error is corrected at this point, the receiver must be reset to resume normal operation.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RXSTATUSEVENT header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	word	The status word that generated the event message (see <i>Table 197: Status Word</i> below)	Enum	4	H
3	bit position	Location of the bit in the status word (see <i>Table 190: Receiver Status</i> on page 882, <i>Table 192: Auxiliary 1 Status</i> on page 884, <i>Table 193: Auxiliary 2 Status</i> on page 886, <i>Table 194: Auxiliary 3 Status</i> on page 887 or <i>Table 196: Auxiliary 4 Status</i> on page 889)	Ulong	4	H+4
4	event	Event type (see <i>Table 198: Event Type</i> on the next page)	Enum	4	H+8
5	description	This is a text description of the event or error	Char[32]	32	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 197: Status Word

Binary	ASCII	Description
0	ERROR	Receiver Error word, see <i>Table 189: Receiver Error</i> on page 881
1	STATUS	Receiver Status word, see <i>Table 190: Receiver Status</i> on page 882
2	AUX1	Auxiliary 1 Status word, see <i>Table 192: Auxiliary 1 Status</i> on page 884
3	AUX2	Auxiliary 2 Status word see <i>Table 193: Auxiliary 2 Status</i> on page 886
4	AUX3	Auxiliary 3 Status word see <i>Table 194: Auxiliary 3 Status</i> on page 887
5	AUX4	Auxiliary 4 Status word see <i>Table 196: Auxiliary 4 Status</i> on page 889

**Table 198: Event Type**

<b>Binary</b>	<b>ASCII</b>	<b>Description</b>
0	CLEAR	Bit was cleared
1	SET	Bit was set

## 3.177 SAFEMODESTATUS

### Safe Mode Status

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides additional information about the state of the receiver in the event that the *Safe Mode* error bit and/or *Reset Loop Detected* status bit are set in the **RXSTATUS** log (see page 878).

The data within this log is set at receiver start up and will not change over time.

For information about *Safe Mode*, see *Safe Mode* in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#), [SMART7 Installation and Operation User Manual](#) or [SMART2 Installation and Operation User Manual](#).

**Message ID:** 2060

**Log Type:** Asynch

**Recommended Input:**

```
log SAFEMODESTATUSA once
```

**Abbreviated ASCII Example:**

```
#SAFEMODESTATUSA,USB1,0,54.0,UNKNOWN,0,0.000,02000020,8e55,16809;
SAFE_MODE_OK,0,"Normal Operation."*484241d0
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	SAFEMODESTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Status	Safe Mode State. See <i>Table 199: Safe Mode States</i> on the next page	Enum	4	H
3	Reset Count	Number of resets since power up or a successful boot	Ulong	4	H+4
4	Description	String for additional information about the Safe Mode State	String	80	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+88
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 199: Safe Mode States

Value	State	Safe Mode Error Bit	Reset Loop Detected Bit	Notes	Recovery Steps
0	SAFE_MODE_OK	0	0	Normal Operation. No reset loop detected.	No action required
1	SAFE_MODE_WARNING	0	1	An unexpected reset was detected. The receiver will operate as normal	No action required
2	SAFE_MODE_DISABLE_SATELLITE_DATA	0	1	Satellite Navigation Data previously saved to NVM is ignored in this state. As the receiver continues to track GNSS satellites, new data will be downloaded. There may be some delay in initial satellite acquisition as this will effectively be a Cold Start, but the receiver will otherwise operate as normal.	No action required
3	SAFE_MODE_DISABLE_NON_COMMUNICATION_NVM	1	1	All data previously saved to NVM that is not related to communication is ignored in this state.  Communication ports (COM, USB, ICOM, etc.) will remain in the configuration previously saved by SAVECONFIG allowing the user to take corrective action.	Depending on what NVM data is causing the problem, a <b>FRESET</b> may resolve the issue.  If a standard <b>FRESET</b> does not resolve the issue, see the <b>FRESET</b> command on page 193 for other NVM targets that may be causing the issue and could be removed.
4	SAFE_MODE_DISABLE_ALL_NVM	1	1	All data previously saved to NVM is ignored in this state.	See recovery steps for SAFE_MODE_DISABLE_NON_COMMUNICATION_NVM.

Value	State	Safe Mode Error Bit	Reset Loop Detected Bit	Notes	Recovery Steps
5	SAFE_MODE_DISABLE_AUTH	1	1	All data previously saved to NVM and all Auth Codes are ignored in this state.	Use the <b>AUTH REMOVE</b> command to remove the offending Auth Code. The <b>AUTHCODES</b> log (see page 491) can be used to determine what Auth Codes are currently loaded.
6	SAFE_MODE_FAILED	1	1	All data previously saved to NVM and all Auth Codes are ignored in this state.	This state is unexpected. The recovery steps for other states may apply.
7	SAFE_MODE_UNEXPECTED_MAIN_FIRMWARE	1	0 or 1	An error related to main firmware loading occurred.	Reload the main firmware.



## 3.178 SATEL4INFO

*Displays the Satel 400 MHz radio configuration settings*

**Platform:** SMART7 with RELAY7

Use this log to display the configuration settings of the Satel 400 MHz radio.

**Message ID:** 2216

**Log Type:** Asynch

**Recommended Input:**

```
log SATEL400INFO
```

**ASCII Example:**

```
#SATEL4INFOA,COM1,0,78.0,FINESTEERING,1997,507154.039,02004000,7499,32768;  
TRIMTALK450S_T,438000000,438000000,12500,1000,TRUE*bf15bfb0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SATEL4INFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	protocol	Refer to <i>Table 200: Compatibility Mode</i> on the next page	ENUM	4	H
3	Freqtx	Transmit frequency in Hz	UINT	4	H+4
4	Freqrx	Receive frequency in Hz	UNIT	4	H+8
5	spacing	Channel spacing in Hz	UNIT	4	H+12
6	txpwr	Maximum transmit power in mW	UNIT	4	H+16
7	Fec	Forward correction error 0 = FALSE 1 = TRUE	ENUM	4	H+20
8	xxxx	32-bit CRC (ASCII or Binary only)	Hex	4	Variable
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 200: Compatibility Mode**

<b>Value</b>	<b>Enum</b>	<b>Mode</b>
0	SATELLINE_3AS	Satellite-3AS
1	PACCREST_4FSK	PacCrest-4FSK
2	PACCREST_GMSK	PacCrest-GMSK
3	TRIMTALK450S_P	TrimTalk450s (PacCrest transmitter)
4	TRIMTALK450S_T	TrimTalk450s (Trimble transmitter)
5	PACCREST_FST	PacCrest-FST

## 3.179 SATEL9INFO

*Displays the Satel UHF radio configuration settings*

**Platform:** SMART7 with RELAY7

Use this log to display the configuration settings of the Satel UHF radio.

**Message ID:** 2220

**Log Type:** Asynch

**Recommended Input:**

```
log SATEL9INFO
```

**ASCII Example:**

```
#SATEL9INFOA,COM1,0,78.5,FINESTEERING,1997,507405.578,02004000,9dd6,32768;
P2MP_RX_SLAVE,4,-1,4,534,1,9,10,FF,FALSE,0,0,0,"1111111111111111"*2d430ec2
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SATEL9INFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	modemmode	Radio role in network	ENUM	4	H
3	Leicachannel	Leica virtual channel <sup>1</sup>	INT	4	H+4
4	novchannel	Novariant virtual channel <sup>2</sup>	INT	4	H+8
5	freqkey	Frequency key	UINT	4	H+12
6	nid	Network ID	UINT	4	H+16
7	minpktsize	Minimum packet size	UINT	4	H+20
8	maxpktsize	Maximum packet size	UINT	4	H+24
9	retrytimeout	Number of failed attempts to receive packets before the connection is dropped	UINT	4	H+28
10	subnet	Multipoint subnet: 00=Roaming FF=Disabled	UINT	4	H+32

<sup>1</sup>If the current configuration does not translate to a valid Leica virtual channel, the reported value will be -1.

<sup>2</sup>If the current configuration does not translate to a valid Novariant virtual channel, the reported value will be -1.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	repeaters	Boolean to specify if repeaters are present in the network	BOOL	4	H+36
12	masterpacketrepeat	How many time master will repeat packets	UINT	4	H+40
13	txpower	The transmitter output power (mW)	UINT	4	H+44
14	hoptablever	Hop table version	UINT	4	H+48
15	freqzone	Frequency zone	FixedUChar Array	16	H+52
16	xxxx	32-bit CRC (ASCII or Binary only)	Hex	4	Variable
17	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 201: ASCII and Binary Values

Type	ASCII	Binary	Description
SATEL9INFO header	-	-	For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.
modemmode	P2MP_MASTER	2	Role in radio network
	P2MP_SLAVE	3	
	P2MP_RX_SLAVE	8	
leicachannel	1-9, -1		Leica virtual channel
novchannel	1-3824, -1		Novariant virtual channel
freqkey	0-14		Frequency key
nid	0-4095		Network ID
minpktsize	0-9		Minimum packet size
maxpktsize	0-9		Maximum packet size
retrytimeout	8-255		Number of failed attempts to receive packets before connection drops off
subnet	0x00, 0xFF		Multipoint subnet: 00=Roaming FF=Disabled
repeaters	FALSE	0	Boolean to specify if repeaters are present in the network
	TRUE	1	
masterpacketrepeat	0-9		Number of times master will repeat packets
txpower	0-1000		The transmitter output power (mW)
hoptablever	0-6		Hop table version
freqzone	"1111111111111111"		Frequency zone

## 3.180 SATELSTATUS

**Delivers status updates for any asynchronous radio operations**

**Platform:** SMART7 with RELAY7

Use this log to display the status of the last operation attempted on the RELAY7 radio. Settings are saved on the radio so it is not necessary to configure the radio for each session. The SATELSTATUS log reports UNDETECTED until the radio is used. If hardware power control is available, OFF may also be reported.



There is no state monitoring, so a radio can be detached after being configured and the status would not update until a radio command is issued.

**Message ID:** 2205

**Log Type:** Asynch

**Recommended Input:**

```
log SATELSTATUS onchanged
```

**ASCII Example:**

```
#SATELSTATUSA,COM1,0,68.5,FINESTEERING,1883,338273.235,03040040,ea7e,32768;
UNDETECTED,OK,""*8b80bd71
```

```
#SATELSTATUSA,COM1,0,68.5,FINESTEERING,1883,338273.235,03040040,ea7e,32768;
READY,OK,""*8b80bd71
```

```
#SATELSTATUSA,COM1,0,68.5,FINESTEERING,1883,338273.235,03040040,ea7e,32768;
ERROR,TIMEOUT,""*8b80bd71
```

```
#SATELSTATUSA,COM1,0,68.5,FINESTEERING,1883,338273.235,03040040,ea7e,32768;
ERROR,COMMAND_FAILED,"SL&FR=900.0000"*8b80bd71
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SATELSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	state	State of the radio. See <i>Table 202: Radio State</i> on the next page.	ENUM	4	H
3	error	Error type. See <i>Table 203: Error Types</i> on the next page.	UINT	4	H+4
4	error_source	Last SATEL radio command issued	String	Variable	H+8
5	xxxx	32-bit CRC (ASCII or Binary only)	Hex	4	Variable
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 202: Radio State**

ASCII	Binary	Description
OFF	0	The radio power is configured OFF
DETECTING	1	The radio is being queried for identity and current configuration
UNDETECTED	2	The radio may be operating but it must be detected before responding to configuration commands
READY	3	The radio is operating and ready to respond to configuration commands
BUSY	4	The radio is being configured. Data may be interrupted
ERROR	5	The radio is in an error state

**Table 203: Error Types**

ASCII	Binary	Description
NONE	0	Error type
COMMAND_FAILED	1	
TIMEOUT	2	

## 3.181 SATVIS2

### *Satellite visibility*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains satellite visibility data for all available systems with additional satellite and satellite system information. One log is output for each available satellite system.



1. The SATVIS2 log is meant to provide a brief overview. The satellite positions and velocities used in the computation of this log are based on Almanac orbital parameters, not the higher precision Ephemeris parameters. BeiDou is an exception, it uses ephemeris only.
2. In the SATVIS2 log output, there may be double satellite number entries. These are GLONASS antipodal satellites in the same orbit plane separated by 180 degrees latitude. Refer to the GLONASS section of *An Introduction to GNSS* available on our website at [novatel.com/an-introduction-to-gnss](http://novatel.com/an-introduction-to-gnss).
3. The SATVIS2 log is generated every 10 seconds. If the log is requested at a faster rate than ontime 10, it will be output at the requested rate, but will only be updated every 10 seconds.

**Message ID: 1043**

**Log Type: Asynch**

**Recommended Input:**

```
log satvis2a onchanged
```

**Abbreviated ASCII Example:**

```
<SATVIS2 USB1 6 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   GPS TRUE TRUE 31
<       29 0 73.9 298.1 1006.524 1006.471
<       5 0 63.2 127.7 829.350 829.298
...
<       21 0 -78.0 245.7 1008.686 1008.634
<       1 0 -82.3 91.7 -329.712 -329.765

<SATVIS2 USB1 5 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   GLONASS TRUE TRUE 24
<       18-3 0 63.2 47.6 -1112.813 -1112.867
<       19+3 0 52.6 301.3 1883.220 1883.167
...
<       7+5 0 -57.3 300.1 -1966.130 -1966.184
<       22-3 0 -74.8 221.7 612.703 612.650

<SATVIS2 USB1 4 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   SBAS TRUE FALSE 4
<       131 0 31.6 183.8 0.016 -0.036
<       138 0 31.4 171.3 0.017 -0.035
<       135 0 30.7 194.0 0.014 -0.038
<       133 0 30.0 199.2 0.013 -0.039

<SATVIS2 USB1 3 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
```



```

< GALILEO TRUE TRUE 23
<     34 0 69.1 59.8 -928.064 -928.116
<     15 0 53.3 267.2 1517.165 1517.113
<     ...
<     12 0 -65.5 86.6 -1047.031 -1047.083
<     33 0 -76.0 240.3 632.424 632.371


<SATVIS2 USB1 2 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   QZSS TRUE TRUE 4
<     194 0 13.5 310.7 203.666 203.614
<     195 0 -16.0 274.7 -306.102 -306.154
<     199 0 -25.6 293.3 2.197 2.145
<     196 0 -47.1 271.8 449.579 449.527

<SATVIS2 USB1 1 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   BEIDOU TRUE TRUE 59
<     43 88 79.1 302.7 545.401 545.349
<     11 0 69.0 83.5 -960.045 -960.097
<     ...
<     13 0 -81.1 244.3 116.464 116.413
<     29 88 -81.6 120.6 -430.185 -430.237

<SATVIS2 USB1 0 54.0 FINESTEERING 2209 513610.000 02000020 a867 16809
<   NAVIC TRUE FALSE 0

```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SATVIS2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Satellite System	GNSS satellite system identifier. See <i>Table 124: Satellite System</i> on page 654	Enum	4	H
3	sat vis	Is satellite visibility valid? 0 = FALSE 1 = TRUE	Enum	4	H+4
4	almanac flag	Was complete almanac used? 0 = FALSE 1 = TRUE	Enum	4	H+8
5	#sat	Number of satellites with data to follow	Ulong	4	H+12

Field	Field type	Description	Format	Binary Bytes	Binary Offset
6	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems. In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2.  For more information, refer to <i>PRN Numbers</i> on page 52	Ulong	4	H+16
7	health	Satellite health   Satellite health values may be found in the applicable Interface Control Document for each system.	Ulong	4	H+20
8	elev	Elevation (degrees)	Double	8	H+24
9	az	Azimuth (degrees)	Double	8	H+32
10	true Doppler	Theoretical Doppler of satellite - the expected Doppler frequency based on a satellite's motion relative to the receiver. It is computed using the satellite's coordinates and velocity along with the receiver's coordinates and velocity (Hz).	Double	8	H+40
11	apparent Doppler	Apparent Doppler for this receiver - the same as Theoretical Doppler above but with clock drift correction added (Hz).	Double	8	H+48
12	Next satellite offset = H + 16 + (#sat x 40)				
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16+ (#sat x 40)
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.182 SATXYZ2

### *Satellite positions in ECEF Cartesian coordinates*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

When combined with a RANGE log, this data set contains the decoded satellite information necessary to compute the solution: satellite coordinates (ECEF), satellite clock correction, ionospheric corrections and tropospheric corrections. See the calculation examples in the usage box below. Only those satellites used in the corresponding PSRPOS solution are reported here. See also *Figure 12: The WGS84 ECEF Coordinate System* on page 540.



The position coordinates are provided in the datum configured by the **OUTPUTDATUM** command (see page 284).

**Message ID: 1451**

**Log Type: Synch**

**Recommended Input:**

```
log satxyz2 ontime 1
```

**Abbreviated ASCII Example:**

```
<SATXYZ2 USB1 0 68.5 FINESTEERING 2210 335347.000 02000020 7513 16809
<    33
<      GPS 5 -5020461.7928 -25680412.1145 3895185.6848 -24674.365 7.451
3.734 0.0 0.0
<      GPS 6 17719519.1461 -7620594.3470 18357851.0315 92492.352 8.923
6.449 0.0 0.0
<      GPS 29 -17763264.5350 -2136772.7366 19599826.2346 -154398.173 5.339
3.151 0.0 0.0
<      GPS 12 -9148026.7576 -24014583.7810 6678087.9193 -65209.710 6.008
3.091 0.0 0.0
<      GPS 9 18287652.1028 730576.0171 19192008.9639 -100642.405 10.945
14.299 0.0 0.0
...
<      QZSS 194 -24965113.5911 24703385.9532 28399888.3498 -74.060 9.319
11.140 0.0 0.0
<      GLONASS 22-3 7699005.7624 -11043302.0832 21669033.1478 -55159.517
7.696 2.922 0.0 0.0
<      GLONASS 12-1 -15776044.3967 -19935215.2933 -2172302.8981 5551.124
11.793 6.465 0.0 0.0
<      GLONASS 14-7 -9924647.1145 5441857.5868 22869748.6581 7687.030
10.285 4.219 0.0 0.0
<      GLONASS 13-2 -18443739.0480 -11030605.9522 13725659.4436 -6740.531
6.909 2.806 0.0 0.0
<      GLONASS 24+2 -24194892.7729 158595.4159 8103819.1464 28295.396
10.608 7.975 0.0 0.0
...
<      GALILEO 26 11908959.5845 -13702223.9079 23377702.1405 -347773.553
7.083 3.176 0.0 0.0
```

```

<          GALILEO 7 22792674.3142 -12903431.3874 13791966.8186 -200807.963
11.538 9.629 0.0 0.0
<          GALILEO 24 -16791994.0871 2450155.5014 24241175.8702 -186539.254
6.992 3.469 0.0 0.0
<          GALILEO 31 -18231691.1561 -17046216.1699 15920139.1514 -148949.282
6.140 2.485 0.0 0.0
<          GALILEO 33 -9401282.9993 -16271727.7340 22863722.0668 -136856.054
5.559 2.128 0.0 0.0
...
<          BEIDOU 37 -18620975.4950 -20106704.6345 -5303071.3088 -273075.077
16.311 9.289 0.0 0.0
<          BEIDOU 46 -23602968.9441 -7543827.2273 12872870.3614 -99460.897
9.808 3.573 0.0 0.0
<          BEIDOU 27 -14195438.5868 -10662737.8554 21501016.4435 217428.319
8.585 2.310 0.0 0.0
<          BEIDOU 42 15931468.6323 639406.2465 22950291.6443 -291213.434 13.427
7.528 0.0 0.0
<          BEIDOU 33 13384960.0662 -19323978.0538 15033146.5546 -289635.111
10.878 3.898 0.0 0.0

```



The OEM7 family defines ionospheric and tropospheric corrections positively which means that ionospheric and tropospheric corrections are added to the geometric ranges or subtracted from the measured pseudoranges. A positive clock offset indicates the clock is running ahead of the reference time. For example:

$$P = p + pd + c(dT - dt) + d(\text{ion}) + d(\text{trop}) + E_p$$

is equivalent to

$$P - c(dT - dt) - d(\text{ion}) - d(\text{trop}) = p + pd + E_p$$

where

P = measured pseudorange

p = geometric range

pd = orbit error

dt = satellite clock offset

dT = receiver clock offset

d (ion) = ionospheric delay

d (trop) = tropospheric delay

c = speed of light

E<sub>p</sub> = noise and multipath



Note that when dual frequency ionosphere corrections are used, the corrections include receiver biases. Consequently, the correction does not provide a measure of the ionosphere delay in an absolute sense.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SATXYZ2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	#sat	Number of satellites to follow	Ulong	4	H
3	System	Satellite system (refer to <i>Table 124: Satellite System</i> on page 654)	Enum	4	H+4
4	Satellite ID	Satellite ID	Ulong	4	H+8
5	X	Satellite X co-ordinates (ECEF,m)	Double	8	H+12
6	Y	Satellite Y co-ordinates (ECEF,m)	Double	8	H+20
7	Z	Satellite Z co-ordinates (ECEF,m)	Double	8	H+28
8	clk corr	Satellite clock correction (m)	Double	8	H+36
9	iono delay	Ionosphere delay (m)	Double	8	H+44
10	tropo delay	Troposphere delay (m)	Double	8	H+52
11	Reserved1		Double	8	H+60
12	Reserved2		Double	8	H+68
13	Next satellite offset = H+4+(#sat x 72)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#sat x 72)
15	[CR][LF]	Sentence terminator (ASCII only)			

## 3.183 SAVEDSURVEYPOSITIONS

### *Saved surveyed positions*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7

This log lists the surveyed positions saved on the receiver.

A surveyed position is saved using the **AUTOSURVEY** command (see page 87) or **SURVEYPOSITION** command (see page 420).

**Message ID:** 1951


**Log Type:** Polled

**Recommended Input:**

```
log savedsurveypositions once
```

**Abbreviated ASCII Example:**

```
<SAVEDSURVEYPOSITIONS USB1 0 52.5 FINESTEERING 2209 514133.104 02000020 ddf2
16809
<      3
<      "MN01" 51.11600000000 -114.03800000000 1065.0000
<      "MN02" 55.11600000000 -114.03800000000 1065.0000
<      "MN03" 55.11600000000 -94.03800000000 65.0000
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SAVEDSURVEYPOSITIONS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#Positions	Number of records to follow.	Ulong	4	H
3	Position ID	ID for the saved position. <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> In the Binary case, the ID string must be null terminated and additional bytes of padding must be added to make the total length of the field 8 bytes.</div>	String	8	H+4
4	Latitude	Latitude of the position (-90 to 90 degrees) where a '-' sign denotes south and a '+' sign denotes north	Double	8	H+12

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
5	Longitude	Longitude of the position (-360 to 360 degrees) where a '-' sign denotes west and a '+' sign denotes east	Double	8	H+20
6	Height	Mean Sea Level height of the position in metres	Double	8	H+28
7	Next reading offset = $H+4+(\#Positions * 32)$				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#Positions * 32)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.184 SBAS0

### *Do not use for safety applications*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This message specifies the PRN if GEO transmitting SBAS corrections is operating in test mode, and should not be used for safety applications for a period of time outlined in the SBAS signal specification.



The GEO transmitting SBAS corrections is operating in test mode, and should not be used for safety-of-life applications.

See how the SBAS0 message relates to the SBAS testing modes in the **SBASCONTROL** command on page 369.

**Message ID:** 976

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS0a onchanged
```

**ASCII Example:**

```
#SBAS0A,COM1,0,72.5,SATTIME,2073,327283.000,1a000000,5a84,15761;130*8390deb3
```



Although the SBAS was designed for aviation users, it supports a wide variety of non-aviation uses including agriculture, surveying, recreation, and surface transportation.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS0 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN message - also PRN not to use	Ulong	4	H
3	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.185 SBAS1

### PRN mask assignments

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The PRN mask is given in SBAS1. The transition of the PRN mask to a new one (which will be infrequent) is controlled with the 2-bit IODP, which sequences to a number between 0 and 3. The same IODP appears in the applicable SBAS2, SBAS3, SBAS4, SBAS5, SBAS7, SBAS24 and SBAS25 messages (SBAS32, SBAS33, SBAS34, SBAS35 and SBAS45). This transition would probably only occur when a new satellite is launched or when a satellite fails and is taken out of service permanently. A degraded satellite may be flagged as a “do not use” satellite temporarily.

**Message ID:** 977

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS1a onchanged
```

**ASCII Example:**

```
#SBAS1A,USB1,0,46.5,SATTIME,2209,514210.000,02000020,5955,16809;135,fffffffff000  
000000000000000000000002a400000000000000000,2*193c8b8a
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS1 message can be logged to view the data breakdown of SBAS frame 1 which contains information about the PRN mask assignment.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS1 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	mask	PRN bit mask	Uchar[27]	28 <sup>1</sup>	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+32
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary log case, an additional 1 byte of padding is added to maintain 4-byte alignment.

## 3.186 SBAS2

### *Fast correction slots 0-12*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS2 are fast corrections for slots 0-12 in the mask of SBAS1. This message may or may not come when SBAS is in testing mode (see the **SBASCONTROL** command on page 369 for details).

**Message ID:** 982

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS2a onchanged
```

**ASCII Example:**

```
#SBAS2A,USB1,0,52.5,SATTIME,2209,514322.000,02000020,e194,16809;133,2,2,2047,0,2047,2,-2,-1,-2,2047,0,2047,2047,0,1,14,5,14,9,5,6,11,14,7,14,14,5,6*34dbdcec
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS2 message can be logged to view the data breakdown of SBAS frame 2 which contains information about fast correction slots 0-12.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS2 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodf	Issue of fast corrections data	Ulong	4	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	prc0	prc(i): Fast corrections (-2048 to +2047) for the PRN in slot i (i = 0-12)	Long	4	H+12
6	prc1		Long	4	H+16
7	prc2		Long	4	H+20
8	prc3		Long	4	H+24
9	prc4		Long	4	H+28
10	prc5		Long	4	H+32
11	prc6		Long	4	H+36
12	prc7		Long	4	H+40
13	prc8		Long	4	H+44
14	prc9		Long	4	H+48
15	prc10		Long	4	H+52
16	prc11		Long	4	H+56
17	prc12		Long	4	H+60
18	udre0	udre(i): User differential range error indicator for the PRN in slot i (i = 0-12) See <i>Table 204: Evaluation of UDREI</i> on the next page for scaling information.	Ulong	4	H+64
19	udre1		Ulong	4	H+68
20	udre2		Ulong	4	H+72
21	udre3		Ulong	4	H+76
22	udre4		Ulong	4	H+80
23	udre5		Ulong	4	H+84
24	udre6		Ulong	4	H+88
25	udre7		Ulong	4	H+92
26	udre8		Ulong	4	H+96
27	udre9		Ulong	4	H+100
28	udre10		Ulong	4	H+104
29	udre11		Ulong	4	H+108
30	udre12		Ulong	4	H+112
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 204: Evaluation of UDREI**

<b>UDREI 1</b>	<b>UDRE metres</b>	<b><math>\sigma^2_{i.udre}</math> metres<sup>2</sup></b>
0	0.75	0.0520
1	1.0	0.0924
2	1.25	0.1444
3	1.75	0.2830
4	2.25	0.4678
5	3.0	0.8315
6	3.75	1.2992
7	4.5	1.8709
8	5.25	2.5465
9	6.0	3.3260
10	7.5	5.1968
11	15.0	20.7870
12	50.0	230.9661
13	150.0	2078.695
14	Not Monitored	Not Monitored
15	Do Not Use	Do Not Use

---

<sup>1</sup>The s2UDRE broadcast in SBAS2, SBAS3, SBAS4, SBAS5, SBAS6 and SBAS24 applies at a time prior to or at the time of applicability of the associated corrections.

## 3.187 SBAS3

### *Fast corrections slots 13-25*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS3 are fast corrections for slots 13-25 in the mask of SBAS1.

**Message ID:** 987

**Log Type:** Asynch

**Recommended Input:**

```
log sbas3a onchanged
```

**ASCII Example:**

```
#SBAS3A,USB1,0,51.0,SATTIME,2209,514339.000,02000020,bff5,16809;131,1,2,2047,-8,-12,2047,10,2047,2,2047,2047,10,2047,-2,7,14,7,8,14,6,14,5,14,14,10,14,5,6
*b79dcab9
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS3 message can be logged to view the data breakdown of SBAS frame 3 which contains information about fast correction slots 13-25.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS3 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodf	Issue of fast corrections data	Ulong	4	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	prc13	prc(i):  Fast corrections (-2048 to +2047) for the PRN in slot i (i = 13-25)	Long	4	H+12
6	prc14		Long	4	H+16
7	prc15		Long	4	H+20
8	prc16		Long	4	H+24
9	prc17		Long	4	H+28
10	prc18		Long	4	H+32
11	prc19		Long	4	H+36
12	prc20		Long	4	H+40
13	prc21		Long	4	H+44
14	prc22		Long	4	H+48
15	prc23		Long	4	H+52
16	prc24		Long	4	H+56
17	prc25		Long	4	H+60
18	udre13	udre(i):  User differential range error indicator for the PRN in slot i (i = 13-25)  See <i>Table 204: Evaluation of UDREI</i> on page 916 for scaling information.	Ulong	4	H+64
19	udre14		Ulong	4	H+68
20	udre15		Ulong	4	H+72
21	udre16		Ulong	4	H+76
22	udre17		Ulong	4	H+80
23	udre18		Ulong	4	H+84
24	udre19		Ulong	4	H+88
25	udre20		Ulong	4	H+92
26	udre21		Ulong	4	H+96
27	udre22		Ulong	4	H+100
28	udre23		Ulong	4	H+104
29	udre24		Ulong	4	H+108
30	udre25		Ulong	4	H+112
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.188 SBAS4

### *Fast correction slots 26-38*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS4 are fast corrections for slots 26-38 in the mask of SBAS1.

**Message ID:** 992

**Log Type:** Asynch

**Recommended Input:**

```
log sbas4a onchanged
```

**ASCII Example:**

```
#SBAS4A,USB1,0,50.5,SATTIME,2209,514359.000,02000020,5b17,16809;131,0,2,2047,
2047,-2,-2,-2,2047,-12,-7,-10,-13,2047,2047,2047,14,14,5,14,6,14,10,10,10,10,
15,15,15*21e3ed26
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS4 message can be logged to view the data breakdown of SBAS frame 4 which contains information about fast correction slots 26-38.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS4 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodf	Issue of fast corrections data	Ulong	4	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	prc26	prc(i):  Fast corrections (-2048 to +2047) for the PRN in slot i (i = 26-38)	Long	4	H+12
6	prc27		Long	4	H+16
7	prc28		Long	4	H+20
8	prc29		Long	4	H+24
9	prc30		Long	4	H+28
10	prc31		Long	4	H+32
11	prc32		Long	4	H+36
12	prc33		Long	4	H+40
13	prc34		Long	4	H+44
14	prc35		Long	4	H+48
15	prc36		Long	4	H+52
16	prc37		Long	4	H+56
17	prc38		Long	4	H+60
18	udre26	udre(i):  User differential range error indicator for the PRN in slot i (i = 26-38)  See <i>Table 204: Evaluation of UDREI</i> on page 916 for scaling information.	Ulong	4	H+64
19	udre27		Ulong	4	H+68
20	udre28		Ulong	4	H+72
21	udre29		Ulong	4	H+76
22	udre30		Ulong	4	H+80
23	udre31		Ulong	4	H+84
24	udre32		Ulong	4	H+88
25	udre33		Ulong	4	H+92
26	udre34		Ulong	4	H+96
27	udre35		Ulong	4	H+100
28	udre36		Ulong	4	H+104
29	udre37		Ulong	4	H+108
30	udre38	Ulong	4	H+112	
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.189 SBAS5

### *Fast correction slots 39-50*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS5 are fast corrections for slots 39-50 in the mask of SBAS1.

**Message ID:** 994

**Log Type:** Asynch

**Recommended Input:**

```
log sbas5a onchanged
```

**ASCII Example:**

```
#SBAS5A,COM1,0,72.5,SATTIME,1093,161480.000,02040020,31d4,209;122,1,3,-7,2047,
2047,2047,-4,2047,2047,2047,9,2047,2047,-3,-2,11,14,14,14,4,14,14,14,5,14,14,4,
2*2bf0109b
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS5 message can be logged to view the data breakdown of SBAS frame 5 which contains information about fast correction slots 39-50.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS5 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodf	Issue of fast corrections data	Ulong	4	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+8

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	prc39	prc(i):  Fast corrections (-2048 to +2047) for the PRN in slot i (i = 39-50)	Long	4	H+12
6	prc40		Long	4	H+16
7	prc41		Long	4	H+20
8	prc42		Long	4	H+24
9	prc43		Long	4	H+28
10	prc44		Long	4	H+32
11	prc45		Long	4	H+36
12	prc46		Long	4	H+40
13	prc47		Long	4	H+44
14	prc48		Long	4	H+48
15	prc49		Long	4	H+52
16	prc50		Long	4	H+56
17	prc51 (Invalid, do not use)		Long	4	H+60
18	udre39	udre(i):  User differential range error indicator for the PRN in slot i (i = 39-50)  See <i>Table 204: Evaluation of UDREI</i> on page 916 for scaling information.	Ulong	4	H+64
19	udre40		Ulong	4	H+68
20	udre41		Ulong	4	H+72
21	udre42		Ulong	4	H+76
22	udre43		Ulong	4	H+80
23	udre44		Ulong	4	H+84
24	udre45		Ulong	4	H+88
25	udre46		Ulong	4	H+92
26	udre47		Ulong	4	H+96
27	udre48		Ulong	4	H+100
28	udre49		Ulong	4	H+104
29	udre50		Ulong	4	H+108
30	udre51 (Invalid, do not use)		Ulong	4	H+112
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.190 SBAS6

### *Integrity message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS6 is the integrity information message. Each message includes an IODF for each fast corrections message. The  $\sigma^2_{UDRE}$  information for each block of satellites applies to the fast corrections with the corresponding IODF.

**Message ID:** 995

**Log Type:** Asynch

**Recommended Input:**

```
log sbas6a onchanged
```

**ASCII Example:**

```
#SBAS6A,COM1,0,57.5,SATTIME,1093,273317.000,02000020,526a,209;122,3,3,3,3,9,14,14,2,3,10,2,14,14,3,14,14,5,14,14,7,14,14,14,14,14,14,3,3,14,14,14,14,3,15,11,11,15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*925a2a9b
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS6 message can be logged to view the data breakdown of SBAS frame 6 which contains information about the integrity message.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS6 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodf2	Issue of fast corrections data	Ulong	4	H+4
4	iodf3	Issue of fast corrections data	Ulong	4	H+8
5	iodf4	Issue of fast corrections data	Ulong	4	H+12
6	iodf5	Issue of fast corrections data	Ulong	4	H+16
7	udre0	udre(i): User differential range error indicator for the PRN in slot i (i = 0-50) See <i>Table 204: Evaluation of UDREI</i> on page 916 for scaling information.	Ulong	4	H+20

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	udre1		Ulong	4	H+24
9	udre2		Ulong	4	H+28
10	udre3		Ulong	4	H+32
11	udre4		Ulong	4	H+36
12	udre5		Ulong	4	H+40
13	udre6		Ulong	4	H+44
14	udre7		Ulong	4	H+48
15	udre8		Ulong	4	H+52
16	udre9		Ulong	4	H+56
17	udre10		Ulong	4	H+60
18	udre11		Ulong	4	H+64
19	udre12		Ulong	4	H+68
20	udre13		Ulong	4	H+72
21	udre14		Ulong	4	H+76
22	udre15		Ulong	4	H+80
23	udre16		Ulong	4	H+84
24	udre17		Ulong	4	H+88
25	udre18		Ulong	4	H+92
26	udre19		Ulong	4	H+96
27	udre20		Ulong	4	H+100
28	udre21		Ulong	4	H+104
29	udre22		Ulong	4	H+108
30	udre23		Ulong	4	H+112
31	udre24		Ulong	4	H+116
32	udre25		Ulong	4	H+120
33	udre26		Ulong	4	H+124
34	udre27		Ulong	4	H+128
35	udre28		Ulong	4	H+132
36	udre29		Ulong	4	H+136

Field	Field type	Description	Format	Binary Bytes	Binary Offset
37	udre30		Ulong	4	H+140
38	udre31		Ulong	4	H+144
39	udre32		Ulong	4	H+148
40	udre33		Ulong	4	H+152
41	udre34		Ulong	4	H+156
42	udre35		Ulong	4	H+160
43	udre36		Ulong	4	H+164
44	udre37		Ulong	4	H+168
45	udre38		Ulong	4	H+172
46	udre39		Ulong	4	H+176
47	udre40		Ulong	4	H+180
48	udre41		Ulong	4	H+184
49	udre42		Ulong	4	H+188
50	udre43		Ulong	4	H+192
51	udre44		Ulong	4	H+196
52	udre45		Ulong	4	H+200
53	udre46		Ulong	4	H+204
54	udre47		Ulong	4	H+208
55	udre48		Ulong	4	H+212
56	udre49		Ulong	4	H+216
57	udre50		Ulong	4	H+220
58	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+224
59	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	al(4)		Ulong	4	H+32
11	al(5)		Ulong	4	H+36
12	al(6)		Ulong	4	H+40
13	al(7)		Ulong	4	H+44
14	al(8)		Ulong	4	H+48
15	al(9)		Ulong	4	H+52
16	al(10)		Ulong	4	H+56
17	al(11)		Ulong	4	H+60
18	al(12)		Ulong	4	H+64
19	al(13)		Ulong	4	H+68
20	al(14)		Ulong	4	H+72
21	al(15)		Ulong	4	H+76
22	al(16)		Ulong	4	H+80
23	al(17)		Ulong	4	H+84
24	al(18)		Ulong	4	H+88
25	al(19)		Ulong	4	H+92
26	al(20)		Ulong	4	H+96
27	al(21)		Ulong	4	H+100
28	al(22)		Ulong	4	H+104
29	al(23)		Ulong	4	H+108
30	al(24)		Ulong	4	H+112
31	al(25)		Ulong	4	H+116
32	al(26)		Ulong	4	H+120
33	al(27)		Ulong	4	H+124
34	al(28)		Ulong	4	H+128
35	al(29)		Ulong	4	H+132
36	al(30)		Ulong	4	H+136
37	al(31)		Ulong	4	H+140
38	al(32)		Ulong	4	H+144

Field	Field type	Description	Format	Binary Bytes	Binary Offset
39	al(33)		Ulong	4	H+148
40	al(34)		Ulong	4	H+152
41	al(35)		Ulong	4	H+156
42	al(36)		Ulong	4	H+160
43	al(37)		Ulong	4	H+164
44	al(38)		Ulong	4	H+168
45	al(39)		Ulong	4	H+172
46	al(40)		Ulong	4	H+176
47	al(41)		Ulong	4	H+180
48	al(42)		Ulong	4	H+184
49	al(43)		Ulong	4	H+188
50	al(44)		Ulong	4	H+192
51	al(45)		Ulong	4	H+196
52	al(46)		Ulong	4	H+200
53	al(47)		Ulong	4	H+204
54	al(48)		Ulong	4	H+208
55	al(49)		Ulong	4	H+212
56	al(50)		Ulong	4	H+216
57	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+220
58	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.192 SBAS9

### *GEO navigation message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS9 provides the GEO navigation message representing the position, velocity and acceleration of the geostationary satellite, in ECEF coordinates and its apparent clock time and frequency offsets.

Also included is the time of applicability, an Issue of Data (IOD) and an accuracy exponent (URA) representing the estimated accuracy of the message. The time offset and time drift are with respect to SBAS Network Time. Their combined effect is added to the estimate of the satellite's transmit time.

**Message ID:** 997

**Log Type:** Asynch

**Recommended Input:**

```
log sbas9a onchanged
```

**ASCII Example:**

```
#SBAS9A,USB1,3,52.0,SATTIME,2209,514470.000,02000020,b580,16809;131,237,82496,
0,-19136930.4000,-37573052.2400,-2757.2000,-0.221875000,-0.001875000,
-0.2160000,-0.0000125,0.0000250,0.000000000,2.607703209e-08,-8.185452316e-12
*e31c3f2e

#SBAS9A,USB1,2,52.0,SATTIME,2209,514470.000,02000020,b580,16809;135,172,82496,
0,-24165841.7600,-34542291.9200,1690.8000,0.700000000,-1.039375000,-1.3040000,
-0.0000750,0.0000125,0.000000000,4.563480616e-08,-1.364242053e-11*e3afca34

#SBAS9A,USB1,1,52.0,SATTIME,2209,514470.000,02000020,b580,16809;138,172,82496,
0,-12514381.3600,-40257277.6000,82561.6000,0.581250000,-1.101250000,23.3200000,
-0.0001250,0.0000250,-0.000437500,2.328306437e-08,-1.091393642e-11*53ac8a8b

#SBAS9A,USB1,0,52.0,SATTIME,2209,514473.000,02000020,b580,16809;133,237,82496,
0,-26616356.6400,-32700988.0000,15871.2000,-0.046250000,-0.230000000,
-0.7200000,-0.0000250,0.0000250,-0.000062500,2.002343535e-08,-4.547473509e-12
*2e70df43
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS9 message can be logged to view the data breakdown of SBAS frame 9 which contains the GEO navigation message.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS9 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H

Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	iodn	Issue of GEO navigation data	Ulong	4	H+4
4	$t_0$	Time of applicability	Ulong	4	H+8
5	ura	URA value	Ulong	4	H+12
6	x	ECEF x coordinate (m)	Double	8	H+16
7	y	ECEF y coordinate (m)	Double	8	H+24
8	z	ECEF z coordinate (m)	Double	8	H+32
9	xvel	X rate of change (m/s)	Double	8	H+40
10	yvel	Y rate of change (m/s)	Double	8	H+48
11	zvel	Z rate of change (m/s)	Double	8	H+56
12	xaccel	X rate of rate change ( $m/s^2$ )	Double	8	H+64
13	yaccel	Y rate of rate change ( $m/s^2$ )	Double	8	H+72
14	zaccel	Z rate of rate change ( $m/s^2$ )	Double	8	H+80
15	$a_{f0}$	Time offset (s)	Double	8	H+88
16	$a_{f1}$	Time drift (s)	Double	8	H+96
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+104
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### 3.193 SBAS10

#### *Degradation factor*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The fast corrections, long term corrections and ionospheric corrections are all provided in the SBAS10 message.

**Message ID:** 978

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS10a onchanged
```

**ASCII Example:**

```
#SBAS10A,USB1,0,52.5,SATTIME,2209,514522.000,02000020,c305,16809;131,54,38,76,256,152,100,311,83,256,6,836,300,0,0,0,000000000000000000000000*028dd85b
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS10 message can be logged to view the data breakdown of SBAS frame 10 which contains information about degradation factors.

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
1	Log header	SBAS10 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	b <sub>rcc</sub>	Estimated noise and round off error parameter	Ulong	4	H+4	0.002
4	C <sub>lrc</sub> _lsb	Maximum round off due to the least significant bit (lsb) of the orbital clock	Ulong	4	H+8	0.002
5	C <sub>lrc</sub> _vl	Velocity error bound	Ulong	4	H+12	0.00005
6	i <sub>lrc</sub> _vl	Update interval for v=1 long term	Ulong	4	H+16	-
7	C <sub>lrc</sub> _v0	Bound on update delta	Ulong	4	H+20	0.002
8	i <sub>lrc</sub> _v1	Minimum update interval v = 0	Ulong	4	H+24	-
9	C <sub>geo</sub> _lsb	Maximum round off due to the lsb of the orbital clock	Ulong	4	H+28	0.0005

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
10	c <sub>geo_v</sub>	Velocity error bound	Ulong	4	H+32	0.00005
11	i <sub>geo</sub>	Update interval for GEO navigation message	Ulong	4	H+36	-
12	c <sub>er</sub>	Degradation parameter	Ulong	4	H+40	0.5
13	c <sub>iono_step</sub>	Bound on ionospheric grid delay difference	Ulong	4	H+44	0.001
14	i <sub>iono</sub>	Minimum ionospheric update interval	Ulong	4	H+48	-
15	c <sub>iono_ramp</sub>	Rate of ionospheric corrections change	Ulong	4	H+52	0.000005
16	rss <sub>udre</sub>	User differential range error flag	Ulong	4	H+56	-
17	rss <sub>iono</sub>	Root sum square flag	Ulong	4	H+60	-
18	spare bits	Spare 88 bits, possibly GLONASS	Hex[11]	11	H+64	-
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+75	-
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

## 3.194 SBAS12

### *SBAS network time and UTC*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS12 contains information bits for the UTC parameters and UTC time standard from which an offset is determined. The UTC parameters correlate UTC time with the SBAS network time rather than with GPS reference time.

**Message ID:** 979

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS12a onchanged
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS12 message can be logged to view the data breakdown of SBAS frame 12 which contains information about time parameters.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS12 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	A <sub>1</sub>	Time drift (s/s)	Double	8	H+4
4	A <sub>0</sub>	Time offset (s)	Double	8	H+12
5	seconds	Seconds into the week (s)	Ulong	4	H+20
6	week	Week number	Ushort	2	H+24
7	dt <sub>ls</sub>	Delta time due to leap seconds	Short	2	H+26
8	wn <sub>lsf</sub>	Week number, leap second future	Ushort	2	H+28
9	dn	Day of the week (the range is 1 to 7 where Sunday = 1 and Saturday = 7)	Ushort	2	H+30
10	dt <sub>lsf</sub>	Delta time, leap second future	Ushort	2	H+32
11	utc id	UTC type identifier	Ushort	2	H+34
12	gpstow	GPS reference time of the week	Ulong	4	H+36
13	gpswn	GPS de-modulo week number	Ulong	4	H+40

Field	Field type	Description	Format	Binary Bytes	Binary Offset
14	glo indicator	Is GLONASS information present? 0 = FALSE 1 = TRUE	Enum	4	H+44
15		Reserved array of hexabytes for GLONASS	Char[10]	12	H+48
16	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+60
17	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.195 SBAS17

### *GEO Almanac message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

Almanacs for all GEOs are broadcast periodically to alert you of their existence, location, the general service provided, status and health.



Unused almanacs have a PRN number of 0 and should be ignored, see *ASCII Example* below.

**Message ID:** 980

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS17a onchanged
```

**ASCII Example:**

```
#SBAS17A,USB1,0,52.0,SATTIME,2209,514511.000,02000020,896c,16809;133,3,0,131,0,-19136000,-37572600,0,0,0,0,0,133,0,-26616200,-32700200,26000,0,0,0,0,135,0,-24167000,-34541000,0,0,0,0,82496*a44a79e8
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS17 message can be logged to view the data breakdown of SBAS frame 17 which contains GEO almanacs.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS17 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	#ents	Number of almanac entries with information to follow	Ulong	4	H+4
4	data id	Data ID type	Ushort	2	H+8
5	entry prn	PRN for this entry	Ushort	2	H+10
6	health	Health bits	Ushort	4 <sup>1</sup>	H+12
7	x	ECEF x coordinate (m)	Long	4	H+16
8	y	ECEF y coordinate (m)	Long	4	H+20

<sup>1</sup>In the binary log case, an additional 2 bytes of padding is added to maintain 4-byte alignment.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
9	z	ECEF z coordinate (m)	Long	4	H+24
10	x vel	X rate of change (m/s)	Long	4	H+28
11	y vel	Y rate of change (m/s)	Long	4	H+32
12	z vel	Z rate of change (m/s)	Long	4	H+36
13...	Next entry = H+8+(#ents x 32)				
14	t0	Time of day in seconds (0 to 86336) Scaling = 64	Ulong	4	H+8+ (#ents x 32)
15	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12+ (#ents x 32)
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.196 SBAS18

### IGP mask

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The ionospheric delay corrections are broadcast as vertical delay estimates, at specified ionospheric grid points (IGPs), applicable to a signal on L1. The predefined IGPs are contained in 11 bands (numbered 0 to 10). Bands 0-8 are vertical bands on a Mercator projection map and bands 9-10 are horizontal bands on a Mercator projection map. Since it is impossible to broadcast IGP delays for all possible locations, a mask is broadcast to define the IGP locations providing the most efficient model of the ionosphere at the time.

**Message ID:** 981

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS18a onchanged
```

**ASCII Example:**

```
#SBAS18A,USB1,0,52.5,SATTIME,2209,514594.000,02000020,f2c0,16809;133,5,9,3,ffff
ffffe000000000fffff00007ffff00007ffffe00007f810000,0*396a66e6
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS18 message can be logged to view the data breakdown of SBAS frame 18 which contains information about ionospheric grid points.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS18 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	#bands	Number of bands broadcast	Ulong	4	H+4
4	band num	Specific band number that identifies which of the 11 IGP bands the data belongs to	Ulong	4	H+8
5	iodi	Issue of ionospheric data	Ulong	4	H+12
6	igp mask	IGP mask	Uchar[26]	28 <sup>1</sup>	H+16
7	spare bit	One spare bit	Ulong	4	H+44

<sup>1</sup>In the binary log case, an additional 2 bytes of padding are added to maintain 4-byte alignment.

---

Field	Field type	Description	Format	Binary Bytes	Binary Offset
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.197 SBAS24

### *Mixed fast/slow corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

If there are 6 or fewer satellites in a block, they may be placed in this mixed correction message. There is a fast data set for each satellite and a UDRE indicator. Each message also contains an IODP indicating the associated PRN mask.

The fast correction (PRC) has a valid range of -2048 to +2047. If the range is exceeded, a don't use indication is inserted into the user differential range error indicator (UDREI) field, see *Table 204: Evaluation of UDREI* on page 916. You should ignore extra data sets not represented in the PRN mask.

The time of applicability (T0) of the PRC is the start of the epoch of the WNT second that is coincident with the transmission at the GEO satellite of the first bit of the message block.

**Message ID: 983**

**Log Type: Asynch**

**Recommended Input:**

```
log SBAS24a onchanged
```

**ASCII Example:**

```
#SBAS24A, COM1, 0, 34.0, SATTIME, 1337, 417108.000, 02000000, 0a33, 1984;134, 2047, 2047,
2047, 2047, -1, -2, 14, 14, 14, 14, 11, 14, 2, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
*76ff954b
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS24 message can be logged to view the data breakdown of SBAS frame 24 which contains mixed fast/slow corrections.

