

1. Introduction	2
1.1 Overview	3
1.2 System components	4
2. System Requirements	5
2.1 Operating System Compatibility	6
3. Setup and Installation	7
3.1 Download, Licensing and Installation Process of the Standalone Version	8
4. MuSNAT: Receiver Background	9
4.1 Fundamentals of Receiver Processing and Architecture	10
4.2 MuSNAT Supported Signals	11
4.2.1 Civil Signals	12
4.2.2 Data Table	14
4.2.3 Restricted/Authorized/Military Signals	15
5. File Format Description	16
5.1 Input File Format Description	18
5.1.1 .musnat File	19
5.1.2 Random Code Family (.rcf) File	20
5.1.3 Sample Meta File (.smf)	21
5.2 Output File Format Description	23
5.2.1 Acquisition Log (.acqlog)	24
5.2.2 Inter-Frequency Bias Log (.ifblog)	25
5.2.3 Position Output Log (.poslog)	26
5.2.4 Navigation Data File (.YYn)	28
5.2.5 RINEX Observation Data File (.YYo)	32
5.2.6 Tracking Dump Log (.dumplog)	33
5.2.7 Multi-Correlator log (.mclog)	36
5.2.8 Receiver Status Output File (*.pvt)	38
5.3 Configuration Structure	39
6. Description of the User Interface	40
6.1 MuSNAT-Core with Graphical User Interface	41
6.1.1 Menu Bar	42
6.1.2 Settings and Configuration Window	43
6.1.3 Tool and Plot Window	44
6.2 MuSNAT-Analyzer	55
6.3 MuSNAT-ASCII	60
6.3.1 How to run/debug MuSNAT-ASCII on Microsoft Visual Studio VS-2019?	61
6.3.2 How to run MuSNAT-ASCII from command prompt?	63
7. Configuration Settings	68
7.1 General Information	69
7.2 Configuration Parameters	70
7.2.1 Section Acquisition	71
7.2.2 Section CentralizedPhaseTracking	72
7.2.3 Section FFTAcquisitionLevel*	73
7.2.4 Section Main	74
7.2.5 Section Navigation	76
7.2.6 Section Navigation Data	77
7.2.7 Section RINEX_Obs	78
7.2.8 Section Receiver*	81
7.2.9 Section Receiver Status	91
7.2.10 Section SPP*	92
7.2.11 Section SpectrumAnalyzer	94

Introduction

The Multi-Sensor Navigation Analysis Tool (MuSNAT) has undergone a transformative journey since its inception at the University of the Bundeswehr Munich (UniBwM) in 2002. Initially rooted in developing operational real-time GNSS receivers, MuSNAT has evolved into a sophisticated software suite designed to explore and implement cutting-edge signal processing and navigation algorithms. The present formulation of MuSNAT was first introduced by [Pany et al. \(2019\)](#), marking the transition from its earlier forms. This evolution has been driven by the aim to address critical challenges in navigation technology and provide researchers with a powerful tool for in-depth analysis.

MuSNAT finds its application across a wide range of domains:

- GNSS Signal and Channel Analysis
- Signal Quality Monitoring
- GNSS Receiver and Navigation Algorithm Development
- Integration with Navigation Technologies like INS and LiDAR
- RTK and PPP
- Galileo Signal Authentication
- Spoofing and Jamming Detection
- Antenna Array Processing
- Handling New GNSS Signals and Pseudolite Signals

This document aims to support present and future users in establishing MuSNAT fundamentals and present use cases for the applications listed above.

Overview

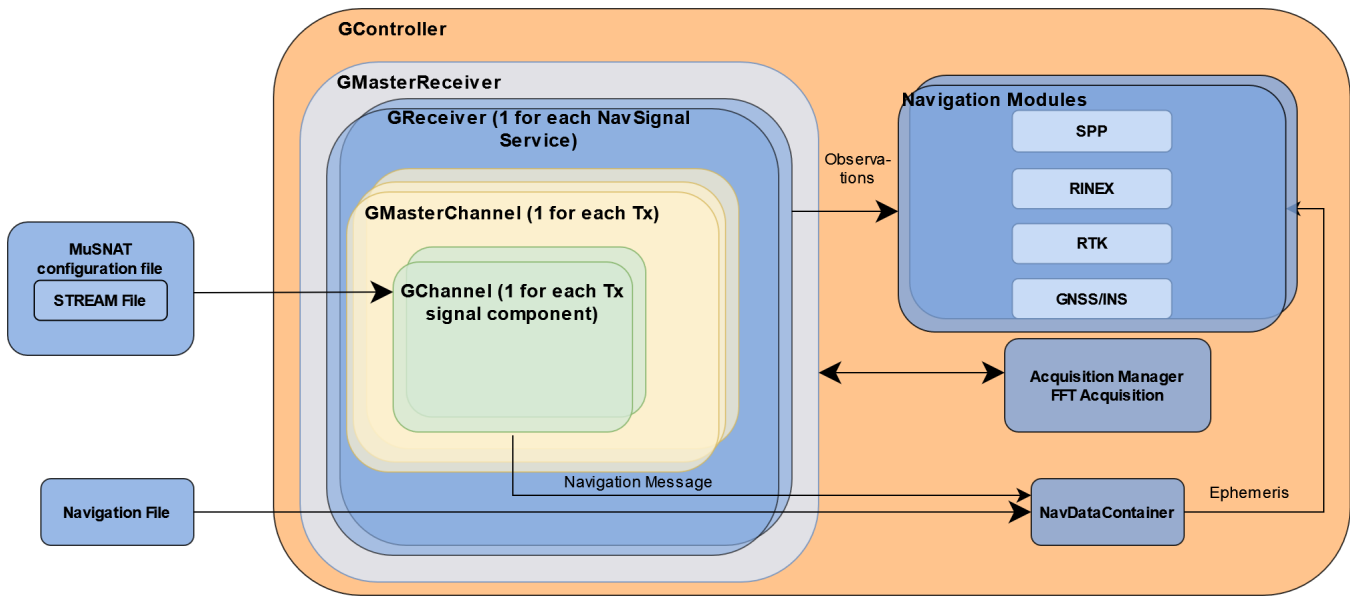
MuSNAT is developed as a Visual Studio 2017 C++/C# project, incorporating acceleration techniques like Intel IPP and CUDA. A console version, with a very limited user interface using ASCII characters, is maintained as a Visual Studio 2019/2022 project with the latest Intel IPP and CUDA support. Various other common C++ libraries are included using the vcpkg library manager and Visual Studio. The software is presently in the process of being adapted for use on Linux.