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
1	Log header	SBAS24 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
3	prc0	Fast corrections (-2048 to +2047) for the PRN in slot i (i = 0-5)	Long	4	H+4	-
4	prc1		Long	4	H+8	-
5	prc2		Long	4	H+12	-
6	prc3		Long	4	H+16	-
7	prc4		Long	4	H+20	-
8	prc5		Long	4	H+24	-
9	udre0	User differential range error indicator for the PRN in slot i (i = 0-5)	Ulong	4	H+28	See Table 204: Evaluation of UDREI on page 916
10	udre1		Ulong	4	H+32	
11	udre2		Ulong	4	H+36	
12	udre3		Ulong	4	H+40	
13	udre4		Ulong	4	H+44	
14	udre5		Ulong	4	H+48	
15	iodp	Issue of PRN mask data	Ulong	4	H+52	-
16	block id	Associated message type	Ulong	4	H+56	
17	iodf	Issue of fast corrections data	Ulong	4	H+60	-
18	spare	Spare value	Ulong	4	H+64	-
19	vel	Velocity code flag	Ulong	4	H+68	-
20	mask1	Index into PRN mask (Type 1)	Ulong	4	H+72	-
21	iode1	Issue of ephemeris data	Ulong	4	H+76	-
22	dx1	Delta x (ECEF)	Long	4	H+80	0.125
23	dy1	Delta y (ECEF)	Long	4	H+84	0.125
24	dz1	Delta z (ECEF)	Long	4	H+88	0.125
25	da <sub>f0</sub>	Delta a <sub>f0</sub> clock offset	Long	4	H+92	2 <sup>-31</sup>
26	mask2	Second index into PRN mask (Type 1)	Ulong	4	H+96	-
27	iode2	Second issue of ephemeris data	Ulong	4	H+100	-
28	ddx	Delta delta x (ECEF)	Long	4	H+104	2 <sup>-11</sup>
29	ddy	Delta delta y (ECEF)	Long	4	H+108	2 <sup>-11</sup>
30	ddz	Delta delta z (ECEF)	Long	4	H+112	2 <sup>-11</sup>

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
31	da <sub>f1</sub>	Delta a <sub>f1</sub> clock offset	Long	4	H+116	2 <sup>-39</sup>
32	t <sub>0</sub>	Applicable time of day	Ulong	4	H+120	16
33	iodp	Issue of PRN mask data	Ulong	4	H+124	-
34	corr spare	Spare value when velocity code is equal to 0	Ulong	4	H+128	-
35	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+132	-
36	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

## 3.198 SBAS25

### *Long term slow satellite corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS25 provides error estimates for slow varying satellite ephemeris and clock errors with respect to WGS84 ECEF coordinates.

**Message ID:** 984

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS25a onchanged
```

**ASCII Example:**

```
#SBAS25A,USB1,0,51.5,SATTIME,2209,514661.000,02000020,b8ff,16809;133,1,16,94,
23,-11,1,-3,0,0,-1,-1,1,-6,5157,2,0,1,31,75,0,12,1,12,0,0,-1,-1,1,0,5157,2,0
*3341fe84
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS25 message can be logged to view the data breakdown of SBAS frame 25 which contains long term slow satellite corrections.

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
1	Log header	SBAS25 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	1st half vel	Velocity code flag (0 or 1)	Ulong	4	H+4	-
4	1st half mask1	Index into PRN mask (Type 1)	Ulong	4	H+8	-
5	1st half iode1	Issue of ephemeris data	Ulong	4	H+12	-
6	1st half dx1	Delta x (ECEF)	Long	4	H+16	0.125
7	1st half dy1	Delta y (ECEF)	Long	4	H+20	0.125

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
8	1st half dz1	Delta z (ECEF)	Long	4	H+24	0.125
9	1st half a <sub>f0</sub>	Delta a <sub>f0</sub> clock offset	Long	4	H+28	2 <sup>-31</sup>
10	1st half mask2	Second index into PRN mask (Type 1) Dummy value when velocity code = 1	Ulong	4	H+32	-
11	1st half iode2	Second issue of ephemeris data Dummy value when velocity code = 1	Ulong	4	H+36	-
12	1st half ddx	Delta delta x (ECEF) when velocity code = 1 Delta x (dx) when velocity code = 0	Long	4	H+40	2 <sup>-11</sup>
13	1st half ddy	Delta delta y (ECEF) when velocity code = 1 Delta y (dy) when velocity code = 0	Long	4	H+44	2 <sup>-11</sup>
14	1st half ddz	Delta delta z (ECEF) when velocity code = 1 Delta z (dz) when velocity code = 0	Long	4	H+48	2 <sup>-11</sup>
15	1st half a <sub>f1</sub>	Delta a <sub>f1</sub> clock offset when velocity code = 1 Delta a <sub>f0</sub> clock offset when velocity code = 0	Long	4	H+52	2 <sup>-39</sup>
16	1st half t <sub>0</sub>	Applicable time of day Dummy value when velocity code = 0	Ulong	4	H+56	16
17	1st half iodp	Issue of PRN mask data	Ulong	4	H+60	-
18	1st half corr spare	Spare value when velocity code = 0 Dummy value when velocity code = 1	Ulong	4	H+64	-
19	2nd half vel	Velocity code flag (0 or 1)	Ulong	4	H+68	-
20	2nd half mask1	Index into PRN mask (Type 1)	Ulong	4	H+72	-
21	2nd half iode1	Issue of ephemeris data	Ulong	4	H+76	-
22	2nd half dx1	Delta x (ECEF)	Long	4	H+80	0.125

Field	Field type	Description	Format	Binary Bytes	Binary Offset	Scaling
23	2nd half dy1	Delta y (ECEF)	Long	4	H+84	0.125
24	2nd half dz1	Delta z (ECEF)	Long	4	H+88	0.125
25	2nd half a <sub>f0</sub>	Delta a <sub>f0</sub> clock offset	Long	4	H+92	2 <sup>-31</sup>
26	2nd half mask2	Second index into PRN mask (Type 1) Dummy value when velocity code = 1	Ulong	4	H+96	-
27	2nd half iode2	Second issue of ephemeris data Dummy value when velocity code = 1	Ulong	4	H+100	-
28	2nd half ddx	Delta delta x (ECEF) when velocity code = 1 Delta x (dx) when velocity code = 0	Long	4	H+104	2 <sup>-11</sup>
29	2nd half ddy	Delta delta y (ECEF) when velocity code = 1 Delta y (dy) when velocity code = 0	Long	4	H+108	2 <sup>-11</sup>
30	2nd half ddz	Delta delta z (ECEF) when velocity code = 1 Delta z (dz) when velocity code = 0	Long	4	H+112	2 <sup>-11</sup>
31	2nd half a <sub>f1</sub>	Delta a <sub>f1</sub> clock offset when velocity code = 1 Delta a <sub>f0</sub> clock offset when velocity code = 0	Long	4	H+116	2 <sup>-39</sup>
32	2nd half t <sub>0</sub>	Applicable time of day Dummy value when velocity code = 0	Ulong	4	H+120	16
33	2nd half iodp	Issue of PRN mask data	Ulong	4	H+124	-
34	2nd half corr spare	Spare value when velocity code = 0 Dummy value when velocity code = 1	Ulong	4	H+128	-
35	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+132	-
36	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-



## 3.199 SBAS26

### *Ionospheric delay corrections*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS26 provides vertical delays (relative to an L1 signal) and their accuracy at geographically defined IGPs identified by the BAND NUMBER and IGP number. Each message contains a band number and a block ID, which indicates the location of the IGPs in the respective band mask.

**Message ID:** 985

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS26a onchanged
```

**ASCII Example:**

```
#SBAS26A,USB1,0,50.5,SATTIME,2209,514696.000,02000020,ec70,16809;131,2,0,15,91,12,73,11,60,9,52,9,44,9,39,9,35,9,34,10,31,11,85,12,70,11,58,10,50,9,43,9,39,9,3,0*d3271683
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS26 message can be logged to view the data breakdown of SBAS frame 26 which contains ionospheric delay corrections

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS26 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	band num	Band number	Ulong	4	H+4
4	block id	Block ID	Ulong	4	H+8
5	#pts	Number of grid points with information to follow	Ulong	4	H+12
6	igp <sub>vde</sub>	IGP vertical delay estimates Scaling = 0.125	Ulong	4	H+16
7	givei	Grid ionospheric vertical error indicator	Ulong	4	H+20
8...	Next #pts entry = H + 16 + (#pts x 8)				
9	iodi	Issue of data - ionosphere	Ulong	4	H+16+ (#pts x 8)

Field	Field type	Description	Format	Binary Bytes	Binary Offset
10	spare	7 spare bits	Ulong	4 <sup>1</sup>	H+20+ (#pts x 8)
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+24+ (#pts x 8)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

---

<sup>1</sup>In the binary case, additional bytes of padding are added after this field to maintain 4-byte alignment for the fields that follow.

## 3.200 SBAS27

### *SBAS service message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

SBAS27 messages apply only to the service provider transmitting the message. The number of service messages indicates the total number of unique SBAS27 messages for the current IODS. Each unique message for that IODS includes a sequential message number. The IODS is incremented in all messages, each time that any parameter in any SBAS27 message is changed.

**Message ID:** 986

**Log Type:** Asynch

**Recommended Input:**

```
log SBAS27a onchanged
```



Each raw SBAS frame gives data for a specific frame decoder number. The SBAS27 message can be logged to view the data breakdown of SBAS frame 27 which contains information about SBAS service messages.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBAS27 header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iods	Issue of slow corrections data	Ulong	4	H+4
4	#messages	Low-by-one count of messages	Ulong	4	H+8
5	message num	Low-by-one message number	Ulong	4	H+12
6	priority code	Priority code	Ulong	4	H+16
7	dudre inside	Delta user differential range error – inside	Ulong	4	H+20
8	dudre outside	Delta user differential range error – outside	Ulong	4	H+24
9...	#reg	Number of regions with information to follow	Ulong	4	H+28
10	lat1	Coordinate 1 latitude	Long	4	H+32
11	lon1	Coordinate 1 longitude	Long	4	H+36
12	lat2	Coordinate 2 latitude	Long	4	H+40

Field	Field type	Description	Format	Binary Bytes	Binary Offset
13	lon2	Coordinate 2 longitude	Long	4	H+44
14	shape	Shape where: 0 = triangle, 1 = square	Ulong	4	H+48
15	Next #reg entry = H+32+(#reg x 20)				
16	Reserved		Ulong	4	H+32+ (#reg x 20)
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36+ (#reg x 20)
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.201 SBASALMANAC

### *SBAS Almanac collection*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains a collection of all current SBAS almanacs that have been decoded by the receiver and may contain almanac data for multiple PRNs. The SBASALMANAC log is populated by the GEO Almanac Message Type 17 which is available in the **SBAS17** log (see page 935). These PRNs are broken out into individual SBAS almanac entries for this message and output individually. If multiple SBAS subsystems (e.g., WAAS, EGNOS, GAGAN, MSAS) are tracked, this message will include almanac data collected from each with the subsystem identified in each message entry. The almanac data contains all of the information required to compute the satellite position as well as health and status information.

The OEM7 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

**Message ID:** 1425

**Log Type:** Asynch

**Recommended Input:**

Log SBASALMANACA onchanged

**ASCII Example:**

```
#SBASALMANACA,USB1,3,50.5,SATTIME,2209,514769.000,02000020,b46b,16809;131,WAAS,
82752,0,0,-19136000,-37572600,0,0,0,0*a1514a7e
#SBASALMANACA,USB1,2,50.5,SATTIME,2209,514769.000,02000020,b46b,16809;133,WAAS,
82752,0,0,-26616200,-32700200,26000,0,0,0*d4db7b3c
#SBASALMANACA,USB1,1,50.5,SATTIME,2209,514769.000,02000020,b46b,16809;135,WAAS,
82752,0,0,-24164400,-34543600,0,0,0,0*f26489ca
#SBASALMANACA,USB1,0,50.5,SATTIME,2209,514720.000,02000020,b46b,16809;138,WAAS,
82496,0,0,-12513800,-40258400,78000,0,0,0*7e01e7f5
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SBASALMANAC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Satellite ID	Satellite ID	Ulong	4	H
3	Variant	System variant (refer to <i>Table 205: SBAS Subsystem Types</i> on the next page)	Enum	4	H + 4
4	Time	Time of day (s)	Ulong	4	H + 8
5	Data ID	Data identification	Ushort	2	H + 12

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	Health	Satellite health	Ushort	2	H + 14
7	X	ECEF X coordinate (m)	Long	4	H + 16
8	Y	ECEF Y coordinate (m)	Long	4	H + 20
9	Z	ECEF Z coordinate (m)	Long	4	H + 24
10	X Velocity	X rate of change (m/s)	Long	4	H + 28
11	Y Velocity	Y rate of change (m/s)	Long	4	H + 32
12	Z Velocity	Z rate of change (m/s)	Long	4	H + 36
13	CRC	32-bit CRC (ASCII and binary only)	Ulong	4	H + 40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 205: SBAS Subsystem Types

ASCII	Binary	Description
NONE	0	No system
UNKNOWN	1	Unknown system
WAAS	2	WAAS system
EGNOS	3	EGNOS system
MSAS	4	MSAS system
GAGAN	5	GAGAN system
QZSS	7	QZSS System

## 3.202 SOFTLOADSTATUS

*Describes the status of the SoftLoad process*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log describes the status of the SoftLoad process.



Status values  $\geq 16$  (ERROR) indicate that an error has occurred during the loading process. Status  $< 16$  (ERROR) are part of normal SoftLoad operation.

**Message ID:** 1235

**Log Type:** Asynch

**Recommended Input:**

```
log softloadstatusa onchanged
```

**ASCII Example:**

```
#SOFTLOADSTATUSA,USB1,0,50.5,UNKNOWN,0,0.000,02000020,2d64,16809;  
NOT_STARTED*248aced3
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SOFTLOADSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	status	Status of the SoftLoad process see <i>Table 206: SoftLoad Status Type</i> below	Enum	4	H
3	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 206: SoftLoad Status Type**

Value	Name	Description
1	NOT_STARTED	SoftLoad process has not begun.
2	READY_FOR_SETUP	SoftLoad process is ready to receive setup information in the form of the <b>SOFTLOADSETUP</b> command or <b>SOFTLOADSREC</b> command with S0 records. Once sufficient setup data has been sent, the process is also ready for the <b>SOFTLOADDATA</b> command.

Value	Name	Description
3	READY_FOR_DATA	SoftLoad process is ready to receive data in the form of the <b>SOFTLOADDATA</b> command or <b>SOFTLOADSREC</b> command with S3 records. Once all data has been sent, send the <b>SOFTLOADCOMMIT</b> command.
4	DATA_VERIFIED	SoftLoad data has passed CRC. This status occurs after a <b>SOFTLOADCOMMIT</b> command.
5	WRITING_FLASH	SoftLoad data is being written to flash. This status occurs after a <b>SOFTLOADCOMMIT</b> command. During a firmware upload, the receiver may remain in this state for 300 seconds or longer.
6	WROTE_FLASH	SoftLoad data has been written to flash.
7	WROTE_AUTHCODE	The embedded AuthCode was successfully written.
8	COMPLETE	SoftLoad process has completed. The next step is to send the <b>RESET</b> command to reset the receiver.
9	VERIFYING_DATA	SoftLoad is verifying the downloaded image.
10	COPIED_SIGNATURE_AUTH	Signature AuthCodes have been copied from the current firmware to the downloaded firmware.
11	WROTE_TRANSACTION_TABLE	The downloaded firmware has been activated and will be executed if the receiver is reset. This status is effectively identical to COMPLETE.
13	PROCESSING_FILE	The <b>SOFTLOADFILE</b> command is processing the firmware file.
16	ERROR	Indicates an internal error in the SoftLoad process. This error is not expected to occur. Contact NovAtel Customer Support for assistance.
17	RESET_ERROR	Error resetting SoftLoad. Reset the receiver and restart the SoftLoad process.
18	BAD_SRECORD	A bad S Record was received. Ensure that S Records are enclosed in double quotes within the <b>SOFTLOADSREC</b> command.
19	BAD_PLATFORM	This data cannot be loaded onto this platform. Ensure that the correct *.shex file for the platform is being used.
20	BAD_MODULE	This module cannot be loaded with SoftLoad. This file must be loaded using WinLoad or a similar loader.
21	BAD_AUTHCODE	Bad AuthCode received for this PSN.
22	NOT_READY_FOR_SETUP	A <b>SOFTLOADSETUP</b> command was entered before a <b>SOFTLOADRESET</b> command or after a <b>SOFTLOADDATA</b> command.
23	NO_MODULE	No data type was entered before a <b>SOFTLOADDATA</b> command was received. Set the data type using the <b>SOFTLOADSETUP</b> command or <b>SOFTLOADSREC</b> command with an "S0~T~" S Record.



Value	Name	Description
24	NO_PLATFORM	No platform was entered before a <b>SOFTLOADDATA</b> command was received. Set the platform using the <b>SOFTLOADSETUP</b> command or <b>SOFTLOADSREC</b> command with an "S0~P~" S Record.
25	NOT_READY_FOR_DATA	A <b>SOFTLOADDATA</b> command was received but the receiver was not ready for it.
26	MODULE_MISMATCH	The SoftLoad data module was changed in the middle of loading. Restart the SoftLoad process using the <b>SOFTLOADRESET</b> command.
27	OUT_OF_MEMORY	SoftLoad has run out of RAM to store the incoming data. Reset the receiver and restart the SoftLoad process.
28	DATA_OVERLAP	SoftLoad data has overlapped. Ensure that the correct address and length is set in the <b>SOFTLOADDATA</b> command or <b>SOFTLOADSREC</b> command.
29	BAD_IMAGE_CRC	CRC of the downloaded image has failed. Ensure that all content from the *.shex file has been successfully downloaded.
30	IMAGE_OVERSIZE	The downloaded image is too big for the intended data module.
31	AUTHCODE_WRITE_ERROR	An error occurred when writing the embedded AuthCode to flash.
32	BAD_FLASH_ERASE	Erasing of the flash failed. This could indicate a failure in the flash hardware.
33	BAD_FLASH_WRITE	Writing to the flash failed. This could indicate a failure in the flash hardware.
34	TIMEOUT	SoftLoad time out has occurred.
35	INCOMPATIBLE_FLASH	Application image that does not support the onboard flash rejected.

## 3.203 SOURCETABLE

### *NTRIP source table entries*

**Platform:** OEM729, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, SMART7-W, SMART7-I

This log outputs the NTRIP source table entries from the NTRIP caster set by the **NTRIPSOURCETABLE** command (see page 281). The entry data field in the first entry is always the header of the retrieved SOURCETABLE. The entry data field in the last entry is the string "ENDSOURCETABLE" which indicates the end of the source table. Entries in between these fields are the real SOURCETABLE entries.



The maximum number of entries is set to 500. If a caster has more than 500 entries, the first 500 entries are output and the last entry does not have the string "ENDSOURCETABLE". This indicates that not all of the caster entries available are shown in the log.

**Message ID:** 1344

**Log Type:** Polled

**Recommended Input:**

```
log sourcetablea once
```

**ASCII Example:**

```
#SOURCETABLEA,COM1,17,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"HTTP/1.1 200 OK;Ntrip-Version: Ntrip/2.0;Ntrip-
Flags: st_filter,st_auth,st_match,st_strict,rtsp,plain_rtp;Server: NTRIP
Caster/2.0.15;Date: Fri, 27 Jan 2017 18:12:01 GMT;Connection: close;Content-
Type: gnss/sourcetable;Content-Length: 2057"*87a7d39d

#SOURCETABLEA,COM1,16,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"CAS;hera.novatel.ca;80,2101;NovAtel;NovAtel;0;CAN;
51;-115;http://www.novatel.com"*e3ec11a0

#SOURCETABLEA,COM1,15,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"NET;GREF;NovAtel;B;N;http://novatel.com;none;novat
el.com;none"*2a6b50eb

#SOURCETABLEA,COM1,14,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"STR;novatel_rtcmv3;Office Roof DL1L2;RTCM 3.0;1033
(10),1005(10),1019(60),1020(60),1003(1),1011(1);2;GPS+GLO;NovAtel;CAN;51;-115;
0;0;NovAtel OEM628;none;B;N;9600;Test"*8a7c760f

#SOURCETABLEA,COM1,13,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"STR;novatel_rtcmm;Office Roof DL1L2;RTCM 2.3;1(1),3
(10),31(1),32(10);0;GPS+GLO;NovAtel;CAN;51;-115;0;0;NovAtel OEM628;none;B;N;
9600;Test"*08c57cb7

#SOURCETABLEA,COM1,12,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"STR;novatel_rtca;Office Roof DL1L2;RTCA;RTCAREF
(10),RTCA1(1),RTCAEPHEM(60);0;GPS;NovAtel;CAN;51;-115;0;0;NovAtel OEM628;none;
B;N;9600;Test"*006997bc

#SOURCETABLEA,COM1,11,84.0,COARSESTEERING,1933,497547.000,02400020,71dd,32768;
"hera.novatel.com:2101",0,0,"STR;novatel_cmrm;Office Roof DL1L2;CMR;CMRREF(10),
```

```
CMROBS (1) , CMRGLOOBS (1) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel OEM628 ; none ;
B ; N ; 9600 ; Test "*"0955ccb7

#SOURCETABLEA, COM1, 10, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; novatel_rtcaobs2 ; Office Roof DL1L2 ; RTCA ;
rtcaref (10) , rtcaobs2 (1) , rtcaephem (60) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel
OEM628 ; none ; B ; N ; 9600 ; Test "*"426e39a5

#SOURCETABLEA, COM1, 9, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; novatel_cmrplus ; Office Roof DL1L2 ; CMR+ ; cmrplus
(1) , cmrobs (1) , cmrgloobs (1) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel OEM628 ;
none ; B ; N ; 9600 ; Test "*"2d5ba56e

#SOURCETABLEA, COM1, 8, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ; "h
era.novatel.com:2101", 0, 0, "STR ; novatel_rtcm2021 ; Office Roof DL1L2 ; RTCM 2.3 ; 3
(10) , 2021 (1) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel
OEM628 ; none ; B ; N ; 9600 ; Test "*"d82df5de

#SOURCETABLEA, COM1, 7, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; novatel_1819 ; Office Roof DL1L2 ; RTCM 2.3 ; 3 (10) ,
22 (10) , 23 (60) , 24 (60) , 1819 (1) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel OEM628 ;
none ; B ; N ; 9600 ; Test "*"7aead153

#SOURCETABLEA, COM1, 6, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ; "h
era.novatel.com:2101", 0, 0, "STR ; novatel_rtcaobs ; Office Roof DL1L2 ; RTCA ; rtcaref
(10) , rtcaobs (1) , rtcaephem (60) ; 2 ; GPS+GLO ; NovAtel ; CAN ; 51 ; -115 ; 0 ; 0 ; NovAtel
OEM628 ; none ; B ; N ; 9600 ; Test "*"530a51c4

#SOURCETABLEA, COM1, 5, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; novatel_novatelx ; Office Roof ; NovatelX ;
novatelobs ; 2 ; GPS+GLO ; NovAel ; CAN ; 51 ; -114 ; 0 ; 0 ; NovAtel OEM628 ; none ; B ; N ; 9600 ;
Test "*"4438c2e2

#SOURCETABLEA, COM1, 4, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; Hyderabad1 ; hyderabad test1 ; unknown ; unknown ;
2 ; GPS+GLO ; NovAtel ; INDIA ; 17 ; 78 ; 0 ; 0 ; NovAtel OEM628 ; none ; B ; N ; 9600 ; Test "*"de6c19f0

#SOURCETABLEA, COM1, 3, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; Hyderabad2 ; hyderabad
test1 ; unknown ; unknown ; 2 ; GPS+GLO ; NovAtel ; INDIA ; 17 ; 78 ; 0 ; 0 ; NovAtel
OEM628 ; none ; B ; N ; 9600 ; Test "*"27e9ee1

#SOURCETABLEA, COM1, 2, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ; "h
era.novatel.com:2101", 0, 0, "STR ; Hyderabad3 ; hyderabad test1 ; unknown ; unknown ;
2 ; GPS+GLO ; NovAtel ; INDIA ; 17 ; 78 ; 0 ; 0 ; NovAtel OEM628 ; none ; B ; N ; 9600 ; Test "*"3ed5941b

#SOURCETABLEA, COM1, 1, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "STR ; Hyderabad4 ; hyderabad test1 ; unknown ; unknown ;
2 ; GPS+GLO ; NovAtel ; INDIA ; 17 ; 78 ; 0 ; 0 ; NovAtel OEM628 ; none ; B ; N ; 9600 ; Test "*"a3a188e2

#SOURCETABLEA, COM1, 0, 84.0, COARSESTEERING, 1933, 497547.000, 02400020, 71dd, 32768 ;
"hera.novatel.com:2101", 0, 0, "ENDSOURCETABLE "*"7758fba9
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SOURCETABLE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	endpoint	NTRIPCASTER Endpoint	String with varied length up to 80 bytes	a <sup>1</sup>	H
3	Reserved1	reserved	Ulong	4	H+a
4	Reserved2	reserved	Ulong	4	H+a+4
5	Entry data	Source table entry data	String with varied length up to 512 bytes	b <sup>1</sup>	H+a+8
6	xxxx	32-bit CRC (ASCII and binary only)	Ulong	4	H+a+b+8
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

## 3.204 TECTONICSCOMPENSATION

### *Tectonics Compensation status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log shows the status of the tectonics compensation.

**Message ID:** 2291

**Log Type:** Asynch

**Recommended Input:**

```
log tectonicscompensationa onchanged
```

**ASCII Example:**

```
#TECTONICSCOMPENSATIONA,USB1,0,51.0,COARSESTEERING,2209,511539.016,02000020,3af4,16809;AVAILABLE,"North America",-0.018,-0.000,-0.006*6dac1168
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TECTONICSCOMPENSATION header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	status	Status of plate tectonics compensation. See <i>Table 207: Tectonics Compensation Status</i> below.	Enum	4	H
3	name	Name of the plate that is being used for determining station velocities.	Char(32)	Variable	H+4
4	xvel	Station velocities for the location of the receiver (metres/year)	Float	4	Variable
5	yvel		Float	4	Variable
6	zvel		Float	4	Variable
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	Variable
8	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

**Table 207: Tectonics Compensation Status**

Binary	ASCII	Description
0	UNAVAILABLE	Tectonics compensation is not available.
1	AVAILABLE	Tectonics compensation is available.

---

Binary	ASCII	Description
2	WARNING	The receiver is either close to a plate boundary or inside a region where the tectonics compensation is not well modeled.
4	OFF_PLATE	The receiver has moved from the plate that was initially selected. The receiver will not change the plate to avoid discontinuities in the positions.

## 3.205 TERRASTARINFO

### *TerraStar subscription information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains details on the TerraStar subscription.

**Message ID:** 1719


**Log Type:** Asynch

**Recommended Input:**

```
log terrastarininfoa onchanged
```

**ASCII Example:**

```
#TERRASTARINFOA,USB1,0,46.0,FINESTEERING,2209,511557.037,02000020,91ea,16809;
"QV579:5420:8616",BUBBLE,80007f00,0,0,0,LOCAL_AREA,51.15028,-114.03056,2
*5ac46d87
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TERRASTARINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	PAC	Product activation code	Char[16]	16	H
3	Type	Subscription type (see <i>Table 208: TerraStar Subscription Type</i> on the next page)	Enum	4	H+16
4	Subscription permissions	Services permitted by the subscription (see <i>Table 209: TerraStar Subscription Details Mask</i> on the next page)  <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;">  Bits in the Reserved areas of this field may be set, but the Reserved bits should be ignored. A receiver can have only one active subscription at a time but a subscription can enable multiple services or bits. For example, "00000700" indicates the receiver has a TerraStar-C subscription though TerraStar-L is also allowed. </div>	Hex	4	H+20

Field	Field type	Description	Format	Binary Bytes	Binary Offset
5	Service End Day	Last day of the year for which service is available. Service expires at the end of this UTC day. For example, if the Service End Year and Day are 2019 and 15, respectively, then the service will expire on January 15, 2019 at 24:00 UTC.	Ulong	4	H+24
6	Service End Year	Year that subscription ends	Ulong	4	H+28
7	Reserved		Ulong	4	H+32
8	Region restriction	For region restricted subscriptions, the type of region restriction (see <i>Table 210: TerraStar Region Restriction</i> on the next page)	Enum	4	H+36
9	Center point latitude	For local area subscriptions, the center point latitude (degrees)	Float	4	H+40
10	Center point longitude	For local area subscriptions, the center point longitude (degrees)	Float	4	H+44
11	Radius	For local area subscriptions, the maximum permitted distance from center point (kilometres)	Ulong	4	H+48
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+52
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 208: TerraStar Subscription Type

ASCII	Binary	Description
UNASSIGNED	0	Decoder has not had an assigned operating mode
TERM	1	Term subscription
MODEL	5	Receiver is operating with an RTK assist enabled model and there is not an active TerraStar subscription installed
BUBBLE	100	Receiver is operating in a TerraStar-permitted subscription-free bubble
INCOMPATIBLE_SUBSCRIPTION	104	Subscription is incompatible with this version of firmware

Table 209: TerraStar Subscription Details Mask

Bit	Mask	Description
0-8	0x000001FF	Reserved



Bit	Mask	Description
9	0x00000200	TerraStar-C service
10	0x00000400	TerraStar-L service
11	0x00000800	RTK ASSIST service
12	0x00001000	RTK ASSIST PRO service
13	0x00002000	TerraStar-C PRO service
14	0x00004000	TerraStar-X Regional service
15-30	0x7FFF8000	Reserved
31	0x80000000	Secure Internet Delivery

Table 210: TerraStar Region Restriction

ASCII	Binary	Description
NONE	0	TerraStar operation has no region restrictions.
GEOGATED	1	TerraStar operation limited to on-land GEOGATED is also the default value reported if there is no subscription
LOCAL_AREA	2	TerraStar operation limited to radius from local area center point
NEARSHORE	3	TerraStar operation limited to on land and near shore (coastal) regions

## 3.206 TERRASTARSTATUS

### *TerraStar decoder and subscription status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains status information for the TerraStar decoder and subscription.

**Message ID:** 1729

**Log Type:** Asynch

**Recommended Input:**

```
log terrastarstatusa onchanged
```

**ASCII Example:**

```
#TERRASTARSTATUSA,USB1,0,50.0,FINESTEERING,2209,515067.036,02000020,32bc,16809;
ENABLE,LOCKED,0,IN_RANGE,DISABLED*a17b5059
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TERRASTARSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Access	Access status. ENABLE (1) if the subscription is valid; DISABLE (0) otherwise	Enum	4	H
3	Sync state	Decoder data synchronization state (see <i>Table 211: Decoder Data Synchronization State</i> below)	Enum	4	H+4
4	Reserved		Ulong	4	H+8
5	Local area status	For local-area subscriptions, indicates if the receiver is within the permitted area (see <i>Table 212: TerraStar Local Area Status</i> on the next page)	Enum	4	H+12
6	Geogating status	Geogating status (see <i>Table 213: TerraStar Geogating Status</i> on the next page)	Enum	4	H+16
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+20
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 211: Decoder Data Synchronization State**

ASCII	Binary	Description
NO_SIGNAL	0	None of the decoders have received data in the last 30 seconds

ASCII	Binary	Description
SEARCH	1	At least one decoder is receiving data and is searching for the format
LOCKED	2	At least one decoder has locked onto the format

Table 212: TerraStar Local Area Status

ASCII	Binary	Description
DISABLED	0	The subscription is not restricted to a local area. This is also the value when there is no subscription.
WAITING_FOR_POSITION	1	Waiting for a position
RANGE_CHECK	16	Checking position against local area region restriction
IN_RANGE	129	Receiver is within the permitted local area
OUT_OF_RANGE	130	Receiver is outside the permitted local area
POSITION_TOO_OLD	255	Position is too old

Table 213: TerraStar Geogating Status

ASCII	Binary	Description
DISABLED	0	The subscription is restricted to a local area or there is no region restriction. This is also the value when there is no subscription.
WAITING_FOR_POSITION	1	Waiting for a position
ONSHORE	129	Receiver is over land
OFFSHORE	130	Receiver is over water
POSITION_TOO_OLD	255	Position is too old
PROCESSING	1000	Geogater is determining status

## 3.207 TILTDATA

### *Tilt filter accelerometer measurements and computed inclination*

**Platform:** SMART7-I, SMART7-W, SMART2-TB

For SMART antenna products with Terrain Compensation, this log contains the instantaneous horizontal accelerometer measurements, as well as the filtered inclination values computed based on the movement of the vehicle.

**Message ID:** 1220

**Log Type:** Synch

**Recommended Input:**

```
log tiltdataa ontime 1
```

**ASCII Example:**

```
#TILTDATAA,COM1,0,73.0,FINESTEERING,2045,504349.000,02000000,16f3,32768;
00000000,0.002557007,0.000229843,0.000000000,0.064623833,-1.836378931,
0.000000000,0*5ec9e0e3
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TILTDATA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	Sensor Status	Sensor reported status data. 0 = Good 1 = Error	Hex	4	H
3	X Acceleration	X-axis acceleration (m/s <sup>2</sup> )	Double	8	H+4
4	Y Acceleration	Y-axis acceleration (m/s <sup>2</sup> )	Double	8	H+12
5	Reserved		N/A	8	H+20
6	X Inclination	Pitch angle (degrees)	Double	8	H+28
7	Y Inclination	Roll angle (degrees)	Double	8	H+36
8	Reserved		N/A	8	H+44
9	Reserved		N/A	4	H+52
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
11	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 3.208 TIME

### *Time data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log provides several time related pieces of information including receiver clock offset and UTC time and offset. It can also be used to determine any offset in the PPS signal relative to GPS reference time.

To find any offset in the PPS signal, log the TIME log 'ontime' at the same rate as the PPS output. For example, if the PPS output is configured to output at a rate of 0.5 seconds (see the **PPSCONTROL** command on page 309) log the TIME log 'ontime 0.5' as follows:

```
log time ontime 0.5
```

The TIME log offset field can then be used to determine any offset in PPS output relative to GPS reference time.



GPS reference time is the receiver's estimate of the true GPS system time. GPS reference time can be found in the header of the TIME log. The relationship between GPS reference time and true GPS system time is:

GPS system time = GPS reference time - offset

**Message ID: 101**

**Log Type: Synch**

**Recommended Input:**

```
log timea ontime 1
```

**ASCII Example:**

```
#TIMEA,USB1,0,50.5,FINESTEERING,2209,515163.000,02000020,9924,16809;VALID,
-2.501488425e-09,6.133312031e-10,-17.99999999630,2022,5,13,23,5,45000,VALID
*1100ad64
```



1. Consider the case where you used the **ADJUST1PPS** command (see page 60) to synchronize two receivers in a primary/secondary relationship to a common external clock. You can use the TIME log after the clock model status is valid to monitor the time difference between the Primary and Secondary receivers.
2. The header of the TIME log gives you the GPS reference time (the week number since January 5th, 1980) and the seconds into that week. The TIME log outputs the UTC offset (offset of GPS system time from UTC time) and the receiver clock offset from GPS system time.

If you want the UTC time in weeks and seconds, take the week number from the header. Then take the seconds into that week, also from the header, and add the correction to the seconds using the 2 offsets. Ensure not to go negative or rollover (go over the total number of seconds, 604800, in a week). In the case of a rollover, add a week and the left over seconds become the seconds into this new week. If negative, subtract a week and the remainder from the seconds of that week become the seconds into this new week.

For example:

```
TIME COM1 0 73.5 FINESTEERING 1432 235661.000 02000000 9924 2616 VALID
-0.000000351 0.000000214 -14.00000000106 2007 6 19 17 27 27000 VALID
```

From the time information above:

GPS reference time = 1432 (GPS reference week), 235661.000 (GPS seconds) from the header.

From the description in *UTC offset* row in the following table:

UTC time = GPS reference time - offset + UTC offset

UTC time

= week 1432, 235661.000 s – (- 0.000000351 (offset) ) - 14.00000000106 (UTC offset)

= week 1432, seconds 235647.00000034994

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TIME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	clock status	Clock model status (not including current measurement data), see <i>Table 101: Clock Model Status</i> on page 549	Enum	4	H
3	offset	Receiver clock offset in seconds from GPS system time. A positive offset implies that the receiver clock is ahead of GPS system time. To derive GPS system time, use the following formula:  GPS system time = GPS reference time - offset.  The GPS reference time can be obtained from the log header.	Double	8	H+4

Field	Field type	Description	Format	Binary Bytes	Binary Offset
4	offset std	Receiver clock offset standard deviation (s)	Double	8	H+12
5	utc offset	The offset of GPS system time from UTC time, computed using almanac parameters. UTC time is GPS reference time plus the current UTC offset minus the receiver clock offset:  UTC time = GPS reference time - offset + UTC offset	Double	8	H+20
6	utc year	UTC year	Ulong	4	H+28
7	utc month	UTC month (0-12) If UTC time is unknown, the value for month is 0.	Uchar	1	H+32
8	utc day	UTC day (0-31) If UTC time is unknown, the value for day is 0.	Uchar	1	H+33
9	utc hour	UTC hour (0-23)	Uchar	1	H+34
10	utc min	UTC minute (0-59)	Uchar	1	H+35
11	utc ms	UTC millisecond (0-60999) Maximum of 60999 when leap second is applied.	Ulong	4	H+36
12	utc status	UTC status 0 = Invalid 1 = Valid 2 = Warning <sup>1</sup>	Enum	4	H+40
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

<sup>1</sup>Indicates that the leap second value is used as a default due to the lack of an almanac.

## 3.209 TIMESYNC

### *Synchronize time between GNSS receivers*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The TIMESYNC log is used in conjunction with the **ADJUST1PPS** command (see page 60) to synchronize the time between GNSS receivers.

**Message ID:** 492

**Log Type:** Synch

**Recommended Input:**

```
log timesynca ontime 1
```

**ASCII Example:**

```
#TIMESYNCA,USB1,0,50.0,FINESTEERING,2209,515196.000,02000020,bd3f,16809;
2209,515196000,FINESTEERING*bd19b2d0
```



The time data embedded in this log represents the time of the most recent 1PPS signal. Refer to *Figure 2: 1PPS Alignment* on page 60 to see the alignment between a Fine and a Cold Clock receiver. Also refer to Transferring Time Between Receivers in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#).

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TIMESYNC header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	week	GPS reference week number	Ulong	4	H
3	ms	Number of milliseconds into the GPS reference week	Ulong	4	H+4
4	time status	GPS reference time Status, see <i>Table 13: GPS Reference Time Status</i> on page 53	Enum	4	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 3.210 TRACKSTAT

### *Tracking status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

The TRACKSTAT log contains an entry for each channel. If there are multiple signal channels for one satellite (for example L1, L2 P(Y), L2C, and L5 for GPS), then there will be multiple entries for that satellite.

As shown in *Table 156: Channel Tracking Status* on page 806 these entries can be differentiated by bit 20, which is set if there are multiple observables for a given satellite, and bits 21-25, which denote the signal type for the observation.

A zero in the PRN/slot of the TRACKSTAT log indicates the channel should be considered idle with the exception of those for GLONASS. A GLONASS channel should only be considered idle if the tracking state is 0 in the channel tracking status word.



For dual antenna receivers, a TRACKSTAT\_1 log can be requested to get TRACKSTAT data from the second antenna. As described in *Table 4: Binary Message Header Structure* on page 38, the message type indicates the log is from the second antenna. To request an ASCII log enter TRACKSTATA\_1 and for a binary log enter TRACKSTATB\_1.

**Message ID: 83**

**Log Type: Synch**

**Recommended Input:**

```
log trackstata ontime 1
```

**Abbreviated ASCII Example:**

```
<TRACKSTAT USB1 0 50.0 FINESTEERING 2209 515256.000 02000020 457c 16809
<   SOL_COMPUTED PPP 5.0 312
<     26 0 1810dc04 22959963.689 1368.049 47.443 1767.330 -0.428 GOOD 0.591
<     26 0 11305c0b 22959969.244 1066.013 45.559 1766.738 0.000 OBSL2 0.000
<     26 0 0230dc0b 22959969.108 1066.013 47.623 1767.281 0.000 OBSL2 0.000
<     26 0 01d0dc04 22959968.716 1021.637 48.652 1771.039 0.000 OBSL5 0.000
<     0 0 02008000 0.000 0.000 0.000 0.000 0.000 NA 0.000
<     2 0 0810dc24 23818626.401 -2643.823 44.784 1766.721 0.344 GOOD 0.836
<     2 0 01305c2b 23818628.678 -2060.130 44.380 1766.721 0.000 OBSL2 0.000
<     0 0 0220c020 0.000 0.000 0.000 0.000 0.000 NA 0.000
<     18 0 0810dc44 21881022.854 2511.170 49.936 1766.761 0.558 GOOD 0.917
<     18 0 01305c4b 21881025.040 1956.757 46.530 1766.761 0.000 OBSL2 0.000
<     18 0 0230dc4b 21881025.958 1956.757 49.643 1765.443 0.000 OBSL2 0.000
<     18 0 01d0dc44 21881030.540 1875.185 53.632 1771.842 0.000 OBSL5 0.000
<     18 0 42109c44 21881024.407 2511.098 49.898 1767.313 0.000 OBSL1 0.000
<     16 0 1810dc64 24867321.907 3173.445 44.028 334.935 0.030 GOOD 0.265
...
<     12 0 080680c1 0.000 -500.000 0.000 0.000 0.000 NA 0.000
<     8 0 080680e1 0.000 -4500.000 0.000 0.000 0.000 NA 0.000
<     11 0 08068101 0.000 -500.000 0.000 0.000 0.000 NA 0.000
<     6 0 0a670924 0.000 -391.971 40.146 1768.092 0.000 NA 0.000
<     1 0 0a670944 0.000 -49.230 44.610 1737.318 0.000 NA 0.000
<     3 0 0a670161 0.000 639.268 0.000 0.000 0.000 NA 0.000
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TRACKSTAT header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	sol status	Solution status (see <i>Table 86: Solution Status</i> on page 523)	Enum	4	H
3	pos type	Position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Enum	4	H+4
4	cutoff	GPS tracking elevation cut-off angle	Float	4	H+8
5	# chans	Number of hardware channels with information to follow	Ulong	4	H+12
6	PRN/slot	Satellite PRN number of range measurement Refer to <i>PRN Numbers</i> on page 52	Short	2	H+16
7	glofreq	(GLONASS Frequency + 7), see <i>GLONASS Slot and Frequency Numbers</i> on page 52	Short	2	H+18
8	ch-tr-status	Channel tracking status (see <i>Table 156: Channel Tracking Status</i> on page 806)	Ulong	4	H+20
9	psr	Pseudorange (m) - if this field is zero but the channel tracking status in the previous field indicates that the card is phase locked and code locked, the pseudorange has not been calculated yet	Double	8	H+24
10	Doppler	Doppler frequency (Hz)	Float	4	H+32
11	C/No	Carrier to noise density ratio (dB-Hz)	Float	4	H+36
12	locktime	Number of seconds of continuous tracking (no cycle slips)	Float	4	H+40
13	psr res	Pseudorange residual from pseudorange position filter (m)	Float	4	H+44
14	reject	Range reject code from pseudorange position filter (see <i>Table 92: Observation Statuses</i> on page 529)	Enum	4	H+48
15	psr weight	Pseudorange filter weighting	Float	4	H+52
16...	Next PRN offset = H+16+(#chans x 40)				
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16 (#chans x 40)
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.211 TRANSFERPORTSTATUS

*Display the state of the USB transfer port*

Platform: PwrPak7



PwrPak7M variants do not support this log.

This log displays the current state of the USB transfer port.

**Message ID: 2114**

**Log Type: Asynch**

**Recommended Input:**

```
log transferportstatusa onchanged
```

**ASCII Example:**

```
#TRANSFERPORTSTATUSA,USB1,0,50.0,FINESTEERING,2209,515351.867,02000020,4b3f,16809;USBSTICK,HOST*c9620422
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TRANSFERPORTSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	USB Detection Type	Type of connection detected See <i>Table 214: USB Detection Type</i> below	Enum	4	H
3	USB Mode	Current USB operation mode See <i>Table 215: USB Mode</i> on the next page	Enum	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 214: USB Detection Type**

Binary	ASCII	Description
0	NONE	Nothing is detected
1	USBSTICK	A flash drive is detected
2	PC	A computer is detected
3	ERROR	This is an error state

**Table 215: USB Mode**

<b>Binary</b>	<b>ASCII</b>	<b>Description</b>
0	DEVICE	The USB port is in device mode
1	HOST	The USB port is in host mode
2	OTG	The USB port is in OTG mode
3	INVALID	The USB port is in an invalid mode
4	NONE	The USB port is not in an operation mode
5	TRANSITION	The USB port operation mode is transitioning

## 3.212 UPTIME

### *Report the running time of the receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the number of seconds the receiver's firmware has been running, after the application of power or after the completion of a reset.

**Message ID:** 1777

**Log Type:** Polled

**Recommended Input:**

```
log uptimea once
```

**ASCII Example:**

```
#UPTIMEA,USB1,0,49.0,FINESTEERING,2209,515393.000,02000020,27d2,16809;3896*171a
bba2
```



In this example, 151639 seconds since power-on = 42.1 hours.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	UPTIME header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Uptime	The number of seconds the receiver has been running after a power up or reset.	Ulong	4	H
3	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.213 USERANTENNA

### *Display user defined antennas*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log lists the stored used defined antennas and their parameters.

**Message ID: 2282**

**Recommended input:**

```
log userantennaa onchanged
```

**ASCII Example:**

```
#USERANTENNA,USB1,0,67.5,FINESTEERING,2211,235864.286,02000020,7fac,16809;
USER_ANTENNA_1,"NOVCUSTOM",2,GPSL1,0.09,0.00,51.74,0.00,-0.03,-0.11,-0.20,0.23,
-0.17,-0.04,0.14,0.26,0.25,0.07,-0.24,-0.54,-0.67,-0.49,-0.02,0.55,0.84,0.47,
GPSL2,-1.54,1.66,52.00,0.00,0.00,0.00,-0.03,-0.13,-0.28,-0.48,-0.61,-0.56,
-0.28,0.23,0.84,1.29,1.31,0.77,-0.16,-0.95,-0.81,0.97*7fbbf0bf
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	USERANTENNA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Antenna type	User defined antenna type See <i>Table 216: User-Defined Antenna Type</i> on the next page	Enum	4	H
3	Antenna name	Name of the user defined antenna	Char[16]	16	H+4
4	Number of frequencies	Number of frequencies for which corrections are stored	Ulong	8	H+20
5	Frequency	The frequency for which the phase center corrections are valid See <i>Table 22: Frequency Type</i> on page 91	Enum	4	H+28
6	NorthOffset	North phase center offset in millimetres	Float	4	H+32
7	EastOffset	East phase center offset in millimetres	Float	4	H+36
8	UpOffset	Up phase center offset in millimetres	Float	4	H+40
9	PCVArray	19 element array of Phase Center Variations, in millimetres, in 5-degree elevation increments from 90 to 0	Float[19]	76	H+44

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
10	Next Frequency = H + 28 + (Number of frequencies x 92)				
11	xxxx	32-bit CRC (ASCII and binary only)	Hex	4	Variable
12	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

**Table 216: User-Defined Antenna Type**

Binary	ASCII
1001	USER_ANTENNA_1
1002	USER_ANTENNA_2
1003	USER_ANTENNA_3
1004	USER_ANTENNA_4
1005	USER_ANTENNA_5

## 3.214 USERI2CRESPONSE

### *Status of USERI2CREAD or USERI2CWRITE Command*

**Platform:** OEM7600, OEM7700, OEM7720

This log reports the status of a previously executed **USERI2CREAD** or **USERI2CWRITE** command. There is one log emitted for each command that is executed.

For the **USERI2CREAD** command (see page 462), this log outputs the data read from the device on the I2C bus and the status of the read operation.

For the **USERI2CWRITE** command (see page 464), the status of the write operation is reported and the data field will always be 0.

**Message ID: 2234**

#### Recommended Input:

```
log USERI2CRESPONSE onnew
```

#### Abbreviated ASCII Example 1:

```
USERI2CREAD 70 4 aabbccdd 12 6789
```

```
<USERI2CRESPONSE COM1 0 84.0 FINESTEERING 1994 257885.895 02000000 e3f6 32768
< 70 aabbccdd OK READ 6789 12 000102030405060708090a0b
```

#### Abbreviated ASCII Example 2:

```
USERI2CWRITE 70 3 aabbcc 8 0001020304050607 12345
```

```
<USERI2CRESPONSE COM1 0 84.0 FINESTEERING 1994 257885.895 02000000 e3f6 32768
< 70 aabbcc OK WRITE 12345 0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	USERI2CRESPONSE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	DeviceAddress	The 7 bit address of the I2C device. Valid values are 0 through 127. For ASCII and Abbreviated commands, this field is a hexadecimal string of two digits. There is no 0x prefix and spaces are not allowed in the string.	Uchar	1 <sup>1</sup>	H
3	RegisterAddress	The actual register address used for the operation. This is a ULONG value in hexadecimal format (without 0x prefix).	Ulong	4	H+4

<sup>1</sup>In the binary case, additional bytes of padding are added after this field to maintain 4-byte alignment for the fields that follow.



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	ErrorCode	Error code for the operation. See <i>Table 217: Error Code</i> below.	Enum	4	H+8
5	OperationMode	Operation mode code. See <i>Table 218: Operation Mode Code</i> on the next page.	Enum	4	H+12
6	TransactionID	This is the copy of Transaction ID provided to the command.	Ulong	4	H+16
7	ReadDataLength	For a Read operation, this is the actual number of bytes read from the I2C device. For a Write operation, this value is always zero.	Ulong	4	H+20
8	ReadData	For a Read operation, this is the data read from the device. For ASCII logs this field is displayed as a string of hexadecimal digits, with two digits per byte. The first byte retrieved from the I2C device is the first byte displayed and so on.  The maximum size of this field is 256 bytes.  When ReadDataLength is zero, this field will be empty.	HEXBYTE ARRAY	Y	H+24

Table 217: Error Code

Binary	ASCII	Description
0	OK	I2C transaction is successful
1	IN_PROGRESS	I2C transaction is currently in progress
2	DATA_TRUNCATION	I2C transaction read data was truncated
3	BUS_BUSY	I2C bus is busy
4	NO_DEVICE_REPLY	No device replied to the I2C transaction request
5	BUS_ERROR	I2C bus error or bus arbitration lost
6	TIMEOUT	I2C transaction has timed out
7	UNKNOWN_FAILURE	I2C transaction has an unexplained failure

**Table 218: Operation Mode Code**

<b>Binary</b>	<b>ASCII</b>	<b>Description</b>
0	NONE	No Operation
1	READ	Read Operation
2	WRITE	Write Operation
3	SHUTDOWN	Shut down Operation

## 3.215 VALIDMODELS

### Valid model information

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log gives a list of valid authorized models available and expiry date information.

If a model has no expiry date, it reports the year, month and day fields as 0s.

**Message ID:** 206

**Log Type:** Asynch

**Recommended Input:**

```
log validmodelsa once
```

**ASCII Example:**

```
#VALIDMODELSA,USB1,0,49.0,FINESTEERING,2209,515587.940,02000020,342f,16809;
1,"FFNRNNCBES1",0,0,0*0d1a3309
```



Use the VALIDMODELS log to output a list of available models for the receiver. Use the **AUTH** command (see page 84), to add a model and the **MODEL** command (see page 266) to change the currently active model. See the **VERSION** log on page 984 for the currently active model

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	VALIDMODELS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	#mod	Number of models with information to follow	Ulong	4	H
3	model	Model name	String [Max16]	Variable <sup>1</sup>	H+4
4	expyear	Expiry year	Ulong	4	Variable Max: H+20
5	expmonth	Expiry month	Ulong	4	Variable Max: H+24
6	expday	Expiry day	Ulong	4	Variable: Max: H+28

<sup>1</sup>In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
7...	Next model offset = H+4+(#mod x variable [max:28])				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#mod x variable [max:28])
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 3.216 VERIPOSINFO

### *Veripos subscription information*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log contains details on the Veripos subscription.

**Message ID:** 1728

**Log Type:** Asynch

**Recommended Input:**

```
log veriposinfoa onchanged
```

**ASCII Example:**

```
#VERIPOSINFOA,USB1,0,49.5,FINESTEERING,2209,511557.037,02000020,2bd7,16809;557954,BUBBLE,80000733,"6"*971fc40a
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	VERIPOSINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Serial number	Receiver serial number	Ulong	4	H
3	Mode	Operating mode (see <i>Table 219: Veripos Operating Mode</i> below)	Enum	4	H+4
4	Details	Subscription details (refer to <i>Table 220: Veripos Subscription Details Mask</i> on the next page)	Hex	4	H+8
5	Service code	Veripos service code	Char[4]	4	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 219: Veripos Operating Mode**

ASCII	Binary	Description
UNASSIGNED	0	Decoder has not had an assigned operating mode
NCC_CONTROLLED	7	Decoder operation disabled by a command from the Network Control Center (NCC)
NO_DISABLE	8	Decoder operation not disabled

ASCII	Binary	Description
BUBBLE	100	Decoder is operating in a Veripos permitted subscription-free bubble
MODEL_DENIED	101	Decoder operation is not permitted on the current firmware model

Table 220: Veripos Subscription Details Mask

Bit	Mask	Description
0	0x00000001	Subscription permits differential positioning (Standard Service)
1-3	0x0000000E	Reserved
4	0x00000010	Subscription permits Ultra PPP positioning
5	0x00000020	Subscription permits Ultra2 PPP positioning
6-7	0x000000C0	Reserved
8	0x00000100	Subscription permits Apex PPP positioning
9	0x00000200	Subscription permits Apex2 PPP positioning
10	0x00000400	Subscription permits Apex5 PPP positioning
11	0x00000800	Subscription permits Apex-AR PPP positioning
12-31	0xFFFFF000	Reserved

## 3.217 VERIPOSSTATUS

### *Veripos decoder and subscription status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log contains status information for the Veripos decoder and subscription.

**Message ID:** 1730

**Log Type:** Asynch

**Recommended Input:**

```
log veriposstatusa onchanged
```

**ASCII Example:**

```
#VERIPOSSTATUSA,USB1,0,49.0,FINESTEERING,2209,515667.036,02000020,0719,16809;
ENABLE,LOCKED*2c005d7e
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	VERIPOSSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	Access	Access status. ENABLE (1) if the subscription is valid; DISABLE (0) otherwise	Enum	4	H
3	Sync state	Decoder data synchronization state (see <i>Table 221: Decoder Data Synchronization State</i> below)	Enum	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 221: Decoder Data Synchronization State**

ASCII	Binary	Description
NO_SIGNAL	0	None of the decoders have received data in the last 30 seconds
SEARCH	1	At least one decoder is receiving data and is searching for the format
LOCKED	2	At lease one decoder has locked onto the format

## 3.218 VERSION

### Version information

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log contains the version information for all components of a system. A component may be hardware (for example, a receiver or data collector) or firmware in the form of applications or data (for example, data blocks for height models or user applications). See *Table 223: Firmware and Boot Version Field Formats* on page 987 for details on the format of key fields.

See also the **VALIDMODELS** log on page 979.

**Message ID:** 37

**Log Type:** Polled

**Recommended Input:**

```
log versiona once
```

**Abbreviated ASCII Example:**

```
<VERSION USB1 0 49.0 FINESTEERING 2209 515707.883 02000020 3681 16809
< 11
< GPSCARD "FFNRNNCBES1" "BMHR17090005E" "OEM7700-1.00" "OM7CR0810RN0000"
"OM7BR0001RBG000" "2021/Dec/06" "11:16:48"
< OEM7FPGA "" "" "" "OMV070001RN0000" "" "" ""
< APPLICATION "" "" "" "EP7AR0810RN0000" "" "2021/Dec/06" "11:16:59"
< DEFAULT_CONFIG "" "" "" "EP7CR0810RN0000" "" "2021/Dec/06" "11:17:11"
< PACKAGE "" "" "" "EP7PR0810RN0000" "" "2021/Dec/06" "11:17:05"
< DB_WWWISO "WWWISO" "0" "" "1.8.0" "" "2022/Apr/11" "15:44:16"
< ENCLOSURE "PWRPAK7-E1" "NMNE17200009B" "" "" "" "" ""
< WIFI "RS9113" "" "" "1.7.12" "" "2021/Dec/06" "11:17:22"
< REGULATORY "US" "" "" "" "" "" ""
< IMUCARD "Epson G320N 125" "00000354" "G320PDGN" "2510" "" "" ""
< WHEELSENSOR "" "" "" "SWS000201RN0000" "" "" ""
```



The VERSION log is a useful log as a first communication with your receiver. Once connected, using NovAtel Connect or a terminal emulator program, log VERSION and check that the output makes sense. Also, ensure that you have the receiver components you expected.

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	Log header	VERSION header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	# comp	Number of components (cards, and so on)	Long	4	H



Field	Field type	Description	Format	Binary Bytes	Binary Offset
3	type	Component type (see <i>Table 222: Component Types</i> on the next page)	Enum	4	H+4
4	model	OEM7 firmware model number e.g., CFNPNNTVN indicates the receiver's current model functionality  When the component type is ENCLOSURE, the model specifies the enclosure type.	Char[16]	16	H+8
5	psn	Product serial number	Char[16]	16	H+24
6	hw version	Hardware version in the format: P-R  Where P = hardware platform R = hardware revision  Example: OEM7700-1.00	Char[16]	16	H+40
7	sw version	Firmware version, see <i>Table 223: Firmware and Boot Version Field Formats</i> on page 987	Char[16]	16	H+56
8	boot version	Boot code version, see <i>Table 223: Firmware and Boot Version Field Formats</i> on page 987	Char[16]	16	H+72
9	comp date	Firmware compile date in the format: YYYY/Mmm/DD  Where YYYY = year Mmm = month DD = day (1-31)  Example: 2018/Jul/10	Char[12]	12	H+88
10	comp time	Firmware compile time in the format: HH:MM:SS  Where: HH = hours MM = minutes SS = seconds  Example: 14:37:01	Char[12]	12	H+100
11...	Next component offset = H + 4 + (#comp x 108)				

Field	Field type	Description	Format	Binary Bytes	Binary Offset
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#comp x 108)
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 222: Component Types

Binary	ASCII	Description
0	UNKNOWN	Unknown component
1	GPSCARD	OEM7 family receiver In an enclosure product this is the receiver card in the enclosure.
2	CONTROLLER	Reserved
3	ENCLOSURE	OEM card enclosure
4-6	Reserved	
7	IMUCARD	IMU integrated in the enclosure
8	USERINFO	Application specific information
12-14	Reserved	
15	WIFI	Wi-Fi radio firmware
16-17	Reserved	
18	RADIO	UHF radio component
19	WWW_CONTENT	Web Server content
20	Regulatory	Regulatory configuration
21	OEM7FPGA	OEM7 FPGA version
22	APPLICATION	Embedded application
23	Package	Package
24	Reserved	
25	DEFAULT_CONFIG	Default configuration data
26	WHEELSENSOR	Wheel sensor in the enclosure
27	EMBEDDED_AUTH	Embedded Auth Code data
981073920 (0x3A7A0000)	DB_HEIGHTMODEL	Height/track model data
981073928 (0x3A7A0008)	DB_WWWISO	Web UI ISO Image
981073930 (0x3A7A000A)	DB_LUA_SCRIPTS	Lua Script ISO Image

Table 223: Firmware and Boot Version Field Formats

Field Format (ASCII)	Description	Example	
NWXYZFFMMRN0000	N = Family Name The Family Name can be: O = OEM	Software Version: OM7CR0500RN0000  Boot Version: OM7BR0001RB0000	In both examples, the Family Name is <b>O</b> .
	WX = Product The Product can be: M7: OEM7 product A7: Agriculture optimized OEM7 product		In both examples, the Product is <b>M7</b> .
	Y = Image Type The Image Type can be: B: boot code M: main firmware application C: combined main firmware application and user application		In the Software Version example, the Image Type is <b>C</b> .  In the Boot Version example, the Image Type is <b>B</b> .
	Z = Signature The Signature can be: R: Officially signed H: High Speed signed		In both examples the Signature is <b>R</b> .
	FF = Feature Release Number		In the Software Version example, the Feature Release Number is <b>05</b> .  In the Boot Version example, the Feature Release Number is <b>00</b> .
	MM = Maintenance Release Number		In the Software Version example, the Maintenance Release Number is <b>00</b> .  In the Boot Version example, the Maintenance Release Number is <b>01</b> .

Field Format (ASCII)	Description	Example	
	<p>R = Release Type</p> <p>The Release Type can be:</p> <p>A: Alpha</p> <p>B: Beta</p> <p>R: Release</p> <p>S: Special</p> <p>E: Engineering Special</p> <p>C: Customer Approved Special</p>		In both examples, the Release Type is <b>R</b> .
	<p>N = Distribution Permit</p> <p>The Distribution Permit can be:</p> <p>N: No restrictions</p> <p>B: Boot Code</p> <p>H: High Speed Build (Requires a permit to use)</p>		<p>In the Software Version example, the Distribution Permit is <b>N</b>.</p> <p>In the Boot Version example, the Distribution Permit is <b>B</b>.</p>
	0000 = Minor Release Indicator		In both examples, the Minor Release Indicator is <b>0000</b> .



The firmware version of a receiver is also presented in a short version (e.g. 7.05.04). The short version is the product number, feature release number and maintenance release number separated by dots. For example, the firmware version OM7MR0504RN0000 has a short version of 7.05.04.

## 3.219 WIFIAPSETTINGS

### *Display the Wi-Fi access point configuration*

Platform: PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this log to display the Wi-Fi access point configuration. If the access point is not currently operational, the log reports the access point configuration to be applied the next time the **WIFIMODE AP** or **WIFIMODE CONCURRENT** command is received.



The term passkey and password are the same.

**Message ID: 2093**

**Log Type: Polled**

**Recommended Input:**

```
LOG WIFIAPSETTINGS
```

**ASCII Example:**

```
#WIFIAPSETTINGS SA, USB1, 0, 49.5, FINESTEERING, 2209, 515774.000, 02000020, fc0e, 16809;
"PwrPak7-NMNE17200009B", "P3YSXG8D", 2P4GHZ, WPA2, CCMP, US, 11, "00:23:a7:b6:de:9c"
*aa09a912
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	WIFIAPSETTINGS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	SSID	SSID of the AP	String [Max 33]	Variable	H
3	passkey	Passkey of the AP	String [Max 65]	Variable	Variable
4	band	Wi-Fi operating band. See <i>Table 224: Wi-Fi Band</i> on the next page.	Enum	4	Variable
5	security protocol	Wireless security protocol. See <i>Table 225: Wi-Fi Security Protocol</i> on the next page.	Enum	4	Variable
6	encryption	Wireless encryption type. See <i>Table 226: Wi-Fi Encryption Type</i> on the next page.	Enum	4	Variable

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	region	Regulatory region. See <i>Table 227: Regulatory Region</i> on the next page.	Enum	4	Variable
8	channel	Wireless channel used by access point to communicate with connected clients.	Int	4	Variable
9	BSSID	BSSID of the AP (MAC of the Wi-Fi interface)	String [Max 18]	Variable	Variable
10	xxxx	32-bit CRC (ASCII or Binary only)	Hex	4	Variable
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 224: Wi-Fi Band

Binary	ASCII	Description
1	2P4GHZ	2.4 GHz

Table 225: Wi-Fi Security Protocol

Binary	ASCII	Description
1	OPEN	Open network (No security)
2	WPA	Wi-Fi Protected Access
3	WPA2	Wi-Fi Protected Access version 2



NovAtel Wi-Fi access points only support the WPA2 security protocol. As a result, the **WIFIAPSETTINGS** log will only report **WPA2**.

NovAtel Wi-Fi Clients support OPEN, WPA, and WPA2 security protocols.

Table 226: Wi-Fi Encryption Type

Binary	ASCII	Description
1	OPEN	Open (no encryption)
2	TKIP	Temporal Key Integrity Protocol (used with WPA)
3	CCMP	AES-based CCMP (Cipher Chaining Message Authentication) used with WPA2



NovAtel Wi-Fi access points only support the WPA2 security protocol. As a result, the **WIFIAPSETTINGS** log will only report **CCMP**.

**Table 227: Regulatory Region**

Binary	ASCII	Description
0	None	Receiver has not been configured to comply with any regional regulatory requirements. Wireless components (e.g. Wi-Fi) will not operate. Contact NovAtel Customer Support.
1	US	United States
2	EU	Europe
3	AU	Australia
4	JP	Japan
5	NZ	New Zealand
6	BR	Brazil

## 3.220 WIFINETLIST

*Display a list of Wi-Fi networks in range of client*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this log to display information about Wi-Fi access points detected by the receiver when configured in either Client or Concurrent mode.

**Message ID:** 2210

**Log Type:** Asynch

**Recommended input:**

```
log wifinetlist onchanged
```

**ASCII Example:**

```
<WIFINETLIST USB1 0 65.5 FINESTEERING 2210 340950.380 02000020 0c90 16809
< 7
< "PwrPak7-NMNE17200016X" -37 11 "00:23:a7:b6:df:7c" WPA2
< "Hexagon" -47 6 "38:17:c3:81:7b:e5" OPEN
< "Geosystems" -47 6 "38:17:c3:81:7b:e6" WPA2
< "PwrPak7-NMPM18040011A" -47 11 "88:da:1a:04:cc:d4" WPA2
< "SM7-NMPX18500007B" -47 11 "88:da:1a:43:73:20" WPA2
< "PwrPak7-NMND21220024J" -51 11 "80:c9:55:0b:da:60" WPA2
< "PwrPak7-NMND18280030D" -54 11 "88:da:1a:07:97:70" WPA2
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	WIFINETLIST header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	num_networks	Number of network records to follow	Ulong	4	H
3	SSID	SSID of scanned access point.	String [Max 33]	variable	H+4
4	RSSI	Absolute value of RSSI (dBm)	Int	4	variable
5	channel	RF channel to use	Ulong	4	variable
6	BSSID	MAC address of the access point.	String [Max 18]	variable	variable
7	security	Security mode of access point. <i>Table 228: Wi-Fi Security Type</i> on the next page	Enum	4	variable



**Table 228: Wi-Fi Security Type**

Binary	ASCII	Description
1	OPEN	Open network (no security)
2	WPA	Wi-Fi protected access
3	WPA2	Wi-Fi protected access II

## 3.221 WIFISTATUS

### *Display status information of receiver Wi-Fi*

**Platform:** PwrPak7, SMART7-I, SMART7-SI, SMART7-W



PwrPak7M and PwrPak7Q variants do not support this command.

Use this log to display the current status of a Wi-Fi Client and or Access Point in addition to information about any connected Access Point and or Clients.

**Message ID:** 2207

**Log Type:** Asynch

**Recommended input:**

```
log wifistatus onchanged
```

**ASCII Example:**

```
#WIFISTATUSA,USB1,0,50.0,UNKNOWN,0,8.029,02000020,8a58,16809;  
ACCESSPOINT_OPERATIONAL,"",0,0,"",OPEN,0*dfc10e04
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	WIFISTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	status	Wi-Fi operational status. Refer to <i>Table 229: Wi-Fi Status</i> on the next page.	String [Max 33]	Variable	H
3	AP SSID	SSID of connected access point. This field is blank if the client is not connected to an access point.	String [Max 33]	Variable	
4	AP RSSI	Value of RSSI (dBm) of connection to access point. This field is zero if the client is not connected to an access point.	Ulong	4	
5	AP channel	Channel use to connect to access point. This field is zero if the client is not connected to an access point.	Ulong	4	
6	AP BSSID	MAC address of connected access point. This field is blank if the client is not connected to an access point.	Ulong	4	

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	AP security	Security mode of connected access point. See <i>Table 228: Wi-Fi Security Type</i> on page 993 in the <b>WIFINETLIST</b> command (see page 992). The value of this field is only valid if the AP SSID field is not blank.	Enum	4	
8	#clients	Number of clients connected to an access point to follow.	Ulong	4	
9	BSSID	MAC address of connected client.	String [Max 18]	Variable	
10...	Next Client				

Table 229: Wi-Fi Status

Binary	ASCII	Description
0	STARTUP	
1	OFF	Wi-Fi hardware is powered off
2	ON	Wi-Fi hardware is powered on but not configured
3	CONFIGURING_ACCESSPOINT	Wi-Fi hardware is being configured as an access point
4	ACCESSPOINT_OPERATIONAL	Wi-Fi hardware is operating as an access point and is ready to accept client connections
5	CONFIGURING_CLIENT	Wi-Fi hardware is being configured as a client station
6	CLIENT_OPERATIONAL	Wi-Fi hardware is operating as a client but is not connected to an access point
7	BOOTING_UP	Wi-Fi hardware is booting up
8	BOOTUP_COMPLETE	Wi-Fi hardware has booted up
9	UPGRADE_REQUIRED	Wi-Fi firmware version does not match required version
10	UPGRADING_FIRMWARE	Wi-Fi firmware is being upgraded
11	UPGRADING_FIRMWARE_10	Wi-Fi firmware update is 10% complete
12	UPGRADING_FIRMWARE_20	Wi-Fi firmware update is 20% complete
13	UPGRADING_FIRMWARE_30	Wi-Fi firmware update is 30% complete

Binary	ASCII	Description
14	UPGRADING_ FIRMWARE_40	Wi-Fi firmware update is 40% complete
15	UPGRADING_ FIRMWARE_50	Wi-Fi firmware update is 50% complete
16	UPGRADING_ FIRMWARE_60	Wi-Fi firmware update is 60% complete
17	UPGRADING_ FIRMWARE_70	Wi-Fi firmware update is 70% complete
18	UPGRADING_ FIRMWARE_80	Wi-Fi firmware update is 80% complete
19	UPGRADING_ FIRMWARE_90	Wi-Fi firmware update is 90% complete
20	UPGRADING_ FIRMWARE_COMPLETE	Wi-Fi firmware update is 100% complete
21	ERROR	An error has occurred
22	CONFIGURING_ CONCURRENT	Wi-Fi hardware is being configured to operate as both an access point and client concurrently
23	CONCURRENT_ OPERATIONAL	Wi-Fi hardware is operating as both an access point and client. The client is not connected to an access point.
24	CONNECTING_TO_AP	Wi-Fi hardware is attempting to connect to an access point
25	CONNECTED_TO_AP	Wi-Fi hardware has successfully connected to an access point
26	CONNECTION_FAILURE	Wi-Fi client failed to connect to an access point. Possible causes include invalid password provided in the <b>WIFINETCONFIG</b> command (see page 478)
27	CONFIGURING_ NETWORK_ PARAMETERS	Wi-Fi client is configuring its network parameters based on active connection to an access point
28	CONNECTION_ REFUSED	Wi-Fi client failed on initial attempt to connect to an access point due to the connection being actively refused by the access point  Possible causes include invalid configuration settings in the <b>WIFINETCONFIG</b> command on page 478
29	PREFERRED_ NETWORK_ MISCONFIGURED	Wi-Fi client cannot connect to preferred access point due to a misconfiguration of network parameters in the <b>WIFINETCONFIG</b> command on page 478
30	CONCURRENT_ CONNECTING_TO_AP	Wi-Fi concurrent client is configuring its network parameters based on active connection to an access point

Binary	ASCII	Description
31	CONCURRENT_ CONFIGURING_ NETWORK_PARAMS	Wi-Fi concurrent client is configuring its network parameters based on active connection to an access point
32	CONCURRENT_ CONNECTED_TO_AP	Wi-Fi concurrent client has successfully connected to an access point
33	DISCONNECTING_ FROM_AP	Wi-Fi client is disconnecting from an access point
34	CONCURRENT_ DISCONNECTING_ FROM_AP	Wi-Fi concurrent is disconnecting from an access point
35	BOOTUP_ CONNECTING_TO_ WIFI_MODULE	
36	BOOTUP_ERROR	

## Chapter 4 SPAN Commands

The commands used to configure GNSS+INS functions are described in the following sections. For information about other available commands, refer to *Commands* on page 58.

## 4.1 ALIGNMENTMODE

### *Set the Alignment Mode*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to set the alignment method used to initialize the SPAN system.

The default ALIGNMENTMODE is AUTOMATIC. In this mode, the first available method to align is used.

Sending the ALIGNMENTMODE command manually overrides the AUTOMATIC setting and changes the options available to complete an alignment.

**Message ID: 1214**

#### **Abbreviated ASCII Syntax:**

```
ALIGNMENTMODE mode
```

#### **Abbreviated ASCII Example:**

```
ALIGNMENTMODE AIDED_TRANSFER
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	ALIGNMENTMODE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	mode	UNAIDED	0	Static coarse alignment or kinematic alignment methods are available.	Enum	4	H
		AIDED_TRANSFER	2	Seed the initial azimuth estimate from the ALIGN solution.			
		AUTOMATIC	3	Seed the full attitude from the ALIGN solution, perform a regular static coarse alignment or perform a kinematic alignment, whichever is possible first.			
		STATIC	4	Static coarse alignment method only.			
		KINEMATIC	5	Kinematic alignment method only.			



If the **ALIGNMENTMODE** selected can use a kinematic alignment (UNAIDED, KINEMATIC or AUTOMATIC), the **SETINSROTATION RBV** command must be sent to the receiver regardless of system configuration and IMU orientation.

It is *strongly recommended* to use the **SETINSROTATION RBV** command to define the orientation of the system for all configurations.



NVM Seed injected (see the **INSSEED** command on page 1021) and commanded (see **SETINITAZIMUTH** command on page 1038) alignments are valid for all alignment modes and will supersede all other options if valid and available.



## 4.2 CONNECTIMU

### *Connects an IMU to a Port*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7

Use this command to specify the type of IMU connected to the receiver and the receiver port used by the IMU.

**Message ID:** 1428

#### Abbreviated ASCII Syntax:

```
CONNECTIMU IMUPort IMUType
```

#### Abbreviated ASCII Example:

```
CONNECTIMU COM2 LN200
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	CONNECTIMU header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	IMUPort	COM1	1	IMU Port is COM port 1	Enum	4	H
		COM2	2	IMU Port is COM port 2			
		COM3	3	IMU Port is COM port 3			
		SPI	7	IMU Port is the SPI port			
		COM4	19	IMU Port is COM port 4			
		COM5	31	IMU Port is COM port 5			
3	IMUType	See <i>Table 230: IMU Type</i> on the next page		IMU Type	Enum	4	H+4



SPI is available only on the OEM7500, OEM7600, OEM7700, and OEM7720.



COM4 and COM5 are available only on the OEM7600, OEM7700 and OEM7720.



COM2 is available only on the OEM719 with hardware revisions 1.04 and earlier. For the OEM719, it is recommended to use COM1 to connect to an IMU .



The IMU-ISA-100C, IMU-FSAS, IMU-HG1900, IMU-LN200, IMU- $\mu$ IMU, IMU-CPT, IMU-KVH-P-1750 and IMU-KVH1750 use RS-422 protocol and must be connected to a receiver port that is configured to use RS-422. Refer to the [OEM7 Installation and Operation User Manual](#) or [PwrPak7 Installation and Operation User Manual](#) for information about which receiver ports support RS-422 and instructions for enabling RS-422.

Table 230: IMU Type

Binary	ASCII	Description
0	UNKNOWN	Unknown IMU type (default)
1	HG1700_AG11	Honeywell HG1700 AG11
4	HG1700_AG17	Honeywell HG1700 AG17
5	HG1900_CA29	Honeywell HG1900 CA29
8	LN200	Northrop Grumman LN200/LN200C
11	HG1700_AG58	Honeywell HG1700 AG58
12	HG1700_AG62	Honeywell HG1700 AG62
13	IMAR_FSAS	iMAR iIMU-FSAS
16	KVH_COTS	KVH CPT IMU
20	HG1930_AA99	Honeywell HG1930 AA99
26	ISA100C	Northrop Grumman Litef ISA-100C
27	HG1900_CA50	Honeywell HG1900 CA50
28	HG1930_CA50	Honeywell HG1930 CA50
31	ADIS16488	Analog Devices ADIS16488
32	STIM300	Sensoror STIM300
33	KVH_1750	KVH 1750 IMU KVH P-1750 IMU
41	EPSON_G320	Epson G320N
52	LITEF_MICROIMU	Northrop Grumman Litef $\mu$ IMU-IC
56	STIM300D	Sensoror STIM300, Direct Connection
58	HG4930_AN01	Honeywell HG4930 AN01
61	EPSON_G370	Epson G370N
62	EPSON_G320_200HZ	Epson G320N – 200 Hz
68	HG4930_AN04	Honeywell HG4930 AN04 – 100 Hz
69	HG4930_AN04_400HZ	Honeywell HG4930 AN04 – 400 Hz



The IMU Type field also supports the legacy ASCII values that contain the "IMU\_" prefix. For example, *LN200* or *IMU\_LN200*. IMUs recently added as SPAN supported devices, such as the LITEF\_MICROIMU and STIM300D, do not support the "IMU\_" prefix.



Values not shown in this table are reserved.

## 4.3 DMICONFIG

### Configure a DMI Input

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to configure a Distance Measurement Instrument (DMI) input.

For more information about using a DMI, see Distance Measurement Instrument in the [OEM7 SPAN Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#).

**Message ID: 2270**

#### Abbreviated ASCII Syntax:

```
DMICONFIG dmi_ID switch [source]
```

#### Abbreviated ASCII Examples:

```
DMICONFIG DMI1 ENABLE IMU
```

```
DMICONFIG DMI1 DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	–	–	DMICONFIG header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	–	H	0
2	DMI_ID	DMI1	0	ID of the DMI sensor	Enum	4	H
3	Switch	DISABLE	0	Disable the DMI sensor	Enum	4	H+4
		ENABLE	1	Enable the DMI sensor for use			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	Source	EXT_COUNT	0	Select how the value of the RAWDMI input will be interpreted.	Enum	4	H+8
		EXT_VELOCITY	1				
		IMU	2	Use <b>ENCLOSURE</b> for the PwrPak7 or CPT7.			
		ENCLOSURE	3	Default = EXT_COUNT			
5	Reserved	Default = OFF	Default = 1	Reserved Field (Optional) If entering, use default values.	Enum	4	H+12

## 4.4 EXTERNALPVAS

### *Enter PVA Update*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



This command should only be used by advanced users of GNSS+INS.



The standard deviations entered using this command must be representative of actual input error.



Relative updates should only be used when they are input at consistent rates.



The **EXTERNALPVAS** command uses a short header if the command is entered in ASCII or Binary.

This command allows the user to provide their own update for INS in full position, velocity, attitude and other updates, and in any combination. The user can also provide height or attitude only updates, along with Zero Velocity Updates (ZUPTs). The position and velocity updates can be entered in local level or ECEF.

The **EXTERNALPVAS** command is designed to provide a method for additional sensor information to be input into the SPAN filter, specifically during GNSS denied environments. This will provide a method to constrain the error growth that is typical in an Inertial Kalman Filter when GNSS observations are unavailable (environments such as: urban canyon, tunnels, jamming etc.). It is important to ensure that the external update and its corresponding standard deviations are accurate and input with minimal latency to provide optimal effectiveness. Entering an external update or its standard deviation inappropriately may have an adverse effect on the SPAN solution.



The default input frame is ECEF. Updates are entered in ECEF unless Local Level is specified using the OptionsMask parameter.

**Message ID: 1463**

#### **Abbreviated ASCII Syntax:**

```
EXTERNALPVAS Position1 Position2 Position3 Velocity1 Velocity2 Velocity3
Attitude1 Attitude2 Attitude3 PosStdDev1 PosStdDev2 PosStdDev3 VelStdDev1
VelStdDev2 VelStdDev3 AttStdDev1 AttStdDev2 AttStdDev3 UpdateMask
OptionsMask
```

#### **Abbreviated ASCII Example: (Local Level Velocity Update)**

```
EXTERNALPVAS 0.0 0.0 0.0 0.4502 10.54 -0.09598 0.0 0.0 0.0 0.0 0.0 0.0 0.01
0.01 0.01 0.0 0.0 0.0 4000 10
```

**Full ASCII Example:** (Local Level Velocity Update)

```
%EXTERNALPVASA,2051,493713.000;0.00000000000,0.00000000000,0.00000000000,
0.1333,17.1424,-0.3311,0.000000000,0.000000000,0.000000000,0.0000,0.0000,
0.0000,0.0119,0.0096,0.0132,0.0000,0.0000,0.0000,00004000,00000010*edced535
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	EXTERNALPVAS header This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Position1			Latitude (degrees), ECEF X-coordinate (m) or X relative coordinate (m)	Double Array [3]	24	H
	Position2			Longitude (degrees), ECEF Y-coordinate (m) or Y relative coordinate (m)			
	Position3			Height (ellipsoidal m), ECEF Z-coordinate (m) or Z relative coordinate (m)			
3	Velocity1			North velocity or velocity along the X-axis (m/s)	Float Array [3]	12	H+24
	Velocity2			East velocity or velocity along the Y-axis (m/s)			
	Velocity3			Up velocity or velocity along the Z-axis (m/s)			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	Attitude1			Pitch – Local Level to SPAN User Output Frame or relative delta from SPAN User Output Frame (degrees)	Float Array [3]	12	H+36
	Attitude2			Roll – Local Level to SPAN User Output Frame or relative delta from SPAN User Output Frame (degrees)			
	Attitude3			Azimuth – Local Level to SPAN User Output Frame or relative delta from SPAN User Output Frame (degrees)			
5	PosStdDev1			Position1 standard deviation (m)	Float Array [3]	12	H+48
	PosStdDev2			Position2 standard deviation (m)			
	PosStdDev3			Position3 standard deviation (m)			
6	VelStdDev1			Velocity1 standard deviation (m/s)	Float Array [3]	12	H+60
	VelStdDev2			Velocity2 standard deviation (m/s)			
	VelStdDev3			Velocity3 standard deviation (m/s)			
7	AttStdDev1			Attitude1 standard deviation (degrees)	Float Array [3]	12	H+72
	AttStdDev2			Attitude2 standard deviation (degrees)			
	AttStdDev3			Attitude3 standard deviation (degrees)			



Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
8	UpdateMask			<p>This mask selects which updates are applied. Setting a bit applies the update. More than one update can be applied at one time.</p> <p>See <i>Table 231: EXTERNALPVAS Updates Mask</i> below for the external update bits that can be used.</p>	HEX Ulong	4	H+84
9	OptionsMask			<p>This mask selects the update options for various updates, such as using an ECEF or Local Level system for the position updates.</p> <p>See <i>Table 232: EXTERNALPVAS Options Mask</i> on the next page for details.</p>	HEX Ulong	4	H+88

Table 231: EXTERNALPVAS Updates Mask

Bit	Mask	Description
0	0x00001	Reserved
1	0x00002	Reserved
2	0x00004	ZUPT Update. No fields are required in the <b>EXTERNALPVAS</b> command for this update.
3	0x00008	Reserved
4	0x00010	Reserved
5	0x00020	External Position Update. This update is entered using Position1 to Position3 in the <b>EXTERNALPVAS</b> command.
6	0x00040	Reserved
7	0x00080	Reserved
8	0x00100	Reserved
9	0x00200	Reserved
10	0x00400	Reserved

Bit	Mask	Description
11	0x00800	Reserved
12	0x01000	Reserved
13	0x02000	Reserved
14	0x04000	External Velocity Update. This update is entered using Velocity1 to Velocity3 in the <b>EXTERNALPVAS</b> command.
15	0x08000	External Attitude Update. This update is entered using Attitude1 to Attitude3 in the <b>EXTERNALPVAS</b> command.
16	0x10000	External Heading Update. This update is entered using Attitude3 in the <b>EXTERNALPVAS</b> command.
17	0x20000	External Height Update. This update is entered using Position3 in the <b>EXTERNALPVAS</b> command.



If both the External Position Update and External Height Update bits are set, only the External Position Update will be applied.

If both the External Attitude Update and External Heading Update bits are set, only the External Attitude Update will be applied.

**Table 232: EXTERNALPVAS Options Mask**

Bit	Mask	Description	Range Value
0	0x00000001	Reserved	
1	0x00000002	Reserved	
2–3	0x0000000C	Position Input Frame	00 – ECEF
			01 – LLH
			10 - Vehicle Frame (Relative update only)
4–5	0x00000030	Velocity Input Frame	00 – ECEF
			01 – LLH
			10 - Vehicle Frame (Absolute update only)
6	0x00000040	Position Update Type	0 – Absolute
			1 – Relative
7	0x00000080	Attitude Update Type	0 – Absolute
			1 – Relative

## 4.5 INPUTGIMBALANGLE

### *Input Gimbal Angles into the Receiver*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to input information about the current mount gimbal angles. Gimbal angles are the angle from the locked mount frame to the current gimbal location. They are input in the mount body frame. See Variable Lever Arm in the [OEM7 SPAN Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#) for details on frame definitions.



It is very important to follow the order of rotations (Z, X, Y) when determining the rotations from the locked mount frame to the current gimbal location.

**Message ID: 1317**

#### **Abbreviated ASCII Syntax:**

```
INPUTGIMBALANGLE XAngle YAngle ZAngle [XUncertainty] [YUncertainty]
[ZUncertainty]
```

#### **Abbreviated ASCII Examples:**

```
INPUTGIMBALANGLE 0.003 -0.1234 12.837
```

```
INPUTGIMBALANGLE 0.003 -0.1234 12.837 0.001 0.001 0.005
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INPUTGIMBALANGLE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	X Angle	±180		Right hand rotation from the locked mount frame X axis to the current gimbal location in degrees.	Double Array [3]	24	H
	Y Angle	±180		Right hand rotation from the locked mount frame Y axis to the current gimbal location in degrees.			
	Z Angle	±180		Right hand rotation from the locked mount frame Z axis to the current gimbal location to in degrees.			
3	X Uncertainty	0 – 180		Uncertainty of X rotation in degrees. Default is 0	Double Array [3]	24	H+24 H+40
	Y Uncertainty	0 – 180		Uncertainty of Y rotation in degrees. Default is 0			
	Z Uncertainty	0 – 180		Uncertainty of Z rotation in degrees. Default is 0			

## 4.6 INSALIGNCONFIG

### Configure ALIGN Parameters for SPAN Receiver

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use the **INSALIGNCONFIG** command to configure ALIGN for a SPAN master receiver to a secondary rover receiver. The command will configure the port on the master and rover and setup corrections at the desired rate. It also attempts to re-establish these corrections should they stop.



#### Important

- This command must be used to define the ALIGN communication between the receivers that SPAN is to configure.
- When using ICOM ports, the Ethernet settings on the SPAN master and rover receiver must be manually configured.
- The master and rover ports must be the same interface type (i.e. Serial to Serial or Ethernet to Ethernet).
- The **INSALIGNCONFIG** command can be used to set the output rate for dual antenna receivers (e.g. OEM7720). In these cases, the port configuration fields are ignored.

**Message ID: 2163**

#### Abbreviated ASCII Syntax:

```
INSALIGNCONFIG masterport [roverport] [baudrate] [outputrate]
```

#### Abbreviated ASCII Example:

```
INSALIGNCONFIG COM1 COM2 230400 5
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSALIGNCONFIG header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	masterport	NOPORT	0	Specify which COM port on the master receiver to use to communicate with an external ALIGN capable receiver.  Selecting NOPORT disables automatic dual antenna configuration.	Enum	4	H
		COM1	1				
		COM2	2				
		COM3	3				
		COM4	19				
		COM5	31				
		ICOM1	23				
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46				
		ICOM6	47				
		ICOM7	48				
3	roverport	COM1	1	Specify which rover COM port is connected to the master receiver  (Default = COM2)	Enum	4	H+4
		COM2	2				
		COM3	3				
		COM4	19				
		COM5	31				
		ICOM1	23				
		ICOM2	24				
		ICOM3	25				
		ICOM4	29				
		ICOM5	46				
		ICOM6	47				
		ICOM7	48				
		4	baudrate				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	outputrate	1, 2, 4, 5, 10		The data rate, in Hz, in which ALIGN will be output (Default = 1 Hz)	Ulong	4	H+12
6	Reserved	Reserved (Default = 0)			Ulong	4	H+16



If entering, use the default value.



The SPAN filter only requires ALIGN updates at 1 Hz. Increasing the output rate (using the *outputrate* field) does not increase performance, it only provides ALIGN logs at higher rates.

## 4.7 INSCALIBRATE

### *Initiate calibration of the INS offsets*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to initiate the calibration of INS offsets.



The RBV calibration requires a valid RBV estimate to be entered prior to initializing the calibration. See the **SETINSROTATION** command on page 1041 for details on entering a RBV estimate.



For optimal SPAN performance when using Dual Antenna with SPAN, an ALIGN offset calibration is required for each unique installation. This calibration refines the IMU to antenna baseline angular offset from the initial estimate derived from the input lever arms.