The core receiver, MuSNAT-Core, is configured using an XML file and initiated through a user-friendly GUI.

MuSNAT		
Feature	Solution	Remark
Operating System	Windows 10/11	Compiles as GUI or as command line version (port of command line version to Linux under preparation)
Programming environment	Microsoft Visual Studio 2019 C++	CUDA, Intel OneAPI, vcpkg and .net for GUI
IF sample file input source	ION SDR Standard and proprietary file readers	proprietary readers faster than ION SDR reader
Real-time sample input	yes, via TCP/IP	server available via LabView for selected NI USRPs
Additional sensors	LiDAR, IMU	LiDAR uses PCL format, IMU proprietary ASCII format, video formats supported but not yet used
Supported GNSS	GPS, Galileo, BeiDou, GLONASS, SBAS, OFDM (LTE, 5G, ...)	nearly all open spreading codes available and at least for each system one navigation message decoder
Acquisition	optimized fast Fourier transform (FFT) method	CPU and GPU supported
Tracking	dot-product from Intel Performance Primitives (CUDA version for massive multi-correlator applications)	computational performance mostly limited by memory bus width
Further features	multi-antenna, signal-generator, primary-secondary tracking, SQL database for logging, vector tracking, GNSS/INS integration, RTKLIB	via Matlab-interface support of Galileo OSNMA/HAS and synthetic aperture processing
Availability	Executable plus data visualizer downloadable via UniBwM (2023)	Source code available for research projects with UniBwM

System components

The figure below provides a high-level description of the MuSNAT software receiver architecture. There are 7 major units: the controller, the master receiver, the receiver, the master channel, the channel, the acquisition manager, and finally, the navigation processor unit.



The **controller unit (GController class)** is the top-level unit that controls the whole MuSNAT processing by reading the IF samples provided from one of the following sources mentioned below and is responsible for starting or stopping the receiver.

1. The standard file reader using the Standard Meta File (.smf) format
2. A file reader based on the ION GNSS SDR standard
3. in real-time via TCP/IP (currently only one stream is supported)

This module also conducts continuous error checks and stops the receiver completely when an error happens.

The **master receiver unit (GMasterReceiver class)** is the responsible of reading the IF samples and performing the IF sample synchronization. The samples are then attached to receiver units (GReceiver classes). The master receiver is also in charge of initiating the pseudorange measurements and passing the measurements to the navigation processor to compute the PVT solution.

The **receiver unit (GReceiver class)** represents a single-frequency receiver. The attached classes to this unit (**GMasterChannel and GChannel classes**) are the ones controlling the whole signal processing block. This includes the reference code (i.e. replica) generation and performing the correlation, the tracking loops, data decoding, and pseudorange measurements. In addition, it provides an interface to the acquisition unit. For each NavSignal service a single receiver unit is initialized. All receiver units run in parallel (multi-threading) and the synchronization between the different receiver units is performed by the master receiver. There exists also some specialized receiver modules attached to this unit such as the **Spectral Analyzer**.

Each receiver unit comprises several master channel units (**GMasterChannel class**), each one tracking one navigation signal. In addition, each master channel unit comprises at least **one channel unit (GChannel class)**, which processes a single signal component. For example, for the Galileo E1-B/C, each master channel initializes two channel units, one for the data component, E1-B, and a second one for the pilot component, E1-C.

The **navigation processor unit** is the last main unit of MuSNAT, which is in charge of retrieving the measurements from the master receiver units and sending them to the corresponding navigation modules. The main function of this unit is to work as interface between the signal processing block and the navigation modules. The navigation modules employed by MuSNAT can be SPP, RINEX, RTK, etc. and therefore, each navigation module performs the corresponding operations. It must be noted that MuSNAT is capable of employing only one or more navigation modules in a serial order as listed within the .musnat configuration file, and therefore, special care is taken when passing the retrieved measurements to the appropriate navigation module.

System Requirements

Before installing MuSNAT, ensure your system meets the following requirements:

- A 64-bit Windows 10 PC
- Adequate hard disk space (over 1 Gb)

Presently the software installation is supported on Windows 10/11.

Operating System Compatibility

Currently, MuSNAT is designated to work on the Windows Operating System. A suitable porting of MuSNAT for Linux and support for its use under virtual environments and docker platforms is under consideration.

Setup and Installation

MuSNATCore and MuSNAT Analyzer are publicly available in this [folder](#). Check further download instructions at the left panel.

Before installing MuSNAT, it is recommended to install Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017, and 2019, as described. If you experience incompatibility with libraries, be advised to download the DLL files in the download folder above.

Installing the Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017, and 2019 is recommended to facilitate a smooth installation process. The redistributable executable, VC_redist.x64.exe, is here: [Microsoft Visual C++ Redistributable](#). (See also [here](#).)

Download, Licensing and Installation Process of the Standalone Version

[The current Desktop Build Version is from 24 July 2023](#)

The MuSNAT software components, namely the MuSNAT CoreProcessGUI and MuSNATProcessAnalyzerGUI, are available as standard Windows Desktop applications.

The CoreProcessGUI installer encompasses essential reference code files and sample configuration files for some cases.