**Message ID: 1882**

#### **Abbreviated ASCII Syntax:**


```
INSCALIBRATE Offset [Trigger] [SDThreshold]
```


#### **Abbreviated ASCII Example:**

```
INSCALIBRATE RBV NEW 1.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSCALIBRATE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	Offset	ANT1	1	<p>Use this option to set the INS calibration offset from the IMU to the primary GNSS antenna</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The ANT1 option is available only on IMU Grade 2 or higher IMUs. See Models and Features in the <a href="#">OEM7 SPAN Installation and Operation User Manual</a>. </div>	Enum	4	H
		ALIGN	8	Use this option to set the INS calibration offset from the IMU Body frame to ALIGN frame rotation.			
		RBV	11	Use this option to set the INS calibration offset from the IMU Body frame to Vehicle frame rotation.			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	Trigger	STOP	0	Ends the INS calibration and applies the resulting estimate to the system.	Enum	4	H+4
		NEW	1	Begins a new calibration, overwriting any previous input.			
		ADD	2	Begins a new calibration, the results of which will be cumulatively averaged with previously calibrated offsets.  If no previous calibration is present, this trigger behaves as the NEW trigger.			
		RESET	3	Resets the current calibration process and removes it from the cumulative average computation. The previous inputs of calibrated values remain.			
4	SDThreshold			Standard Deviation Threshold (default for lever arm calibration = 0.10 m) (default for RBV and ALIGN calibration = 0.5 degrees)	Float	4	H+8

## 4.8 INSCOMMAND

### *INS Control Command*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to enable or disable INS. When INS is disabled, no INS position, velocity or attitude is output (however IMU data is still available). Also, INS aiding of tracking reacquisition is disabled. If the command is used to disable INS and then re-enable it, the INS system has to go through its alignment procedure (equivalent to issuing a **RESET** command). See System Start-Up and Alignment Techniques in the [OEM7 SPAN Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#) or [SMART7 Installation and Operation User Manual](#) for information about the SPAN alignment procedures.

**Message ID:** 379

**Abbreviated ASCII Syntax:**

```
INSCOMMAND action
```

**Abbreviated ASCII Example:**

```
INSCOMMAND ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSCOMMAND header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	Action	RESET	0	Resets the GNSS/INS alignment without clearing the INS biases.	Enum	4	H
		DISABLE	1	Disables INS navigation.			
		ENABLE	2	Enables INS navigation where alignment initialization starts again if the Action parameter was previously set to DISABLE.			
		START_NO_TIME	3	Raw IMU data will begin to flow upon system startup. IMU data collection can begin before the receiver has a GNSS solution. (default)			
		START_FINE_TIME	4	RAWIMU data will only be output after the system reaches FINESTEERING.			
		RESTART	5	Resets the GNSS/INS alignment and clears the INS biases.			

## 4.9 INSSEED

### *INS Seed Configuration*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This command enables or disables the saving and restoration of the last known SPAN solution from NVM.

**Message ID:** 1906




**Abbreviated ASCII Syntax:**

```
INSSEED Command [Validation]
```

**Abbreviated ASCII Example:**

```
INSSEED ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSSEED header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Command	DISABLE	0	Disable the INS seed functionality	Enum	4	H
		ENABLE	1	Enable the INS seed functionality			
		CLEAR	2	Clear the currently saved seed value so it will not be used until re-saved			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	Validation	VALIDATE	0	Validate INS Seed data using GNSS solution before injecting (default)	Enum	4	H+4
		INJECT	1	Force an NVM seed value (if available) to be used, without any validation.   Using this option to force the seed to be used can result in an unstable INS solution if the vehicle has moved. For advanced users only.			
4	Reserved (Default = 0)   If entering, use the default value.				Ulong	4	H+8
5	Reserved (Default = 0)   If entering, use the default value.				Ulong	4	H+12

## 4.10 INSTHRESHOLDS

### *Change the INS\_HIGH\_VARIANCE Threshold*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



This command should only be used by advanced users of GNSS+INS.

The **INSTHRESHOLDS** command allows you to customize the criteria by which the system reports the inertial solution status. This criteria is used to determine whether the solution status is reported as `INS_SOLUTION_GOOD` or `INS_HIGH_VARIANCE`.

This command is useful in situations where system dynamics are known to be challenging.

**Message ID: 1448**




#### Abbreviated ASCII Syntax:

```
INSTHRESHOLDS Threshold_Configuration
```

#### Abbreviated ASCII Example:

```
INSTHRESHOLDS DEFAULT
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSTHRESHOLDS header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Threshold_Configuration	DEFAULT	0	Standard INS status threshold settings	Enum	4	H
		LOW	1	Low INS status threshold settings (only checks the Attitude standard deviation)			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	Reserved (Default = 0)				Double	8	H+4
	 If entering, use the default value.						
4	Reserved (Default = 0)				Double	8	H+12
	 If entering, use the default value.						
5	Reserved (Default = 0)				Double	8	H+20
	 If entering, use the default value.						



## 4.11 INSZUPT

### *Request Zero Velocity Update*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to manually perform a Zero Velocity Update (ZUPT).

NovAtel's SPAN Technology System does ZUPTs automatically. It is not necessary to use this command under normal circumstances.



This command should only be used by advanced users of GNSS/INS and only when the system is truly stationary. Applying a ZUPT while moving will result in severe instability of the solution.

**Message ID: 382**

**Abbreviated ASCII Syntax:**

INSZUPT

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	INSZUPT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Reserved Default = TRUE This parameter is optional when using abbreviated ASCII syntax.				BOOL	4	H

## 4.12 RELINSAUTOMATION

### *Enables Relative INS on the Rover*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to configure the Relative INS plug and play feature on the rover receiver. RELINSAUTOMATION enables/disables the plug and play feature, sets the rover COM port to which the master receiver is connected, sets the baud rate for communication, sets the correction transfer rate and enables/disables sending the HEADINGEXTB/HEADINGEXT2B log back to the master receiver.

On issuing this command at the rover receiver, the rover will automatically sync with the master receiver and configure it to send corrections at the specified baud rate and specified data rate.



The recommended method for configuring Relative INS is to use the **RELINSCONFIG** command (see page 1028).



This command should only be issued at the rover receiver.



if the rover receiver is not connected to the master receiver using a serial COM port, use the **RELINSCONFIG** command (see page 1028).



To use the **RELINSAUTOMATION** command, the receiver requires a Relative INS capable model.

**Message ID: 1763**

#### Abbreviated ASCII Syntax:

```
RELINSAUTOMATION option [comport] [baudrate] [datarate] [headingextb_option]
```

#### Abbreviated ASCII Example:

```
RELINSAUTOMATION enable com2 230400 10 on
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	RELINSAUTOMATION header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	option	ENABLE	0	Enables or disables the plug and play feature.	Enum	4	H
		DISABLE	1				
3	comport	COM1	1	The COM port on the rover receiver to which the master receiver is connected. (default = COM2)	Enum	4	H+4
		COM2	2				
		COM3	3				
4	baudrate	9600, 19200, 38400, 57600, 115200, 230400, 460800		The baud rate used for communication between the master and rover receivers.	Ulong	4	H+8
5	datarate	1, 2, 4, 5, 10 or 20 Hz		The rate at which corrections are transferred between the receivers. (default = 10 Hz)	Ulong	4	H+12
6	headingextb_option	ON OFF		Enables or disables sending the HEADINGEXTB/ HEADINGEXT2B log back to the master receiver. (default = ON)	Enum	4	H+16

## 4.13 RELINSCONFIG

### Configure Relative INS

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to configure Relative INS on this receiver.



To use the **RELINSCONFIG** command, the receiver requires a Relative INS capable model.

**Message ID: 1797**

#### Abbreviated ASCII Syntax:

```
RELINSCONFIG enable rxtype [port] [baud] [rateinhz]
```

#### Abbreviated ASCII Example:

```
RELINSCONFIG ENABLE ROVER COM2 230400 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	RELINSCONFIG header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	enable	DISABLE	0	Enables or disables the Relative INS functionality.	Enum	4	H
		ENABLE	1				
3	rxtype	ROVER	1	Defines the receiver as the master or rover in a Relative INS configuration.	Enum	4	H+4
		MASTER	2				
4	port	See Table 233: COM Ports on the next page		Communication port used to communicate with the other receiver. (default = COM2)	Enum	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
5	baud	9600, 19200, 38400, 57600, 115200, 230400, 460800		The baud rate used for communication between the master and rover receivers. (default = 230400)	Ulong	4	H+12
6	rateinhz	1, 2, 4, 5, 10 or 20 Hz		The rate at which corrections are transferred between the receivers. (default =10 Hz)	Ulong	4	H+16

Table 233: COM Ports

Decimal	ASCII	Description
1	COM1	COM port 1
2	COM2	COM port 2
3	COM3	COM port 3
13	USB1	USB port 1
14	USB2	USB port 2
15	USB3	USB port 3
19	COM4	COM port 4
23	ICOM1	IP virtual COM port 1
24	ICOM2	IP virtual COM port 2
25	ICOM3	IP virtual COM port 3
29	ICOM4	IP virtual COM port 4
31	COM5	COM port 5
46	ICOM5	IP virtual COM port 5
47	ICOM6	IP virtual COM port 6
48	ICOM7	IP virtual COM port 7
49	SCOM1	Scripted application COM port 1
50	SCOM2	Scripted application COM port 2
51	SCOM3	Scripted application COM port 3
52	SCOM4	Scripted application COM port 4

## 4.14 SETALIGNMENTORIENTATION

### *Enable or disable orientation verification*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This command enables or disables the orientation verification condition for INS alignment. If the detected enclosure frame roll or pitch exceeds the defined thresholds, INS alignment will not occur.

This command is specifically designed for use on smart antenna products where the GNSS antenna is rotating with the IMU and the customer wants to ensure the enclosure is in the proper orientation (right side up) prior to SPAN completing alignment.

**Message ID: 2164**

#### **Abbreviated ASCII Syntax:**

```
SETALIGNMENTORIENTATION switch [Roll/Pitch Threshold]
```

#### **Abbreviated ASCII Example:**

```
SETALIGNMENTORIENTATION ENABLE 60.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETALIGNMENT ORIENTATION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Switch	Disable	0	Disable orientation check on INS alignment.	Enum	4	H
		Enable	1	Enable orientation check on INS alignment.			
3	Roll/Pitch Threshold	0.0 - 90.0		Roll and pitch lower limit for preventing INS alignment (degrees, default = 60.0).	Double	8	H+4

## 4.15 SETALIGNMENTVEL

### *Set the Minimum Kinematic Alignment Velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use the **SETALIGNMENTVEL** command to adjust the minimum required velocity for a kinematic alignment.

Useful in cases, such as helicopters, where alignment velocity should be increased to prevent a poor alignment before the vehicle/aircraft is able to flight straight and level.

**Message ID: 1397**

#### **Abbreviated ASCII Syntax:**

```
SETALIGNMENTVEL velocity
```

#### **Abbreviated ASCII Example**

```
SETALIGNMENTVEL 5.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETALIGNMENTVEL header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Velocity	Minimum: 0.2 m/s (Default is 5 m/s)		The minimum velocity, in m/s, required to kinematically align.	Double	8	H

## 4.16 SETHEAVEWINDOW

### *Set Heave Filter Length*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to control the length of the heave filter. This filter determines the heave (vertical displacement) of the IMU, relative to a long term level surface. This command does not control the DELAYEDHEAVE filter window.

**Message ID:** 1383

**Abbreviated ASCII Syntax:**

```
SETHEAVEWINDOW filterlength
```

**Abbreviated ASCII Example:**

```
SETHEAVEWINDOW 35
```

Field	Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETHEAVEWINDOW header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	FilterLength	Integer (1 – 300 s) (default = 20 s)		This filter length will be used in the heave filter. Typically, set the filter length to 5 x wave period	Long	4	H



## 4.17 SETIMUEVENT

### *IMU Event Selection Command*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7

This command is used to select the event used for timing the IMU data. IMUs that use a Time of Validity (ToV) signal send a pulse (using an Event IN) to the receiver. IMUs that synchronize the IMU clock with the receiver clock require a signal from the receiver (using an Event OUT).


**Message ID:** 1965

#### Abbreviated ASCII Syntax:

```
SETIMUEVENT Direction Event
```

#### Abbreviated ASCII Example:

```
SETIMUEVENT OUT EVENT3
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Command header	–	–	SETIMUEVENT header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	–	H	0
2	Direction	IN	1	Direction of event  IN2 is only available on the OEM7500 and is intended for IMUs with multiple ToVs.	Enum	4	H
		OUT	2				
		IN2	3				
3	Event	OFF	1	The event to assign to the direction.  The DEFAULT event is EVENT2 for the IN direction and EVENT1 for the OUT direction.	Enum	4	H+4
		DEFAULT	2				
		EVENT1	3				
		EVENT2	4				
		EVENT3	5				
		EVENT4	6				

## 4.18 SETIMUPORTPROTOCOL

### *Sets the Protocol Used for the IMU Serial Port*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7

Use the **SETIMUPORTPROTOCOL** command to change the IMU serial port to use either RS-232 or RS-422 protocol. This overrides the default configured internally when the **CONNECTIMU** command is sent.



Before changing the IMU serial port protocol:

1. Make sure the receiver port connected to the IMU is capable of RS-422 protocol. Refer to the [OEM7 Installation and Operation User Manual](#) for information about the receiver serial ports.
2. The IMU data message is input into the receiver at that particular protocol.

**Message ID: 1767**

#### Abbreviated ASCII Syntax:

```
SETIMUPORTPROTOCOL SerialProtocol
```

#### Abbreviated ASCII Example:

```
SETIMUPORTPROTOCOL RS422
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETIMUPORTPROTOCOL header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	SerialProtocol	RS232 RS422		The protocol for the IMU serial port.	Enum	4	H

## 4.19 SETIMUSPECS

### *Specify Error Specifications and Data Rate*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7



This command should only be used by advanced users of GNSS/INS.

Use the **SETIMUSPECS** command to specify the error specifications and data rate for the desired IMU. If the default specs for the supported models are different than the unit used then this command can be used to override the default values.

This command is only available for the following IMUs:

- Honeywell HG1930 (default specifications are for the AA99/CA50 model)
- Honeywell HG1900 (default specifications are for the CA29/CA50 model)

**Message ID: 1295**

#### **Abbreviated ASCII Syntax:**

```
SETIMUSPECS DataRate AccelBias AccelVRW GyroBias GyroARW AccelSFError
GyroSFError [DataLatency]
```

#### **Abbreviated ASCII Example: (iMAR-FSAS Specs)**

```
SETIMUSPECS 200 1 .0198 0.75 0.0028 300 300 2.5
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETIMUSPECS header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	DataRate	100 Hz to 400 Hz		Data rate of the IMU	Ushort	2	H
3	AccelBias	-		Total accelerometer bias in milli-g	Double	8	H+2

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	AccelVRW	-		Accelerometer velocity random walk in m/s/rt-hr	Double	8	H+10
5	GyroBias	-		Total gyroscope bias in deg/hr	Double	8	H+18
6	GyroARW	-		Gyroscope angular random walk in deg/rt-hr	Double	8	H+26
7	AccelSFError	> 0		Accelerometer scale factor error in parts per million. Optional. Default = 1000 ppm.	Ulong	4	H+34
8	GyroSFError	> 0		Gyroscopic scale factor error in parts per million. Optional. Default = 1000 ppm.	Ulong	4	H+38
9	DataLatency	> 0		Time delay in milliseconds from the time of validity of the IMU data to the time the input pulse is received by the SPAN enabled receiver. This may include filtering delays, processing delays and transmission times depending on the timing method (TOV, ASYNC, SYNC) and the internal IMU handling. Optional. Default = 0.0.	Double	8	H+42

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
10	Reserved	SCALE_ DEFAULT	0	Reserved. If using ASCII format, enter SCALE_ DEFAULT. If using binary format, enter 0.	Enum	4	H+50
11	CRC	-		32-bit CRC	Hex	4	H+54

## 4.20 SETINITAZIMUTH

### *Set Initial Azimuth and Standard Deviation*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to start SPAN operation with a previously known azimuth. Azimuth is the weakest component of a coarse alignment and is also the easiest to know from an external source (i.e., like the azimuth of roadway). When using this command, SPAN operation through alignment will appear the same as with a usual coarse alignment. Roll and pitch is determined using averaged gyro and accelerometer measurements. The input azimuth is used rather than what is computed by the normal coarse alignment routine.

- Input azimuth values must be accurate for good system performance.
- Sending **SETINITAZIMUTH** resets the SPAN filter. Following realignment, vehicle dynamics are required for the filter to re-converge. Bridging performance is poor before filter convergence.
- The entered azimuth angle is with respect to the configured output frame. This is generally the vehicle frame unless a User Frame offset has been configured using the **SETINSROTATION** command (see page 1041). All offsets should be entered before entering the **SETINITAZIMUTH** command.
- This command is not save configurable and must be re-entered after each start-up. The command can be entered at any time and will be used automatically when the system is ready to begin alignment.



Azimuth is positive in a clockwise direction when looking towards the z-axis origin.

**Message ID: 863**

**Abbreviated ASCII Syntax:**

```
SETINITAZIMUTH azimuth azSTD
```

**Abbreviated ASCII Example:**

```
SETINITAZIMUTH 90 5
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETINITAZIMUTH header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	azimuth	0 to 360		Input azimuth angle (degrees)	Double	8	H
3	azSTD	1 to 25		Input azimuth standard deviation angle (degrees)	Float	4	H+8

## 4.21 SETINSPROFILE

### *Sets filter behavior depending on system environment*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This command sets specific filter behavior depending on the environment the system is installed in. The DEFAULT profile is the legacy setting from earlier SPAN products. The other profiles make changes specific to that environment.

See INS Profiles in the [OEM7 SPAN Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#) for a detailed description of each profile's effect.

**Message ID:** 1944

#### Abbreviated ASCII Syntax:

```
SETINSPROFILE profile
```

#### Abbreviated ASCII Example:

```
SETINSPROFILE LAND
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETINSPROFILE header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	Profile	Default	0	Default INS profile with standard SPAN behavior.	Enum	4	H
		LAND	1	INS profile for land vehicles			
		MARINE	2	INS profile for marine vehicles			
		FIXEDWING	3	INS profile for fixed wing aircraft			
		FOOT	4	INS profiles for walking/backpack applications			
		VTOL	5	INS profile for vertical takeoff and landing vehicles (UAVs, helicopters, etc.)			
		RAIL	6	INS profile for trains			
		AGRICULTURE	7	INS profile for agriculture applications			



## 4.22 SETINSROTATION

**Specifies rotational offsets between the IMU frame and other reference frames**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use the **SETINSROTATION** command to specify rotational offsets between the IMU frame and other reference frames, such as the vehicle frame or an ALIGN baseline. Offsets must be entered as the rotation from the IMU body frame, to the frame of interest. The order of rotations is Z, X, Y. All rotations are right handed.



- It is very important to follow the order of rotations (Z, X, Y) when determining the rotations from IMU body frame to frame of interest.
- It is strongly recommended to enter meaningful standard deviations when entering an ALIGN or RBV rotation.



To specify translational offsets between frames, see the **SETINSTRANSULATION** command on page 1044.

**Message ID: 1921**

**Abbreviated ASCII Syntax:**

```
SETINSROTATION INSRotation XRotation YRotation ZRotation [XRotationSD]
[YRotationSD] [ZRotationSD]
```

**Abbreviated ASCII Example:**

```
SETINSROTATION RBV 0 0 90 3.0 3.0 3.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETINSROTATION header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	INSRotation	<i>Table 234: Rotational Offset Types on page 1043</i>		Rotational offset to be set.	Enum	4	H



Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	XRotation	±180		X rotation offset from IMU origin (degrees)	Float Array [3]	12	H+4
	YRotation	±180		Y rotation offset from IMU origin (degrees)			
	ZRotation	±180		Z rotation offset from IMU origin (degrees)			
4	XRotationSD	0.25 to 45		Optional X rotation offset standard deviation (degrees). Applicable for RBV and ALIGN rotations. (Default = 3.0)	Float Array [3]	12	H+16
	YRotationSD	0.25 to 45		Optional Y rotation offset standard deviation (degrees). Applicable for RBV and ALIGN rotations. (Default = 3.0)			
	ZRotationSD	0.25 to 45		Optional Z rotation offset standard deviation (degrees). Applicable for RBV and ALIGN rotations. (Default = 3.0)			
5	Reserved (Default = 0)				Long	4	H+28
 If entering, use the default value.							

Table 234: Rotational Offset Types

ASCII Value	Binary Value	Description
USER	4	Rotation from the IMU body frame to the user output frame. This offset shifts the attitude information in the INSPVA, INSATT, INSATTQS, INSPVACMP, IMURATEPVA, GIMBALLEDPVA, PASHR and TSS1 logs, along with their short header and extended versions. As well it changes the output frame for the CORRIMUS, IMURATECORRIMUS, INSVELUSER, RELINSPVA and SYNCRELINSPVA logs.
MARK1	5	Rotation from the IMU body frame to the desired output for MARK1. This offset rotates the attitude information in the MARK1PVA log.
MARK2	6	Rotation from the IMU body frame to the desired output for MARK2. This offset rotates the attitude information in the MARK2PVA log.
ALIGN	8	Rotation from the IMU body frame to an ALIGN dual antenna solution. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  When using a dual antenna ALIGN solution with SPAN, this offset will be calculated automatically if translational offsets to both the primary and secondary GNSS antennas are provided using the <b>SETINSTRANSLATION</b> command on the next page. </div>
MARK3	9	Rotation from the IMU body frame to the desired output for MARK3. This offset rotates the attitude information in the MARK3PVA log.
MARK4	10	Rotation from the IMU body frame to the desired output for MARK4. This offset rotates the attitude information in the MARK4PVA log.
RBV	11	Rotation from the IMU body frame to the vehicle frame.
RBM	12	Rotation from the IMU body frame to the gimbal mount body frame.

## 4.23 SETINSTRANSALATION

**Specifies translational offsets between the IMU frame and other reference frames**

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use the **SETINSTRANSALATION** command to specify translational offsets between the IMU frame and other reference frames, including GNSS antennas or the desired output frame. Offsets must be entered as the vector from the IMU, to the frame or position of interest. Offsets can be entered either in the IMU body frame, or the vehicle frame; offsets in the vehicle frame will be automatically rotated into the IMU body frame using the best available IMU Body to Vehicle Rotation (RBV).

For details on entering the RBV rotation or other angular offsets, see the **SETINSROTATION** command on page 1041.

**Message ID: 1920**

### Abbreviated ASCII Syntax:

```
SETINSTRANSALATION INStranslation XTranslation YTranslation ZTranslation
[XTranslationSD] [YTranslationSD] [ZTranslationSD] [InputFrame]
```

### Abbreviated ASCII Example:

```
SETINSTRANSALATION USER 1.0 2.0 3.0 0.05 0.05 0.05 VEHICLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETINSTRANSALATION header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	InsTranslation	See Table 235: Translation Offset Types on page 1046		Translation offset to be set	Enum	4	H
3	XTranslation	±100		X translation offset from IMU origin (m)	Float Array [3]	12	H+4
	YTranslation	±100		Y translation offset from IMU origin (m)			
	ZTranslation	±100		Z translation offset from IMU origin (m)			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	XTranslationSD	0 to 10		Optional X translation offset standard deviation (m)	Float Array [3]	12	H+16
	YTranslationSD	0 to 10		Optional Y translation offset standard deviation (m)			
	ZTranslationSD	0 to 10		Optional Z translation offset standard deviation (m)			
5	InputFrame	<i>Table 236: Translation Input Frame on the next page</i>		Optional input frame for translation offset values	Enum	4	H+28



For the ANT1, ANT2, EXTERNAL and GIMBAL translations, the standard deviation defaults are set to 10% of the translation value (up to a max of 10 metres).



The offset standard deviation values are not optional when entering a value for the Input Frame.



For USER, MARK1, MARK2, MARK3 and MARK4 translations, standard deviations values are not used in the solution.



Large standard deviations can lead to an inaccurate position solution. Therefore, it is highly recommended to measure translation offsets as accurately as possible and to manually enter translation offset standard deviations that reflect that accuracy.



It is important to keep in mind that as the user offset is moved further from the IMU center of navigation, the PVA solution will become noisier due to the projection of angular changes over a longer distance.

Table 235: Translation Offset Types


ASCII Value	Binary Value	Description
ANT1	1	Offset from the IMU center of navigation to the phase center of the primary GNSS antenna.  <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;">  Do not use the ANT1 option (SETINSTRANSFORMATION ANT1) on a SMART7-S. The correct lever arm value has been set at the factory. </div>
ANT2	2	Offset from the IMU center of navigation to the phase center of the secondary GNSS antenna.
EXTERNAL	3	Offset from the IMU center of navigation to the external position source location. This offset type is for use with the <b>EXTERNALPVAS</b> command (see page 1006).
USER	4	Translation from the IMU center of navigation to the user output location. This offset shifts the position and velocity information in the INSPVA, INSPOS, INSVEL, INSVELUSER, INSATT, INSSPD, IMURATEPVA, INSPVACMP and GIMBALLEDPVA logs along with their short header and extended versions. As well it changes the relative offset point of the HEAVE, SYNCHEAVE, DELAYEDHEAVE, RELINSPVA and SYNCRELINSPVA logs.
MARK1	5	Translation from the IMU center of navigation to the MARK1 output location. This offset shifts the position and velocity information in the MARK1PVA log.
MARK2	6	Translation from the IMU center of navigation to the MARK2 output location. This offset shifts the position and velocity information in the MARK2PVA log.
GIMBAL	7	Translation from the IMU center of navigation to the gimbal mount center of rotation.
MARK3	9	Translation from the IMU center of navigation to the MARK3 output location. This offset shifts the position and velocity information in the MARK3PVA log.
MARK4	10	Translation from the IMU center of navigation to the MARK4 output location. This offset shifts the position and velocity information in the MARK4PVA log.

Table 236: Translation Input Frame

ASCII Value	Binary Value	Description
IMUBODY	0	Offset is provided in the IMU enclosure frame. Default: IMUBODY

ASCII Value	Binary Value	Description
VEHICLE	1	<p>Offset is provided in the vehicle frame.</p> <p>Offsets entered in the vehicle frame will be automatically rotated into the IMU frame using the best available RBV (rotation from IMU Body to Vehicle) information when required.</p> <p>Vehicle frame offsets should only be used if the RBV is known accurately, either through user measurement or calibration.</p> <p>The order of entry for vehicle frame offsets and the RBV rotation does not matter.</p>

## 4.24 SETINSUPDATE

### *Enable/Disable INS Filter Updates*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



This command should only be used by advanced users of GNSS+INS.

Use this command to enable or disable the available INS filter updates.

**Message ID: 1821**

**Abbreviated ASCII Syntax:**

```
SETINSUPDATE INSUpdate Trigger
```

**Abbreviated ASCII Example:**

```
SETINSUPDATE ZUPT DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETINSUPDATE header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	INSUpdate	POS	0	Position updates	Enum	4	H
		ZUPT	1	Zero velocity updates			
		PSR	2	Pseudorange updates			
		ADR	3	Carrier phase updates			
		DOPPLER	4	Doppler updates			
		ALIGN	5	Heading updates			
		DMI	6	Distance measuring instrument (wheel sensor) updates			



Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	Trigger	DISABLE	0	Disable the INS update specified in the INSUpdate field.	Enum	4	H+4
		ENABLE	1	Enable the INS update specified in the INSUpdate field.			

## 4.25 SETMAXALIGNMENTTIME

### *Set a Time Limit for Static Course Alignment*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to set a maximum time limit allowed for static coarse alignments. Coarse alignments typically take under 60 seconds, but in heavy vibration conditions they can take much longer trying to compensate for the vibration induced noise. This command is used to cap the time to a specific length.



This command is for advanced users only.  
Alignment accuracy cannot be guaranteed if the alignment time is capped using this command.

**Message ID: 1800**

#### Abbreviated ASCII Syntax:

```
SETMAXALIGNMENTTIME switch [duration]
```

#### Abbreviated ASCII Example:

```
SETMAXALIGNMENTTIME ENABLE 90
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETMAXALIGNMENTTIME header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	switch	DISABLE	0	Disables the static alignment time limit.	Enum	4	H
		ENABLE	1	Enables the static alignment time limit.			
3	duration	30 - 300		Maximum static alignment time in seconds. Default is 180.	Ulong	4	H+4

## 4.26 SETRELINSOUTPUTFRAME

### *Sets the Relative INS Output Frame*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use this command to change the frame of the output solution provided in the RELINSPVA and SYNCRELINSPVA logs. See **RELINSPVA** log on page 1174 and **SYNCRELINSPVA** log on page 1178 for information about these logs.

See in the [OEM7 SPAN Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#) for information about the Relative INS functionality.

**Message ID:** 1775

#### Abbreviated ASCII Syntax:

```
SETRELINSOUTPUTFRAME OutputFrame [DiffCriteria]
```

#### Abbreviated ASCII Example:

```
SETRELINSOUTPUTFRAME ECEF TRUE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETRELINSOUTPUTFRAME header This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	OutputFrame	ROVER	1	Frame of the output solution in the RELINSPVA and SYNCRELINSPVA logs.	Enum	4	H
		MASTER	2	ROVER – the output frame of the rover INS solution MASTER – the output frame of the master INS solution			
		ECEF	3	ECEF – Earth Centered Earth Fixed			
		LOCALLEVEL	4	LOCALLEVEL – Local level The default is the ROVER.			
3	DiffCriteria	FALSE	0	The delta solution is computed as Rover minus Master. (default)	Bool	1	H+4
		TRUE	1	The delta solution is computed as Master minus Rover.			

## 4.27 SETUPSENSOR

### *Add a new sensor object*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to add a new sensor object to the system. A sensor object consists of an ID, an Event\_Out line and an Event\_In line. This is intended as a simplified way to set up triggering to and from a sensor rather than configuring all connections independently. It also allows for event pulses to be sent to a sensor at specific GPS times (see the **TIMEDEVENTPULSE** command on page 1056).

**Message ID: 1333**

#### **Abbreviated ASCII Syntax:**

```
SETUPSENSOR SensorID EventOut OPP OAP EventIn EIC IPP ITB ITG
```

#### **Abbreviated ASCII Example:**

```
SETUPSENSOR SENSOR3 MARK1 POSITIVE 2 MARK4 EVENT POSITIVE 0 2
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	SETUPSENSOR header  This field contains the command name for abbreviated ASCII or the message header for ASCII (see page 33) or Binary (see page 37).	-	H	0
2	Sensor ID	SENSOR1	0	The sensor to configure.	Enum	4	H
		SENSOR2	1				
		SENSOR3	2				
3	EventOut	MARK1	0	Associate a specific MARK Event_Out line to this sensor configuration.	Enum	4	H+4
		MARK2	1				
		MARK3	2				
		MARK4	3				
4	OPP	NEGATIVE	0	Mark output pulse polarity	Enum	4	H+8
		POSITIVE	1				

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
5	OAP	2 - 500		Mark output active period in milliseconds. Value must be divisible by 2.	Ulong	4	H+12
6	EventIn	MARK1	0	Associate a specific MARK Event_In line to this sensor configuration.	Enum	4	H+16
		MARK2	1				
		MARK3	2				
		MARK4	3				
7	EIC	DISABLE	0	Event in control	Enum	4	H+20
		EVENT	1				
8	IPP	NEGATIVE	0	Mark input pulse polarity	Enum	4	H+24
		POSITIVE	1				
9	ITB	-99999999 to 99999999		Mark input time bias in milliseconds	Long	4	H+28
10	ITG	2 to 3599999		Mark input time guard in milliseconds	Ulong	4	H+32



The Event\_In and Event\_Out options available are dependent on the receiver used in the SPAN system. For information about the Event lines supported, see the Strobe Specifications for the receiver in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#), or [CPT7 Installation and Operation User Manual](#).



MARK4 is available only on SPAN systems with an OEM7600, OEM7700 or OEM7720 receiver.

## 4.28 TAGNEXTMARK

### *Tags the Next Incoming Mark Event*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to tag the next incoming mark event on the selected mark with a 32-bit number. This is available in the **TAGGEDMARK1PVA**, **TAGGEDMARK2PVA**, **TAGGEDMARK3PVA** and **TAGGEDMARK4PVA** log (see page 1181) to easily associate the PVA log with a supplied event.

**Message ID:** 1257

#### Abbreviated ASCII Syntax:

```
TAGNEXTMARK Mark Tag
```

#### Abbreviated ASCII Example:

```
TAGNEXTMARK MARK1 1234
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	TAGNEXTMARK header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0
2	Mark	MARK1	0	Event line	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				
3	Tag	-	-	Tag for next mark event	Ulong	4	H+4



The Mark options available are dependent on the receiver used in the SPAN system. For information about the Event lines supported, see the Strobe Specifications for the receiver in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#).

## 4.29 TIMEDEVENTPULSE

### *Add a new camera event*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this command to add a new camera event to the system. **TIMEDEVENTPULSE** sends a pulse on the sensor MARK output at the selected GPS time and sets the trigger on the sensor MARK input to be tagged with an event ID (see the **TAGGEDMARK1PVA**, **TAGGEDMARK2PVA**, **TAGGEDMARK3PVA** and **TAGGEDMARK4PVA** log on page 1181). The lines connected to each sensor are configured using the **SETUPSENSOR** command (see page 1053).



A maximum of 10 unprocessed events can be buffered into the system. A **TIMEDEVENTPULSE** command must be entered at least 1 second prior to the requested event time.



The receiver will accept the **TIMEDEVENTPULSE** command only if the receiver time status is **FINESTEERING**.

**Message ID: 1337**

#### **Abbreviated ASCII Syntax:**

```
TIMEDEVENTPULSE SensorID GPSWeek GPSSeconds [Event ID]
```

#### **Abbreviated ASCII Example:**

```
TIMEDEVENTPULSE -1 1617 418838 100
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Command header	-	-	TIMEDEVENTPULSE header  This field contains the command name for abbreviated ASCII or the message header for <i>ASCII</i> (see page 33) or <i>Binary</i> (see page 37).	-	H	0



Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
2	SensorID	ALL	-1 (0xffffffff)	The sensor(s) affected by the trigger command.	Long	4	H
		SENSOR1	0x01	The decimal representation of the combination of bits 0-2 can be used to select a combination of active sensors (e.g. 5 [101] will select sensors 1 and 3).			
		SENSOR2	0x02				
		SENSOR3	0x04				
3	GPSWeek	0 - MAX Ulong		The GPS week that triggers the event.	Ulong	4	H+4
4	GPSSeconds	0 - 604800		The GPS week seconds that triggers the event.	Double	8	H+8
5	Event ID	0- MAX Ulong		The event's identifier, used to tag the TAGGEDMARKxPVA logs if a sensor input is enabled. Optional Default = 0	Ulong	4	H+16

## Chapter 5 SPAN Logs

The SPAN specific logs follow the same general logging scheme as normal OEM7 Family logs. They are available in ASCII or binary formats and are defined as being either synchronous or asynchronous. All the logs in this chapter are used only with the SPAN system.

For information on other available logs and output logging, refer to *Logs* on page 481.

One difference from the standard OEM7 Family logs is there are two possible headers for the ASCII and binary versions of the logs. Which header is used for a given log is described in the log definitions in this chapter. The reason for the alternate short headers is that the normal OEM7 binary header is quite long at 28 bytes. This is nearly as long as the data portion of many of the INS logs and creates excess storage and baud rate requirements. Note that the INS related logs contain a time tag within the data block in addition to the time tag in the header. The time tag in the data block should be considered the exact time of applicability of the data. All INS Position, Velocity and Attitude logs can be obtained at a rate of up to 200 Hz. The standard deviation and update logs are available once per second.



Each ASCII log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters, for example, **\*1234ABCD[CR][LF]**. This value is a 32-bit CRC of all bytes in the log, excluding the '#' or '%' identifier and the asterisk preceding the four checksum digits. See also *Description of ASCII and Binary Logs with Short Headers* on page 48.

*Table 237: Inertial Solution Status* on page 1081 shows the status values included in the INS position, velocity and attitude output logs. If the IMU is connected properly and a good status value is not being received, check the hardware setup to ensure it is properly connected. This situation can be recognized in the RAWIMU data by observing accelerometer and gyro values which are not changing with time.



### Logging Restriction Important Notice

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format and, if possible, the short header binary format (available on most INS logs).

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing  
RAWIMUXSB ONNEW
    - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate.
  - Real time INS solution  
IMURATEPVA ONNEW or IMURATEPVAS ONNEW
- Other possible INS solution logs available at high rates are: INSPVASB, INSPOSSB, INSVELSB, INSATTSB



The periods available when using the ONTIME trigger are 0.005 (200 Hz), 0.01 (100 Hz), 0.02 (50 Hz), 0.05, 0.1, 0.2, 0.25, 0.5, 1, and any integer number of seconds.

## 5.1 Logs with INS or GNSS Data

There are several logs in the system designed to output the best available solution as well as many logs that output only a specific solution type (PSR, RTK, INS, etc). The table below lists the logs that can provide either a GNSS solution or an INS solution. Most of these derive from the solution the system picks as the best solution. SPAN systems also have a secondary best solution that derives from the GNSS solution only (**BESTGNSSPOS** log (see page 1060) and **BESTGNSSVEL** log (see page 1062)). The position output from these logs is at the phase center of the antenna.

Log	Log Format	GNSS/INS
BESTPOS	NovAtel	YES
BESTVEL	NovAtel	YES
BESTUTM	NovAtel	YES
BESTXYZ	NovAtel	YES
GPGGA	NMEA	YES
GPGLL	NMEA	YES
GPRMC	NMEA	YES
GPVTG	NMEA	YES

## 5.2 BESTGNSSPOS

### *Best GNSS Position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the best available GNSS position (without INS) computed by the receiver. In addition, it reports several status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used.

With the system operating in an RTK mode, this log reflects the latest low latency solution for up to 60 seconds after reception of the last base station observations. After this 60 second period, the position reverts to the best solution available and the degradation in accuracy is reflected in the standard deviation fields. If the system is not operating in an RTK mode, pseudorange differential solutions continue for the time specified in the **PSRDIFFTIMEOUT** command (see page 320).



BESTGNSSPOS always outputs positions at the antenna phase center.

**Message ID:** 1429

**Log Type:** Synch


**Recommended Input:**

```
log bestgnssposa ontime 1
```

**ASCII Example:**

```
#BESTGNSSPOSA,USB1,0,65.0,FINESTEERING,2209,487877.000,02000020,c292,16809;
SOL_COMPUTED,PPP,51.15043712234,-114.03067870773,1097.3556,-17.0001,WGS84,
0.0176,0.0123,0.0317,"TSTR",17.000,0.000,42,38,38,37,00,00,3f,37*5fea7d4b
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTGNSSPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Sol Status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	Pos Type	Position type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	Lat	Latitude (degrees)	Double	8	H+8
5	Lon	Longitude (degrees)	Double	8	H+16
6	Hgt	Height above mean sea level (metres)	Double	8	H+24

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
7	Undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum  <div style="border: 1px solid black; padding: 5px; width: fit-content;">  <p>When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.</p> </div>	Float	4	H+32
8	Datum ID	Datum ID 61 = WGS84 63 = USER	Enum	4	H+36
9	Lat $\sigma$	Latitude standard deviation (metres)	Float	4	H+40
10	Lon $\sigma$	Longitude standard deviation (metres)	Float	4	H+44
11	Hgt $\sigma$	Height standard deviation (metres)	Float	4	H+48
12	Stn ID	Base station ID	Char[4]	4	H+52
13	Diff_age	Differential age in seconds	Float	4	H+56
14	Sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite solutions used in solution	Uchar	1	H+65
17	#solnL1SVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 90: Extended Solution Status</i> on page 527)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see <i>Table 89: Galileo and BeiDou Signal-Used Mask</i> on page 526)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see <i>Table 88: GPS and GLONASS Signal-Used Mask</i> on page 526)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.3 BESTGNSSVEL

### *Best Available GNSS Velocity Data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the best available GNSS velocity information (without INS) computed by the receiver. In addition, it reports a velocity status indicator, which is useful to indicate whether or not the corresponding data is valid. The velocity measurements sometimes have a latency associated with them. The time of validity is the time tag in the log minus the latency value.

The velocity is typically computed from the average change in pseudorange over the time interval or the RTK Low Latency filter. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the BESTGNSSVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the positioning filters are updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency is reduced by increasing the update rate of the positioning filter used by requesting the BESTGNSSVEL or BESTGNSSPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz reduces the velocity latency to 0.005 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

A valid solution with a latency of 0.0 indicates the instantaneous Doppler measurement was used to calculate velocity.

**Message ID:** 1430

**Log Type:** Synch

**Recommended Input:**

```
log bestgnssvela ontime 1
```

**ASCII Example:**

```
#BESTGNSSVELA,USB1,0,64.0,FINESTEERING,2209,488172.000,02000020,00b0,16809;  
SOL_COMPUTED,PPP,0.250,12.000,0.0025,23.890175,0.0021,0.0*f40aab04
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	Log header	BESTGNSSVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Sol Status	Solution status, see <i>Table 86: Solution Status</i> on page 523	Enum	4	H
3	Vel Type	Velocity type, see <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	Latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+8

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
5	Age	Differential age	Float	4	H+12
6	Hor Spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	Trk Gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	Vert Spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.4 CORRIMUDATA

### *Corrected IMU Measurements*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



The **CORRIMUS** log on page 1068 was introduced in firmware version OEM 7.07.0x and is designed to replace the **CORRIMUDATA** and **CORRIMUDATAS** logs. It is recommended to upgrade to **CORRIMUS** log on page 1068 when collecting corrected IMU measurements synchronously. The **CORRIMUDATA** and **CORRIMUDATAS** logs are still available, but will eventually be phased out.

The CORRIMUDATA log contains the raw IMU data corrected for gravity, the earth's rotation and estimated sensor errors. The values in this log are incremental values, accumulated over the logging interval of CORRIMUDATA, in units of radians for the attitude rate and m/s for the accelerations. Data output is not in the IMU Body frame, but is automatically rotated into the user configured output frame (configured with the **SETINSROTATION** command (see page 1041), default Vehicle frame).



The short header format, CORRIMUDATAS, is recommended, as it is for all high data rate logs.

CORRIMUDATA can be logged with the ONTIME trigger, up to a rate of 200 Hz.



Since the CORRIMUDATA log is synchronous, if you log at a rate less than the full data rate of the IMU, the corrected IMU data is accumulated to match the requested time interval. For asynchronous full rate data, see the **IMURATECORRIMUS** log on page 1074.



To obtain the instantaneous rates of acceleration (in m/s/s) or rotation (in rad/s) from the output values of measurements per sample rate (m/s/sample and rad/sample), multiply the output values by the CORRIMUDATA logging rate in Hz.

**Message ID: 812**

**Log Type: Synch**

**Recommended Input:**

```
log corrimudatab ontime 0.01
```

**Example log:**

```
#CORRIMUDATAA,USB1,0,65.0,FINESTEERING,2209,488261.000,02000020,e5fc,16809;2209,488261.000000000,-0.0000097404099474,0.0000046348926633,0.0000137198213235,-0.0000719383196058,0.0003131269933030,0.0005105002543702*0c6740ba
```



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CORRIMUDATA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x axis rotation (right-handed) (rad/sample)	Double	8	H+12
5	RollRate	About y axis rotation (right-handed) (rad/sample)	Double	8	H+20
6	YawRate	About z axis rotation (right-handed) (rad/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+56
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.5 CORRIMUDATAS

### Short Corrected IMU Measurements

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



The **CORRIMUS** log on page 1068 was introduced in firmware version OEM 7.07.0x and is designed to replace the **CORRIMUDATA** and **CORRIMUDATAS** logs. It is recommended to upgrade to **CORRIMUS** log on page 1068 when collecting corrected IMU measurements synchronously. The **CORRIMUDATA** and **CORRIMUDATAS** logs are still available, but will eventually be phased out.

This log is the short header version of the **CORRIMUDATA** log (see page 1064).



To obtain the instantaneous rates of acceleration (in m/s/s) or rotation (in rad/s) from the output values of measurements per sample rate (m/s/sample and rad/sample), multiply the output values by the **CORRIMUDATAS** logging rate in Hz.

**Message ID: 813**

**Log Type: Synch**

**Recommended Input:**

```
log corrimudatasb ontime 0.01
```

**Example log:**

```
%CORRIMUDATASA, 2209, 488331.000; 2209, 488331.000000000, -0.0000016434879909,
0.0000053888857361, -0.0000077180775912, -0.0001338946509402,
-0.0013131748634920, -0.0002721905080061*bd5ba919
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CORRIMUDATAS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x-axis rotation (right-handed) (rad/sample)	Double	8	H+12
5	RollRate	About y-axis rotation (right-handed) (rad/sample)	Double	8	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	YawRate	About z-axis rotation (right-handed) (rad/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x-axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y-axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z-axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+56
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.6 CORRIMUS

### *Corrected IMU Measurements*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S



The **CORRIMUS** log replaces the **CORRIMUDATA** and **CORRIMUDATAS** logs.

The **CORRIMUS** log contains the raw IMU data corrected for gravity, the earth's rotation and estimated sensor errors. The values in this log are incremental values, accumulated over the logging interval of **CORRIMUS**, in units of radians for the attitude rate and m/s for the accelerations. Data output is not in the IMU Body frame, but is automatically rotated into the user configured output frame (configured with the **SETINSROTATION** command (see page 1041), default is Vehicle frame).

The **CORRIMUS** log uses the short header format and can be used for high data rate logs. **CORRIMUS** can be logged with the ONTIME trigger, up to a rate of 200 Hz.



Since the **CORRIMUS** log is synchronous, if you log at a rate less than the full data rate of the IMU, the corrected IMU data is accumulated to match the requested time interval. For asynchronous full rate data, see the **IMURATECORRIMUS** log on page 1074.



To obtain the instantaneous rates of acceleration (in m/s/s) or rotation (in rad/s) from the output values of measurements per sample rate (m/s/sample and rad/sample); multiply the output values by: **IMUDataRate (Hz) / IMUDataCount**.



Due to IMU data being handled asynchronously and being accumulated into a synchronous log there are numerous use cases that cause an inconsistent number of samples being used in the accumulated output.

**Message ID: 2264**

**Log Type: Synch**

**Recommended Input:**

```
log corrimusb ontime 0.01
```

**Example log:**

```
%CORRIMUSA,2209,488407.000;112,-0.0000242405098038,0.0000287475625144,
-0.0000016866224967,0.0015291995267506,-0.0013533739686903,-0.0018536657202395
,8.168,0*2135fe3d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	CORRIMUS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	–	H	0
2	IMUDataCount	Count of the number of IMU Samples used in each log output accumulation.	Ulong	4	H
3	PitchRate	About x-axis rotation (right-handed) (rad/sample)	Double	8	H+4
4	RollRate	About y-axis rotation (right-handed) (rad/sample)	Double	8	H+12
5	YawRate	About z-axis rotation (right-handed) (rad/sample)	Double	8	H+20
6	LateralAcc	INS Lateral Acceleration (along x-axis) (m/s/sample)	Double	8	H+28
7	LongitudinalAcc	INS Longitudinal Acceleration (along y-axis) (m/s/sample)	Double	8	H+36
8	VerticalAcc	INS Vertical Acceleration (along z-axis) (m/s/sample)	Double	8	H+44
9	Reserved		Float	4	H+52
10	Reserved		Ulong	4	H+56
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+60
12	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

## 5.7 DELAYEDHEAVE

### *Delayed Heave Filter*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log contains the value of the delayed heave filter. The delayed heave value differs from the heave value in that delayed heave uses forward and backward smoothing, while heave uses backward smoothing only. This heave solution is calculated at the location entered in the **SETINSTRANSLATION USER** command.



The **DELAYEDHEAVE** log will not contain valid data until the delayed heave window conditions have been met. You must have an inertial solution to use this log.

**Message ID:** 1709

**Log Type:** Synch

**Recommended Input:**

```
log delayedheaveb ontime 0.1
```

**ASCII example:**

```
#DELAYEDHEAVEA,USB1,0,59.5,FINESTEERING,2209,494152.000,02000020,27a3,16809;
0.000000000,1.000000000*01dbc61c
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	DELAYEDHEAVE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Delayed Heave	Delayed heave value	Double	8	H
3	Std. Dev.	Standard deviation of the delayed heave value	Double	8	H+8
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
5	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.8 GIMBALLEDPVA

### *Display Gimballed Position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

Use the GIMBALLEDPVA log to view the re-calculated position, velocity and attitude of the gimbal null position whenever a new **INPUTGIMBALANGLE** command (see page 1011) is received.