To get started with MuSNAT, you can download the software components from the following links:

- MuSNATCoreGUI: [MuSNAT Core](#)
- MuSNATAnalyzer for Data Visualization: [MuSNAT Analyzer](#)

Run both installer packages to install MuSNAT Core and MuSNAT Analyzer on your PC. As the installer have not been verified by Microsoft you need to accept security exceptions on your PC.

Please consider running MuSNAT with administrator rights, especially as log files are generated during processing. **Installing MuSNAT in a non-Windows-controlled folder, such as c:\MuSNATCore and c:\MuSNATAnalyzer, is also highly recommendable.**

MuSNAT supports using CUDA to access your NVidia Graphics card to accelerate the acquisition process. In this case, please install the executable with CUDA on its name. The version without CUDA support is also provided.

It's worth noting that MuSNAT-Core will necessitate a License Activation key associated with the workstation. When launching MuSNAT for the first time, you will be prompted for license activation, including a unique client key specific to your workstation. For inquiries about licensing, please contact Georg Betz at [ISTA/LRT 9.2 \(Georg.Betz@unibw.de\)](#).

MuSNAT: Receiver Background

This chapter aims to offer foundational information for using the MuSNAT while also offering a deeper understanding of the operational principles underlying the MuSNAT software receiver.

Fundamentals of Receiver Processing and Architecture

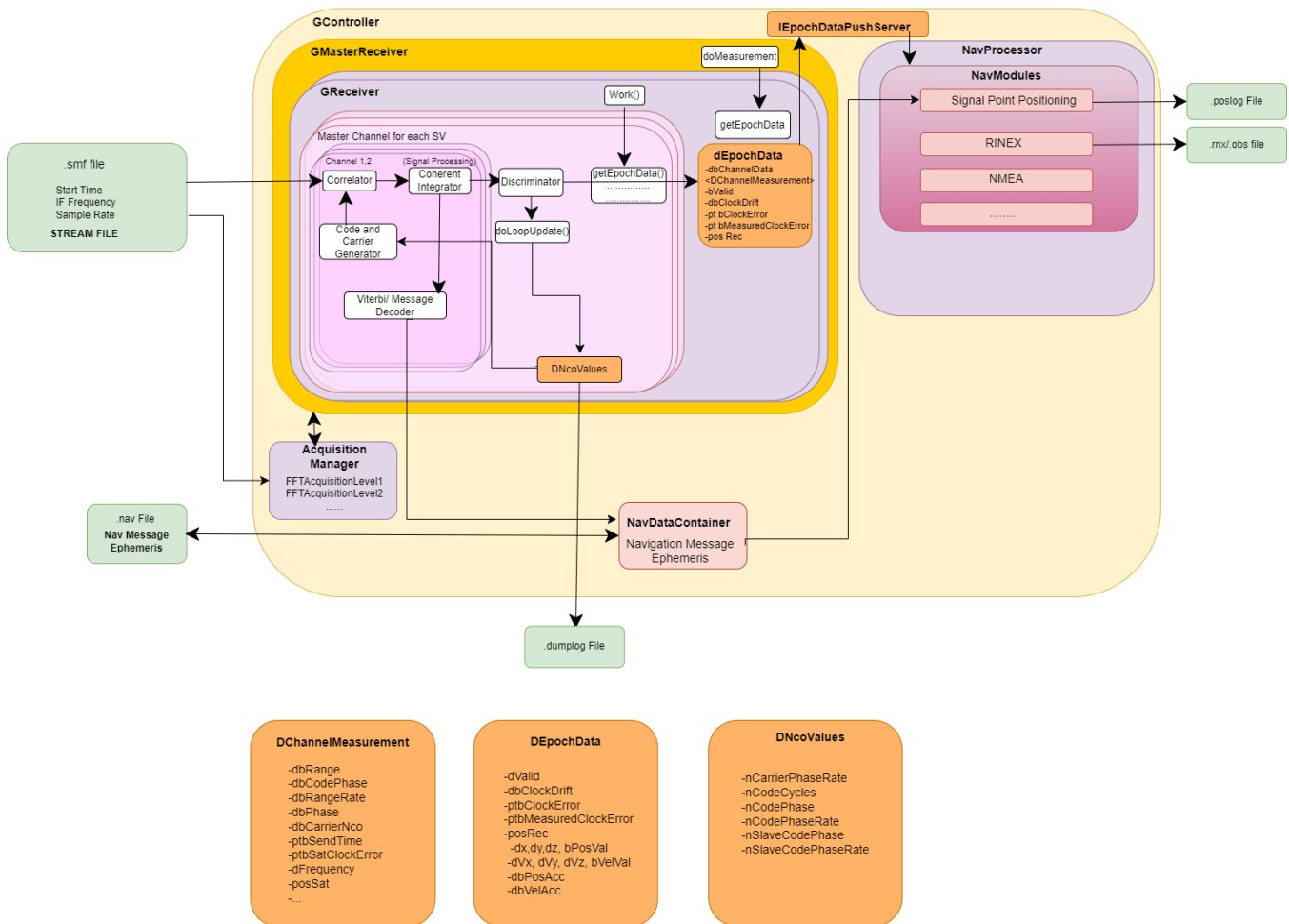
Like typical GNSS receivers, the MuSNAT software receiver undergoes signal processing steps such as acquisition, tracking, navigation message decoding, and Position, Velocity, and Time (PVT) computation. It carries out signal processing exclusively through software on a conventional CPU. The receiver can handle IF samples from prerecorded data streams retrieved from a Front-End and real-time sample input for subsequent post-processing.

The software architecture and data flow are depicted in Figure below. As previously mentioned, the software receiver receives digital IF sample data in real-time or retrieved from preexisting data files. Upon receiving IF sample data from an external source, the GNSS signal processing is internally managed by a central entity known as the "Master Receiver." This "Master Receiver" contains multiple units (receivers), each tasked with tracking signals associated with specific GNSS services, for example, GPS L1 C/A.

Each individual "Receiver" unit consists of a "Master Channel" for each SV and various "Channels," each dedicated to tracking a particular GNSS satellite signal. These "Receivers" employ a versatile tracking scheme that can be customized for each unit. Among the "Receiver" units, one notable distinction is their navigation message decoders.

To enhance signal acquisition control, dedicated units have been established. These units are structured as follows:

- An overarching entity referred to as the "Acquisition". Before performing an FFT acquisition, the receiver tries to acquire the signals by computing the signal's code phase and Doppler from the last position fix (so-called vector acquisition). The estimates are treated as input to the tracking loops, which are in charge of improving the estimates to reduce measurement errors. If the accuracy of those measurements is precise enough, the values are used to start tracking.
- Should acquisition remain unattainable through this methodology, the receiver proceeds with the optimized FFT acquisition technique referring to the "FFTAcquisitionLevel*", where * represents the acquisition unit described in [Section FFTAcquisitionLevel*](#).
- Continuing with the workflow, the "Navigation Processor" retrieves measurements from the "Master Receiver" and then forwards them to one or more designated "Navigation Modules." These modules execute a spectrum of tasks, including single-point positioning, generating RINEX output, and monitoring signal quality. The user defines the activation or deactivation of particular navigation modules through a configuration (.musnat) file. A more in-depth understanding of the functions and features of this module shall be given in the dedicated section.



MuSNAT Supported Signals

The table provides an overview of the current state of GNSS signal implementation in the MuSNAT software receiver, distinguishing between different levels of implementation completeness.

Civil Signals

System	Band	Frequency [MHz]	Signal	Bw [MHz]	Modulation	Min. Rcv. [dBW]	MuSNAT	Comment	Development Status	In ALLGNSS-Configurator	Testdata id	
GPS	L1	1575.42	C/A	2	BPSK(1)	-158.5	ACQ + TRK+ NAV		ready			
			L1C-D	4	TMBOC(6,1,4 /33)	-163.0	TRK	L1C-decoder not ready (will be implemented)	ready			
			L1C-P	4	TMBOC(6,1,4 /33)	-158.5	TRK		ready			
	L2	1227.60	L2C-M	2	BPSK(1)mux	-163.0	ACQ + TRK + NAV	CNAV decoder needs more testing	ready			
			L2C-L	2	BPSK(1)mux	-163.0	TRK	handover, no acquisition possible, L2C-L config needs more testing	ready			
	L5	1176.45	L5I-D	20	BPSK(10)	-157.9	ACQ+TRK	CNAV decoder needs more testing	ready			
L5Q-P			20	BPSK(10)	-157.9	ACQ+TRK		ready				
GLONASS [GLO]	L1	1602.0+k-0.5625	L1OF(C /A)	1	BPSK(0.5)	-161.0	ACQ+TRK+ NAV		ready			
			1600.955	L1OC-D	2	BPSK(1) mux	?	-	no code, no dec.	need adaptation		
				L1OC-P	4	BOC(1,1) mux	?	-	no code	need adaptation		
	1575.42	L1OCM	?	BOC(1,1)	?	-	no code, no dec.	need adaptation				
	L2	1246.0+k-0.4375	L2OF(C /A)	1	BPSK(0.5)	-161.0	ACQ+TRK+ NAV		ready			
			1248.06	L2OC-D	2	BPSK(1) mux	?	-		need adaptation		
				L2OC-P	4	BOC(1,1) mux	?	-		need adaptation		
	L3	1202.025	L3OC-D	20	BPSK(10)	?	-	no code, no dec.	need adaptation			
			L3OC-P	20	BPSK(10)	?	-		need adaptation			
			L3OCM	20	BPSK(10)	?	-	no code, no dec.	need adaptation			
	L5	1176.45	L5OCM	20	BPSK(10)	?	-	no code, no dec.	need adaptation			
	Galileo [GAL]	E1	1575.42	E1-OS-D	4	CBOC(6.1.1 /11)	-160.0	ACQ+TRK+ NAV		ready		
E1-OS-P				4	CBOC(6.1.1 /11)	-160.0	ACQ+TRK		ready			
E5ab		1191.795	E5ab	50	AltBOC(15,10)	-155.0	TRK	handover	ready			
E5a		1176.45	E5a-D	20	BPSK(10)	-158.0	ACQ+TRK	no F/NAV dec.	ready			
E5b		1207.14	E5b-D	20	BPSK(10)	-158.0	ACQ+TRK	no DF I/NAV dec.	ready			
			E5b-P	20	BPSK(10)	-158.0	ACQ+TRK		ready			
E6		1278.75	E6-CS-D	10	BPSK(5)	-158.0	ACQ+TRK		ready			
	E6-CS-P		10	BPSK(5)	-158.0	ACQ+TRK	only if not encrypted, encrypted in future (CAS)	ready				
Beidou [BDS]	B1	1561.098	B1I	4	BPSK(2)	-163.0	TRK	no decoder	ready			
			1575.42	B1C-D	32	BOC(1,1)	-159.0 (MEO) & -161.0 (IGSO)	ACQ+TRK+ NAV		ready		
				B1C-P	32	QMBOC(6,1,4 /33)		ACQ+TRK	BOC(1,1) or BOC(6,1) tracked combined with B1C-D	ready		
	B2	1176.45	B2a-D	20	BPSK(10)	-156.0 (MEO) & -158.0	-	ICD available	need adaptation			