This log should only be requested with the ONCHANGED or ONNEW trigger.



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 1321

**Log Type:** Asynch

**Recommended Input:**

```
log gimballedpvaa onnew
```

**ASCII Example:**

```
#GIMBALLEDPVAA,USB1,0,58.0,FINESTEERING,2209,492397.728,02000020,65c2,16809;
2209,492397.728000000,51.15043715731,-114.03067884053,1080.3608,-0.0013,0.0007,
0.0012,-0.317910813,0.298591270,157.994258109,INS_SOLUTION_GOOD*a8ca2418
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	GIMBALLEDPVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GPS week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal height [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction	Double	8	H+36
8	East Velocity	Velocity in an easterly direction	Double	8	H+44

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
9	Up Velocity	Velocity in an upward direction	Double	8	H+52
10	Roll	Right-handed rotation from local level around the y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around the x-axis in degrees	Double	8	H+68
12	Azimuth	Right-handed rotation from local level around the z-axis in degrees	Double	8	H+76
13	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-



## 5.9 HEAVE

### Heave Filter Log

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log provides vessel heave computed by the integrated heave filter. Refer also to information in the **SETHEAVEWINDOW** command on page 1032. This log is asynchronous, but is available at approximately 10 Hz. This heave solution is calculated at the location entered in the **SETINSTRANSLATION USER** command.



You must have an inertial solution to use this log.

**Message ID:** 1382

**Log Type:** Asynch

**Recommended Input:**

```
log heaveb onnew
```

**Example:**

```
#HEAVEA,USB1,0,65.0,FINESTEERING,2209,488914.441,02000020,0fc8,16809;
2209,488914.440742000,0.016366779*511a7948
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	HEAVE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Heave	Instantaneous heave in metres	Double	8	H+12
5	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+20
6	[CR][LF]	Sentence Terminator (ASCII Only)	-	-	-

## 5.10 IMURATECORRIMUS

### *Asynchronous Corrected IMU Data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the same information as the **CORRIMUS** log (see page 1068), but is available asynchronously at the full rate of the IMU.



Using this log consumes significant system resources and should only be used by experienced users. However, using this log consumes less resources than logging the synchronous CORRIMUDATAS log at the same rate.



To obtain the instantaneous rates of acceleration (in m/s/s) or rotation (in rad/s) from the output values of measurements per sample rate (m/s/sample and rad/sample), multiply the output values by the IMU data rate in Hz.



This log should only be requested with the ONNEW trigger.

**Message ID:** 1362

**Log Type:** Asynch

**Recommended Input:**

```
log imuratecorrimbusb onnew
```

**Example log:**

```
%IMURATECORRIMUSA,2209,489768.026;2209,489768.026275000,-0.000000298,-0.000000690,-0.000001034,-0.000043357,0.000009959,-0.000012101*36ef5bdd
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IMURATECORRIMUS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x axis rotation (rad/sample)	Double	8	H+12
5	RollRate	About y axis rotation (rad/sample)	Double	8	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	YawRate	About z axis rotation (right-handed) (rad/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x-axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y-axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z-axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+60
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.11 IMURATEPVA

### *Asynchronous INS Position, Velocity and Attitude*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the same information as the **INSPVA** log (see page 1101), but is available asynchronously at the full rate of the IMU.



Using this log consumes significant system resources and should only be used by experienced users. However, using this log consumes less resources than logging the synchronous INSPVA log at the same rate.



This log should only be requested with the ONNEW trigger.



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 1778

**Log Type:** Asynch

**Recommended Input:**

```
log imuratepvaa onnew
```

**ASCII Example:**

```
#IMURATEPVAA,USB1,0,62.5,FINESTEERING,2209,489768.026,02000020,81ba,16809;
2209,489768.026275000,51.15043706787,-114.03067862997,1080.3569,-0.0021,0.0003,
-0.0009,-0.314843800,0.320501757,158.167631576,INS_SOLUTION_GOOD*d377f22f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IMURATEPVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.12 IMURATEPVAS

### *Asynchronous INS Position, Velocity and Attitude*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the same information as the **INSPVAS** log (see page 1105), but is available asynchronously at the full rate of the IMU.



Using this log consumes significant system resources and should only be used by experienced users. However, using this log consumes less resources than logging the synchronous INSPVAS log at the same rate.



This log should only be requested with the ONNEW trigger.



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID: 1305**

**Log Type: Asynch**

**Recommended Input:**

```
log imuratepvasa onnew
```

**ASCII Example:**

```
%IMURATEPVASA,2209,489768.026;2209,489768.026275000,51.15043706787,-114.03067862997,1080.3569,-0.0021,0.0003,-0.0009,-0.314843800,0.320501757,158.167631576,INS_SOLUTION_GOOD*bf2d3f18
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	IMURATEPVAS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North	Double	8	H+76
13	Status	INS Status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.13 INSATT

### *INS Attitude*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the most recent attitude measurements computed by the SPAN filter. This attitude definition may not correspond to other definitions of the terms pitch, roll and azimuth. By default, the output attitude is with respect to the vehicle frame. If the attitude output is desired with respect to another frame of reference, use the **SETINSROTATION USER** command (see the **SETINSROTATION** command on page 1041) to configure the user output frame offset rotation.

**Message ID:** 263

**Log Type:** Synch

**Recommended Input:**

```
log insatta ontime 1
```

**ASCII Example:**

```
#INSATTA,USB1,0,65.5,FINESTEERING,2209,489963.000,02000020,c6a7,16809;2209,489963.000000000,-0.325157913,0.303361140,158.066428095,INS_SOLUTION_GOOD*13a0c733
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSATT header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right-handed rotation from local level around y-axis in degrees.	Double	8	H+12
5	Pitch	Right-handed rotation from local level around x-axis in degrees.	Double	8	H+20
6	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North. This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on the next page.	Enum		H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex		H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



Table 237: Inertial Solution Status

Binary	ASCII	Description
0	INS_INACTIVE	IMU logs are present, but the alignment routine has not started; INS is inactive.
1	INS_ALIGNING	INS is in alignment mode.
2	INS_HIGH_VARIANCE	The INS solution uncertainty contains outliers and the solution may be outside specifications. <sup>1</sup> The solution is still valid but you should monitor the solution uncertainty in the <b>INSSTDEV</b> log (see page 1116). It may be encountered during times when GNSS is absent or poor.
3	INS_SOLUTION_GOOD	The INS filter is in navigation mode and the INS solution is good.
6	INS_SOLUTION_FREE	The INS Filter is in navigation mode and the GNSS solution is suspected to be in error. The inertial filter will report this status when there are no available updates (GNSS or other) being accepted and used.
7	INS_ALIGNMENT_COMPLETE	The INS filter is in navigation mode, but not enough vehicle dynamics have been experienced for the system to be within specifications.
8	DETERMINING_ORIENTATION	INS is determining the IMU axis aligned with gravity.
9	WAITING_INITIALPOS	The INS filter has determined the IMU orientation and is awaiting an initial position estimate to begin the alignment process.
10	WAITING_AZIMUTH	The INS filter has orientation, initial biases, initial position and valid roll/pitch estimated. Will not proceed until initial azimuth is entered.
11	INITIALIZING_BIASES	The INS filter is estimating initial biases during the first 10 seconds of stationary data.
12	MOTION_DETECT	The INS filter has not completely aligned, but has detected motion.
14	WAITING_ALIGNMENTORIENTATION	The INS filter is waiting to start alignment until the current Vehicle Frame roll and pitch estimates are within the configured threshold of the expected orientation (set by the <b>SETALIGNMENTORIENTATION</b> command on page 1030). <b>Note:</b> This requires an accurate RBV rotation to be configured, see the <b>SETINSROTATION</b> command on page 1041.

<sup>1</sup>The solution uncertainty threshold levels can be adjusted using the **INSTHRESHOLDS** command on page 1023.

## 5.14 INSATTQS

### Short INS Quaternion Attitude

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the attitude from the INSATT log, but the rotation from local level is given as a Quaternion rather than Euler Angles. The quaternion takes the form:

$$\mathbf{q}_i^b = [w \ x \ y \ z]^T$$

The element  $w$  is the rotational component, defining the magnitude of the rotation to be performed. The elements  $x$ ,  $y$ , and  $z$  are the vector portion of the rotation, which define the axis about which the rotation is to be performed.

If  $\theta$  is the rotational angle, and the axis of rotation is defined by the vector  $\mathbf{v} = [v_x \ v_y \ v_z]^T$ , then the elements of the quaternion can be written as:

$$w = \cos \frac{\theta}{2}$$

$$x = v_x \sin \frac{\theta}{2}$$

$$y = v_y \sin \frac{\theta}{2}$$

$$z = v_z \sin \frac{\theta}{2}$$

**Message ID:** 2118

**Log Type:** Synch

**Recommended Input:**

```
log insattqsa ontime 1
```

**ASCII Example:**

```
%INSATTQSA, 2209, 490002.000; 2209, 490002.000000000, 0.190321570, -0.002193610, -0.003030327, -0.981714676, INS_SOLUTION_GOOD*a67493fa
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSATTQS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	Quaternion w	Quaternion rotation from local level, w component	Double	8	H+12
5	Quaternion x	Quaternion rotation from local level, x component	Double	8	H+20
6	Quaternion y	Quaternion rotation from local level, y component	Double	8	H+28
7	Quaternion z	Quaternion rotation from local level, z component	Double	8	H+36
8	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+44
9	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+48
10	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.15 INSATTS

### Short INS Attitude

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSATT** log (see page 1080).

**Message ID:** 319

**Log Type:** Synch

**Recommended Input:**

```
log insattsa ontime 1
```

**ASCII Example:**

```
%INSATTSA, 2209, 490065.000; 2209, 490065.000000000, -0.316596979, 0.329117050, 158.038475648, INS_SOLUTION_GOOD*85ec8455
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSATTS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+12
5	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+20
6	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.16 INSATTX

### *Inertial Attitude – Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log includes the information from the **INSATT** log (see page 1080), as well as information about the attitude standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.



The INSATTX log is a large log and is not recommend for high rate logging.

If you want to use high rate logging, log the **INSATTS** log at a high rate and the **INSSTDEVS** log ontime 1.

**Message ID:** 1457

**Log Type:** Synch

**Recommended Input:**

```
log insattxa ontime 1
```

**ASCII Example:**

```
#INSATTXA,USB1,0,66.0,FINESTEERING,2209,490101.000,02000020,5d25,16809;  
INS_SOLUTION_GOOD,INS_PPP,-0.312617220,0.320251425,158.010043982,0.0297,  
0.0297,0.0580,13001041,0*6557406d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSATTX header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	INS Status	Solution status See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H
3	Pos Type	Position type See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	Roll	Roll in Local Level (degrees)	Double	8	H+8
5	Pitch	Pitch in Local Level (degrees)	Double	8	H+16
6	Azimuth	Azimuth in Local Level (degrees) This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+24

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	Roll $\sigma$	Roll standard deviation (degrees)	Float	4	H+32
8	Pitch $\sigma$	Pitch standard deviation (degrees)	Float	4	H+36
9	Azimuth $\sigma$	Azimuth standard deviation (degrees)	Float	4	H+40
10	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> below	Hex	4	H+44
11	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+48
12	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 238: Extended Solution Status**

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Position update	0 = Unused 1 = Used
	1	0x00000002	Phase update	0 = Unused 1 = Used
	2	0x00000004	Zero velocity update	0 = Unused 1 = Used
	3	0x00000008	Wheel sensor update	0 = Unused 1 = Used
N1	4	0x00000010	ALIGN (heading) update	0 = Unused 1 = Used
	5	0x00000020	External position update	0 = Unused 1 = Used
	6	0x00000040	INS solution convergence flag	0 = Not converged 1 = Converged
	7	0x00000080	Doppler update	0 = Unused 1 = Used

Nibble	Bit	Mask	Description	Range Value
N2	8	0x00000100	Pseudorange update	0 = Unused 1 = Used
	9	0x00000200	Velocity update	0 = Unused 1 = Used
	10	0x00000400	Reserved	
	11	0x00000800	Dead reckoning update	0 = Unused 1 = Used
N3	12	0x00001000	Phase wind up update	0 = Unused 1 = Used
	13	0x00002000	Course over ground update	0 = Unused 1 = Used
	14	0x00004000	External velocity update	0 = Unused 1 = Used
	15	0x00008000	External attitude update	0 = Unused 1 = Used
N4	16	0x00010000	External heading update	0 = Unused 1 = Used
	17	0x00020000	External height update	0 = Unused 1 = Used
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	
N5	20	0x00100000	Reserved	
	21	0x00200000	Reserved	
	22	0x00400000	Reserved	
	23	0x00800000	Reserved	
N6	24	0x01000000	Turn on biases estimated	0 = Static turn-on biases not estimated (starting from zero) 1 = Static turn-on biases estimated
	25	0x02000000	Alignment direction verified	0 = Not verified 1 = Verified
	26	0x04000000	Alignment Indication 1	0 = Not set, 1 = Set Refer to <i>Table 239: Alignment Indication</i> below
	27	0x08000000	Alignment Indication 2	0 = Not set, 1 = Set Refer to <i>Table 239: Alignment Indication</i> below

Nibble	Bit	Mask	Description	Range Value
N7	28	0x10000000	Alignment Indication 3	0 = Not set, 1 = Set Refer to <i>Table 239: Alignment Indication</i> below
	29	0x20000000	NVM Seed Indication 1	0 = Not set, 1 = Set Refer to <i>Table 240: NVM Seed Indication</i> on the next page
	30	0x40000000	NVM Seed Indication 2	0 = Not set, 1 = Set Refer to <i>Table 240: NVM Seed Indication</i> on the next page
	31	0x80000000	NVM Seed Indication 3	0 = Not set, 1 = Set Refer to <i>Table 240: NVM Seed Indication</i> on the next page

Table 239: Alignment Indication

Bits 28-26 Values <sup>1</sup>	Hex Value	Completed Alignment Type
000	0x00	Incomplete Alignment
001	0x01	Static
010	0x02	Kinematic
011	0x03	Dual Antenna
100	0x04	User Command
101	0x05	NVM Seed

Table 240: NVM Seed Indication

Bit 31-29 Values	Hex Value	NVM Seed Type
000	0x00	NVM Seed Inactive
001	0x01	Seed stored in NVM is invalid
010	0x02	NVM Seed failed validation check
011	0x03	NVM Seed is pending validation (awaiting GNSS)
100	0x04	NVM Seed Injected (includes error model data)
101	0x05	NVM Seed data ignored due to a user-commanded filter reset or configuration change
110	0x06	NVM Seed error model data injected

<sup>1</sup>The values from left to right are Bit 28, Bit 27 and Bit 26.

<sup>2</sup>The values from left to right are Bit 31, Bit 30 and Bit 29.



## 5.17 INSCALSTATUS

### Offset calibration status

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log reports the status and estimated values of the currently running offset calibration.

**Message ID:** 1961

**Log Type:** Asynch

**Abbreviated ASCII Syntax:**

```
log inscalstatus onchanged
```

**ASCII Example:**

```
#INSCALSTATUSA,USB1,0,51.5,FINESTEERING,2209,494041.015,02010020,c0db,16809;  
RBV,0.0000,0.0000,-0.0000,45.0000,45.0000,45.0000,INSUFFICIENT_SPEED,0*62a2839c
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	INSCALSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Offset Type	Type of offset (see <i>Table 241: Offset Type</i> on the next page).	Enum	4	H
3	X axis offset	IMU body frame X-axis offset (metres or degrees)	Float	4	H+4
4	Y axis offset	IMU body frame Y-axis offset (metres or degrees)	Float	4	H+8
5	Z axis offset	IMU body frame Z-axis offset (metres or degrees)	Float	4	H+12
6	X uncertainty	IMU body frame X-axis offset uncertainty (metres or degrees)	Float	4	H+16
7	Y uncertainty	IMU body frame Y-axis offset uncertainty (metres or degrees)	Float	4	H+20
8	Z uncertainty	IMU body frame Z-axis offset uncertainty (metres or degrees)	Float	4	H+24
9	Source Status	Source from which offset values originate (see <i>Table 242: Source Status</i> on the next page).	Enum	4	H+28

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
10	Multi-line Calibration Count	Counter for number of completed calibrations cumulatively averaged.	Ulong	4	H+32
11	xxxx	32-bit CRC (ASCII and Binary only).	Hex	4	H+36
12	[CR][LF]	Sentence terminator (ASCII only).	-	-	-



Units for the axis offset and uncertainty values (fields 3-8) are in metres for translational offset components and degrees for rotational offset components.

Table 241: Offset Type

Binary	ASCII	Description
1	ANT1	Primary IMU to antenna lever arm
8	ALIGN	Align offset
11	RBV	IMU body to vehicle offset

Table 242: Source Status

Binary	ASCII	Description
1	FROM_NVM	Offset values originate from saved parameters in NVM
2	CALIBRATING	Offset values originate from a currently running calibration process
3	CALIBRATED	Offset values originate from a completed calibration process
4	FROM_COMMAND	Offset values originate from a user command
5	Reserved	
6	FROM_DUAL_ANT	Offset values originate from a dual antenna Align solution
7	INS_CONVERGING	Offset values originate from initial input values. Calibration process on hold until INS solution is converged.
8	INSUFFICIENT_SPEED	Offset values originate from a currently running calibration process. Further estimation on hold due to insufficient speed.
9	HIGH_ROTATION	Offset values originate from a currently running calibration process. Further estimation on hold due to high vehicle rotations.

## 5.18 INSCONFIG

### *Determine required settings for post-processing or system analysis*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the single message required to determine all required settings for post-processing or system analysis. This log is asynchronous and published for any change to the included fields. It is intended to be recorded occasionally though it could be updated frequently at system startup. The **INSCONFIG** log may also be logged synchronously with the ONTIME trigger, but this log should **NOT** be requested at a high rate (limited to 1 Hz).

**Message ID:** 1945


**Log Type:** Asynch

**Recommended Input:**

```
log insconfig onchanged
```

**ASCII Example:**

```
#INSCONFIGA,USB1,0,66.5,FINESTEERING,2209,338183.000,02000020,bba4,16809;
EPSON_G320,5,50,20,DEFAULT,7fd1bf,AUTOMATIC,ROVER,FALSE,0,0,0,17,0,0,0,0,0,0,0,
1,RBV,IMUBODY,0.0000,0.0000,0.0000,3.0000,3.0000,3.0000,FROM_COMMAND*996d7655
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	INSCONFIG header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	IMU Type	IMU type. See <i>Table 230: IMU Type</i> on page 1002.	Enum	4	H
3	Mapping	Mapping / Orientation	Uchar	1	H+4
4	Initial Alignment Velocity	Minimum Alignment Velocity entered by the user (see the <b>SETALIGNMENTVEL</b> command on page 1031).  Velocity (m/s) is scaled by 10 for 10 cm/s precision	Uchar	1	H+5
5	Heave Window	Length of the heave window in seconds (if set using the <b>SETHEAWEWINDOW</b> command on page 1032)	Ushort	2	H+6
6	Profile	Profile setting (see the <b>SETINSPROFILE</b> command on page 1039)	Enum	4	H+8

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
7	INS Enabled Updates	Enabled INS updates (see <i>Table 247: INS Update Values</i> on page 1122)	Hex	4	H+12
8	Alignment Mode	Alignment mode configured on the system (see the <b>ALIGNMENTMODE</b> command on page 999)	Enum	4	H+16
9	Relative INS Output Frame	The user specified output frame of the Relative INS Vector (see <b>SETRELINSOUTPUTFRAME</b> command on page 1051) If not specified, the default value appears.	Enum	4	H+20
10	Relative INS Output Direction	The User specified Output direction of the Relative INS Vector (From or To Master-Rover) (see the <b>SETRELINSOUTPUTFRAME</b> command on page 1051). If not specified, the default value appears. TRUE if From Master, FALSE (Default) if From Rover	Bool	4	H+24
11	INS Receiver Status	Lower byte- INS Reset. Corresponds numerically to the INS Reset as described by the <b>INSResetEnum</b> Second byte- = 0x01 if an IMU Communication Error (Receiver status bit 17). = 0x00 otherwise. Other values are reserved for future use. Upper 2 bytes - reserved.	Hex	4	H+28
12	INS Seed Enabled	INS Seed Enable setting (see the <b>INSSEED</b> command on page 1021) Enabled = 1, Disabled = 0	Uchar	1	H+32
13	INS Seed Validation	INS Seed Validation setting (see the <b>INSSEED</b> command on page 1021)	Uchar	1	H+33
14	Reserved 1		N/A	2	H+34
15	Reserved 2		N/A	4	H+36
16	Reserved 3		N/A	4	H+40
17	Reserved 4		N/A	4	H+44

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
18	Reserved 5		N/A	4	H+48
19	Reserved 6		N/A	4	H+52
20	Reserved 7		N/A	4	H+56
21	Number of Translations	Number of translation entries to follow	Ulong	4	H+60
22	Translation	Translation to follow (see <i>Table 235: Translation Offset Types</i> on page 1046)	Enum	4	variable
23	Frame	Frame of translation (IMUBODY or VEHICLE)	Enum	4	variable
24	X Offset	X Offset	Float	4	variable
25	Y Offset	Y Offset	Float	4	variable
26	Z Offset	Z Offset	Float	4	variable
27	X Uncertainty	X Uncertainty	Float	4	variable
28	Y Uncertainty	Y Uncertainty	Float	4	variable
29	Z Uncertainty	Z Uncertainty	Float	4	variable
30	Translation Source	Source of translation (see <i>Table 242: Source Status</i> on page 1091)	Enum	4	variable
Next Translation					
variable	Number of Rotations	Number of rotation entries to follow	Ulong	4	variable
variable	Rotation	Rotation to follow (see <i>Table 234: Rotational Offset Types</i> on page 1043)	Enum	4	variable
variable	Frame	Frame of rotation (IMUBODY or VEHICLE)	Enum	4	variable
variable	X Rotation	X Rotation	Float	4	variable
variable	Y Rotation	Y Rotation	Float	4	variable
variable	Z Rotation	Z Rotation	Float	4	variable
variable	X Rotation Std Dev	X Rotation offset standard deviation (degrees)	Float	4	variable
variable	Y Rotation STD Dev	Y Rotation offset standard deviation (degrees)	Float	4	variable
variable	Z Rotation STD Dev	Z Rotation offset standard deviation (degrees)	Float	4	variable

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
variable	Rotation Source	Source of rotation (see <i>Table 242: Source Status</i> on page 1091)	Enum	4	variable
	Next Rotation				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.19 INSDATUMINFO

### *Datum information for INS position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7, SMART2

This log reports the datum and co-ordinate epoch of the solution for all INS logs with position output (**INSPVA**, **INSPVAX**, **INSPVACMP**, **INSPOS**, **INSPOSX**, **MARKxPVA** and **TAGGEDMARKxPVA**).

**Message ID:** 2384

**Log Type:** Asynch

**Recommended Input:**

```
log insdatuminfoa onchanged
```

**ASCII Example:**

```
#INSDATUMINFOA,USB1,0,66.0,FINESTEERING,2209,490200.000,02000020,f6f5,16809;
"ITRF2014",1165,2022.363,GOOD,0.000000000,0*21d0a220
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSDATUMINFO header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	–	H	0
2	datum	Datum of the position being output by the INS position logs.	Char[32]	variable	H
3	epsg_code	EPSG code of the datum.	Ulong	4	variable
4	epoch	Co-ordinate epoch (decimal year). Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	variable
5	status	Transformation status. See <i>Table 85: Transformation Status</i> on page 518.	Enum	4	variable
6	reserved	Reserved for future use	Float	4	variable
7	reserved	Reserved for future use	Ulong	4	variable
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
9	[CR][LF]	Sentence terminator (ASCII only)	–	–	–

## 5.20 INSPOS

### *INS Position*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the most recent position measurements in WGS84 coordinates and includes an INS status indicator. The log reports the position at the IMU center, unless the **SETINSTRLANSATION USER** command was issued. See the **SETINSTRLANSATION** command on page 1044.



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 265

**Log Type:** Synch

**Recommended Input:**

```
log insposa ontime 1
```

**ASCII Example:**

```
#INSPOSA,USB1,0,66.5,FINESTEERING,2209,490404.000,02000020,5b52,16809;2209,
490404.000000000,51.15043709813,-114.03067863568,1080.3587,INS_SOLUTION_GOOD
*18ee6937
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPOS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude[degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 5.21 INSPOSS

### Short INS Position

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSPOS** log (see page 1097).



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 321

**Log Type:** Synch

**Recommended Input:**

```
log inspossa ontime 1
```

**ASCII Example:**

```
%INSPOSSA,2209,490447.000;2209,490447.000000000,51.15043708257,-114.03067868674,1080.3587,INS_SOLUTION_GOOD*94c38497
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPOSS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.22 INSPOSX

### *Inertial Position – Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log includes the information from the INSPOS log, as well as information about the position standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.



The INSPOSX log is a large log and is not recommend for high rate logging.

If you want to use high rate logging, log the **INSPOSS** log at a high rate and the **INSSTDEVS** log ontime 1.



This log provides the position information in the user datum.

To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 1459

**Log Type:** Synch

**Recommended Input:**

```
log insposxa ontime 1
```

**ASCII example:**

```
#INSPOXAX,USB1,0,67.5,FINESTEERING,2209,490485.000,02000020,7e79,16809;  
INS_SOLUTION_GOOD,INS_PPP,51.15043706246,-114.03067867866,1097.3569,  
-17.0001,0.1819,0.1819,0.1812,13000045,0*2b90352c
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPOSX header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	INS Status	Solution status See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H
3	Pos Type	Position type See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	Lat	Latitude	Double	8	H+8
5	Long	Longitude	Double	8	H+16

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	Height	Height above sea level (m)	Double	8	H+24
7	Undulation	Undulation (m)	Float	4	H+32
8	Lat $\sigma$	Latitude standard deviation	Float	4	H+36
9	Long $\sigma$	Longitude standard deviation	Float	4	H+34
10	Height $\sigma$	Height standard deviation	Float	4	H+44
11	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+48
12	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+52
13	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+54
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 87: Position or Velocity Type* on page 524), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see **BESTGNSSPOS** log on page 1060).

## 5.23 INSPVA

### *INS Position, Velocity and Attitude*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log allows INS position, velocity and attitude, with respect to the SPAN frame, to be collected in one log, instead of using three separate logs. Refer to the **INSATT** log (see page 1080) for an explanation of how the SPAN frame may differ from the IMU enclosure frame.



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 507

**Log Type:** Synch

**Recommended Input:**

```
log inspvaa ontime 1
```

**ASCII Example:**

```
#INSPVAA,USB1,0,67.5,FINESTEERING,2209,490558.000,02000020,18bc,16809;2209,490558.000000000,51.15043714042,-114.03067871718,1080.3548,0.0051,-0.0014,-0.0012,-0.296402993,0.311887972,157.992156267,INS_SOLUTION_GOOD*cc698020
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.24 INSPVACMP

### *Compressed version of the INSPVA log for CAN*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the **INSPVA** log (see page 1101) information compressed as a NMEA 2000 fast packet message for CAN output. The PGN of this message is configured using the **PGNCONFIG** command on page 290. The information in the table below is the data portion of the INSPVACMP NMEA 2000 message. Refer to the *NATIONAL MARINE ELECTRONICS ASSOCIATION (NMEA) NMEA2000(r) rev 2.0 - January/2013* document for full details on this message output and the NMEA 2000 fast packet protocol.

**Message ID: 1889**

**Log Type: Synch**

**Abbreviated ASCII Syntax:**

```
LOG CCOM1 INSPVACMP ONTIME 1
```

Field	Field Type	Description	Format	Binary Bytes	Units	Scale Factor
1	Message Time	Seconds from week start	Ulong	4	seconds	0.001
2	INS Status	INS solution status (see <i>Table 237: Inertial Solution Status</i> on page 1081)	UChar	1	N/A	N/A
3	GNSS Position Type	GNSS position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	UChar	1	N/A	N/A
4	Latitude	Latitude (WGS84)	5 byte Long	5	degrees	180/2 <sup>39</sup>
5	Longitude	Longitude (WGS84)	5 byte Long	5	degrees	180/2 <sup>39</sup>
6	Height	Ellipsoidal Height (WGS84)	Long	4	metres	0.0001
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction)	Short	2	m/s	0.002
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction)	Short	2	m/s	0.002
9	Up Velocity	Velocity in an up direction	Short	2	m/s	0.002
10	Roll	Right-handed rotation from local level around y axis	Short	2	degrees	0.01

Field	Field Type	Description	Format	Binary Bytes	Units	Scale Factor
11	Pitch	Right-handed rotation from local level around x axis	Short	2	degrees	0.01
12	Azimuth	Left-handed rotation around z-axis clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Ushort	2	degrees	0.01
13	Azimuth Rate	Rate of change of the Azimuth	Short	2	degrees/s	0.01

## 5.25 INSPVAS

### Short INS Position, Velocity and Attitude

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSPVA** log (see page 1101).



This log provides the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 508

**Log Type:** Synch

**Recommended Input:**

```
log inspvasa ontime 1
```

**ASCII Example:**

```
%INSPVASA,2209,490712.000;2209,490712.000000000,51.15043713712,-114.03067872932,1080.3580,0.0014,-0.0007,0.0015,-0.293782412,0.302470613,157.989827209,INS_SOLUTION_GOOD*a87ab88b
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPVAS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude [degrees]	Double	8	H+12
5	Longitude	Longitude [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from north This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.26 INSPVASDCMP

### *Standard deviation information for the INSPVACMP CAN message*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the standard deviation information for the **INSPVACMP** log (see page 1103) information compressed in a NMEA 2000 fast packet message for CAN output. The PGN of this message is configured using the **PGNCONFIG** command on page 290. The information in the table below is the data portion of the INSPVASDCMP NMEA 2000 message. Refer to the *NATIONAL MARINE ELECTRONICS ASSOCIATION (NMEA) NMEA2000(r) rev 2.0 - January/2013* document for full details on this message output and the NMEA 2000 fast packet protocol.

**Message ID: 1890**

**LogType: Synch**

**Abbreviated ASCII Syntax:**

```
LOG CCOM1 INSPVASDCMP ONTIME 1
```

Field	Field Type	Description	Format	Binary Bytes	Units	Scale Format
1	Week	GNSS Week	Ushort	2	weeks	1
2	Message Time	Seconds from week start	Ulong	4	seconds	0.001
3	Sigma Latitude	Latitude Standard Deviation	Ushort	2	metres	0.001
4	Sigma Longitude	Longitude Standard Deviation	Ushort	2	metres	0.001
5	Sigma Height	Height Standard Deviation	Ushort	2	metres	0.001
6	Sigma North Velocity	North Velocity Standard Deviation	Ushort	2	m/s	0.001
7	Sigma East Velocity	East Velocity Standard Deviation	Ushort	2	m/s	0.001
8	Sigma Up Velocity	Up Velocity Standard Deviation	Ushort	2	m/s	0.001
9	Sigma Roll	Roll Standard Deviation	Ushort	2	degrees	0.01
10	Sigma Pitch	Pitch Standard Deviation	Ushort	2	degrees	0.01
11	Sigma Azimuth	Azimuth Standard Deviation	Ushort	2	degrees	0.01
12	Update Counter	Elapsed time since the last ZUPT or position update	Uchar	1	seconds	1

Field	Field Type	Description	Format	Binary Bytes	Units	Scale Format
13	Position Type	GNSS position type (see <i>Table 87: Position or Velocity Type</i> on page 524)	Uchar	1	N/A	N/A
14	Extended Solution Status	Extended Solution Status (see <i>Table 238: Extended Solution Status</i> on page 1086)	Hex	4	N/A	N/A
15	ALIGN Age	Elapsed time since a valid ROVERPOS solution was available	Uchar	1	seconds	1

## 5.27 INSPVAX

### *Inertial PVA – Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log includes the information from the INSPVA log, as well as information about the position standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.



The INSPVAX log is a large log and is not recommend for high rate logging.  
If you want to use high rate logging, log the **INSPVAS** log (see page 1105) at a high rate and the **INSSTDEV** log on page 1116 ontime 1.



This log provides the position information in the user datum.  
To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 1465

**Log Type:** Synch

**Recommended Input:**

```
log inspvaxa ontime 1
```

**ASCII example:**

```
#INSPVAXA,USB1,0,65.5,FINESTEERING,2209,490782.000,02000020,46eb,16809;  
INS_SOLUTION_GOOD,INS_PPP,51.15043710672,-114.03067871892,1097.3598,-17.0001,  
-0.0016,0.0002,0.0029,-0.308665944,0.297893298,157.960833016,0.1816,0.1816,  
0.1808,0.0018,0.0018,0.0016,0.0292,0.0292,0.0582,13000045,0*a74644a2
```

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log header	INSPVAX header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	INS Status	Solution status See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H
3	Pos Type	Position type See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	Lat	Latitude (degrees)	Double	8	H+8

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
5	Long	Longitude (degrees)	Double	8	H+16
6	Height	Height above mean sea level (m)	Double	8	H+24
7	Undulation	Undulation (m)	Float	4	H+32
8	North Vel	North velocity (m/s)	Double	8	H+36
9	East Vel	East velocity (m/s)	Double	8	H+44
10	Up Vel	Up velocity (m/s)	Double	8	H+52
11	Roll	Roll in Local Level (degrees)	Double	8	H+60
12	Pitch	Pitch in Local Level (degrees)	Double	8	H+68
13	Azimuth	Azimuth in Local Level (degrees) This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
14	Lat $\sigma$	Latitude standard deviation (m)	Float	4	H+84
15	Long $\sigma$	Longitude standard deviation (m)	Float	4	H+88
16	Height $\sigma$	Height standard deviation (m)	Float	4	H+92
17	North Vel $\sigma$	North velocity standard deviation (m/s)	Float	4	H+96
18	East Vel $\sigma$	East velocity standard deviation (m/s)	Float	4	H+100
19	Up Vel $\sigma$	Up velocity standard deviation (m/s)	Float	4	H+104
20	Roll $\sigma$	Roll standard deviation (degrees)	Float	4	H+108
21	Pitch $\sigma$	Pitch standard deviation (degrees)	Float	4	H+112
22	Azimuth $\sigma$	Azimuth standard deviation (degrees)	Float	4	H+116
23	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+120
24	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+124
25	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+126
26	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 87: Position or Velocity Type* on page 524), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see **BESTGNSSPOS** log on page 1060).

## 5.28 INSSEEDSTATUS

### Status of INS Seed

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log reports the current status of the INS Seed. See INS Seed / Fast INS Initialization in the [OEM7 SPAN Installation and Operation User Manual](#), [CPT7 Installation and Operation User Manual](#) or [SMART7 Installation and Operation User Manual](#) for more information about an INS Seed.

**Message ID:** 2129

**Log Type:** Asynch

**Abbreviated ASCII Syntax:**

```
log insseedstatusa onnew
```

**Example:**

```
#INSSEEDSTATUSA,USB1,0,65.5,FINESTEERING,2209,493827.000,02000020,0d1f,16809;
IGNORED,ALLVALID,0.3286,0.0293,352.9208,-1632848.2546,-3662158.6358,
4944901.2915,-17.0001,0,0,0,0*db3c246f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSSEEDSTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Injection Status	Status of the INS Seed being injected into the solution. See <i>Table 243: Injection Status</i> on the next page	Enum	4	H
3	Validity Status	Flag to indicate if current seed data in NVM is valid. See <i>Table 244: Validity Status</i> on the next page	Bool	4	H+4
4	Pitch	IMU frame pitch angle (degrees)	Float	4	H+8
5	Roll	IMU frame roll angle (degrees)	Float	4	H+12
6	Azimuth	IMU frame azimuth angle (degrees)	Float	4	H+16
7	PositionX	ECEF-based x-coordinate	Double	8	H+20
8	PositionY	ECEF-based y-coordinate	Double	8	H+28
9	PositionZ	ECEF-based z-coordinate	Double	8	H+36
10	Undulation	Geoid undulation	Float	4	H+44
11	Reserved		Ulong	4	H+48

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
12	Reserved		Ulong	4	H+52
13	Reserved		Ulong	4	H+56
14	Reserved		Ulong	4	H+60
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+64
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

**Table 243: Injection Status**

Binary	ASCII	Description
0	NOT_INITIALIZED	INS Seed has not been injected into the solution
1	INVALID	Valid INS Seed was not found in non-volatile memory
2	FAILED	INS Seed has failed validation and has been discarded
3	PENDING	INS Seed is awaiting validation
4	INJECTED	INS Seed alignment data has successfully been injected (including error model data)
5	IGNORED	INS Seed was pending, but has been ignored due to a user commanded filter reset or configuration change
6	ERRORMODELINJECTED	INS Seed error model data has successfully been injected

**Table 244: Validity Status**

Binary	ASCII	Description
0	INVALID	INS Seed in NVM is not valid
1	ALLVALID	INS Seed in NVM is valid
2	ERRORMODELVALID	INS Seed error model in NVM is valid (alignment data is not valid)



## 5.29 INSSPD

### *INS Speed*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the most recent speed measurements in the horizontal and vertical directions and includes an INS status indicator.

**Message ID:** 266

**Log Type:** Synch

**Recommended Input:**

```
log insspda ontime 1
```

**ASCII Example:**

```
#INSSPDA,USB1,0,65.5,FINESTEERING,2209,490925.000,02000020,0a39,16809;2209,490925.000000000,247.102818,0.0021,-0.0007,INS_SOLUTION_GOOD*714af7ba
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSSPD header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees Track over ground is best used when the vehicle is moving. When the vehicle is stationary, position error can make the direction of motion appear to change randomly.	Double	8	H+12
5	Horizontal Speed	Magnitude of horizontal speed in m/s.	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in m/s where a positive value indicates speed upward and a negative value indicates speed downward.	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.30 INSSPDS

### Short INS Speed

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSSPD** log (see page 1114).

**Message ID:** 323

**Log Type:** Synch

**Recommended Input:**

```
log insspdsa ontime 1
```

**ASCII Example:**

```
%INSSPDSA,2209,490957.000;2209,490957.000000000,354.197009,0.0045,0.0028,
INS_SOLUTION_GOOD*253f4d87
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSSPDS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees. Track over ground is best used when the vehicle is moving. When the vehicle is stationary, position error can make the direction of motion appear to change randomly.	Double	8	H+12
5	Horizontal Speed	Magnitude of horizontal speed in m/s.	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in m/s where a positive value indicates speed upward and a negative value indicates speed downward.	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.31 INSSTDEV

### *INS PVA standard deviations*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log displays the INS PVA standard deviations.

**Message ID:** 2051

**Log Type:** Synch

**Abbreviated ASCII Syntax:**

```
log insstdev ontime 1
```

**ASCII Example:**

```
#INSSTDEVA,USB1,0,66.0,FINESTEERING,2209,491004.000,02000020,2396,16809;0.1813,
0.1813,0.1806,0.0018,0.0018,0.0017,0.0292,0.0291,0.0577,13000045,0,0,7fd1bf,0
*b490ddee
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	INSSTDEV header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Latitude $\sigma$	Latitude standard deviation (m)	Float	4	H
3	Longitude $\sigma$	Longitude standard deviation (m)	Float	4	H+4
4	Height $\sigma$	Height standard deviation (m)	Float	4	H+8
5	North Velocity $\sigma$	North velocity standard deviation (m/s)	Float	4	H+12
6	East Velocity $\sigma$	East velocity standard deviation (m/s)	Float	4	H+16
7	Up Velocity $\sigma$	Up velocity standard deviation (m/s)	Float	4	H+20
8	Roll $\sigma$	Roll standard deviation (degrees)	Float	4	H+24
9	Pitch $\sigma$	Pitch standard deviation (degrees)	Float	4	H+28
10	Azimuth $\sigma$	Azimuth standard deviation (degrees)	Float	4	H+32
11	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Ulong	4	H+36

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
12	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+40
13	Reserved		Ushort	2	H+42
14	Reserved		Ulong	4	H+44
15	Reserved		Ulong	4	H+48
16	xxxx	32-bit CRC (ASCII and Binary only).	Hex	4	H+52
17	[CR][LF]	Sentence terminator (ASCII only).	-	-	-



The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 87: Position or Velocity Type* on page 524), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see **BESTGNSSPOS** log on page 1060).

## 5.32 INSSTDEVS

### Short INS PVA standard deviations

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSSTDEV** log (see page 1116).

**Message ID:** 2052

**Log Type:** Synch

**Abbreviated ASCII Syntax:**

```
log insstdevs ontime 1
```

**ASCII Example:**

```
%INSSTDEVSA,2209,491032.000;0.1812,0.1812,0.1805,0.0018,0.0018,0.0017,0.0291,0.0291,0.0575,13000045,0,0,7fd1bf,0*b6d40807
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Log header	INSSTDEVS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Latitude $\sigma$	Latitude standard deviation (m)	Float	4	H
3	Longitude $\sigma$	Longitude standard deviation (m)	Float	4	H+4
4	Height $\sigma$	Height standard deviation (m)	Float	4	H+8
5	North Velocity $\sigma$	North velocity standard deviation (m/s)	Float	4	H+12
6	East Velocity $\sigma$	East velocity standard deviation (m/s)	Float	4	H+16
7	Up Velocity $\sigma$	Up velocity standard deviation (m/s)	Float	4	H+20
8	Roll $\sigma$	Roll standard deviation (degrees)	Float	4	H+24
9	Pitch $\sigma$	Pitch standard deviation (degrees)	Float	4	H+28
10	Azimuth $\sigma$	Azimuth standard deviation (degrees)	Float	4	H+32
11	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Ulong	4	H+36
12	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+40

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
13	Reserved		Ushort	2	H+42
14	Reserved		Ulong	4	H+44
15	Reserved		Ulong	4	H+48
16	xxxx	32-bit CRC (ASCII and Binary only).	Hex	4	H+52
17	[CR][LF]	Sentence terminator (ASCII only).	-	-	-



The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 87: Position or Velocity Type* on page 524), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see **BESTGNSSPOS** log on page 1060).

## 5.33 INSUPDATESTATUS

### *INS Update Status*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the most recent INS update information used by the system. It provides information about what updates were performed in the INS filter at the specified time.

Multiple **INSUPDATESTATUS** logs may be published each epoch. For example, any DMI or external updates will trigger an independent, asynchronous log. These asynchronous logs may show that other updates are not being used because they are the only valid update applied at the specified time.

In the ASCII example below, the first INSUPDATESTATUS log shows 22 phase and 24 doppler updates were applied at 416218.000 and the second INSUPDATESTATUS log shows the DMI update was USED at 416218.002.

**Message ID:** 1825

**Log Type:** Asynch

**Recommended Input:**

```
log insupdatestatus onchanged
```

**ASCII Example:**

```
#INSUPDATESTATUSA,USB1,0,66.0,FINESTEERING,2209,491101.000,02000020,78f1,16809;
INS_PPP,0,0,0,INACTIVE,INACTIVE,13000045,007fd1bf,0,0*9ed62c60
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSUPDATESTATUS header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	PosType	Type of GNSS solution used for the last INS filter update. See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H
3	NumPSR	Number of raw pseudorange observations used in the last INS filter update.	Integer	4	H+4
4	NumADR	Number of raw phase observations used in the last INS filter update.	Integer	4	H+8
5	NumDOP	Number of raw doppler observations used in the last INS filter update.	Integer	4	H+12
6	DMI Update Status	Distance measurement instrument (wheel sensor) status. See <i>Table 245: DMI Update Status</i> below.	Enum	4	H+16

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	ALIGN Update Status	Status of the ALIGN update during the last INS filter update. See <i>Table 246: ALIGN Update Values</i> below	Enum	4	H+20
8	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+24
9	INS Enabled Updates	Enabled INS updates. See <i>Table 247: INS Update Values</i> on the next page	Hex	4	H+28
10	Reserved		Ulong	4	H+32
11	Reserved		Ulong	4	H+36
12	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 245: DMI Update Status**

Binary	ASCII	Description
0	INACTIVE	The DMI sensor is not connected.
1	ACTIVE	The DMI sensor is connected but not used.
2	USED	The DMI sensor is connected and used in the INS solution.
3	RESERVED	Reserved
4	BAD_MISC	The measurement of the DMI sensor resulted in bad misclosure and was rejected.
5	HIGH_ROTATION	The measurement of the DMI sensor was not used because the system is undergoing high rotation.

**Table 246: ALIGN Update Values**

Binary	ASCII	Description
0	INACTIVE	An ALIGN update was not available.
1	ACTIVE	ALIGN updates are running, but the epoch is not used as an update. When all other rejection criteria pass, an ALIGN update will still only be applied once every 5 seconds (20 seconds when stationary).
2	USED	The update for that epoch was taken.
5	BAD_MISC	ALIGN updates are running, but was not performed this epoch due to a large disagreement with filter estimates.



Table 247: INS Update Values

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Position update	0 = Disabled 1 = Enabled
	1	0x00000002	Phase update	0 = Disabled 1 = Enabled
	2	0x00000004	Zero velocity update	0 = Disabled 1 = Enabled
	3	0x00000008	Wheel sensor update	0 = Disabled 1 = Enabled
N1	4	0x00000010	ALIGN update	0 = Disabled 1 = Enabled
	5	0x00000020	External position update	0 = Disabled 1 = Enabled
	6	0x00000040	Reserved	
	7	0x00000080	Doppler update	0 = Disabled 1 = Enabled
N2	8	0x00000100	Pseudorange update	0 = Disabled 1 = Enabled
	9	0x00000200	Velocity update	0 = Disabled 1 = Enabled
	10	0x00000400	Reserved	
	11	0x00000800	Dead reckoning update	0 = Disabled 1 = Enabled
N3	12	0x00001000	Phase wind up update	0 = Disabled 1 = Enabled
	13	0x00002000	Course over ground update	0 = Disabled 1 = Enabled
	14	0x00004000	External velocity update	0 = Disabled 1 = Enabled
	15	0x00008000	External attitude update	0 = Disabled 1 = Enabled
N4	16	0x00010000	External heading update	0 = Disabled 1 = Enabled
	17	0x00020000	External height update	0 = Disabled 1 = Enabled
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	

## 5.34 INSVEL

### *INS Velocity*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains the most recent North, East and Up velocity vector values, with respect to the local level frame and also includes an INS status indicator.

**Message ID:** 267

**Log Type:** Synch

**Recommended Input:**

```
log insvela ontime 1
```

**ASCII Example:**

```
#INSVELA,USB1,0,67.0,FINESTEERING,2209,491133.000,02000020,29aa,16809;2209,491133.000000000,-0.0035,0.0010,0.0004,INS_SOLUTION_GOOD*73237b3a
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSVEL header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North in m/s	Double	8	H+12
5	East Velocity	Velocity East in m/s	Double	8	H+20
6	Up Velocity	Velocity Up in m/s	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.35 INSVELS

### Short INS Velocity

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **INSVEL** log (see page 1124).