			B2a-P	20	BPSK(10)	(IGSO)	-	ICD available	need adaptation		
		1207.14	PPP-B2b-I	20	BPSK(10)	-160.0	-	ICD available	need adaptation		
	B3	1268.52	B3I	20	BPSK(10)	-163.0	TRK	no decoder	ready		
QZS	L1	1575.42	C/A	2	BPSK(1)-I	-158.5	ACQ+TRK	Block I, may adapt GPS dec.	need adaptation		
					BPSK(1)-Q	-158.5	ACQ+TRK	Block II	need config		
			L1C-D	4	BOC(1,1)-I	-163.0	ACQ+TRK	Block I, may adapt GPS dec.	need adaptation		
					TMBOC(6,1,4/33)-I	-163.0	-	Block II	need adaptation		
			L1C-P	4	BOC(1,1)-Q	-158.2	ACQ+TRK	Block I	need config		
					TMBOC(6,1,4/33)-I	-158.2	-	Block II	need adaptation		
	SAIF	2	BPSK(1)	-161.0	-	SBAS, may use GPS SBAS dec.	need config				
	L2	1227.60	L2 CM	2	BPSK(1) mux	-163.0	ACQ+TRK+NAV	uses GPS L2C dec.	need config		
			L2 CL	2	BPSK(1) mux	-163.0	ACQ+TRK		need config		
	L6	1278.75	L61 (LEX)	10	BPSK(5) mux	-158.7	-	codes avail, no CSK mod	need adaptation		
				10	BPSK(5) mux	-158.7	-	codes avail, no CSK mod	need adaptation		
			L62	10	BPSK(5) mux	-159.8	-	codes avail, no CSK mod	need adaptation		
10				BPSK(5) mux	-159.8	-	codes avail, no CSK mod	need adaptation			
L5	1176.45	L5I	20	BPSK(10)	-157.0	ACQ+TRK	no dec.	need adaptation			
		L5Q	20	BPSK(10)	-157.0	ACQ+TRK		need config			
IRNSS	L5	1176.45	SPS	48	BPSK(1)	-159.0	ACQ+TRK	no dec.	need adaptation		
	S	2492.028	SPS	32	BPSK(1)	-162.3	ACQ+TRK	no dec.	need adaptation		
	L1C	-	-	-	-	-	ACQ+TRK	no dec.	ready		
SBAS	L1	1575.42	C/A-I	2	BPSK(1)	-161.0	ACQ+TRK+NAV	WAAS, EGNOS codes and Basic SBAS decoder available	need config		
			C/A-Q	2	BPSK(1)	-161.0	ACQ+TRK+NAV		need config		
	L5	1176.45	L5I	20	BPSK(10)	-158.5	-	no codes, no dec.	need adaptation		
			L5Q	20	BPSK(10)	-161.0	-	optional signal component	need adaptation		

Legend Development Status	
ready	=ready to use
need config	=need just adaption of the configuration file, create ranging code
need adaption	=need adaption of existing elements to e.g implement decoders
large modification	=need large modification of the source code e.g. to support new signal types

Data Table	
Location of Testsignals for MuSNAT Receivers	
ID	Path

Data Table

Location of Testsignals for MuSNAT Receivers

ID	Path

Restricted/Authorized/Military Signals

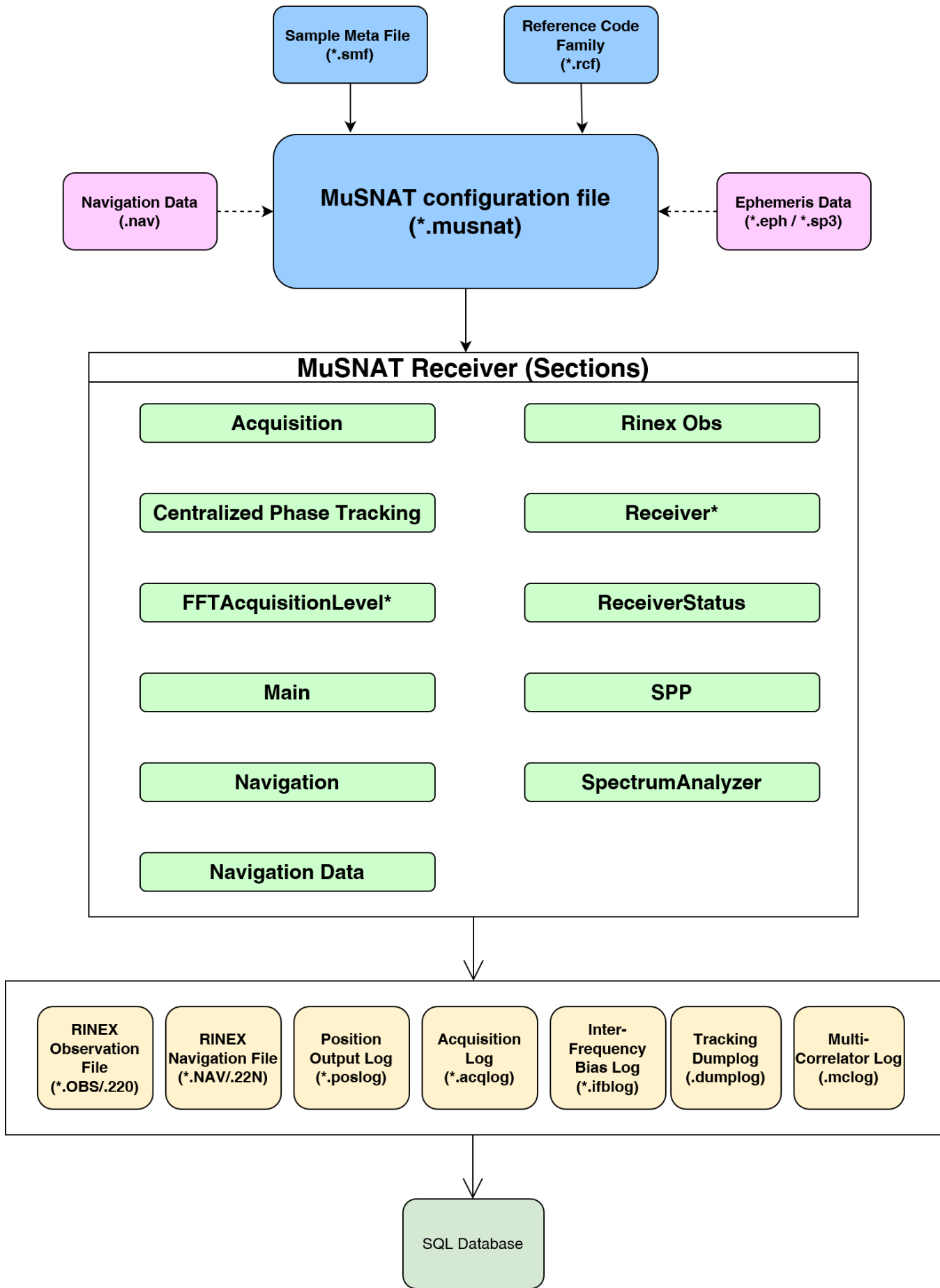
System	Band	Frequency (MHz)	Signal	BW (MHz)	Modulation	Min. Rcv. (dBW)	MUSNAT	Comment	Development Status
GPS	L1	1575.42	P(Y)	20	BPSK(10)	-161,5			-
			M-D	30	BOC(10,5) mux	-158.0			-
			M-P	30	BOC(10,5) mux	-158.0			-
	L2	1227.60	P(Y)	20	BPSK(10)	-164.5	TRK	code-less with L1	ready
			M-D	30	BOC(10,5) mux	-164.0			-
			M-P	30	BOC(10,5) mux	-164.0			-
GLONASS	L1	1602.0+k-0.5625	L1SF(P)	10	BPSK(5)	n/a			-
		1600,955	L1SC	10	BOC(5,2,5)	?			-
	L2	1246.0+k-0.4375	L2SF(P)	10	BPSK(5)	n/a			-
		1248.06	L2SC	14	BOC(5,2,5)	?			-
GALILEO	E1	1575.42	E1-PRS	34	BOC(15,2,5)	n/a			-
	E6	1278.75	E6-PRS	30	BOC(10,5)	n/a			-
BEIDOU BDS-2	B1-2	1561.098	B1-2-RS-Q	4	BPSK(2)	n/a			-
	B3	1268.52	B3-RS-I	20	BPSK(10)	n/a			-
			B3-RS-Q	20	BPSK(10)	n/a			-
	B2b	1207.14	B2b-RS-Q	20	BPSK(10)	n/a			-
IRNSS	L5	1176.45	RS-D-Q	32	BOC(5,2)	-156.0			-
			RS-P-I	32	BOC(5,2)	-159.0			-
	S	2492.028	RS-D	32	BOC(5,2)	-159.3			-
			RS-P	32	BOC(5,2)	-162.3			-
BDS	B1	1575.420	B1A	?	BOC(14,2)	?			-
		1561.098	B1Q	4	BPSK(2)	?			-
	B2	1207.14	PPP-B2b Q	20	?	?			-
	B3	1268.52	B3A	?	BOC(15,2,5)	?			-
			B3Q	20	BPSK(10)	?			-