**Message ID:** 324

**Log Type:** Synch

**Recommended Input:**

```
log insvelsa ontime 1
```

**ASCII Example:**

```
%INSVELSA,2209,491166.000;2209,491166.000000000,-0.0033,-0.0004,-0.0003,
INS_SOLUTION_GOOD*24448c97
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSVELS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North m/s	Double	8	H+12
5	East Velocity	Velocity East m/s	Double	8	H+20
6	Up Velocity	Velocity Up m/s	Double	8	H+28
7	Status	INS status, see <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.36 INSVELUSER

### *INS velocity in user output frame*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log outputs INS Velocity information in the user defined output frame as per the **SETINSROTATION USER** command. The default, when no USER rotation is defined, is the vehicle frame assuming a valid RBV rotation has been entered.



The velocity information will be translated to the reference point defined by the **SETINSTRANSULATION USER** command.



The *Slip Angle Validity* flag provides an indication of the confidence in the slip angle calculation based on the solution quality. Included in the quality indicators for a valid slip angle is a check for minimum vehicle speed. The user should consider the velocity standard deviations when using both the velocity and slip angle values.



The range of the Slip Angle is -180 to 180 degrees. When the Slip Angle is positive, the vehicle is slipping to the driver's right (starboard) assuming the user frame Y axis coincides with the vehicle forward direction of travel. When the Slip Angle is negative, the vehicle is slipping to the driver's left (port). This corresponds to the sign of the velocity in the user frame X axis.

**Message ID: 2318**

**Log Type: Synch**

**Recommended Input:**

```
LOG INSVELUSER ONTIME 1
```

**Abbreviated ASCII Example:**

```
#INSVELUSERA,USB1,0,66.5,FINESTEERING,2209,491218.000,02000020,9ac8,16809;
INS_SOLUTION_GOOD,-0.0015,-0.0008,-0.0007,0.0018,0.0018,0.0016,-119.785,0,0,
00000000*89b473ba
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSVELUSER header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	INS Status	Solution status See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H
3	X-Axis Velocity	Velocity in the Output Frame X-Axis (Across Track for Vehicle)	Double	8	H+4
4	Y-Axis Velocity	Velocity in the Output Frame Y-Axis (Along Track for Vehicle)	Double	8	H+12
5	Z-Axis Velocity	Velocity in the Output Frame Z-Axis (Vertical for Vehicle)	Double	8	H+20
6	X-Axis Velocity StDev	Velocity Standard Deviation in the Output Frame X-Axis (Across Track for Vehicle)	Float	4	H+28
7	Y-Axis Velocity StDev	Velocity Standard Deviation in the Output Frame Y-Axis (Along Track for Vehicle)	Float	4	H+32
8	Z-Axis Velocity StDev	Velocity Standard Deviation in the Output Frame Z-Axis (Vertical for Vehicle)	Float	4	H+36
9	Slip Angle	Heading angle offset from actual motion over ground (Track over Ground) to Output Frame ( $\text{Heading}_{\text{GroundTrack}} - \text{Heading}_{\text{UserFrame}}$ )	Float	4	H+40
10	Reserved	Reserved	Ulong	4	H+44
11	Reserved	Reserved	Ulong	4	H+48
12	Extended Velocity Status	Extended Velocity Status, see <i>Table 248: Extended Velocity Status</i> on the next page.	Hex Ulong	4	H+52
13	XXXX	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
14	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–

**Table 248: Extended Velocity Status**

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Slip Angle Validity	0 = Invalid 1 = Valid
	1	0x00000002	Reserved	
	2	0x00000004	Reserved	
	3	0x00000008	Reserved	
N1	4	0x00000010	Reserved	
	5	0x00000020	Reserved	
	6	0x00000040	Reserved	
	7	0x00000080	Reserved	
N2	8	0x00000100	Reserved	
	9	0x00000200	Reserved	
	10	0x00000400	Reserved	
	11	0x00000800	Reserved	
N3	12	0x00001000	Reserved	
	13	0x00002000	Reserved	
	14	0x00004000	Reserved	
	15	0x00008000	Reserved	

## 5.37 INSVELX

### *Inertial Velocity – Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log includes the information from the INSVEL log, as well as information about the velocity standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.



The INSVELX log is a large log and is not recommend for high rate logging.

If you want to use high rate logging, log the **INSVELS** log at a high rate and the **INSSTDEVS** log ontime 1.

**Message ID:** 1458

**Log Type:** Synch

**Recommended Input:**

```
log insvelxa ontime 1
```

**ASCII example:**

```
#INSVELXA,USB1,0,66.5,FINESTEERING,2209,491263.000,02000020,1a1f,16809;
INS_SOLUTION_GOOD,INS_PPP,-0.0007,-0.0009,0.0017,0.0018,0.0018,0.0017,
13000045,0*b2e79695
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	INSVELX header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.		H	0
2	INS Status	Solution status See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H
3	Pos Type	Position type See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+4
4	North Vel	North velocity (m/s)	Double	8	H+8
5	East Vel	East velocity (m/s)	Double	8	H+16
6	Up Vel	Up velocity (m/s)	Double	8	H+24
7	North Vel $\sigma$	North velocity standard deviation (m/s)	Float	4	H+32

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
8	East Vel $\sigma$	East velocity standard deviation (m/s)	Float	4	H+36
9	Up Vel $\sigma$	Up velocity standard deviation (m/s)	Float	4	H+40
10	Ext sol stat	Extended solution status See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+44
11	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+48
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



## 5.38 MARK1PVA, MARK2PVA, MARK3PVA and MARK4PVA

### *Position, Velocity and Attitude at Mark Input Event*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

These logs output position, velocity and attitude information, with respect to the IMU Body frame, when an event is received on the Mark input. If the **SETINSTRANSULATION** command (see page 1044) and **SETINSROTATION** command (see page 1041) have been entered with a MARKx parameter, the MARKxPVA log will contain the solution translated, and then rotated, by the values provided in the commands (e.g. **SETINSTRANSULATION MARK1** and **SETINSROTATION MARK1** commands for the MARK1PVA log). See the **SETINSTRANSULATION** command on page 1044 and **SETINSROTATION** command on page 1041.



The MARKxPVA logs available are dependent on the receiver used in the SPAN system. For information about the Event lines supported, see the Strobe Specifications for the receiver in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#).



These logs should only be requested with the ONCHANGED or ONNEW trigger.



These logs provide the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:** 1067 (MARK1PVA)  
1068 (MARK2PVA)  
1118 (MARK3PVA)  
1119 (MARK4PVA)

**Log Type:** Asynch

**Recommended Input:**

```
log mark1pva onnew
log mark2pva onnew
log mark3pva onnew
log mark4pva onnew
```

**Abbreviated ASCII Example:**

```
#MARK1PVAA, COM1, 0, 74.5, FINESTEERING, 1732, 247231.455, 02040020, 5790, 12002; 1732,
247231.454623850, 51.11693182283, -114.03885213810, 1047.4525, 0.0004, 0.0004,
-0.0006, 0.847121689, 1.124640813, 278.577037489, INS_SOLUTION_GOOD*5a6b060e
#MARK2PVAA, COM1, 0, 74.5, FINESTEERING, 1732, 247232.271, 02040020, 2425, 12002; 1732,
247232.271459820, 51.11693179023, -114.03885206704, 1047.4529, 0.0004, -0.0011,
-0.0007, 0.837101074, 1.134127754, 278.346498557, INS_SOLUTION_GOOD*08209ec0
```

```
#MARK3PVAA,COM1,0,74.5,FINESTEERING,1732,247232.271,02040020,2425,12002;1732,
247232.271459820,51.11693179023,-114.03885206704,1047.4529,0.0004,-0.0011,
-0.0007,0.837101074,1.134127754,278.346498557,INS_SOLUTION_GOOD*08209ec0
```

```
#MARK4PVAA,COM1,0,74.5,FINESTEERING,1732,247232.271,02040020,2425,12002;1732,
247232.271459820,51.11693179023,-114.03885206704,1047.4529,0.0004,-0.0011,
-0.0007,0.837101074,1.134127754,278.346498557,INS_SOLUTION_GOOD*08209ec0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	MARK1PVA / MARK2PVA / MARK3PVA / MARK4PVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week at Mark input	Ulong	4	H
3	Seconds	Seconds from week at Mark input	Double	8	H+4
4	Latitude	Latitude at Mark input (degrees)	Double	8	H+12
5	Longitude	Longitude at Mark input (degrees)	Double	8	H+20
6	Height	Ellipsoidal height at Mark input (metres)	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction) at Mark input (metres/second)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction) at Mark input (metres/second)	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark input (metres/second)	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees at Mark input (degrees)	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees at Mark input (degrees)	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North at Mark input (degrees)	Double	8	H+76
13	Status	INS Status, see <i>Table 237: Inertial Solution Status</i> on page 1081 at Mark input	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.39 PASHR

### *NMEA, Inertial Attitude Data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

The PASHR log uses a UTC time, calculated with default parameters, to output NMEA messages without waiting for a valid almanac. The UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters and sets the UTC time to VALID. For more information about NMEA, refer to *NMEA Standard Logs* on page 728. The PASHR log contains only INS derived attitude information and is only filled when an inertial solution is available.



As of firmware version 7.03.00, an INS status flag (field 12) has been added to the PASHR log. This change was made to match the industry accepted form of the message. Previous firmware versions on OEM7 and OEM6 do not output this field.

**Message ID:** 1177

**Log Type:** Synch

**Recommended Input:**

```
log pashr ontime 1
```

**Example:**

```
$PASHR,163029.000,158.09,T,-0.30,+0.31,+0.01,0.029,0.029,0.059,1,1*3B
```

Field	Structure	Description	Symbol	Example
1	\$PASHR	Log header	---	\$PASHR
2	Time	UTC Time	hhmmss.sss	195124.00
3	Heading	Heading value in decimal degrees The heading is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	HHH.HH	305.30
4	True Heading	T displayed if heading is relative to true north.	T	T
5	Roll	Roll in decimal degrees. The ± sign will always be displayed.	RRR.RR	+0.05
6	Pitch	Pitch in decimal degrees. The ± sign will always be displayed.	PPP.PP	-0.13
7	Heave	Instantaneous heave in metres. The ± will always be displayed	Heave	+0.01
8	Roll Accuracy	Roll standard deviation in decimal degrees.	rr.rrr	0.180
9	Pitch Accuracy	Pitch standard deviation in decimal degrees.	pp.ppp	0.185

Field	Structure	Description	Symbol	Example
10	Heading Accuracy	Heading standard deviation in decimal degrees.	hh.hhh	4.986
11	GPS Update Quality Flag	0 = No position 1 = All non-RTK fixed integer positions 2 = RTK fixed integer position	1	1
12	INS Status Flag	0 = All SPAN Pre-Alignment INS Status 1 = All SPAN Post-Alignment INS Status - These include: INS_ALIGNMENT_COMPLETE, INS_SOLUTION_GOOD, INS_HIGH_VARIANCE, INS_SOLUTION_FREE	1	1
13	Checksum	Checksum	*XX	*2B
14	[CR][LF]	Sentence terminator		[CR][LF]

## 5.40 RAWDMI

### *Raw Distance Measurement Instrument (DMI) Measurements*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

This log contains the raw measurements from the Distance Measurement Instruments (DMI) connected to the receiver and enabled by the **DMICONFIG** command (see page 1004).

RAWDMI can also be used as a command. If DMI inputs are not supported on the OEM7 receiver or IMU, the DMI wheel ticks can be collected by a separate device and sent to the receiver using RAWDMI as a command.



The RAWDMI command does not need to be sent if the SPAN system has a PwrPak7 receiver or one of the following NovAtel enclosure IMUs: IMU-ISA-100C, IMU-ENC-LN200, IMU-HG1900, IMU-μIMU-IC, IMU-IGM-A1, IMU-IGM-S1, IMU-FSAS or IMU-CPT. For these devices, connect the DMI to the DMI inputs on the PwrPak7 or IMU and SPAN system will collect the wheel tick.

**Message ID:** 2269

**Log Type:** Synch

**Recommended Input:**

```
log rawdmia onnew
```

**Example log:**

```
#RAWDmia, COM1, 0, 24.0, FINESTEERING, 2048, 427043.137, 02004048, b411, 32768; 2297, 0, 0, 0, 00000001*40c6ec74
```

**Command Abbreviate ASCII Syntax:**

```
RAWDMI DMI1 DMI2 DMI3 DMI4 Mask
```

**Command Example:**

```
RAWDMI 1000 0 0 0 1 (indicates 1000 wheel ticks from DMI1)
```

```
RAWDMI 0 0 0 2000 8 (indicates 2000 wheel ticks from DMI4)
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWDMI header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
2	DMI1	DMI Input Value for DMI1	Long Array [4]	16	H
	DMI2	DMI Input Value for DMI2 (Inactive - support coming in a future release)			
	DMI3	DMI Input Value for DMI3 (Inactive - support coming in a future release)			
	DMI4	DMI Input Value for DMI4 (Inactive - support coming in a future release)			
3	Mask	Mask to indicate which DMI inputs are used in the solution. Bit 0 = DMI1 Bit 1 = DMI2 Bit 2 = DMI3 Bit 3 = DMI4	Long	4	H+16
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20
5	[CR][LF]	Sentence Terminator (ASCII only)	–	–	–



The DMI Input Values are interpreted based on the configuration of that input in the **DMICONFIG** command (see page 1004).



Only DMI1 is currently supported.

## 5.41 RAWIMU

### *Raw IMU Data*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log contains an IMU status indicator and the measurements from the accelerometers and gyros with respect to the IMU enclosure frame. If logging this data, consider the **RAWIMUS** log (see page 1161) to reduce the amount of data.



The change in velocity (acceleration) and angle (rotation rate) scale factors for each IMU type can be found in *Table 263: Raw IMU Scale Factors* on page 1164. Multiply the appropriate scale factor by the count value for the velocity (field 5-7) and angle (field 8-10) increments.



To obtain acceleration in m/s/s or rotation rate in rad/s, multiply the velocity/rotation increments by the output rate of the IMU:

- 100 Hz for HG1700, HG1900, HG1930 and HG4930
- 125 Hz for STIM300, G320N, PwrPak7-E1, PwrPak7D-E1 and SMART7-S
- 200 Hz for ISA-100C, iMAR-FSAS, LN200, KVH 1750, KVH P-1750, ADIS16488, G370N, G320N\_200Hz, PwrPak7-E2 and PwrPak7D-E2

The units of acceleration and rotation rate will depend on the IMU Scale Factors.



This log is output in the IMU Body frame.



#### **Logging Restriction Important Notice**

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format and, if possible, the short header binary format (available on most INS logs).

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing  
RAWIMUXSB ONNEW
    - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate.
  - Real time INS solution  
IMURATEPVA ONNEW or IMURATEPVAS ONNEW
- Other possible INS solution logs available at high rates are: INSPVASB, INSPOSSB, INSVELSB, INSATTSB

**Message ID: 268**

**Log Type: Asynch****Recommended Input:**

```
log rawimua onnew
```

**ASCII Example:**

```
#RAWIMUA,USB1,0,65.5,FINESTEERING,2209,491571.654,02000020,8358,16809;2209,491571.653515,ea59fe00,329181823,-2557458,4291497,234324,232588,-1010826*4b79604d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWIMU header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	IMU Status	<p>The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs.</p> <p>For the raw IMU status, see one of the following tables:</p> <ul style="list-style-type: none"> <li>• <i>Table 249: iIMU-FSAS IMU Status</i> on the next page</li> <li>• <i>Table 250: HG1700 IMU Status</i> on page 1141</li> <li>• <i>Table 251: LN200 IMU Status</i> on page 1143</li> <li>• <i>Table 252: ISA-100C IMU Status</i> on page 1144</li> <li>• <i>Table 253: IMU-CPT IMU Status</i> on page 1145</li> <li>• <i>Table 254: KVH 1750 and KVH P-1750 IMU Status</i> on page 1147</li> <li>• <i>Table 255: HG1900 and HG1930 IMU Status</i> on page 1148</li> <li>• <i>Table 256: HG4930 AN01 IMU Status</i> on page 1150</li> <li>• <i>Table 257: HG4930 AN04 IMU Status</i> on page 1151</li> <li>• <i>Table 258: ADIS16488 and IMU-IGM-A1 IMU Status</i> on page 1153</li> <li>• <i>Table 259: STIM300 and IMU-IGM-S1 IMU Status</i> on page 1154</li> <li>• <i>Table 260: <math>\mu</math>IMU IMU Status</i> on page 1156</li> <li>• <i>Table 261: G320N IMU Status</i> on page 1157</li> <li>• <i>Table 262: G370N IMU Status</i> on page 1159</li> </ul>	Hex Ulong	4	H+12
5	Z Accel Output	Change in velocity count along z axis	Long	4	H+16
6	-(Y Accel Output)	<p>-(Change in velocity count along y axis)</p> <p>A negative value implies the output is along the positive y-axis marked on the IMU. A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.</p>	Long	4	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	X Accel Output	Change in velocity count along x axis	Long	4	H+24
8	Z Gyro Output	Change in angle count around z axis Right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis). Right-handed A negative value implies a positive right-handed rotation about the y-axis marked on the IMU. A positive value implies a negative right-handed rotation about the y-axis marked on the IMU.	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 249: iIMU-FSAS IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	Gyro warm-up	0 = Passed, 1 = Failed
	5	0x00000020	Gyro self-test active	0 = Passed, 1 = Failed
	6	0x00000040	Gyro status bit set	0 = Passed, 1 = Failed
	7	0x00000080	Gyro time-out command interface	0 = Passed, 1 = Failed
N2	8	0x00000100	Power-up built-in test (PBIT)	0 = Passed, 1 = Failed
	9	0x00000200	Reserved	
	10	0x00000400	Interrupt	0 = Passed, 1 = Failed
	11	0x00000800	Reserved	
N3	12	0x00001000	Warm-up	0 = Passed, 1 = Failed
	13	0x00002000	Reserved	
	14	0x00004000		
	15	0x00008000	Initiated built-in test (IBIT)	0 = Passed, 1 = Failed

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000	Accelerometer	0 = Passed, 1 = Failed
	19	0x00080000	Accelerometer time-out	0 = Passed, 1 = Failed
N5	20	0x00100000	Reserved	
	21	0x00200000	Gyro initiated BIT	0 = Passed, 1 = Failed
	22	0x00400000	Gyro self-test	0 = Passed, 1 = Failed
	23	0x00800000	Gyro time-out	0 = Passed, 1 = Failed
N6	24	0x01000000	Analog-to-Digital (AD)	0 = Passed, 1 = Failed
	25	0x02000000	Test mode	0 = Passed, 1 = Failed
	26	0x04000000	Software	0 = Passed, 1 = Failed
	27	0x08000000	RAM/ROM	0 = Passed, 1 = Failed
N7	28	0x10000000	Reserved	
	29	0x20000000	Operational	0 = Passed, 1 = Failed
	30	0x40000000	Interface	0 = Passed, 1 = Failed
	31	0x80000000	Interface time-out	0 = Passed, 1 = Failed

Table 250: HG1700 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002	Reserved	
	2	0x00000004	Reserved	
	3	0x00000008	Reserved	
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed

Nibble	Bit	Mask	Description	Range Value
N2	8	0x00000100	Reserved	
	9	0x00000200	Reserved	
	10	0x00000400	Reserved	
	11	0x00000800	Reserved	
N3	12	0x00001000	Reserved	
	13	0x00002000	Reserved	
	14	0x00004000	Reserved	
	15	0x00008000	Reserved	
N4	16	0x00010000	Reserved	
	17	0x00020000	Reserved	
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	
N5	20	0x00100000	Reserved	
	21	0x00200000	Reserved	
	22	0x00400000	Reserved	
	23	0x00800000	Reserved	
N6	24	0x01000000	Reserved	
	25	0x02000000	Reserved	
	26	0x04000000	Reserved	
	27	0x08000000	IMU Status	0 = Passed, 1 = Failed
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	IMU Status	0 = Passed, 1 = Failed
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	IMU Status	0 = Passed, 1 = Failed

Table 251: LN200 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	IMU Status	0 = Passed, 1 = Failed
	1	0x00000002	IMU Status	0 = Passed, 1 = Failed
	2	0x00000004	IMU Status	0 = Passed, 1 = Failed
	3	0x00000008	IMU Status	0 = Passed, 1 = Failed
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed
N2	8	0x00000100	IMU Status	0 = Passed, 1 = Failed
	9	0x00000200	IMU Status	0 = Passed, 1 = Failed
	10	0x00000400	IMU Status	0 = Passed, 1 = Failed
	11	0x00000800	IMU Status	0 = Passed, 1 = Failed
N3	12	0x00001000	IMU Status	0 = Passed, 1 = Failed
	13	0x00002000	IMU Status	0 = Passed, 1 = Failed
	14	0x00004000	IMU Status	0 = Passed, 1 = Failed
	15	0x00008000	Reserved	
N4	16	0x00010000	Reserved	
	17	0x00020000	Reserved	
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	
N5	20	0x00100000	Reserved	
	21	0x00200000	Reserved	
	22	0x00400000	Reserved	
	23	0x00800000	Reserved	
N6	24	0x01000000	IMU Status	0 = Passed, 1 = Failed
	25	0x02000000	IMU Status	0 = Passed, 1 = Failed
	26	0x04000000	IMU Status	0 = Passed, 1 = Failed
	27	0x08000000	IMU Status	0 = Passed, 1 = Failed

Nibble	Bit	Mask	Description	Range Value
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	Reserved	
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	Reserved	

Table 252: ISA-100C IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Maintenance Indication	0 = Normal, 1 = System Maintenance Indicator
	1	0x00000002	Accelerometers Invalid	0 = Normal, 1 = Invalid
	2	0x00000004	Accelerometer X Warning	0 = Normal, 1 = Warning
	3	0x00000008	Accelerometer Y Warning	0 = Normal, 1 = Warning
N1	4	0x00000010	Accelerometer Z Warning	0 = Normal, 1 = Warning
	5	0x00000020	Accelerometer X NOGO	0 = Normal, 1 = NOGO
	6	0x00000040	Accelerometer Y NOGO	0 = Normal, 1 = NOGO
	7	0x00000080	Accelerometer Z NOGO	0 = Normal, 1 = NOGO
N2	8	0x00000100	Reset Occurred	0 = Normal, 1 = First Message after ISA-100C Reset
	9	0x00000200	Gyroscopes Invalid	0 = Normal, 1 = Invalid
	10	0x00000400	Gyroscope X Warning	0 = Normal, 1 = Warning
	11	0x00000800	Gyroscope Y Warning	0 = Normal, 1 = Warning
N3	12	0x00001000	Gyroscope Z Warning	0 = Normal, 1 = Warning
	13	0x00002000	Gyroscope X NOGO	0 = Normal, 1 = NOGO
	14	0x00004000	Gyroscope Y NOGO	0 = Normal, 1 = NOGO
	15	0x00008000	Gyroscope Z NOGO	0 = Normal, 1 = NOGO

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) 1 LSB = 3.90625e <sup>-3</sup> Celsius Temperature Range +/- 128 Celsius	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 253: IMU-CPT IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Gyro X Status	1 = Valid, 0 = Invalid
	1	0x00000002	Gyro Y Status	1 = Valid, 0 = Invalid
	2	0x00000004	Gyro Z Status	1 = Valid, 0 = Invalid
	3	0x00000008	Unused	Set to 0
N1	4	0x00000010	Accelerometer X Status	1 = Valid, 0 = Invalid
	5	0x00000020	Accelerometer Y Status	1 = Valid, 0 = Invalid
	6	0x00000040	Accelerometer Z Status	1 = Valid, 0 = Invalid
	7	0x00000080	Unused	Set to 0



Nibble	Bit	Mask	Description	Range Value		
N2	8	0x00000100	IMU Data Sequence Counter read in a Ushort.   Increments for each message and resets to 0 after 127.			
	9	0x00000200				
	10	0x00000400				
	11	0x00000800				
N3	12	0x00001000				
	13	0x00002000				
	14	0x00004000				
	15	0x00008000				
N4	16	0x00010000			Unused	
	17	0x00020000				
	18	0x00040000				
	19	0x00080000				
N5	20	0x00100000				
	21	0x00200000				
	22	0x00400000				
	23	0x00800000				
N6	24	0x01000000				
	25	0x02000000				
	26	0x04000000				
	27	0x08000000				
N7	28	0x10000000				
	29	0x20000000				
	30	0x40000000				
	31	0x80000000				



Table 254: KVH 1750 and KVH P-1750 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Gyro X Status	1 = Valid, 0 = Invalid
	1	0x00000002	Gyro Y Status	1 = Valid, 0 = Invalid
	2	0x00000004	Gyro Z Status	1 = Valid, 0 = Invalid
	3	0x00000008	Unused	Set to 0
N1	4	0x00000010	Accelerometer X Status	1 = Valid, 0 = Invalid
	5	0x00000020	Accelerometer Y Status	1 = Valid, 0 = Invalid
	6	0x00000040	Accelerometer Z Status	1 = Valid, 0 = Invalid
	7	0x00000080	Unused	Set to 0
N2	8	0x00000100	IMU Data Sequence Counter read in a Ushort.  Increments for each message and resets to 0 after 127.	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		


Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) Rounded to the nearest degree  Example: <RAWIMU COM1 0 75.0 FINESTEERING 1813 514207.000 00000020 fa9a 45836 1813 514207.0000000000 <b>00260077</b> 32164 -47 -305 1 -10 0	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		IMU status = 00260077
	27	0x08000000		Temperatures bytes = 0026 Decimal value = 38 degrees C
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 255: HG1900 and HG1930 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed

Nibble	Bit	Mask	Description	Range Value
N2	8	0x00000100	Reserved	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000	Reserved	
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000	Reserved	
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	IMU Status	0 = Passed, 1 = Failed
	25	0x02000000	Reserved	
	26	0x04000000	IMU Status	0 = Passed, 1 = Failed
	27	0x08000000	IMU Status	0 = Passed, 1 = Failed
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	IMU Status	0 = Passed, 1 = Failed
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	Reserved	

Table 256: HG4930 AN01 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	IMU Status	0 = Passed, 1 = Failed
	1	0x00000002	Reserved	
	2	0x00000004		
	3	0x00000008	Gyro Status	0 = Passed, 1 = Failed
N1	4	0x00000010	Accelerometer Status	0 = Passed, 1 = Failed
	5	0x00000020	Reserved	
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	Reserved	
N2	8	0x00000100	<p>MU temperature reading as follows:</p> <p>Signed 2-byte integer (SHORT, two's complement)</p> <p>0°C = 0x0000</p> <p>1LSB = 0.01 °C</p>	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000		
	17	0x00020000		 Available only on firmware versions 7.08.02 and later
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Reserved	
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		


Nibble	Bit	Mask	Description	Range Value
N7	28	0x10000000	Reserved	
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		



The values in *Table 256: HG4930 AN01 IMU Status* on the previous page also apply to the legacy CPT7.

**Table 257: HG4930 AN04 IMU Status**

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	IMU Status	0 = Passed, 1 = Failed
	1	0x00000002	Reserved	
	2	0x00000004		
	3	0x00000008	Gyro Status	0 = Passed, 1 = Failed
N1	4	0x00000010	Accelerometer Status	0 = Passed, 1 = Failed
	5	0x00000020	Reserved	
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	Reserved	

Nibble	Bit	Mask	Description	Range Value		
N2	8	0x00000100	MU temperature reading as follows: Signed 2-byte integer (SHORT, two's complement) 0°C = 0x0000 1LSB = 0.01 °C			
	9	0x00000200				
	10	0x00000400				
	11	0x00000800				
N3	12	0x00001000				
	13	0x00002000				
	14	0x00004000				
	15	0x00008000				
N4	16	0x00010000			 Available only on firmware versions 7.08.02 and later	
	17	0x00020000				
	18	0x00040000				
	19	0x00080000				
N5	20	0x00100000				
	21	0x00200000				
	22	0x00400000				
	23	0x00800000				
N6	24	0x01000000	Reserved			
	25	0x02000000				
	26	0x04000000				
	27	0x08000000				
N7	28	0x10000000	Reserved			
	29	0x20000000				
	30	0x40000000				
	31	0x80000000				



The values in *Table 257: HG4930 AN04 IMU Status* on the previous page also apply to the CPT7 and CPT7700.

**Table 258: ADIS16488 and IMU-IGM-A1 IMU Status**

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Alarm Status Flag	
	1	0x00000002	Reserved	
	2	0x00000004		
	3	0x00000008	SPI Communication Error	0 = Passed, 1 = Failed
N1	4	0x00000010	Sensor Over-Range	0 = Passed, 1 = One of more sensors over- ranged
	5	0x00000020	Initial Self Test Failure	0 = Passed, 1 = Failed
	6	0x00000040	Flash Memory Failure	0 = Passed, 1 = Failed
	7	0x00000080	Processing Overrun	0 = Passed, 1 = Failed
N2	8	0x00000100	Self Test Failure – X-axis gyro	0 = Passed, 1 = Failed
	9	0x00000200	Self Test Failure – Y-axis gyro	0 = Passed, 1 = Failed
	10	0x00000400	Self Test Failure – Z-axis gyro	0 = Passed, 1 = Failed
	11	0x00000800	Self Test Failure – X-axis accelerometer	0 = Passed, 1 = Failed
N3	12	0x00001000	Self Test Failure – Y-axis accelerometer	0 = Passed, 1 = Failed
	13	0x00002000	Self Test Failure – Z-axis	0 = Passed, 1 = Failed
	14	0x00004000	Reserved	
	15	0x00008000		

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) 25°C = 0x0000 1 LSB = 0.00565°C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 259: STIM300 and IMU-IGM-S1 IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Gyro status	0 = OK, 1 = X-channel
	1	0x00000002		0 = OK, 1 = Y-channel
	2	0x00000004		0 = OK, 1 = Z-channel
	3	0x00000008		0 = OK, 1 = Error in measurement channel (Bits 0-2 flag the error channels)
N1	4	0x00000010		0 = OK, 1 = Overload (Bits 0-2 flag the error channels)
	5	0x00000020		0 = OK, 1 = Outside operating conditions
	6	0x00000040		0 = OK, 1 = Startup
	7	0x00000080		0 = OK, 1 = System integrity error



Nibble	Bit	Mask	Description	Range Value
N2	8	0x00000100	Accelerometer Status	0 = OK, 1 = X-channel
	9	0x00000200		0 = OK, 1 = Y-channel
	10	0x00000400		0 = OK, 1 = Z-channel
	11	0x00000800		0 = OK, 1 = Error in measurement channel (Bits 0-2 flag the error channels)
N3	12	0x00001000		0 = OK, 1 = Overload (Bits 0-2 flag the error channels)
	13	0x00002000		0 = OK, 1 = Outside operating conditions
	14	0x00004000		0 = OK, 1 = Startup
	15	0x00008000		0 = OK, 1 = System integrity error
N4	16	0x00010000		Temperature of the X gyro sensor 0°C = 0x0000 1 LSB = 2 <sup>-8</sup> °C
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 260:  $\mu$ IMU IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Reset Acknowledged	0 = Normal, 1 = Reset
	1	0x00000002	Gyros Not Initialized	0 = Normal, 1 = Not Initialized
	2	0x00000004	Gyro X Warning	0 = Normal, 1 = Warning
	3	0x00000008	Gyro Y Warning	0 = Normal, 1 = Warning
N1	4	0x00000010	Gyro Z Warning	0 = Normal, 1 = Warning
	5	0x00000020	Gyro X NOGO	0 = Normal, 1 = NOGO
	6	0x00000040	Gyro Y NOGO	0 = Normal, 1 = NOGO
	7	0x00000080	Gyro Z NOGO	0 = Normal, 1 = NOGO
N2	8	0x00000100	Reserved	
	9	0x00000200	Accels Not Initialized	0 = Normal, 1 = Not Initialized
	10	0x00000400	Accel X Warning	0 = Normal, 1 = Warning
	11	0x00000800	Accel Y Warning	0 = Normal, 1 = Warning
N3	12	0x00001000	Accel Z Warning	0 = Normal, 1 = Warning
	13	0x00002000	Accel X NOGO	0 = Normal, 1 = NOGO
	14	0x00004000	Accel Y NOGO	0 = Normal, 1 = NOGO
	15	0x00008000	Accel Z NOGO	0 = Normal, 1 = NOGO

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) 1 LSB = 3.90625e <sup>-3</sup> °C Temperature Range +/- 128 °C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 261: G320N IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Error All	0 = Normal, 1 = Sensor Failure
	1	0x00000002	Reserved	
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010		
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		

Nibble	Bit	Mask	Description	Range Value
N2	8	0x00000100		
	9	0x00000200	Accel Z - New Data	New Data = 1, No Data = 0
	10	0x00000400	Accel Y - New Data	New Data = 1, No Data = 0
	11	0x00000800	Accel X - New Data	New Data = 1, No Data = 0
N3	12	0x00001000	Gyro Z - New Data	New Data = 1, No Data = 0
	13	0x00002000	Gyro Y - New Data	New Data = 1, No Data = 0
	14	0x00004000	Gyro X - New Data	New Data = 1, No Data = 0
	15	0x00008000	Temperature - New Data	New Data = 1, No Data = 0
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte integer (SHORT, two's complement) 34.9876012 °C = 0x0000 1 LSB = -0.0037918 °C  Conversion: Temperature [°C] = -0.0037918 * (VALUE - 2634) + 25 OR Temperature [°C] = -0.0037918 * VALUE + 34.9876012	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		



The values in *Table 261: G320N IMU Status* on the previous page also apply to the PwrPak7-E1, PwrPak7D-E1 and SMART7-S.

Table 262: G370N IMU Status

Nibble	Bit	Mask	Description	Range Value
N0	0	0x00000001	Error All	0 = Normal, 1 = Sensor Failure
	1	0x00000002	Run Time Diagnostics	0 = Normal, 1 = At least one abnormality detected in run time self-diagnosis result
	2	0x00000004	Reserved	
	3	0x00000008		
N1	4	0x00000010		
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	Range Over	0 = Normal, 1 = At least one over range condition detected
	9	0x00000200	Accel Z - New Data	New Data = 1, No Data = 0
	10	0x00000400	Accel Y - New Data	New Data = 1, No Data = 0
	11	0x00000800	Accel X - New Data	New Data = 1, No Data = 0
N3	12	0x00001000	Gyro Z - New Data	New Data = 1, No Data = 0
	13	0x00002000	Gyro Y - New Data	New Data = 1, No Data = 0
	14	0x00004000	Gyro X - New Data	New Data = 1, No Data = 0
	15	0x00008000	Temperature - New Data	New Data = 1, No Data = 0

Nibble	Bit	Mask	Description	Range Value
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte integer (SHORT, two's complement) 34.9876012 °C = 0x0000 1 LSB = -0.0037918 °C  Conversion: Temperature [°C] = -0.0037918 * (VALUE - 2634) + 25 OR Temperature [°C] = -0.0037918 * VALUE + 34.9876012	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		



The values in *Table 262: G370N IMU Status* on the previous page also apply to the PwrPak7-E2 and PwrPak7D-E2.

## 5.42 RAWIMUS

### Short Raw IMU Data

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is the short header version of the **RAWIMU** log (see page 1137).



The change in velocity (acceleration) and angle (rotation rate) scale factors for each IMU type can be found in *Table 263: Raw IMU Scale Factors* on page 1164. Multiply the appropriate scale factor by the count value for the velocity (field 5-7) and angle (field 8-10) increments.



To obtain acceleration in m/s/s or rotation rate in rad/s, multiply the velocity/rotation increments by the output rate of the IMU:

- 100 Hz for HG1700, HG1900, HG1930 and HG4930
- 125 Hz for STIM300, G320N, PwrPak7-E1, PwrPak7D-E1 and SMART7-S
- 200 Hz for ISA-100C, iMAR-FSAS, LN200, KVH 1750, KVH P-1750, ADIS16488, G370N, PwrPak7-E2 and PwrPak7D-E2

The units of acceleration and rotation rate will depend on the IMU Scale Factors.



This log is output in the IMU Body frame.



#### Logging Restriction Important Notice

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format and, if possible, the short header binary format (available on most INS logs).

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing  
RAWIMUXSB ONNEW
    - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate.
  - Real time INS solution  
IMURATEPVA ONNEW or IMURATEPVAS ONNEW
- Other possible INS solution logs available at high rates are: INSPVASB, INSPOSSB, INSVELSB, INSATTSB

**Message ID: 325**

**Log Type: Asynch**

**Recommended Input:**

```
log rawimusa onnew
```

**ASCII Example:**

```
%RAWIMUSA,2209,491612.254;2209,491612.253588,ea5afe00,329303960,-2512321,
4410363,309705,607017,-1009773*1fb76c28
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWIMUS header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	IMU Status	<p>The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs.</p> <p>For the raw IMU status, see one of the following tables:</p> <ul style="list-style-type: none"> <li>• <i>Table 249: iIMU-FSAS IMU Status</i> on page 1140</li> <li>• <i>Table 250: HG1700 IMU Status</i> on page 1141</li> <li>• <i>Table 251: LN200 IMU Status</i> on page 1143</li> <li>• <i>Table 252: ISA-100C IMU Status</i> on page 1144</li> <li>• <i>Table 253: IMU-CPT IMU Status</i> on page 1145</li> <li>• <i>Table 254: KVH 1750 and KVH P-1750 IMU Status</i> on page 1147</li> <li>• <i>Table 255: HG1900 and HG1930 IMU Status</i> on page 1148</li> <li>• <i>Table 256: HG4930 AN01 IMU Status</i> on page 1150</li> <li>• <i>Table 257: HG4930 AN04 IMU Status</i> on page 1151</li> <li>• <i>Table 258: ADIS16488 and IMU-IGM-A1 IMU Status</i> on page 1153</li> <li>• <i>Table 259: STIM300 and IMU-IGM-S1 IMU Status</i> on page 1154</li> <li>• <i>Table 260: <math>\mu</math>IMU IMU Status</i> on page 1156</li> <li>• <i>Table 261: G320N IMU Status</i> on page 1157</li> <li>• <i>Table 262: G370N IMU Status</i> on page 1159</li> </ul>	Hex Ulong	4	H+12
5	Z Accel Output	Change in velocity count along z axis	Long	4	H+16
6	-(Y Accel Output)	<p>-(Change in velocity count along y axis)</p> <p>A negative value implies the output is along the positive y-axis marked on the IMU. A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.</p>	Long	4	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	X Accel Output	Change in velocity count along x axis	Long	4	H+24
8	Z Gyro Output	Change in angle count around z axis Right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis) Right-handed A negative value implies a positive right-handed rotation about the y-axis marked on the IMU. A positive value implies a negative right-handed rotation about the y-axis marked on the IMU.	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 263: Raw IMU Scale Factors

	Gyroscope Scale Factor	Acceleration Scale Factor
HG1700-AG58 HG1900-CA29/CA50 HG1930-AA99/CA50	$2.0^{-33}$ rad/LSB	$2.0^{-27}$ ft/s/LSB
HG1700-AG62	$2.0^{-33}$ rad/LSB	$2.0^{-26}$ ft/s/LSB
HG4930-AN01 HG4930-AN04 CPT7 CPT7700	$2.0^{-33}$ rad/LSB	$2.0^{-29}$ m/s/LSB
IMU-CPT KVH 1750 KVH P-1750	$0.1 / (3600.0 \times 256.0)$ rad/LSB	$0.05 / 2^{15}$ m/s/LSB
IMU-FSAS	$0.1 \times 2^{-8}$ arcsec/LSB	$0.05 \times 2^{-15}$ m/s/LSB
LN-200	$2^{-19}$ rad/LSB	$2^{-14}$ m/s/LSB
ISA-100C $\mu$ IMU	$1.0 \times 10^{-9}$ rad/LSB	$2.0 \times 10^{-8}$ m/s/LSB
ADIS16488 IMU-IGM-A1	$720 / 2^{31}$ deg/LSB	$200 / 2^{31}$ m/s/LSB
STIM300 IMU-IGM-S1	$2^{-21}$ deg/LSB	$2^{-22}$ m/s/LSB

	Gyroscope Scale Factor	Acceleration Scale Factor
G320N PwrPak7-E1 PwrPak7D-E1 SMART7-S	$(0.008/65536)/(DataRate)$ deg/LSB	$(0.200/65536)*(ONE\_G/1000)/(DataRate)$ m/s/LSB
G370N PwrPak7-E2 PwrPak7D-E2	$(0.0151515/65536)/200$ deg/LSB	$(0.400/65536)*(ONE\_G/1000)/200$ m/s/LSB



LSB = Least Significant Bit  
 DataRate = 125 or 200 depending on configuration  
 ONE\_G = 9.80665 m/s/s

## 5.43 RAWIMUSX

### *IMU Data Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This is the short header version of the extended RAWIMUX log intended for use with post-processing. The extended version includes IMU information that is used by the NovAtel Inertial Explorer post-processing software.



The change in velocity (acceleration) and angle (rotation rate) scale factors for each IMU type can be found in *Table 263: Raw IMU Scale Factors* on page 1164. Multiply the appropriate scale factor by the count value for the velocity (field 7-9) and angle (field 10-12) increments.



To obtain acceleration in m/s/s or rotation rate in rad/s, multiply the velocity/rotation increments by the output rate of the IMU:

- 100 Hz for HG1700, HG1900, HG1930 and HG4930
- 125 Hz for STIM300, G320N, PwrPak7-E1, PwrPak7D-E1 and SMART7-S
- 200 Hz for ISA-100C, iMAR-FSAS, LN200, KVH 1750, KVH P-1750, ADIS16488, G370N, PwrPak7-E2 and PwrPak7D-E2

The units of acceleration and rotation rate will depend on the IMU Scale Factors.



This log is output in the IMU Body frame.



#### **Logging Restriction Important Notice**

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format and, if possible, the short header binary format (available on most INS logs).

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing  
RAWIMUXSB ONNEW
    - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate.
  - Real time INS solution  
IMURATEPVA ONNEW or IMURATEPVAS ONNEW
- Other possible INS solution logs available at high rates are: INSPVASB, INSPOSSB, INSVELSB, INSATTSB

**Message ID: 1462**

**Log Type: Asynch****Recommended Input:**

```
log rawimusxb onnew
```

**ASCII example:**

```
%RAWIMUSXA,2209,491674.014;04,41,2209,491674.013699,ea59fe00,329308016
,-2706935,4508389,196192,417138,-1000837*038422a7
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWIMUSX header For information about short log headers, see <i>Description of ASCII and Binary Logs with Short Headers</i> on page 48.	-	H	0
2	IMU Info	IMU Info Bits Bit 0: If set, an IMU error was detected. Check the IMU Status field for details. Bit 1: If set, the IMU data is encrypted and should not be used. Bits 2 to 7: Reserved	Hex Uchar	1	H
3	IMU Type	IMU Type identifier. See <i>Table 230: IMU Type</i> on page 1002.	Uchar	1	H+1
4	GNSS Week	GNSS Week	Ushort	2	H+2
5	GNSS Week Seconds	Seconds from week start	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	IMU Status	<p>The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs.</p> <p>For the raw IMU status, see one of the following tables:</p> <ul style="list-style-type: none"> <li>• <i>Table 249: iIMU-FSAS IMU Status</i> on page 1140</li> <li>• <i>Table 250: HG1700 IMU Status</i> on page 1141</li> <li>• <i>Table 251: LN200 IMU Status</i> on page 1143</li> <li>• <i>Table 252: ISA-100C IMU Status</i> on page 1144</li> <li>• <i>Table 253: IMU-CPT IMU Status</i> on page 1145</li> <li>• <i>Table 254: KVH 1750 and KVH P-1750 IMU Status</i> on page 1147</li> <li>• <i>Table 255: HG1900 and HG1930 IMU Status</i> on page 1148</li> <li>• <i>Table 256: HG4930 AN01 IMU Status</i> on page 1150</li> <li>• <i>Table 257: HG4930 AN04 IMU Status</i> on page 1151</li> <li>• <i>Table 258: ADIS16488 and IMU-IGM-A1 IMU Status</i> on page 1153</li> <li>• <i>Table 259: STIM300 and IMU-IGM-S1 IMU Status</i> on page 1154</li> <li>• <i>Table 260: <math>\mu</math>IMU IMU Status</i> on page 1156</li> <li>• <i>Table 261: G320N IMU Status</i> on page 1157</li> <li>• <i>Table 262: G370N IMU Status</i> on page 1159</li> </ul>	Hex Ulong	4	H+12
7	Z Accel	Change in velocity count along Z-axis.	Long	4	H+16
8	-(Y Accel)	<p>-(Change in velocity count along y-axis.)</p> <p>A negative value implies the output is along the positive y-axis marked on the IMU. A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.</p>	Long	4	H+20
9	X Accel	Change in velocity count along x axis.	Long	4	H+24

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
10	Z Gyro	Change in angle count around z axis. Right-handed	Long	4	H+28
11	-(Y Gyro)	- (Change in angle count around y axis.) Right-handed A negative value implies a positive right-handed rotation about the y-axis marked on the IMU. A positive value implies a negative right-handed rotation about the y-axis marked on the IMU.	Long	4	H+32
12	X Gyro	Change in angle count around x axis. Right-handed	Long	4	H+36
13	XXXX	32-bit CRC (ASCII, Binary, and Short Binary only)	Hex	4	H+40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.44 RAWIMUX

### *IMU Data Extended*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log is an extended version of the RAWIMU log intended for use with post-processing. The extended version includes IMU information that is used by the NovAtel Inertial Explorer post-processing software.



The change in velocity (acceleration) and angle (rotation rate) scale factors for each IMU type can be found in *Table 263: Raw IMU Scale Factors* on page 1164. Multiply the appropriate scale factor by the count value for the velocity (field 7-9) and angle (field 10-12) increments.