File Format Description

This diagram provides insight into how input and output files engage with the MuSNAT receiver, elucidating the key components or modules that form the core of the receiver's functionality.

The core file is the *.**musnat** configuration file, where users input necessary files (illustrated in blue blocks) like Sample Meta File (*.**smf**) and the Reference Code Family (*.**rcf**). Additionally, users can include optional data (illustrated in pink blocks), such as navigation and ephemeris information.

Following a series of operations (depicted in green blocks), the receiver generates several output files (outlined in yellow blocks) that provide information about signal processing and position computation.



Input File Format Description

There are three main input files required for standard GNSS processing in MuSNAT. The format of these files shall be explained in the subsections below.

.musnat File

The configuration settings for the MuSNAT software receiver are outlined in files with the extension `*.musnat`, which utilizes the eXtensible Markup Language (XML) format to represent the configuration and can be directly modified. In addition to this, changes to the configuration parameters could also be done through [the GUI](#). The MuSNAT GUI briefly describes each parameter, reducing the risk of selecting inconsistent values and ensuring proper file structure and logic adherence. While direct file modification is possible, maintaining the compatible XML format and avoiding the use of incorrect parameters is crucial in such cases.

The organization or structure of various sections within the MuSNAT receiver configuration file is detailed in the [upcoming section](#). An example of the configuration file is provided ([SimpleGpsL1.musnat](#)), through which it is possible to replay the recorded signals. The `MuSNAT.defs.xml` file contains all the parameters utilized in the configuration file, and a thorough explanation of these settings is given in the [configuration parameters](#) section.

[Example *.musnat-file \(SimpleGPSL1\)](#)

Random Code Family (.rcf) File

The Random Code Family file stores vital parameters related to the replica signals used for various functions, such as acquisition, P-correlator, D-correlator, and W-correlator. These parameters are structured in the form of an XML tree.

It is important to note that the "Random Code Family" file itself does not contain the actual GNSS signal replicas; instead, it refers to secondary files with a .ref extension, which house the replica signals in ASCII format. The "Random Code File" also provides information about the characteristics and properties of the replica sequences found in these secondary files. At receiver startup the .ref files are converted into a binary format (.bin files) to ensure faster reading of files. It is important to delete the .bin files after a change in the .ref files to ensure consistency between binary and ASCII replica files.

The signals outlined and referenced in the "Random Code Family" are crucial in signal tracking, acquisition, and multi-correlator calculations. Their usage varies depending on the specific configuration of the MuSNAT receiver unit. To specify which reference signal to use, you must set the index number defined in the "Random Code File" within the corresponding MuSNAT receiver unit's settings using "Receiver*.CodeFamilyIndex." Typically, this index number differentiates between different PRN code families (e.g., GPS C/A versus L2CM) and various tracking schemes (e.g., differing correlator spacing).

Except for "nNumberOfFamilies", all "Random Code File" parameters given in the following Table are so-called "option arrays" and contain one entry per code family.