To obtain acceleration in m/s/s or rotation rate in rad/s, multiply the velocity/rotation increments by the output rate of the IMU:

- 100 Hz for HG1700, HG1900, HG1930 and HG4930
- 125 Hz for STIM300, G320N, PwrPak7-E1, PwrPak7D-E1 and SMART7-S
- 200 Hz for ISA-100C, iMAR-FSAS, LN200, KVH 1750, KVH P-1750, ADIS16488, G370N, PwrPak7-E2 and PwrPak7D-E2

The units of acceleration and rotation rate will depend on the IMU Scale Factors.



This log is output in the IMU Body frame.



### **Logging Restriction Important Notice**

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format and, if possible, the short header binary format (available on most INS logs).

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing  
RAWIMUXSB ONNEW
    - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate.
  - Real time INS solution  
IMURATEPVA ONNEW or IMURATEPVAS ONNEW
- Other possible INS solution logs available at high rates are: INSPVASB, INSPOSSB, INSVELSB, INSATTSB

**Message ID: 1461**



**Log Type: Asynch****Recommended Input:**

```
log rawimuxb onnew
```

**ASCII example:**

```
#RAWIMUXA,USB1,0,64.5,FINESTEERING,2209,491740.110,02000020,0dc5,16809;04,41,
2209,491740.109817,ea57fe00,329394068,-2396829,4481425,321199,257329,-936077
*cf9c035d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RAWIMUX header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	IMU Info	IMU Info Bits Bit 0: If set, an IMU error was detected. Check the IMU Status field for details. Bit 1: If set, the IMU data is encrypted and should not be used. Bits 2 to 7: Reserved	Hex Uchar	1	H
3	IMU Type	IMU Type identifier. See <i>Table 230: IMU Type</i> on page 1002.	Uchar	1	H+1
4	GNSS Week	GNSS Week	Ushort	2	H+2
5	GNSS Week Seconds	Seconds from week start	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	IMU Status	<p>The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs.</p> <p>For the raw IMU status, see one of the following tables:</p> <ul style="list-style-type: none"> <li>• <i>Table 249: iIMU-FSAS IMU Status</i> on page 1140</li> <li>• <i>Table 250: HG1700 IMU Status</i> on page 1141</li> <li>• <i>Table 251: LN200 IMU Status</i> on page 1143</li> <li>• <i>Table 252: ISA-100C IMU Status</i> on page 1144</li> <li>• <i>Table 253: IMU-CPT IMU Status</i> on page 1145</li> <li>• <i>Table 254: KVH 1750 and KVH P-1750 IMU Status</i> on page 1147</li> <li>• <i>Table 255: HG1900 and HG1930 IMU Status</i> on page 1148</li> <li>• <i>Table 256: HG4930 AN01 IMU Status</i> on page 1150</li> <li>• <i>Table 257: HG4930 AN04 IMU Status</i> on page 1151</li> <li>• <i>Table 258: ADIS16488 and IMU-IGM-A1 IMU Status</i> on page 1153</li> <li>• <i>Table 259: STIM300 and IMU-IGM-S1 IMU Status</i> on page 1154</li> <li>• <i>Table 260: <math>\mu</math>IMU IMU Status</i> on page 1156</li> <li>• <i>Table 261: G320N IMU Status</i> on page 1157</li> <li>• <i>Table 262: G370N IMU Status</i> on page 1159</li> </ul>	Hex Ulong	4	H+12
7	Z Accel	Change in velocity count along Z-axis.	Long	4	H+16
8	-(Y Accel)	<p>-(Change in velocity count along y-axis.)</p> <p>A negative value implies the output is along the positive y-axis marked on the IMU. A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.</p>	Long	4	H+20
9	X Accel	Change in velocity count along x axis.	Long	4	H+24

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
10	Z Gyro	Change in angle count around z axis. Right-handed	Long	4	H+28
11	-(Y Gyro)	- (Change in angle count around y axis.) Right-handed A negative value implies a positive right-handed rotation about the y-axis marked on the IMU. A positive value implies a negative right-handed rotation about the y-axis marked on the IMU.	Long	4	H+32
12	X Gyro	Change in angle count around x axis. Right-handed	Long	4	H+36
13	XXXX	32-bit CRC (ASCII, Binary, and Short Binary only)	Hex	4	H+40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.45 RELINSPVA

### *Relative INSPVA log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the relative offset between the Master and Rover Inertial Solutions. The output solution provides the offset of where the local station is with respect to the other station.



This log should only be requested with the ONNEW trigger.



To use the **RELINSPVA** log, the receiver requires a Relative INS capable model.

**Message ID:** 1446

**Log Type:** Asynch

**Recommended Input:**

```
LOG RELINSPVAA ONNEW
```

**ASCII example:**

```
#RELINSPVAA,COM1,0,61.0,FINESTEERING,1805,245074.000,02000000,2338,45757;BODY,
9.285958662,-0.755483058,0.079229338,0.001739020,-0.000126304,0.001525848,
0.321033045,0.669367786,4.466250181,0.000000000,"b81V",INS_ALIGNMENT_COMPLETE,
"B20C",INS_ALIGNMENT_COMPLETE,NARROW_INT,00000000*a114ce3c
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	RELINSPVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Output Frame	The current output frame (IMU body, ECEF or local level frame). The output frame is specified using the <b>SETRELINSOUTPUTFRAME</b> command (see page 1051)	Enum	4	H
3	DeltaPosX	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
4	DeltaPosY	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+12
5	DeltaPosZ	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+20
6	DeltaVelX	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+28
7	DeltaVelY	Difference in velocity between two receivers (m/s). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+36
8	DeltaVelZ	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+44
9	DeltaRoll	Difference in roll between the two receivers (degrees).	Double	8	H+52
10	DeltaPitch	Difference in pitch between the two receivers (degrees).	Double	8	H+60

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
11	DeltaHeading	Difference in heading between the two receivers (degrees).	Double	8	H+68
12	Diff Age	Differential age in seconds.	Float	4	H+76
13	Rover ID	Rover receiver ID string.	Char[4]	4	H+80
14	Rover INSStatus	INS status of the rover receiver. See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
15	Master ID	Master receiver ID string.	Char[4]	4	H+88
16	Master INSStatus	INS status of the master receiver. See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+92
17	RTK Status	Status of the current RTK vector between master and rover. See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+96
18	ExtStatus	Extended solution status. See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+100
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+104
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.46 SYNCHEAVE

### *Synchronous Log Containing the Instantaneous Heave Value*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Synchronous heave is available up to the rate of the IMU. It can also be logged 'on time' at lower rates.

This log also includes information about the estimated accuracy of the heave value through the standard deviation of the heave. This heave solution is calculated at the location entered in the **SETINSTRLATION USER** command.



You must have an inertial solution to use this log.

Refer also to information in the **SETHEAVEWINDOW** command on page 1032.

**Message ID: 1708**

**Log Type: Synch**

**Recommended Input:**

```
log syncheaveb ontime 0.05
```

**ASCII example:**

```
#SYNCHEAVEA,USB1,0,66.5,FINESTEERING,2209,491905.000,02000020,552e,16809;0.012799581,0.031871900*01d2be8c
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SYNCHEAVE header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Heave	Instantaneous heave value (metres)	Double	8	H
3	Std. Dev.	Standard deviation of the heave value (metres)	Double	8	H+8
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
5	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.47 SYNCRELINSPVA

### *Synchronous Relative INSPVA log*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides the relative offset between the master and rover inertial solutions. The output solution provides the offset of where the local station is with respect to the other station.



To use the **SYNCRELINSPVA** log, the receiver requires a Relative INS capable model.



This log is designed to provide synchronous, relative Position, Velocity and Attitude information, propagating the information between matched corrections between the master and remote solutions. It is highly recommended that the highest rate of corrections be used at all times for the most precise and robust performance.

**Message ID:** 1743

**Log Type:** Synch

**Recommended Input:**

```
LOG SYNCRELINSPVAA ONTIME 1
```

**ASCII example:**

```
#SYNCRELINSPVAA,COM1,0,72.5,FINESTEERING,1805,247243.000,02000000,e9c7,13005;
BODY,8.141080733,-2.779177478,2.045421773,-0.001464009,-0.001038329,
0.002323548,0.409467974,0.715633909,-6.204731538,0.000000000,"B81V",INS_
ALIGNMENT_COMPLETE,"B20C",INS_ALIGNMENT_COMPLETE,INS_PSRSP,00000000*e270f5c8
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	SYNCRELINSPVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Output Frame	The current output frame (IMU body, ECEF or local level frame). The output frame is specified using the <b>SETRELINSOUTPUTFRAME</b> command (see page 1051)	Enum	4	H



Field	Field Type	Description	Format	Binary Bytes	Binary Offset
3	DeltaPosX	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+4
4	DeltaPosY	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+12
5	DeltaPosZ	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+20
6	DeltaVelX	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+28
7	DeltaVelY	Difference in velocity between two receivers (m/s). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+36

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
8	DeltaVelZ	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+44
9	DeltaRoll	Difference in roll between the two receivers (degrees).	Double	8	H+52
10	DeltaPitch	Difference in pitch between the two receivers (degrees).	Double	8	H+60
11	DeltaHeading	Difference in heading between the two receivers (degrees).	Double	8	H+68
12	Diff Age	Differential age in seconds.	Float	4	H+76
13	Rover ID	Rover receiver ID string.	Char[4]	4	H+80
14	Rover INSStatus	INS status of the rover receiver. See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
15	Master ID	Master receiver ID string.	Char[4]	4	H+88
16	Master INSStatus	INS status of the master receiver. See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+92
17	RTK Status	Status of the current RTK vector between master and rover. See <i>Table 87: Position or Velocity Type</i> on page 524	Enum	4	H+96
18	ExtStatus	Extended solution status. See <i>Table 238: Extended Solution Status</i> on page 1086	Hex	4	H+100
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+104
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## 5.48 TAGGEDMARK1PVA, TAGGEDMARK2PVA, TAGGEDMARK3PVA and TAGGEDMARK4PVA

### *Position, Velocity and Attitude at a Tagged Mark Request*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700



TAGGEDMARKxPVA contains the same information as MARKxPVA with the addition of a unique identifying number (tag).



The TAGGEDMARKxPVA logs available are dependent on the receiver used in the SPAN system. For information about the Event lines supported, see the Strobe Specifications for the receiver in the [OEM7 Installation and Operation User Manual](#), [PwrPak7 Installation and Operation User Manual](#) or [CPT7 Installation and Operation User Manual](#).

The user specifies a TAG for the upcoming TAGGEDMARKxPVA via the **TAGNEXTMARK** command (see page 1055). That tag shows up at the end of this message, which is otherwise identical to the MARKXPVA message.



These logs should only be requested with the ONCHANGED or ONTIME trigger.



These logs provide the position information in the user datum. To determine the datum being used, log the **INSDATUMINFO** log (see page 1096).

**Message ID:**  
**1258 (TAGGEDMARK1PVA)**  
**1259 (TAGGEDMARK2PVA)**  
**1327 (TAGGEDMARK3PVA)**  
**1328 (TAGGEDMARK4PVA)**

**Log Type: Synch**

**Recommended Input:**

```
log taggedmark1pva onnew
log taggedmark2pva onnew
log taggedmark3pva onnew
log taggedmark4pva onnew
```

**Abbreviated ASCII Example:**

```
#TAGGEDMARK1PVAA, COM1, 0, 63.0, FINESTEERING, 1732, 247787.965, 024c0020, ae1e, 12002;
1732, 247787.964913500, 51.11693231436, -114.03884974751, 1046.9481, 0.0001, 0.0007,
0.0004, 1.090392628, 0.766828598, 244.413950146, INS_SOLUTION_GOOD, 1234*34fda4f4
```

```
#TAGGEDMARK2PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,020500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88

#TAGGEDMARK3PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,020500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88

#TAGGEDMARK4PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,020500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	TAGGEDMARK1PVA / TAGGEDMARK2PVA / TAGGEDMARK3PVA / TAGGEDMARK4PVA header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	Week	GNSS Week at Mark input	Ulong	4	H
3	Seconds into Week	GNSS Seconds at Mark input	Double	8	H+4
4	Latitude	Latitude at Mark input (degrees)	Double	8	H+12
5	Longitude	Longitude at Mark input (degrees)	Double	8	H+20
6	Height	Height at Mark input (metres)	Double	8	H+28
7	North Velocity	North Velocity at Mark input (metres/second)	Double	8	H+36
8	East Velocity	East Velocity at Mark input (metres/second)	Double	8	H+44
9	Up Velocity	Up Velocity at Mark input (metres/second)	Double	8	H+52
10	Roll	Roll at Mark input (degrees)	Double	8	H+60
11	Pitch	Pitch at Mark input (degrees)	Double	8	H+68
12	Azimuth	Azimuth at Mark input (degrees)	Double	8	H+76
13	Status	INS Status at Mark input. See <i>Table 237: Inertial Solution Status</i> on page 1081	Enum	4	H+84
14	Tag	Tag ID from the <b>TAGNEXTMARK</b> command (see page 1055), if any (default = 0)	Ulong	4	H+88
15	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+92
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

## 5.49 TSS1

### *TSS1 Protocol for Heave, Roll and Pitch*

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700, SMART7-S

This log provides heave, roll and pitch information in TSS1 protocol.



This message is in a different format than any other log output by the SPAN system.

**Message ID:** 1456

**Log Type:** Synch

**Recommended Input:**

```
log tss1a ontime 1
```

**Message Format:**

```
:XXXXXXXXSMHHHHQMRRRRSMPPPP<CR><LF>
```

**ASCII Example:**

```
:00FFCA -0003F-0325 0319
```

Field	Field Type	Description	Symbol	Example
1	Log header	TSS1 header	:	:
2	Horizontal Acceleration	Horizontal acceleration from 0 to 9.81 m/s <sup>2</sup> . Shown as a one byte unsigned hex number where the least significant bit = 3.83 cm/s <sup>2</sup> .	XX	00
3	Vertical Acceleration	Vertical acceleration from -20.48 to +20.48 m/s <sup>2</sup> . Shown as a two byte hex number where the least significant bit = 0.0625 cm/s <sup>2</sup> .	AAAA	FFCA
4	Space Character	A space delimiter.	S	
5	Heave Polarity	Space if positive. Minus sign (-) if negative.	M	-
6	Heave	Heave value from -99.99 to +99.99 m. Shown as a four digit integer where the least significant bit = 0.01 m.	HHHH	0003
7	Status Flag	F if INS Active. H if INS has not completed an alignment.	Q	F

Field	Field Type	Description	Symbol	Example
8	Roll Polarity	Space if positive. Minus sign (-) if negative.	M	-
9	Roll	Roll value from -99.99 to +99.99 degrees. Shown as a four digit integer where the least significant bit = 0.01 degrees.	RRRR	0325
10	Space Character	A space delimiter.	S	
11	Pitch Polarity	Space if positive. Minus sign (-) if negative.	M	
12	Pitch	Pitch value from -99.99 to +99.99 degrees. Shown as a four digit integer where the least significant bit = 0.01 degrees.	PPPP	0319
13	[CR][LF]	Sentence terminator	<CR><LF>	

## 5.50 VARIABLELEVERARM

### Display Variable Lever Arm Details

**Platform:** OEM719, OEM729, OEM7500, OEM7600, OEM7700, OEM7720, PwrPak7, CPT7, CPT7700

Use this log to redisplay the re-calculated variable lever arm whenever a new **INPUTGIMBALANGLE** command is received. This message is output in the IMU body frame.



The **VARIABLELEVERARM** log will not be published or used internally unless a **SETINSROTATION RBM** command is sent. See the **SETINSROTATION** command on page 1041.



This log should only be requested with the ONCHANGED or ONNEW trigger.

**Message ID:** 1320

**Log Type:** Asynch

**Recommended Input:**

```
log variableleverarma onnew
```

**ASCII Example:**

```
#VARIABLELEVERARMA,USB1,0,63.5,FINESTEERING,2209,494406.137,02000020,4efe,16809;0.000,0.000,0.000,0.000,0.000,0.000*25915585
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log header	VARIABLELEVERARM header For information about log headers, see <i>ASCII</i> on page 33, <i>Abbreviated ASCII</i> on page 35 or <i>Binary</i> on page 37.	-	H	0
2	XOffset	IMU body frame x-axis offset	Double	8	H
3	YOffset	IMU body frame y-axis offset	Double	8	H+8
4	ZOffset	IMU body frame z-axis offset	Double	8	H+16
5	XUncert	X-axis uncertainty in metres	Double	8	H+24
6	YUncert	Y-axis uncertainty in metres	Double	8	H+32
7	ZUncert	Z-axis uncertainty in metres	Double	8	H+40
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+48
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

## Chapter 6 Responses

The receiver is capable of outputting several responses for various conditions. Most responses are error messages to indicate when something is not correct.

The output format of the messages is dependent on the format of the input command. If the command is input as abbreviated ASCII, the output will be abbreviated ASCII. The same rule applies for both ASCII and binary formats.

**Table 264: Response Messages**

ASCII Message	Binary Message ID	Meaning
OK	1	Command was received correctly
Requested log does not exist	2	The log requested does not exist
Not enough resources in system	3	The request has exceeded a limit (for example, the maximum number of logs are being generated)
Data packet doesn't verify	4	Data packet is not verified
Command failed on receiver	5	Command did not succeed in accomplishing requested task
Invalid Message ID	6	The input message ID is not valid
Invalid Message. Field = x	7	Field x of the input message is not correct
Invalid Checksum	8	The checksum of the input message is not correct. Only applies to ASCII and binary format messages.
Message missing field	9	A field is missing from the input message
Array size for field x exceeds max	10	Field x contains more array elements than allowed
parameter x is out of range	11	Field x of the input message is outside the acceptable limits
Message Id already exists in system	12	Message Id already exists in system
Debug token unknown	13	Debug token unknown
Trigger x not valid for this log	14	Trigger type x is not valid for this type of log
Authcode table full - Reload Software	15	Too many authcodes are stored in the receiver. The receiver firmware must be reloaded
Invalid date format	16	This error is related to the inputting of authcodes. Indicates the date attached to the code is not valid



ASCII Message	Binary Message ID	Meaning
Invalid Authcode entered	17	The authcode entered is not valid
No matching model to remove	18	The model requested for removal does not exist
Not valid Auth code for that Model	19	The model attached to the authcode is not valid
Channel is invalid	20	The selected channel is invalid
Requested rate is invalid	21	The requested rate is invalid
Word has no mask for this type	22	The word has no mask for this type of log
Channels locked due to error	23	Channels are locked due to error (the receiver must be reset or power cycled to track satellites again)
Injected time invalid	24	Injected time is invalid
Com port not supported	25	The COM or USB port is not supported
Message is incorrect	26	The message is invalid
Invalid PRN	27	The PRN is invalid
PRN not locked out	28	The PRN is not locked out
PRN lockout list is full	29	PRN lockout list is full
PRN already locked out	30	The PRN is already locked out
Message timed out	31	Message timed out
Unknown COM port requested	33	Unknown COM or USB port requested
Hex string not formatted correctly	34	Hex string not formatted correctly
Invalid baud rate	35	The baud rate is invalid
Message is invalid for this model	36	Message is invalid for this model of receiver
Could Not Save Configuration	38	Could Not Save Configuration
Too Many Configuration Items	39	Too Many Configuration Items
Command only valid if in NVM Fail mode	40	Command is only valid if NVM is in fail mode

<b>ASCII Message</b>	<b>Binary Message ID</b>	<b>Meaning</b>
Invalid offset	41	The offset is invalid
File conflict	43	File conflict
File not found	44	File not found
File open	45	File open
File not open	46	File not open
Invalid DOS FileName	47	Invalid DOS File name
File channel in use	48	File channel in use
File close fail	50	File close fail
Disk not present	51	Disk not present
Disk error	52	Disk error
Disk full	53	Disk full
NVM Write Fail	74	NVM Write Fail
NVM Read Fail	75	NVM Read Fail
Not allowed for input	77	Not allowed for input
Maximum number of user messages reached	78	Maximum number of user messages has been reached
User message decryption failed	79	User message decryption failed
GPS precise time is already known	84	GPS precise time is already known
The message could not be created	87	The message could not be created
Not enough memory to start application	113	Not enough memory to start application
No data available	114	No data available
Invalid handshaking	117	Invalid handshaking
Message name already exists	118	Message name already exists
Invalid message name	119	Invalid message name
The datatype is invalid	120	The data type is invalid
Message ID is reserved	121	Message ID is reserved

ASCII Message	Binary Message ID	Meaning
Message size too large	122	Message size too large
Invalid Security Key	126	Invalid security key
Hardware not available	127	Hardware not available
Requested pulse width is invalid	131	Requested pulse width is invalid
Coarse time is not achieved yet	133	Coarse time is not achieved yet
Invalid Config Code	134	Invalid Config Code
ConfigCode table full - Reload Software	135	Config Code table is full. Reload the software.
Unknown Object Type	136	Unknown object type
This operation is not valid at this time	137	This operation is not valid at this time
User VARF in use	140	User VARF in use
Must enable CLOCKADJUST	141	Must enable CLOCKADJUST. See the <b>CLOCKADJUST</b> command on page 115 for information about enabling.
Disk busy	142	Disk busy
Invalid Word Input Argument	143	Invalid Word Input Argument
Parameter %d is not valid for this model	148	The parameter specified is not valid for this model
IMU SPECS LOCKED FOR THIS IMU TYPE	150	SPAN allows the default specifications for a select few IMUs to be modified to support different variants. However, most IMU specifications are not allowed to change.
Invalid interface mode. Parameter %d	151	The specified Interface mode parameter is not valid.
COMMAND INVALID FOR THIS IMU	154	The entered command cannot be used with the configured IMU. For example, the <b>INSCALIBRATE ANT1</b> command is not valid for lower quality IMUs.
IMU protocol is locked for this IMU type	155	IMU protocol is locked for this IMU type
IMU TYPE IS NOT SUPPORTED WITH CURRENT MODEL	157	A firmware model upgrade is required to use the requested IMU ( <b>CONNECTIMU</b> command on page 1001).

ASCII Message	Binary Message ID	Meaning
Trigger start time is invalid	159	Trigger start time is invalid
Sensor is not initialized	160	Sensor is not initialized
TRIGGER BUFFER IS FULL	161	The <b>TIMEEVENTPULSE</b> command (see page 1056) limit of 10 events has been reached, and a new event cannot be set until an event is cleared.
Board has not achieved finesteering	162	The receiver has not achieved finesteering
SETUPSENSOR COMMAND IS LOCKED	163	The <b>SETUPSENSOR</b> command (see page 1053) cannot be modified because there are remaining trigger events queued.
Invalid Profile Name	165	Invalid Profile Name
Maximum Number Profiles Exceeded	166	The maximum number of profiles is exceeded
Failed To Delete Profile	167	Failed to delete the profile
Profile Name Already Exists	168	Profile name already exists
Total Profile Commands Size Over Limit	169	Total Profile commands size over limit
Cannot Change Profile When Activated	170	Cannot change a Profile when it is activated
Signature Authcode Copy Fail	171	Signature Authcode copy fail
Maximum Number of Profile Commands Exceeded	172	The maximum number of <b>PROFILE</b> commands exceeded
Profile Active, Could Not Save Configuration	173	Profile active, could not save configuration
Current PPP position has bad status and cannot be used for seeding	178	Current PPP position has bad status and cannot be used for seeding
PPP seed position failed integrity check	179	PPP seed position failed integrity check
Invalid password	180	Invalid password
Too many files	181	Too many files

ASCII Message	Binary Message ID	Meaning
Encryption key output is not allowed	186	Encryption key output is not allowed
Secure port requires login	187	Secure port requires login
NMEA2000/J1939 stack is already running on the CAN port	188	NMEA2000/J1939 stack is already running on the CAN port
No saved PPP seed position	191	No saved PPP seed position
System type is invalid for this model	192	System type is invalid for this model
Command is not supported for this model	193	Command is not supported for this model
Position Averaging Not Started	194	Position averaging not started
Not in GLIDE mode	200	Not in GLIDE mode
PPP seeding invalid in forced dynamics mode	201	PPP seeding invalid in forced dynamics mode
Wrong combination of parameters	202	Wrong combination of parameters
Invalid Calibration Request	203	Invalid calibration request
Active Gimbal Detected	204	Active gimbal detected
Authcode table full - Use auth erase_table	205	Authcode table full. An authcode must be removed before another authcode can be added. Refer to the <b>AUTH</b> command (see page 84) for instructions on removing authcodes and cleaning up the authcode table.
Profile Not Running - Profile should be activated	206	Profile not running - Profile should be activated
ID provided is already in use	208	ID provided is already in use
ID provided does not exist	209	ID provided does not exist
Calibration already in progress	210	Calibration already in progress

ASCII Message	Binary Message ID	Meaning
Filter cannot be enabled due to channel speed settings	211	Filter cannot be enabled due to channel speed settings
Notch Filter and Frequency are mismatching	212	Notch filter and frequency are mismatching
Filter can not cascade	213	Filter can not cascade
There is no RF filter applied	214	There is no RF filter applied
ID provided should be 4 character long	215	ID provided should be 4 characters long
Invalid subscription code	216	Invalid subscription code
Subscription table full	217	Subscription table full
Network id does not match subscription code	218	Network ID does not match the subscription code
Subscription not found	219	Subscription not found
Subscription not active	220	Subscription not active
Cannot activate expired subscription	221	Cannot activate expired subscription
Maximum number of logs exceeded. No new log added.	222	Maximum number of logs exceeded. No new log added.
Seed is too far in the past	223	Seed is too far in the past
Final log request must use the ONCE trigger	224	Final log request must use the ONCE trigger
Channel invalid for region x	225	Channel invalid for region x
Region not set	226	Region not set
Estimated RBV must be entered first	227	Initial RBV estimate is required before RBV calibration
Command failed because WIFIALIGNAUTOMATION is enabled	240	Command failed because WIFIALIGNAUTOMATION is enabled. See the <b>WIFIALIGNAUTOMATION</b> command on page 469.
Specified network not enabled with WIFINETCONFIG command	241	Specified network not enabled with WIFINETCONFIG command. See the <b>WIFINETCONFIG</b> command on page 478.

## APPENDIX A Example of Bit Parsing a RANGEEMP4 Log

The following takes a sample RANGEEMP4 log and breaks it down into its raw components.

Data was captured in both RANGE and in RANGEEMP4 logs which are shown here for reference. These logs were captured at a rate of 4 Hz such that the Reference and Differential parts of the RANGEEMP4 could be explained.



Some of the RANGEEMP4 values will have some very slight differences (at the millicycle level) compared to the equivalent RANGE log data due to truncating the double values into integers.

Here are two RANGE logs to reference against once the RANGEEMP4 logs have been determined:

```
RANGE COM1 0 88.5 FINESTEERING 1919 507977.000 02000020 5103 32768
22
27 0 21761200.335 0.036 -114355879.993103 0.006 1121.758 50.0 876.785
18109c04
27 0 21761202.795 0.128 -89108485.029683 0.007 874.097 44.2 862.386 11303c0b
27 0 21761200.306 0.007 -85395622.838987 0.004 837.685 51.7 865.845 01d03c04
21 0 21214757.684 0.027 -111484302.588995 0.005 -1107.624 52.6 888.968
08109c24
21 0 21214757.049 0.122 -86870882.607297 0.006 -863.084 44.6 874.389 01303c2b
10 0 21540290.811 0.027 -113194996.162910 0.005 2288.688 52.6 889.905
08109c44
10 0 21540293.632 0.110 -88203904.731314 0.006 1783.394 45.6 868.725 01303c4b
10 0 21540289.869 0.006 -84528728.138216 0.004 1709.022 53.0 872.386 01d03c44
15 0 21776375.653 0.032 -114435625.391762 0.007 -1814.485 50.9 879.586
18109c64
15 0 21776376.038 0.129 -89170616.457446 0.007 -1413.886 44.1 862.706
11303c6b
18 0 20493192.703 0.031 -107692454.149639 0.007 212.747 51.1 891.550 08109c84
18 0 20493191.933 0.105 -83916195.494946 0.007 165.777 45.9 874.710 01303c8b
61 9 20375330.794 0.104 -108956045.737322 0.006 -3039.481 46.8 891.931
08119ca4
61 9 20375332.806 0.083 -84743599.055547 0.007 -2364.042 34.0 876.813
00b13cab
55 4 22748433.080 0.146 -121432681.638722 0.009 4061.119 43.9 416.032
18119cc4
55 4 22748438.602 0.021 -94447660.068923 0.009 3158.651 46.0 415.562 00b13ccb
38 8 19781617.845 0.058 -105744080.698106 0.004 -2024.611 51.8 893.563
18119ce4
38 8 19781623.453 0.032 -82245418.313339 0.005 -1574.698 42.2 878.833
00b13ceb
39 3 19968976.955 0.055 -106558290.405759 0.004 2248.713 52.3 875.210
08119d04
39 3 19968980.676 0.019 -82878686.553631 0.005 1749.000 46.9 870.890 00b13d0b
54 11 19507573.213 0.059 -104388964.028915 0.005 1289.410 51.8 894.613
08119d24
54 11 19507576.477 0.017 -81191427.275619 0.004 1002.874 48.0 878.832
10b13d2b
```

```

RANGE COM1 0 88.5 FINESTEERING 1919 507977.250 02000020 5103 32768
22
27 0 21761146.982 0.036 -114355599.642256 0.006 1121.140 49.9 877.035
18109c04
27 0 21761149.447 0.122 -89108266.573995 0.007 873.616 44.6 862.636 11303c0b
27 0 21761146.957 0.007 -85395413.484293 0.004 837.294 51.8 866.095 01d03c04
21 0 21214810.390 0.027 -111484579.560955 0.005 -1108.100 52.6 889.218
08109c24
21 0 21214809.754 0.120 -86871098.429369 0.005 -863.454 44.8 874.639 01303c2b
10 0 21540181.949 0.027 -113194424.080322 0.005 2288.176 52.6 890.155
08109c44
10 0 21540184.767 0.111 -88203458.952394 0.006 1782.995 45.4 868.975 01303c4b
10 0 21540181.003 0.006 -84528300.928648 0.004 1708.751 53.0 872.636 01d03c44
15 0 21776461.990 0.032 -114436079.084785 0.006 -1814.956 50.9 879.836
18109c64
15 0 21776462.375 0.129 -89170969.984233 0.007 -1414.253 44.1 862.956
11303c6b
18 0 20493182.598 0.031 -107692401.054068 0.007 212.183 51.2 891.800 08109c84
18 0 20493181.833 0.110 -83916154.122137 0.007 165.338 45.6 874.960 01303c8b
61 9 20375472.914 0.104 -108956805.696703 0.006 -3040.142 46.9 892.181
08119ca4
61 9 20375474.924 0.084 -84744190.134355 0.007 -2364.555 33.9 877.063
00b13cab
55 4 22748242.897 0.150 -121431666.427728 0.009 4060.804 43.7 416.282
18119cc4
55 4 22748248.421 0.021 -94446870.460803 0.009 3158.405 46.0 415.812 00b13ccb
38 8 19781712.549 0.059 -105744586.938646 0.004 -2025.149 51.8 893.813
18119ce4
38 8 19781718.158 0.032 -82245812.055601 0.005 -1575.117 42.3 879.083
00b13ceb
39 3 19968871.615 0.055 -106557728.318448 0.004 2248.162 52.3 875.460
08119d04
39 3 19968875.343 0.019 -82878249.374953 0.005 1748.571 46.8 871.140 00b13d0b
54 11 19507512.994 0.059 -104388641.780659 0.005 1288.778 51.7 894.863
08119d24
54 11 19507516.256 0.016 -81191176.637999 0.005 1002.383 48.1 879.082
10b13d2b

```

Here are the equivalent RANGECMP4 logs which will be broken down into their individual components:

```

#RANGECMP4A, COM1, 0, 88.5, FINESTEERING, 1919, 507977.000, 02000020, fb0e, 32768;295,
030000421204000000009200df7688831f611fd87ca0b03a00638bbdf7b82f49b080fd0ec0ff1f0
91f8214ff4d4d00a1009cbf1751f6911f5141f87fd9571a96dbd7040c8090f87f0080fcf722fe9b
fa8a49a8ff4f299d7f96fb9afefc771800fcffd0063f02cde01f3c7dd3fffb75240886f5fa2b0ff9
1f57f00003edf8b78868c882878014065dbf7d3ed6b722680d5fc0f00a4c08730fe7fecf8bffa3f
003008000000002001f03fa019f8136a11273649b8fcefab9c434c7b89e71560dbfe070030b2e04
fd841f33125320b80b0ecef5ee21243ac0bb03e0ffc36a813fb13bbe5791a0f5ff9e3bdbffbb87
f0cb8064f03f0000e4b67dd15bc5f4a50a3a006ca72fdee53ec86405b2c0fffa3fa450f725d5bfe
d7c49b1fb0fb16b45a87a9adb0740cbfe0700*7DD8F893

```

```

#RANGECMP4A, COM1, 0, 88.5, FINESTEERING, 1919, 507977.250, 02000020, fb0e, 32768;239,
030000421204000000009200dff688831f6102005500e70162dc977c004015c07988840f6101803
a805921cedf8b80002011207080e5f6351f003804081c2200be0808005c01620808725f93028057
801822dae0476000a00f207180fef6251700e803401c62f3bdc8060052013009986f5f220200540
04ca2053ec408005401ca870180410000000000980ff6306fec408004801de07c8692f51028051
80f721b2e04f600040152081804ef7102500600540202205fe040a0086013a0938780f610200618

```



```
04e224edbdb68002010c0498030f7411d0018047812a2d47d090a004c01a609c8544f6202805200
6a02*48E189A2
```

## A.1 Reference Log Decoding

The RANGECMP4 log at time 507977.0 will be decoded first:

```
#RANGECMP4A, COM1, 0, 88.5, FINESTEERING, 1919, 507977.000, 02000020, fb0e, 32768;295,
030000421204000000009200df7688831f611fd87ca0b03a00638bbdf7b82f49b080fd0ec0ff1f0
91f8214ff4d4d00a1009cbf1751f6911f5141f87fd9571a96dbd7040c8090f87f0080fcf722fe9b
fa8a49a8ff4f299d7f96fb9afe9c771800fcffd0063f02cde01f3c7dd3fffb75240886f5fa2b0ff9
1f57f00003edf8b78868c882878014065dbf7d3ed6b722680d5fc0f00a4c08730fe7fecf8bffa3f
003008000000002001f03fa019f8136a11273649b8fcefab9c434c7b89e71560dbfe070030b2e04
fd841f33125320b80b0ecef5ee21243ac0bb03e0ffc36a813fb13bbe5791a0f5ff9e3bdbffbb87
f0cb8064f03f0000e4b67dd15bc5f4a50a3a006ca72fdee53ec86405b2c0fffa3fa450f725d5bfe
d7c49b1fb0fb16b45a87a9adb0740cbfe0700*7DD8F893
```

Since this log falls on a whole second (507977.000), it is a Reference log.

At the start of the RANGECMP4 log is the identifier for how many bytes are in the log. In this case, there are 295 bytes. The rest of the message is compressed binary data and is transmitted as LSB first so the bytes must be swapped before processing.

### A.1.1 Reference Header

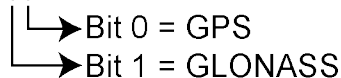
The Reference Header is sent once per message. See *Table 168: Header* on page 825 in the RANGECMP4 log section.

Decoding the bits starting with the first bytes:

*GNSS Field* (16 bits)

- Grab the first 2 bytes (16 bits) = 0x0300
- Swap the bytes = 0x0003
- 0x0003 in binary form = 0000 0000 0000 0011

```
0000 0000 0000 0011
```



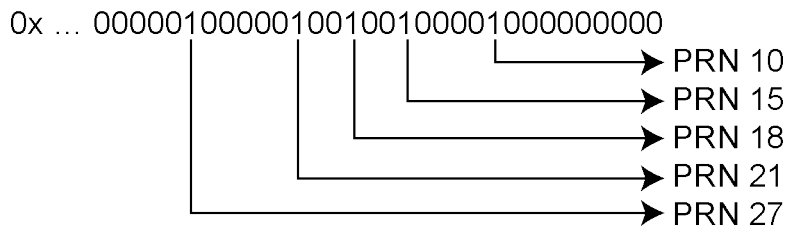
In this example the receiver was configured to track only GPS and GLONASS systems. If other systems had been in the configuration and tracked, they would have shown here.

### A.1.2 Reference Satellite and Signal Block: GPS

This block is sent once for each bit set to 1 in the GNSS field (See *Table 168: Header* on page 825). As identified by the above GNSS field, the first system (right to left) is the GPS System. Use *Table 169: Satellite and Signal Block* on page 825 to determine what satellites and signals data are contained in this GPS system:

*GPS Satellites field* (64 bits)

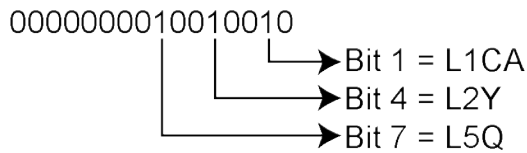
- Grab the next 8 bytes (64 bits) = 0x0042120400000000
- Swap the bytes = 0x0000000004124200
- 0x0000000004124200 in binary form =



- The 1's above identify that there are 5 tracking GPS PRNs.

*GPS Signals field (16 bits)*

- Grab the next 2 bytes (16 bits) = 0x9200
- Swap the bytes = 0x0092
- 0x0092 in binary form =



- The 1's above identify that there are 3 tracking GPS signals: L1CA, L2Y, and L5Q.

*GPS Included Signals field (5 PRNs x 3 Signals = 15 bits – Therefore need 2 bytes)*

- Up to the point of processing the Included Signals field, the bytes are aligned such that the bits start and end within each batch of bytes. After processing this step, it is quite common for the Included Signals Field (mxn matrix) to not be divisible by 8 so bytes not processed will need to be carried over to the next section depending on the size of the matrix.
- Grab the next 2 bytes (16 bits) = 0xdf76
- Swap the bytes = 0x76df
- 0x76df in binary form = 0111011011011111
- Only need 15 of the 16 bits = X111011011011111
- This bit string breaks down into 5 rows (PRNs) and 3 columns (signals) as specified by the mxn (PRN x signals) parameters. Take the bit string and break it up into sets of 3 starting at the MSB. This will result with the lowest PRN being at the bottom row of the stack and the first signal (L1CA) being the far right column.

```

111
011
011
011
111

```

- This stack can be further broken apart to identify the PRNs vs. their Signals:

PRN	L5Q	L2Y	L1CA
27	1	1	1
21	0	1	1
18	0	1	1
15	0	1	1
10	1	1	1

### A.1.3 Reference Measurement Block Header: GPS

This block is sent once for each bit set to 1 in the Satellites field found in *Table 169: Satellite and Signal Block* on page 825. Now that the PRN's signals have been determined, the next step is to determine the specifics of the first PRN (10) and its list of signals (L1CA, L2Y, L5Q). Working from bottom right to upper left of the PRN/Signal chart above, each 1 represents a signal for a PRN. Use *Table 170: Measurement Block Header* on page 826 to determine the contents of each field:

*GPS PRN 10* (first PRN found in the Satellites field)

We will grab enough bytes to process the whole Measurement Block Header. If this was a GLONASS System, a total of 9 bits would be required for this step (1 bit for the Data Format Flag, 3 bits for the Ref Data Block ID, plus 5 bits for the GLONASS Frequency Number). Since this is a GPS system, only 4 bits in total are required (1 bit for the Data Format Flag and 3 bits for the Ref Data Block ID).

There was 1 bit not processed in the last byte so that byte will be carried forward. Only 4 bits need to be looked at for this step so grab the next byte as well:

- Use the last byte (0x76) plus the next byte (0x88)= 0x7688
- Swap the bytes = 0x8876
- 0x8876 in binary form = 1000100001110110
- Ignore the 7 processed bits from the last step = 100010000XXXXXX
- Ignore the 5 MSB bits leaving 4 bits for processing =

XXXXX0000XXXXXX

0 = Data Format Flag (1 bit)  
000 = Ref Data Block (3 bits)

The Data Format Flag identifies that this batch of data is Reference (0) data.  
The Ref Data Block ID is 0x000.



The 5 MSBs have not been processed so this byte will be carried forward.

The Data Format Flag identifies if the upcoming data is Reference or Differential data. By default every log that was published on a whole second will always be Reference logs. Logs between seconds will be Differential logs but could be Reference logs depending on the compression calculations. If a discontinuity occurred that made it impossible for a Differential calculation to fit within the Differential Constraints, it will revert to a Reference log.

### A.1.4 Reference Measurement Block: GPS

This block is sent once for each bit set to 1 in the Included Signals Field found in *Table 169: Satellite and Signal Block* on page 825. Use *Table 171: Primary Reference Signal Measurement Block* on page 827 and *Table 172: Secondary Reference Signals Measurement Block* on page 828 to determine the contents of each field:

A Measurement Block for a single PRN will look like the following:

Primary Parity Flag  
Primary ½ Cycle Slip Flag  
Primary C/No  
Primary Lock Time  
Primary Pseudorange Std Deviation

Primary Phasorange Std Deviation  
Primary Pseudorange  
Primary Phasorange - Primary Pseudorange (determines the Phasorange for the 1<sup>st</sup> Signal)  
Primary Doppler

2<sup>nd</sup> Parity Flag  
2<sup>nd</sup> ½ Cycle Slip Flag  
2<sup>nd</sup> C/No  
2<sup>nd</sup> Lock Time  
2<sup>nd</sup> Pseudorange Std Deviation  
2<sup>nd</sup> Phasorange Std Deviation  
2<sup>nd</sup> Pseudorange - Primary Pseudorange (determines the Pseudorange for the 2<sup>nd</sup> Signal)  
2<sup>nd</sup> Phasorange - 2<sup>nd</sup> Pseudorange (determines the Phasorange for the 2<sup>nd</sup> Signal)  
2<sup>nd</sup> Doppler - Primary Doppler (determines the Doppler for the 2<sup>nd</sup> Signal)

3<sup>rd</sup> Parity Flag  
3<sup>rd</sup> ½ Cycle Slip Flag  
3<sup>rd</sup> C/No  
3<sup>rd</sup> Lock Time  
3<sup>rd</sup> Pseudorange Std Deviation  
3<sup>rd</sup> Phasorange Std Deviation  
3<sup>rd</sup> Pseudorange - Primary Pseudorange (determines the Pseudorange for the 3<sup>rd</sup> Signal)  
3<sup>rd</sup> Phasorange - 3<sup>rd</sup> Pseudorange (determines the Phasorange for the 3<sup>rd</sup> Signal)  
3<sup>rd</sup> Doppler - Primary Doppler (determines the Doppler for the 3<sup>rd</sup> Signal)

...

### **A.1.5 Reference Primary Signal Measurement Block: GPS PRN 10 – L1CA**

The next bytes collected will be for the GPS PRN 10 - L1CA signal data. This is the primary signal of the PRN since it is the first signal. As a result, its Measurement Block consists of 111 bits as listed in *Table 171: Primary Reference Signal Measurement Block* on page 827. Since 111 bits takes up a lot of space, these bits will be split into two groups from *Table 171: Primary Reference Signal Measurement Block* on page 827: the top 25 bits for signal info followed by the bottom 86 bits for signal data.

The signal info section (top 25 bits) is processed as follows:

- With 5 bits left unprocessed from the previous byte, we calculate  $25 - 5 = 20$  bits which rounds up to 3 bytes. Therefore the previous last byte (0x88) plus the next 3 bytes will be needed.

- Use the last byte (0x88) plus grab 3 bytes (x831f61) = 0x88831f61
- Swap the bytes = 0x611f8388
- 0x611f8388 in binary form = 01100001000111111000001110001000
- The previous step used the 3 LSBs = 01100001000111111000001110001XXX
- 25 bits are needed so ignore the 4 MSBs =

```

xxxx0001000111111000001110001xxx
                                1   = Parity Flag
                                0   = ½ Cycle Slip Flag
                        10000011100   = C/No
                1111                  = Lock Time
        0001                          = Pseudorange Std Deviation
0001                                  = Phasorange Std Deviation
    
```

- Parity flag is a 1 (Parity Known)
- ½ Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
0x10000011100b = 1052 x Scaling Factor of 0.05  
= 52.60 dBHz
- The Lock Time value is:  
0x1111b = 15 which means that this signal has been locked for 262144 ms or more.
- The Pseudorange Std Deviation value is:  
0x0001b = 1 which means: 0.020 m < PSR Std Dev <= 0.030 m using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
0x0001b = 1 which means: 0.0039 < ADR Std Dev <= 0.0052 cycles using *Table 177: ADR Std Dev* on page 833.

The signal data section (bottom 86 bits) is processed as follows:

- With 4 bits unprocessed from the previous byte, we calculate 86 – 4 = 82 bits = 11 bytes (2 bits will not be processed in the last byte).
  - Use the last byte (0x61) plus grab 11 bytes (0x1fd87ca0b03a00638bbdf7)  
= 0x611fd87ca0b03a00638bbdf7
  - Swap the bytes = 0xf7bd8b63003ab0a07cd81f61
  - 0xf7bd8b63003ab0a07cd81f61 in binary form =  
111 0111 1011 1101 1000 1011 0110 0011 0000 0000 0011 1010 1011 0000 1010 0000 0111 1100  
1101 1000 0001 1111 0110 0001
  - Only need 86 bits. Ignore last 4 LSBs and first 6 MSBs =

```

xxxx xx11 1011 1101 1000 1011 0110 0011 0000 0000 0011 1010 1011 0000 1010 0000 0111 1100 1101 1000 0001 1111 0110 xxxx
                                0 1010 0000 0111 1100 1101 1000 0001 1111 0110
                                0000 0000 0011 1010 1011 000
11 1011 1101 1000 1011 0110 0011
                                = 1st Pseudo
                                = 1st Phase - 1st Pseudo
                                = 1st Doppler
    
```

- Use *Table 171: Primary Reference Signal Measurement Block* on page 827 to identify if a 2's Complement Conversion is needed as well as what Scale Factor should be used before these binary numbers are used in the following calculations.
- The 1st (Primary) Pseudorange is processed by:

1st Pseudorange =  $0x0101000000111110011011000000111110110b \times \text{Scaling Factor}$

1st Pseudorange =  $43080581622 \times 0.0005$

L1CA Pseudorange for PRN 10 =  $21540290.811 \text{ m}$

- The 1st (Primary) Phasorange is a 2's Complement number (as identified by the Range column in *Table 171: Primary Reference Signal Measurement Block* on page 827) so it is processed in the following manner:

1st Phasorange – 1st Pseudorange =  $2\text{'s Complement}(0x0000000001110101011000b) \times \text{Scaling Factor}$

1st Phasorange –  $21540290.811 \text{ m} = 7512 \times 0.0001$

L1CA Phasorange =  $21540291.5622 \text{ m}$

- Convert this to ADR to check against the original RANGE log:

ADR =  $1\text{st Phasorange} \times \text{Frequency} \times (-1)/\text{Speed Of Light}$

ADR =  $21540291.5622 \text{ m} \times 1575420000 \text{ Hz} \times (-1)/299792458 \text{ m/s}$

L1CA ADR for PRN 10 =  $-113194996.1627158 \text{ cycles}$



In the range logs, PSR and ADR have opposite signs.