Parameter	Description	Unit
nNumberOfFamilies	Number of code families contained within the file. The following entries are lists of the length "nNumberOfFamilies".	-
vnCodeLength	Number of subchips for one replica signal including oversampling	subchip
vnNoOfCodes	Number of replica signals in this family (usually equal to the number of different PRN codes)	-
vnFirstPRN	First PRN number (usually 1, but an offset might be applied for a system whose first PRN number > 1, e.g. SBAS).	-
vnOversampling	Subchips per chip (integer value; in this context one sample is called subchip)	subchip
vnCodePeriod_ms	duration of reference signal in milliseconds (integer)	milli-second
vdDlISlope	Factor used to achieve a unit slope for the code discriminator (applied when waveform-based tracking is used).	subchip
vdClip	Threshold; If the code discriminator exceeds this threshold then it will be clipped	
vdPullIn	Pull-in region of the code discriminator. If 3 times the standard deviation of the predicted sent time is smaller than the pull-in region, then the signal can be assumed to be acquired. Used when the receiver predicts the code phase of a signal in acquisition mode from its internal PVT.	subchip
vdNominalCodeAccuracy	Parameter of a simple model to compute the code pseudorange accuracy. Model: accuracy = $vdNominalCodeAccuracy * \exp(-(C/N_0 - 45)/20)$;	m
vsInfiniteFileNames	Name of file containing *.ref replica signals used for acquisition for all PRNs of the code family	-
vsCarrierFileNames	Name of file containing *.ref replica used to compute the Pcorrelator for all PRNs of the code family	-
vsCodeFileNames	Name of file containing *.ref replica used to compute the Dcorrelator (e.g. early-late, double-delta or similar replica) for all PRNs of the code family	-
vsMonitorFileNames	Name of file containing signal type used to compute the W-multicorrelator for all PRNs of the code family	-
vsFrontendModels	A *.rcf-file containing a model for the signal correlation function and its derivatives. It is only used for the multipath estimating discriminator of MuSNAT. See section III.3 of https://athene-forschung.unibw.de/doc/139125/139125.pdf Current implementation only for GPS CA. It is generated and validated using synthetic signals (YASST)	-

Example: [Code Family \(.rcf\) File](#).

Sample Meta File (.smf)

The following is a snippet of a sample .smf file containing the primary information about the sample stream recorded by the front end.

```

SwRxData.smf

#SignalParameters
nStreams,i,1
dSampleTimeOffset,d,0.0000000000000000
dSensorTimeOffset,d,0.0000000000000000
vnAntennaId,vi,1,0
vnBandNo,vi,1,0
vnFileFormat,vi,1,7
vnFrontendNo,vi,1,0
vnPacketSize,vi,1,679936
vnSampleRateN,vi,1,1
vnSampleRateZ,vi,1,20000000
vdCarrierFrequency,vd,1,1575420000.00000
vdIntermediateFrequency,vd,1,5000445.88565834
vsFileName,vs,1,SwRxData_Band0_FE0_ANT0_f1575420000.stream
vsSensorFile,vs,1,SwRxData.laf
ptCoarseStartTime,pt,2,1353167222,0.997633601478824
    
```

The sample meta-file employs a general line format as follows:

- For a single value: [Parameter name],[Parameter type],[Value]
- For a vector of n values: [Parameter name],[Parameter type],[Number of values “n”],[Value 1], [Value 2], ..., [Value n]
- A vector of values is identified by a “v” at the start of the parameter type or if the parameter type is “pt”. The following parameter types are used:
 - “i” – Integer number
 - “d” – Double number
 - “s” – String
 - “v[k]” – Vector of type [k], for instance “vi” corresponds to the vector of integers
 - “pt” – Time represented by two numbers: integer number of whole seconds and double number for fractional seconds

The description of the parameters used within the .smf file is as follows:

Parameter	Description	Unit	Notes
nStreams	Number of IF signal streams, typically equal to the number of frequency bands of the front	-	
dSampleTimeOffset	Time offset added to IF sample time tags, used to change the time of the first sample	s	
dSensorTimeOffset	Time offset added to sensor data time tags, used to change the time of the first sample	s	
vnAntennaId	Identifier for the Antenna (only used for data labeling)	-	
vnBandNo	Identifier for the frontend internal RF chain (only used for data labeling)	-	
vnFileFormat	IF sample format, i.e., the number of bits used to represent one sample: - “0” ... 8-bit real-valued IF signal - “1” ... 1.5-bit real-valued IF signal -> -1, 0, 1 - “7” ... 2 bit-real-valued IF signal -> -3, -1, 1, 3 - “100” ... 2-bit two-complement I/Q baseband - “101” ... 4-bit two-complement I/Q baseband - “102” ... 8-bit two-complement I/Q baseband - “103” ... 12-bit two-complement I/Q baseband - “104” ... 16-bit two-complement I/Q baseband - “110” ... 2-bit sign/mag. I/Q baseband - “111” ... 4-bit sign/mag. I/Q baseband - “112” ... 8-bit sign/mag. I/Q baseband - “113” ... 12-bit sign/mag. I/Q baseband - “125” ... 32-bit float I/Q baseband	-	
vnFrontEndNo	Identifier for the frontend recording the samples (only used for data labeling)	-	
vnPacketSize	Number of samples per packet (note: packet size must result in equal time per packet for all streams)	-	
vnSampleRateN	Sample rate represented as rational number $fs=Z/N$	-	
vnSampleRateZ	Sample rate represented as rational number $fs=Z/N$	Hz	
vdCarrierFrequency	Nominal carrier frequency (e.g. L1, L2 or L5 frequency, a possible frequenc offset can be specified in Receiver”.FrequencyOffset)	Hz	

vdIntermediateFrequency	Nominal intermediate frequency	Hz	
vsFileName	Name of the binary sample files	-	
vsSensorFile	Name of the binary sensor data file	-	optional
ptCoarseStartTime	Estimated time of the first IF sample*	s	Always represented as xxxxxx, yyyyyy where, xxxxxx is the whole number yyyyyy is the floating point / the fractional part

* - Number of seconds from 6 January 1980 to the IF sample recording (or the simulated date of IF sample recording)

The duration calculation can be performed through the following link: [Time Duration Calculator: Time between two dates/times \(timeanddate.com\)](http://timeanddate.com)

Output File Format Description

The table below outlines the essential explanations for the main file attributes used in the output log descriptions.