- The 1st (Primary) Doppler is a 2's Complement number (as identified by the Range column in *Table 171: Primary Reference Signal Measurement Block* on page 827) so it is processed in the following manner:

1st Doppler(m/s) =  $2\text{'s Complement}(0x11101111011000101101100011b) \times \text{Scaling Factor}$

1st Doppler(m/s) =  $-4,355,229 \times 0.0001$

L1CA Doppler(m/s) =  $-435.5229 \text{ m/s}$

Convert the Doppler to Hz:

1st Doppler(Hz) =  $1\text{st Doppler(m/s)} \times \text{Frequency} \times (-1)/\text{Speed Of Light}$

L1CA Doppler(Hz) for PRN 10 =  $2288.6883 \text{ Hz}$

1st Doppler(Hz) =  $-435.5229 \text{ m/s} \times 1575420000 \text{ Hz} \times (-1)/299792458 \text{ m/s}$

### A.1.6 Reference Secondary Signals Measurement Block: GPS PRN 10 – L2Y

Signal L1CA was the 1st signal (Primary Signal) of the three PRN 10 signals found in this RANGEEMP4 log data. L1CA's data is now used to determine the L2Y's signals data. Since this is the second signal block of this PRN, its data will be processed by using *Table 172: Secondary Reference Signals Measurement Block* on page 828.

With 6 bits left unprocessed from the previous byte, we will require  $82 - 6 = 76$  bits which rounds up to 10 bytes.

- Use the last byte (0xf7) plus grab the next 10 bytes (0xb82f49b080fd0ec0ff1f)  
=  $0xf7b82f49b080fd0ec0ff1f$
- Swap the bytes =  $0x1ffc00efd80b0492fb8f7$
- $0x1ffc00efd80b0492fb8f7$  in binary form =  
0001 1111 1111 1111 1100 0000 0000 1110 1111 1101 1000 0000 1011 0000 0100 1001 0010 1111 1011  
1000 1111 0111
- Only need 78 bits. The 2 LSBs are ignored as they were already processed above and the 4 MSBs are


ignored so there is a total of 82 bits to process

```

xxxx 1111 1111 1111 1100 0000 0000 1110 1111 1101 1000 0000 1011 0000 0100 1001 0010 1111 1011 1000 1111 01xx
                                                                 1 = Parity Flag
                                                                 0 = 1/2 Cycle Slip Flag
                                                                 = C/No
                                                                 = Lock Time
                                                                 = Pseudorange Std Deviation
                                                                 = Phaserange Std Deviation
                                                                 = Pseudo - 1st Pseudo
                                                                 = Phase - Pseudo
                                                                 = Doppler - 1st Doppler
                                                                 011 1000 1111
                                                                 111 1
                                                                 010 1
                                                                 001 0
00 0000 0000 1110 1111 1101 1
1111 1111 1111 11
    
```

Use *Table 172: Secondary Reference Signals Measurement Block* on page 828 to identify if a 2's Complement Conversion is needed as well as what Scale Factor should be used before these binary numbers are used in the following calculations.

- Parity flag is a 1 (Parity Known)
- 1/2 Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
 $0x01110001111b = 911 \times \text{Scaling factor of } 0.05$   
 $= 45.55 \text{ dBHz}$
- The Lock Time value is:  
 $0x1111b = 15$  which means that this signal has been locked for 262144 ms or more.
- The Pseudorange Std Deviation value is:  
 $0x0101b = 5$  which means:  $0.099 \text{ m} < \text{PSR Std Dev} \leq 0.148 \text{ m}$  using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
 $0x0010b = 2$  which means:  $0.0052 < \text{ADR Std Dev} \leq 0.0070$  cycles using *Table 177: ADR Std Dev* on page 833.
- The L2Y Pseudorange is a 2's Complement number (as identified by the Range column in *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is processed in the following manner:  
 $\text{Pseudorange} - 1\text{st Pseudorange} = 2\text{'s Complement}(0x00000001011000001001b) \times \text{Scaling Factor}$   
 $\text{Pseudorange} - 21540290.811 \text{ m} = 5641 \times 0.0005$   
 $\text{L2Y Pseudorange} = 21540293.6315 \text{ m}$
- The L2Y Phaserange is a 2's Complement number (as identified by the Range column in *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is calculated in the following manner:  
 $\text{Phaserange} - \text{Pseudorange} = 2\text{'s Complement}(0x00000000001110111111011b) \times \text{Scaling Factor}$   
 $\text{Phaserange} - 21540293.6315 \text{ m} = 7675 \times 0.0001$   
 $\text{L2Y Phaserange} = 21540294.399 \text{ m}$
- Convert this to ADR to check against the original RANGE log:  
 $\text{ADR} = \text{Phaserange} \times \text{Frequency} \times (-1) / \text{Speed Of Light}$   
 $\text{ADR} = 21540294.399 \text{ m} \times 1227600000 \text{ Hz} \times (-1) / 299792458 \text{ m/s}$   
 $\text{L2Y ADR for PRN 10} = -88203904.73002626 \text{ cycles}$

 In the range logs, PSR and ADR have opposite signs.

- The L2Y Doppler is a 2's Complement number (as identified by the Range Column in *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is calculated in the following manner:

Doppler(m/s) – 1st Doppler(m/s) = 2's Complement(0x11111111111111b) x Scaling Factor  
 Doppler(m/s) – (-435.5229 m/s) = (-1) x 0.0001  
 L2Y Doppler(m/s) = -435.5228 m/s

Convert the Doppler to Hz:

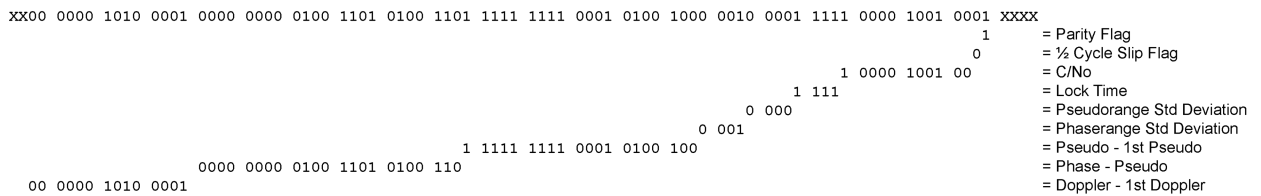
Doppler(Hz) = Doppler(m/s) x Frequency \* (-1)/Speed Of Light  
 Doppler(Hz) = -435.5228 m/s x 1227600000 Hz \* (-1)/299792458 m/s  
 L2Y Doppler(Hz) for PRN 10 = 1783.3938 Hz

**A.1.7 Reference Third Signals Measurement Block: GPS PRN 10 – L5Q**

Signal L1CA was the 1st signal (Primary Signal) of the three PRN 10 signals found in this RANGEEMP4 log data. L1CA's data is now used to determine the L5Q's signals data. Since this is the third signal block of this PRN, its data will be processed using *Table 172: Secondary Reference Signals Measurement Block* on page 828.

With 4 bits left unprocessed from the previous byte, we will require 82 – 4 = 78 bits which rounds up to 10 bytes.

- Use the last byte (0x1f) plus grab the next 10 bytes (0x091f8214ff4d4d00a100) = 0x1f091f8214ff4d4d00a100
- Swap the bytes = 0x00a1004d4dff14821f091f
- 0x00a1004d4dff14821f091f in binary form =  
 0000 0000 1010 0001 0000 0000 0100 1101 0100 1101 1111 1111 0001 0100 1000 0010 0001 1111 0000 1001 0001 1111
- Only need 78 bits. The 4 LSBs are ignored as they were already processed above and the 2 MSBs are ignored so there is a total of 82 bits to process



Use *Table 172: Secondary Reference Signals Measurement Block* on page 828 to identify if a 2's Complement Conversion is needed as well as what Scale Factor should be used before these binary numbers are used in the following calculations.

- Parity flag is a 1 (Parity Known)
- 1/2 Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
 0x10000100100b = 1060 x Scaling Factor of 0.05  
 = 53.00 dBHz
- The Lock Time value is:  
 0x1111b = 15 which means that this signal has been locked for 262144 ms or more.
- The Pseudorange Std Deviation value is:  
 0x0000b = 0 which means: PSR Std Dev <= 0.020 m using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
 0x0001b = 1 which means: 0.0039 < ADR Std Dev <= 0.0052 cycles using *Table 177: ADR Std Dev* on page 833.



- The L5Q Pseudorange is a 2's Complement number (as identified by Range column in *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is processed in the following manner:

Pseudorange – 1st Pseudorange = 2's Complement(0x11111111100010100100b) x Scaling Factor

Pseudorange – 21540290.811 m = (-1884) x 0.0005

L5Q Pseudorange = 21540289.869 m

- The L5Q Phaserange is a 2's Complement number (as identified by the Range column in *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is calculated in the following manner:

Phaserange – Pseudorange = 2's Complement(0x00000000010011010100110b) \* Scaling Factor

Phaserange – 21540289.869 m = 9894 \* 0.0001

L5Q Phaserange = 21540290.8584 m

- Convert this to ADR to check against the original RANGE log:

ADR = Phaserange \* Frequency \* (-1)/Speed Of Light

ADR = 21540290.8584 m \* 1176450000 Hz \* (-1)/299792458 m/s

L5Q ADR for PRN 10 = -84528728.13886692 cycles



In the range logs, PSR and ADR have opposite signs.

- The L5Q Doppler is a 2's Complement number (as identified by the Range column *Table 172: Secondary Reference Signals Measurement Block* on page 828) so it is calculated in the following manner:

Doppler(m/s) – 1st Doppler(m/s) = 2's Complement(0x00000010100001b) x Scaling Factor

Doppler(m/s) – (-435.5229 m/s) = 80 x 0.0001

L5Q Doppler(m/s) = -435.5149 m/s

Convert the Doppler to Hz:

Doppler(Hz) = Doppler(m/s) x Frequency \* (-1)/Speed Of Light

Doppler(Hz) = -435.5149 m/s x 1176450000 Hz \* (-1)/299792458 m/s

L5Q Doppler(Hz) for PRN 10 = 1709.054 Hz

This concludes the processing of the signals present for PRN 10.

The next PRN as identified in the GPS Included Signals Field is PRN 15 with 2 signals. Processing of this data would be handled as described above, starting with the 4 bit Measurement Block followed by the individual signals. This would be followed by PRN 18, 21, and 27. Processing these remaining PRNs and their signals would use up the next 870 bits as shown below:

Bits required for remaining GPS PRNs and Signals:

PRN 15

- 4 bits Measurement Block header
- 111 bits - 1st Signal
- 82 bits - 2nd Signal

PRN 18

- 4 bits Measurement Block header
- 111 bits - 1st Signal
- 82 bits - 2nd Signal

## PRN 21

- 4 bits Measurement Block header
- 111 bits - 1st Signal
- 82 bits - 2nd Signal

## PRN 27

- 4 bits Measurement Block header
- 111 bits - 1st Signal
- 82 bits - 2nd Signal
- 82 bits - 3rd Signal

Total = 870 bits

There are 2 bits left unprocessed from the last byte of PRN 10's processing so 868 more bits (109 bytes) are required. After processing the remaining GPS data, there will be 4 bits left from the last byte to start off the next system (GLONASS as identified by the GNSS field in the Header).

After the last GPS bit, the GLONASS system will then be processed since it was identified as the next system by the GNSS field in the Header.

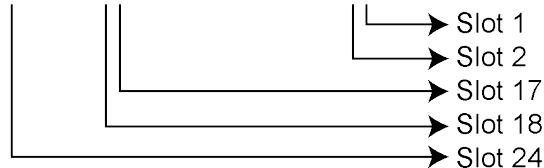
### A.1.8 Reference Satellite and Signal Block: GLONASS

This block is sent once for each bit set to 1 in the GNSS field found in *Table 168: Header* on page 825. As identified by the above GNSS field, the second system (right to left) is the GLONASS System. Use *Table 169: Satellite and Signal Block* on page 825 to determine what satellites slots and signals data are contained in this GLONASS System:

GLONASS Satellites field (64 bits)

- Grab the next 8 bytes (64 bits) = 0x3f0030080000000020
- Swap the bytes = 0x20000000000830003f
- 0x20000000000830003f in binary form =  
001000000000000000000000000000000100000110000000000000111111
- Mask out the used 4 LSBs =  
00100000000000000000000000000000010000011000000000000011XXXX
- Determine the required 64 bits =

XXXX00100000110000000000000011XXXX



- The 1's above identify that there are 5 tracking GLONASS Slots.
- The present GLONASS satellite PRNs/Slot ID's (when between 1 to 24) are therefore (37 + Slot ID):


Slot 1 = PRN 38

Slot 2 = PRN 39

Slot 17 = PRN 54

Slot 18 = PRN 55

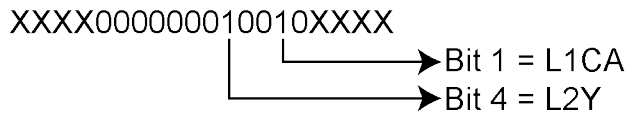
Slot 24 = PRN 61



If the GLONASS Slot ID was between 43 and 64, this would represent a GLONASS satellite that has an unknown Slot ID and is instead assigned a temporary one based upon 64 minus the unadjusted GLONASS Frequency Number (0 to 20). This Slot ID will be updated once the actual PRN/Slot ID has been determined.

*GLONASS Signals field (16 bits)*

- Append the next 2 bytes (0x01f0) to the last byte (0x20) = 0x2001f0
- Swap the bytes = 0x0f0120
- 0x0f0120 in binary form = 11110000000100100000
- Ignore the processed bits = 1111000000010010XXXX
- Determine the required 16 bits =



- The 1's above identify that there are 2 tracking GLONASS signals: L1CA and L2P.

*GLONASS Included Signals field (5 Slot ID's x 2 Signals = 10 bits)*

- Append the next byte (0x3f) to the last byte (0xf0) = 0xf03f
- Swap the bytes = 0x3ff0
- 0x3ff0 in binary form = 0011111111110000
- Ignore the processed bits = 001111111111XXXX
- Determine the required 10 bits = XX1111111111XXXX
- This bit string breaks down into 5 rows (Slots) and 2 columns (signals) as specified by the mxn (Slot IDs x signals) parameters. Take the bit string and break it up into sets of 2 starting at the MSB. This will result with the lowest Slot ID being at the bottom row of the stack and the first signal (L1CA) being the far right column.

```

11
11
11
11
11
    
```

- This stack can be further broken apart to identify the Slot ID's vs. their Signals:

SLOT	L2P	L1CA
24	1	1
18	1	1
17	1	1
2	1	1
1	1	1

### A.1.9 Reference Measurement Block Header: GLONASS PRN 38

(Slot 1 which was the first Slot found in the Satellites Field)

We will grab enough bytes to process the whole Measurement Block Header. Since this is a GLONASS System, a total of 9 bits will be required for this step (1 bit for the Data Format Flag, 3 bits for the Ref Data Block ID, plus 5 bits for the GLONASS Frequency Number).

With 2 bits left unprocessed from the previous byte, we will require  $9 - 2 = 7$  bits which rounds up to 1 byte:

- Use the last byte (0x3f) plus the next byte (0xa0) = 0x3fa0
- Swap the bytes = 0xa03f
- 0xa03f in binary form = 1010000000111111
- Ignore the 6 processed bits from the last step = 1010000000XXXXXX
- Ignore the 1 MSB bits leaving 9 bits for processing =

```
x010000000XXXXXX
      0          = Data Format Flag (1 bit)
     000        = Ref Data Block (3 bits)
    01000      = GLONASS Freq Number (5 bits)
```

The Data Format Flag identifies that this batch of data is Reference (0) data.

The Ref Data Block ID is 0x000.

The GLONASS Frequency Number is 8 (adjusted to 1). When calculating the GLONASS Carrier frequency, this value (0 to 20) will be adjusted to its -7 to +13 value and then multiplied by that frequencies delta. Note that this field only appears in the Reference data and will not be found in the Differential data.



**Special Case:** When the Slot ID is between 43 and 63, the Slot ID of the GLONASS satellite is unknown. In order to keep track of which satellite it is for these calculations, the Frequency Number is used to assign this GLONASS Satellite a temporary Slot ID based on the GLONASS Frequency Numbers binary value of 0 to 20.

### A.1.10 Reference Primary Signal Measurement Block: GLONASS PRN 38 – L1CA

The next bytes collected will be for the GLONASS PRN 38 - L1CA signal data. This is the primary signal of the satellite since it is the first signal. As a result, its Measurement Block consists of 111 bits as listed in *Table 171: Primary Reference Signal Measurement Block* on page 827. Since 111 bits takes up a lot of space, these bits will be split into two groups from *Table 171: Primary Reference Signal Measurement Block* on page 827: the top 25 bits for signal info followed by the bottom 86 bits for signal data.

The signal info section (top 25 bits) is processed as follows:

- With 1 bit left unprocessed from the previous byte, we calculate  $25 - 1 = 24$  bits which equals 3 bytes. Therefore the previous last byte (0xa0) plus the next 3 bytes will be needed.
  - Use the last byte (0xa0) plus grab 3 bytes (x19f813) = 0xa019f813
  - Swap the bytes = 0x13f819a0
  - 0x13f819a0 in binary form = 00010011111110000001100110100000
  - The previous step used the 7 LSBs = 0001001111111000000110011XXXXXX

- Need 25 bits which is exactly what is left over:

```

0001001111111000000110011xxxxxx
          1           = Parity Flag
          1           = ½ Cycle Slip Flag
        10000001100   = C/No
          1111        = Lock Time
         0011         = Pseudorange Std Deviation
        0001         = Phaserange Std Deviation
    
```

- Parity flag is a 1 (Parity Known)
- ½ Cycle Slip flag is a 1 (Cycle Slip Present)
- C/No is:  
 $0x10000001100b = 1036 \times \text{Scaling factor of } 0.05$   
 $= 51.80 \text{ dBHz}$
- The Lock Time value is:  
 $0x1111b = 15$  which means that this signal has been locked for 262144 ms or more.
- The Pseudorange Std Deviation value is:  
 $0x0011b = 3$  which means:  $0.045 \text{ m} < \text{PSR Std Dev} \leq 0.066 \text{ m}$  using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
 $0x0001b = 1$  which means:  $0.0039 < \text{ADR Std Dev} \leq 0.0052 \text{ cycles}$  using *Table 177: ADR Std Dev* on page 833.

The signal data section (bottom 86 bits) is processed as follows:

- With no unprocessed bits from the previous byte, we need 86 bits which rounds up to 11 bytes.

- Grab 11 bytes =  $0x6a11273649b8fcefab9c43$
- Swap the bytes =  $0x439cabeffcb8493627116a$
- $0x439cabeffcb8493627116a$  in binary form =

```

0100 0011 1001 1100 1010 1011 1110 1111 1111 1100 1011 1000 0100 1001 0011 0110 0010 0111
0001 0001 0110 1010
    
```

- Only need 86 bits. Ignore first 2 MSBs =

```

xx00 0011 1001 1100 1010 1011 1110 1111 1111 1100 1011 1000 0100 1001 0011 0110 0010 0111 0001 0001 0110 1010
          0 1001 0011 0110 0010 0111 0001 0001 0110 1010 = 1st Pseudo
          1111 1111 1100 1011 1000 010          = 1st Phase - 1st Pseudo
00 0011 1001 1100 1010 1011 1110          = 1st Doppler
    
```

- Use *Table 171: Primary Reference Signal Measurement Block* on page 827 to identify if a 2's Complement Conversion is needed as well as what Scale Factor should be used before these binary numbers are used in the following calculations.
- The 1st (Primary) Pseudorange is processed by:  
 $1\text{st Pseudorange} = 0x0100100110110001001110001000101101010b \times \text{Scaling Factor}$   
 $1\text{st Pseudorange} = 39563235690 \times 0.0005$   
 $L1CA \text{ Pseudorange for PRN } 38 = 19781617.845 \text{ m}$
- The 1st (Primary) Phaserange is a 2's Complement number (as identified by the Range column in *Table 171: Primary Reference Signal Measurement Block* on page 827) so it is processed in the following

manner:

1st Phaserange – 1st Pseudorange = 2's Complement(0x1111111110010111000010b) \* Scaling Factor

1st Phaserange – 19781617.845 m = -6718 \* 0.0001

L1CA Phaserange = 19781617.1732 m

- Convert this to ADR to check against the original RANGE log:

ADR = 1st Phaserange \* (Carrier Frequency + Frequency Number \* 562500 Hz) \* (-1)/Speed Of Light

ADR = 19781617.1732 m \* (1602000000 Hz + 1 \* 562500 Hz) \* (-1)/299792458 m/s

ADR = 19781617.1732 m \* 1602562500 Hz \* (-1)/299792458 m/s

L1CA ADR for PRN 38 = -105744080.6970745 cycles



In the range logs, PSR and ADR have opposite signs.

- The 1st (Primary) Doppler is a 2's Complement number (as identified by the Range column in *Table 171: Primary Reference Signal Measurement Block* on page 827) so it is processed in the following manner:

1st Doppler(m/s) = 2's Complement(0x00001110011100101010111110b) x Scaling Factor

1st Doppler(m/s) = 3787454 m/s x 0.0001

L1CA Doppler(m/s) = 378.7454 m/s

Convert the Doppler to Hz:

1st Doppler(Hz) = 1st Doppler(m/s) x (Carrier Frequency + Frequency Number \* 562500 Hz) \* (-1)/Speed Of Light

1st Doppler(Hz) = 378.7454 m/s x (1602000000 Hz + 1 \* 562500 Hz) \* (-1)/299792458 m/s

1st Doppler(Hz) = 378.7454 m/s x 1602562500 Hz \* (-1)/299792458 m/s

L1CA Doppler(Hz) for PRN 38 = -2024.6112 Hz

The rest of the GLONASS Reference Signals are handled in a similar manner as described in the above GPS section.

## A.2 Differential Log Decoding

Logs not falling on a whole second are most likely Differential logs which are processed differently than the Reference logs. It is possible for a sub-second RANGEEMP4 log to be a Reference log if the data contained within it did not fit the tight Differential Compression requirements.

Differential logs use the reference data of the same signal unlike reference logs which uses the first signal to define the other signals.

The next RANGEEMP4 log is at time 507977.250:

```
#RANGEEMP4A, COM1, 0, 88.5, FINESTEERING, 1919, 507977.250, 02000020, fb0e, 32768;239,
030000421204000000009200dff688831f6102005500e70162dc977c004015c07988840f6101803
a805921cedf8b80002011207080e5f6351f003804081c2200be0808005c01620808725f93028057
801822dae0476000a00f207180fef6251700e803401c62f3bdc8060052013009986f5f220200540
04ca2053ec408005401ca870180410000000000980ff6306fec408004801de07c8692f51028051
80f721b2e04f600040152081804ef7102500600540202205fe040a0086013a0938780f610200618
04e224edbdb68002010c0498030f7411d0018047812a2d47d090a004c01a609c8544f6202805200
6a02*48E189A2
```

At the start of the RANGEEMP4 log is the identifier for how many bytes are in the log. In this case, there are 239 bytes (just under 20% less than a Reference Log). The rest of the message is compressed binary data and is transmitted as LSB first so the bytes must be swapped before processing.

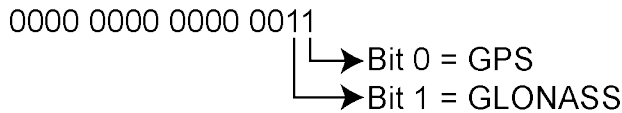
### A.2.1 Differential Header

The Differential Header is sent once per message (See *Table 168: Header* on page 825).

Decoding the bits starting with the first bytes:

*GNSS field* (16 bits)

- Grab the first 2 bytes (16 bits) = 0x0300
- Swap the bytes = 0x0003
- 0x0003 in binary form =



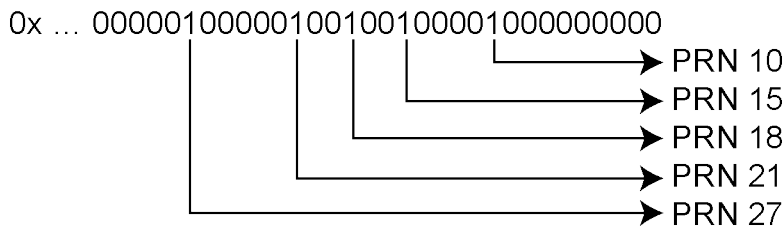
In this example the receiver was configured to track only GPS and GLONASS systems. If other systems had been in the configuration and tracked, they would have shown here.

### A.2.2 Differential Satellite and Signal Block

This block is sent once for each bit set to 1 in the GNSS field found in *Table 168: Header* on page 825. As identified by the above GNSS field, the first system (right to left) is the GPS System. Use *Table 169: Satellite and Signal Block* on page 825 to determine what satellites and signals data are contained in this GPS System:

*GPS Satellites field* (64 bits)

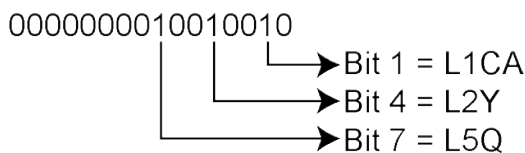
- Grab the next 8 bytes (64 bits) = 0x0042120400000000
- Swap the bytes = 0x...000000004124200
- 0x000000004124200 in binary form =



- The 1's above identify that there are 5 tracking GPS PRNs.

*GPS Signals field* (16 bits)

- Grab the next 2 bytes (16 bits) = 0x9200
- Swap the bytes = 0x0092
- 0x0092 in binary form =



- The 1's above identify that there are 3 tracking GPS signals: L1CA, L2Y, and L5Q.

*GPS Included Signals field* (5 PRNs x 3 Signals = 15 bits – therefore need 2 bytes)

Up to the point of processing the Included Signals field, the bytes are aligned such that the bits start and end within each batch of bytes. After processing this step, it is quite common for the Included Signals field (mxn matrix) to not be divisible by 8 so bytes not processed will need to be carried over to the next section depending on the size of the matrix.

- Grab the next 2 bytes (16 bits) = 0xdf6
- Swap the bytes = 0xf6df
- 0xf6df in binary form = 1111011011011111
- Only need 15 of the 16 bits = X111011011011111
- This bit string breaks down into 5 rows (PRNs) and 3 columns (signals) as specified by the mxn (PRN x signals) parameters. Take the bit string and break it up into sets of 3 starting at the MSB. This will result with the lowest PRN being at the bottom row of the stack and the first signal (L1CA) being the far right column.

```

111
011
011
011
111

```

- This stack can be further broken apart to identify the PRNs vs. their Signals:

PRN	L5Q	L2Y	L1CA
27	1	1	1
21	0	1	1
18	0	1	1
15	0	1	1
10	1	1	1

### A.2.3 Differential Measurement Block Header

This block is sent once for each bit set to 1 in the Satellites field found in *Table 169: Satellite and Signal Block* on page 825. Now that the PRN's signals have been determined, the next step is to determine the specifics of the first PRN (10) and its list of signals (L1CA, L2Y, L5Q). Working from bottom right to upper left of the PRN/Signal chart above, each 1 represents a signal for a PRN. Use *Table 170: Measurement Block Header* on page 826 to determine the contents of each field:

*GPS PRN 10* (first PRN found in the Satellites field)

We will grab enough bytes to process the whole Measurement Block Header. If this was a GLONASS system, a total of 9 bits would be required at this step (1 bit for the Data Format Flag, 3 bits for the Ref Data Block ID, plus 5 bits for the GLONASS Frequency Number). Since this is a GPS system, only 4 bits in total are required (1 bit for the Data Format Flag and 3 bits for the Ref Data Block ID).

There was 1 bit not processed in the last byte so that byte will be carried forward. Only 4 bits need to be looked at for this step so grab the next byte as well:

- Use the last byte (0xf6) plus the next byte (0x88)= 0xf688
- Swap the bytes = 0x88f6
- 0x88f6 in binary form = 1000 1000 1111 0110
- Ignore the processed bits from the last step = 1000 1000 1XXX XXXX



- Ignore the 5 MSB bits leaving 4 bits for processing =

```

xxxx x000 1xxx xxxx
          1          = Data Format Flag (1 bit)
          000        = Ref Data Block (3 bits)

```

The Data Format Flag identifies that this batch of data is Differential (1) data.

The Ref Data Block ID is 0x000. The Ref Data Block ID here identifies that this differential data will be calculated from the Reference data that had a Ref Data Block ID equaling 000 (which was determined in the RANGEEMP4 log at time 507977.00 seconds).



The 5 MSBs have not been processed so this byte will be carried forward.

Logs between seconds will be Differential logs but could be Reference logs depending on the compression calculations. If a discontinuity occurred that made it impossible for a Differential calculation to fit within the Differential Constraints, it will revert to a Reference log.

#### A.2.4 Differential Measurement Block

This block is sent once for each bit set to 1 in the Included Signals field found in *Table 169: Satellite and Signal Block* on page 825. Use *Table 173: Primary Differential Signal Measurement Block* on page 829 and *Table 174: Secondary Differential Signals Measurement Block* on page 831 to determine the contents of each field:

A Measurement Block for a single PRN will look like the following:

```

Primary Parity Flag
Primary ½ Cycle Slip Flag
Primary C/No
Primary Lock Time
Primary Pseudorange Std Deviation
Primary Phaserange Std Deviation
Primary Pseudorange
Primary Phaserange - Primary Pseudorange (determines the Phaserange for the 1st Signal)
Primary Doppler

2nd Parity Flag
2nd ½ Cycle Slip Flag
2nd C/No
2nd Lock Time
2nd Pseudorange Std Deviation
2nd Phaserange Std Deviation
2nd Pseudorange - Primary Pseudorange (determines the Pseudorange for the 2nd Signal)
2nd Phaserange – 2nd Pseudorange (determines the Phaserange for the 2nd Signal)
2nd Doppler – Primary Doppler (determines the Doppler for the 2nd Signal)

```

- 3<sup>rd</sup> Parity Flag
- 3<sup>rd</sup> ½ Cycle Slip Flag
- 3<sup>rd</sup> C/No
- 3<sup>rd</sup> Lock Time
- 3<sup>rd</sup> Pseudorange Std Deviation
- 3<sup>rd</sup> Phasorange Std Deviation
- 3<sup>rd</sup> Pseudorange - Primary Pseudorange (determines the Pseudorange for the 3<sup>rd</sup> Signal)
- 3<sup>rd</sup> Phasorange – 3<sup>rd</sup> Pseudorange (determines the Phasorange for the 3<sup>rd</sup> Signal)
- 3<sup>rd</sup> Doppler – Primary Doppler (determines the Doppler for the 3<sup>rd</sup> Signal)
- ...

### A.2.5 Differential Primary Signal Measurement Block GPS PRN 10 – L1CA

The next bytes collected will be for the GPS PRN 10 - L1CA signal data. Since this is the primary signal of the PRN, its Measurement Block consists of 78 bits as listed in *Table 173: Primary Differential Signal Measurement Block* on page 829.

The signal info section (top 25 bits) is processed as follows:

- With 5 bits left from the previous byte, we calculate  $25 - 5 = 20$  bits which rounds up to 3 bytes. Therefore the previous last byte (0x88) plus the next 3 bytes will be needed.

- Use the last byte (0x88) plus grab 3 bytes (x831f61) = 0x88831f61
- Swap the bytes = 0x611f8388
- 0x611f8388 in binary form  
= 0110 0001 0001 1111 1000 0011 1000 1000

- Only need 25 bits. The last byte uses the 5 MSBs and the first byte ignores the 4 MSBs

XXXX	0001	0001	1111	1000	0011	1000	1XXX	
							1	= Parity Flag
							0	= ½ Cycle Slip Flag
				1000	0011	100		= C/No
			1111					= Lock Time
		0001						= Pseudorange Std Deviation
	0001							= Phasorange Std Deviation

- Parity flag is a 1 (Parity Known)
- ½ Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
0x10000011100b = 1052 x Scaling factor of 0.05  
= 52.60 dBHz
- The Lock Time value is:  
0x1111b = 15 which means that this signal has been locked for 262144 ms or more.

- The Pseudorange Std Deviation value is:  
0x0001b = 1 which means: 0.020 m < PSR Std Dev <= 0.030 m using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
0x0001b = 1 which means: 0.0039 < ADR Std Dev <= 0.0052 cycles using *Table 177: ADR Std Dev* on page 833 Table 10.
- For the following calculations, the time difference between the Differential Log and the Reference log is 0.25 seconds as shown below:  
Time Difference = Current Log Time – Reference log Time  
= 507977.250 - 507977.000  
= 0.250 seconds

The signal data section (bottom 53 bits) is processed as follows:

- With 4 bits unprocessed from the previous byte, we calculate 53 – 4 = 49 bits = 7 bytes (7 bits will not be processed in the last byte).


- Use the last byte (0x61) plus grab 7 bytes (0x02005500e70162)  
= 0x6102005500e70162
- Swap the bytes = 0x6201e70055000261
- 0x6201e70055000261 in binary form =  
0110 0010 0000 0001 1110 0111 0000 0000 0101 0101 0000 0000 0000 0010 0110 0001
- Only need 53 bits. Ignore last 4 LSBs and first 7 MSBs =

```

xxxx xxx0 0000 0001 1110 0111 0000 0000 0101 0101 0000 0000 0000 0010 0110 xxxx
                                000 0000 0000 0010 0110      = 1st Pseudo - Predicted Pseudo
                                000 0000 0101 0101 0        = 1st Phase - Predicted Phase
0 0000 0001 1110 0111 0      = 1st Doppler - Ref Doppler
    
```

- Use *Table 173: Primary Differential Signal Measurement Block* on page 829 to identify if a 2's Complement Conversion is needed as well as what Scale Factor should be used before these binary numbers are used in the following calculations.
- The 1st (Primary) Differential Pseudorange is processed by:  
Predicted Pseudorange = Reference 1st Pseudorange + (1st Doppler x TimeDifference)  
= 21540181.930275 m  
= 21540290.811 m + ((-435.5229 m/s) x 0.250 s)  
1st DiffPseudorange – Predicted Pseudorange = 0x0000000000000100110b x Scaling Factor  
1st DiffPseudorange – 21540181.930275 m = 38 x 0.0005  
L1CA Pseudorange for PRN 10 = 21540181.949275 m
- The 1st (Primary) Differential Phaserange is a 2's Complement number (as identified by the Range column in *Table 173: Primary Differential Signal Measurement Block* on page 829) so it is processed in the following manner:  
Predicted Phaserange = Reference 1st DiffPhaserange + (1st Doppler x TimeDifference)  
= 21540291.5622 m + ((-435.5229 m/s) x 0.250 s)  
= 21540182.681475 m  
1st DiffPhaserange – Predicted Phaserange = 2's Complement(0x0000000010101010b) \* Scaling Factor  
1st DiffPhaserange – 21540182.681475 m = 170 \* 0.0001  
L1CA Phaserange = 21540182.698475 m
- Convert this to ADR to check against the original RANGE log:

ADR = 1st DifPhaserange \* Frequency \* (-1)/Speed Of Light  
 ADR = 21540182.698475 m \* 1575420000 Hz \* (-1)/299792458 m/s  
 L1CA ADR for PRN 10 = -113194424.0799796 cycles

 In the range logs, PSR and ADR have opposite signs.

- The 1st (Primary) Differential Doppler is a 2's Complement number (as identified by the Range column in *Table 173: Primary Differential Signal Measurement Block* on page 829) so it is processed in the following manner:

1st DiffDoppler(m/s)- Reference 1st Doppler = 2's Complement(0x00000001111001110b) x Scaling Factor

1st DiffDoppler(m/s) – (-435.5229 m/s) = 974 x 0.0001

L1CA Doppler(m/s) = -435.4255 m/s

Convert the Doppler to Hz:

1st DiffDoppler(Hz) = 1st DiffDoppler(m/s) x Frequency \* (-1)/Speed Of Light

1st DiffDoppler(Hz) = -435.4255 m/s x 1575420000 Hz \* (-1)/299792458 m/s

L1CA Doppler(Hz) for PRN 10 = 2288.1764464 Hz

### A.2.6 Differential Secondary Signals Measurement Block GPS PRN 10 – L2Y

Unlike Reference logs which always reflect back to the initial signal for their computations, Differential logs uses the last Reference log data of the same signal for its calculations.

- With 7 bits unprocessed from the previous byte, we will require  $74 - 7 = 67$  bits which rounds up to 9 bytes.
  - Use the last byte (0x62) plus grab the next 9 bytes (0xdc977c004015c07988) = 0x62dc977c004015c07988
  - Swap the bytes = 0x8879c01540007c97dc62
  - 0x8879c01540007c97dc62 in binary form =  
 1000 1000 0111 1001 1100 0000 0001 0101 0100 0000 0000 0000 0111 1100 1001 0111 1101 1100  
 0110 0010
  - Only need 74 bits. The 1 LSB is ignored as it was already processed above and the 5 MSBs are ignored so there is a total of 74 bits to process

```

xxxx x000 0111 1001 1100 0000 0001 0101 0100 0000 0000 0000 0111 1100 1001 0111 1101 1100 0110 001x
                                                                 1 = Parity Flag
                                                                 0 = ½ Cycle Slip Flag
                                                                 01 1100 0110 0 = C/No
                                                                 11 11 = Lock Time
                                                                 01 01 = Pseudorange Std Deviation
                                                                 00 10 = Phaserange Std Deviation
                                                                 0 0000 0000 0000 0111 11 = Pseudo - Predicted Pseudo
                                                                 0 0000 0001 0101 010 = Phase - Predicted Phase
000 0111 1001 110 = Doppler - Ref Doppler
    
```

- Parity flag is a 1 (Parity Known)
- ½ Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
 0x01110001100b = 908 x Scaling Factor of 0.05  
 = 45.4 dBHz
- The Lock Time value is:  
 0x1111b = 15 which means that this signal has been locked for 262144 ms or more.

- The Pseudorange Std Deviation value is:  
0x0101b = 5 which means: 0.099 m < PSR Std Dev <= 0.148 m using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
0x0010b = 2 which means: 0.0052 < ADR Std Dev <= 0.0070 cycles using *Table 177: ADR Std Dev* on page 833.
- The L2Y Pseudorange is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is processed in the following manner:  
  

$$\text{Predicted Pseudorange} = \text{Reference 2nd Pseudorange} + (\text{2nd Doppler} \times \text{TimeDifference})$$

$$= 21540293.6315 \text{ m} + ((-435.523 \text{ m/s}) \times 0.250 \text{ s})$$

$$= 21540184.75075 \text{ m}$$
  

$$\text{DiffPseudorange} - \text{Predicted Pseudorange} = \text{2's Complement}(0x0000000000000011111b) \times \text{Scaling Factor}$$

$$\text{DiffPseudorange} - 21540184.75075 \text{ m} = 31 \times 0.0005$$

$$\text{L2Y Pseudorange} = 21540184.76625 \text{ m}$$
- The L2Y Phaserange is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is calculated in the following manner:  
  

$$\text{Predicted Phaserange} = \text{Reference 2nd DiffPhaserange} + (\text{2nd Doppler} \times \text{TimeDifference})$$

$$= 21540294.399 \text{ m} + ((-435.523 \text{ m/s}) \times 0.250 \text{ s})$$

$$= 21540185.51825 \text{ m}$$
  

$$\text{DiffPhaserange} - \text{Predicted Phaserange} = \text{2's Complement}(0x0000000010101010b) \times \text{Scaling Factor}$$

$$\text{DiffPhaserange} - 21540185.51825 \text{ m} = 170 \times 0.0001$$

$$\text{L2Y Phaserange} = 21540185.53525 \text{ m}$$
- Convert this to ADR to check against the original RANGE log:  
  

$$\text{ADR} = \text{Phaserange} \times \text{Frequency} \times (-1)/\text{Speed Of Light}$$

$$\text{ADR} = 21540185.53525 \text{ m} \times 1227600000 \text{ Hz} \times (-1)/299792458 \text{ m/s}$$

$$\text{L2Y ADR for PRN 10} = -88203458.95116848 \text{ cycles}$$



In the range logs, PSR and ADR have opposite signs.

- The L2Y Doppler is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is calculated in the following manner:  
  

$$\text{DiffDoppler(m/s)} - \text{Ref 2nd Doppler(m/s)} = \text{2's Complement}(0x00001111001110b) \times \text{Scaling Factor}$$

$$\text{DiffDoppler(m/s)} - (-435.5229 \text{ m/s}) = (974) \times 0.0001$$

$$\text{L2Y Doppler(m/s)} = -435.4255 \text{ m/s}$$
  
 Convert the Doppler to Hz:  
  

$$\text{Doppler(Hz)} = \text{Doppler(m/s)} \times \text{Frequency} \times (-1)/\text{Speed Of Light}$$

$$\text{Doppler(Hz)} = -435.4255 \text{ m/s} \times 1227600000 \text{ Hz} \times (-1)/299792458 \text{ m/s}$$

$$\text{L2Y Doppler(Hz) for PRN 10} = 1782.994633 \text{ Hz}$$

### A.2.7 Differential Third Signals Measurement Block GPS PRN 10 – L5Q

Unlike Reference logs which always reflect back to the initial signal for their computations, Differential logs uses the last Reference log data of the same signal for its calculations.

- With 3 bits unprocessed from the previous byte, we will require  $74 - 3 = 71$  bits which rounds up to 9 bytes.
  - Use the last byte (0x88) plus grab the next 9 bytes (0x 840f6101803a805921) = 0x88840f6101803a805921
  - Swap the bytes = 0x2159803a8001610f8488
  - 0x2159803a8001610f8488 in binary form =  
0010 0001 0101 1001 1000 0000 0011 1010 1000 0000 0000 0001 0110 0001 0000 1111 1000 0100 1000 1000
  - Only need 74 bits. The 3 LSBs are ignored as they were already processed and the 3 MSBs are ignored so there is a total of 74 bits to process.

```

XXXX 0001 0101 1001 1000 0000 0011 1010 1000 0000 0000 0001 0110 0001 0000 1111 1000 0100 1000 1XXX
                                     1 = Parity Flag
                                     0 = 1/2 Cycle Slip Flag
                                     1000 0100 100 = C/No
                                     1111 = Lock Time
                                     0000 = Pseudorange Std Deviation
                                     0001 = Phasorange Std Deviation
                                     000 0000 0000 0001 0110 = Pseudo - Predicted Pseudo
                                     000 0000 0011 1010 1 = Phase - Predicted Phase
0 0001 0101 1001 1
    
```

- Parity flag is a 1 (Parity Known)
- 1/2 Cycle Slip flag is a 0 (Cycle Slip Not Present)
- C/No is:  
 $0x10000100100b = 1060 \times \text{Scaling factor of } 0.05$   
 $= 53.0 \text{ dBHz}$
- The Lock Time value is:  
 $0x1111b = 15$  which means that this signal has been locked for 262144 ms or more.
- The Pseudorange Std Deviation value is:  
 $0x0000b = 0$  which means: PSR Std Dev  $\leq 0.020$  m using *Table 178: Pseudorange Std Dev* on page 834.
- The ADR Std Deviation value is:  
 $0x0001b = 1$  which means:  $0.0039 < \text{ADR Std Dev} \leq 0.0052$  cycles using *Table 177: ADR Std Dev* on page 833.
- The L5Q Pseudorange is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is processed in the following manner:
 

Predicted Pseudorange = Reference 3rd Pseudorange + (3rd Doppler x TimeDifference)  
 $= 21540289.869 \text{ m} + ((-435.5149 \text{ m/s}) \times 0.250 \text{ s})$   
 $= 21540180.990275 \text{ m}$

DiffPseudorange – Predicted Pseudorange = 2's Complement(0x000 0000 0000 0001 0110b) x Scaling Factor  
 DiffPseudorange – 21540180.990275 m =  $22 \times 0.0005$   
 L5Q Pseudorange = 21540181.001275 m
- The L5Q Phasorange is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is calculated in the following manner:

$$\begin{aligned} \text{Predicted Phaserange} &= \text{Reference 3rd DiffPhaserange} + (\text{3rd Doppler} \times \text{TimeDifference}) \\ &= 21540290.8584 \text{ m} + ((-435.5149 \text{ m/s}) \times 0.250 \text{ s}) \\ &= 21540181.979675 \text{ m} \end{aligned}$$

$$\begin{aligned} \text{DiffPhaserange} - \text{Predicted Phaserange} &= 2\text{'s Complement}(0x000000001110101b) \times \text{Scaling Factor} \\ \text{DiffPhaserange} - 21540181.979675 \text{ m} &= 117 \times 0.0001 \\ \text{L5Q Phaserange} &= 21540181.991375 \text{ m} \end{aligned}$$

- Convert this to ADR to check against the original RANGE log:

$$\begin{aligned} \text{ADR} &= \text{Phaserange} \times \text{Frequency} \times (-1) / \text{Speed Of Light} \\ \text{ADR} &= 21540181.991375 \text{ m} \times 1176450000 \text{ Hz} \times (-1) / 299792458 \text{ m/s} \\ \text{L5Q ADR for PRN 10} &= -84528300.92127641 \text{ cycles} \end{aligned}$$



In the range logs, PSR and ADR have opposite signs.

- The L5Q Doppler is a 2's Complement number (as identified by the Range column in *Table 174: Secondary Differential Signals Measurement Block* on page 831) so it is calculated in the following manner:

$$\begin{aligned} \text{DiffDoppler(m/s)} - \text{Ref 3rd Doppler(m/s)} &= 2\text{'s Complement}(0x00001010110011b) \times \text{Scaling Factor} \\ \text{DiffDoppler(m/s)} - (-435.5149 \text{ m/s}) &= 691 \times 0.0001 \\ \text{L5Q Doppler(m/s)} &= -435.4458 \text{ m/s} \end{aligned}$$

Convert this to Hz:

$$\begin{aligned} \text{Doppler(Hz)} &= \text{Doppler(m/s)} \times \text{Frequency} \times (-1) / \text{Speed Of Light} \\ \text{Doppler(Hz)} &= -435.4458 \text{ m/s} \times 1176450000 \text{ Hz} \times (-1) / 299792458 \text{ m/s} \\ \text{L5Q Doppler(Hz) for PRN 10} &= 1708.78285 \text{ Hz} \end{aligned}$$

This concludes the decoding of the Differential Log for PRN 10 (signals L1CA, L2Y, and L5Q). The rest of the decoding for the other PRNs and systems are handled in the same manner.