Attribute	Description
File Extension	The default file extension of the output file used within the delivered MuSNAT configuration scenarios
File Sort	Use "MuSNAT specific" for files designed for the MuSNAT "Standard" for common standard files
File Type	Use "Output" for output only file
File Version	Identifier for the file version number
Output activation setting	MuSNAT configuration setting to active or deactivate the file output
Output path setting	MuSNAT configuration setting to define the location and name of the output file
Important configuration settings	Important settings within the MuSNAT configuration regarding the log file
Rate of file generation	Rate of generation of the log file, for example one file or daily files
Number of files	Number of files which are generated when the output is activated, for example one file, one file per MuSNAT run or one file per satellite
Data output rate into file	Rate at which data is stored into the file, for example once at MuSNAT start-up or every second
Applicable reference documents	Reference documents for log files, especially for files of file sort "Standard"
External Tools	Additional Tools for further processing / plotting the output files

Acquisition Log (.acqlog)

The described binary file format is created for acquisition type 4, which can be found in the configuration parameter "FFTAcquisitionLevel". AcquisitionType." This binary file holds a two-dimensional array of correlation function values recorded during signal acquisition experiments and saved in binary format. The grid is defined by the time delays of PRN codes and the frequency shifts (Doppler frequency offsets) between the incoming signal and the generated reference signal.

File Extension	*.acqlog
File Sort	MuSNAT specific
File Type	Output
File Version	1.0
Output activation setting:	FFTAcquisitionLevel*.GenerateReports FFTAcquisitionLevel*.LogFileAcqMode
Output path setting	FFTAcquisitionLevel*.SatSignalStatistics
Rate of file generation	Separate files created for each acquisition trial, file generation stops for every satellite signal if it was successfully acquired and handed over to the tracking routine
Number of files	One file per acquisition trial per satellite
Data output rate into the file	Once immediately after file creation
External Tool:	\\WorkingData\MatlabTools\+MuSNAT\plotAcqLog.m (Matlab plotting tool)

-
- 2-dimensional grid of correlation function values
 - Doppler frequencies and PRN code phases for the grid points
 - Satellite azimuth and elevation, if they could be estimated before acquisition from almanac or assistance data
 - Doppler value, PRN code phase, and power estimates of the acquired signal
 - Detection threshold
 - MuSNAT internal data: acquisition level (1 or 2), receiver ID
 - Satellite system (for example, GPS), satellite service (for example, L1 C/A) and satellite PRN number
 - Receiver time in GPS week and GPS second
 - Most recent estimate of the receiver clock error

Inter-Frequency Bias Log (.ifblog)

Within this file, a continuous list of Inter-Frequency Biases (IFBs) is contained, calculated by the "Single Point Positioning" (SPP) navigation modules. These biases are computed between different satellite services of GNSS and are primarily comprised of two components:

1. Signal delay within the receiver hardware, dependent on the signal's path.
2. Discrepancies in system time offset between different GNSS systems, either disregarded or unknown. IFBs are calculated concerning the GNSS service tracked in the initial receiver unit, as specified by the configuration parameter

Main.ReceiverSectionNames, which typically corresponds to GPS L1 C/A with valid observations. The calculated IFB is also influenced by signal processing factors, such as discriminator properties (e.g., narrow or wide correlator spacing). These IFBs are determined at the primary measurement rate and are stored within the log file for reference.

File Extension	*.ifblog
File Sort	MuSNAT / SX-3 specific
File Type	Output
File Version	1.0
Output activation setting:	SPP*.CreatelfbLogFile
Output path setting	SPP*.SppLogFile (equal to the position output but using the extension .ifblog)
Rate of file generation	One file, created at receiver start (time tagging possible)
Number of files	One file per navigation module with activated output
Data output rate into the file	Continuously, defined by overall measurement rate

Field Description:

Field	Field Name	Description
1	GPS Week	GPS time week number
2	GPS second	GPS time second [s]
	
2n+1	IdentifierGNSS service, e.g., L1CA	Estimated inter-frequency bias of the nth considered GNSS service [m]
2n+2	Std. Dev	Estimated standard deviation of the inter-frequency bias of the nth considered GNSS service [m] A standard deviation value set to -1 indicates two special cases: - For the first logged service, typically GPS L1 C/A, the IFB can not be calculated as the first service defines the IFB reference, all calculated IFBs relate to this reference service - For all other services "-1" indicates the IFB could not be calculated due to lack of observed satellites of this service

Position Output Log (.poslog)

In this file, a continuous list of Position, Time, and Velocity (PVT) values is encompassed and calculated by the receiver navigation modules. The PVT calculation is achievable independently within the following types of navigation modules:

1. Single Point Positioning (SPP): Multiple independent SPP navigation modules, utilizing different GNSS services, are possible.
2. Low Visibility Positioning (LVP): This module employs a simple Kalman filter.
3. EGNOS: This module is based on Satellite-Based Augmentation System (SBAS) for positioning.
4. GnsInsKalman: This positioning module integrates GNSS positions and Inertial Measurement Unit (IMU) data. These modules can operate simultaneously, with each active module generating its position output log file. The PVT values are calculated at the primary measurement rate, typically one second, and stored within the log file.

File Extension	*.poslog
File Sort	MuSNAT specific
File Type	Output
File Version	1.0
Output activation setting:	SPP*.CreateSppLogFile LVP.CreateLvpLogFile EGNOS.CreateEgnosLogFile GnsInsKalman.CreateGnsInsKalmanLogFile
Output path setting	SPP*.SppLogFile LVP.LogFile EGNOS.EgnosLogFile GnsInsKalman.PoslogGnsInsFilterFilePath
Rate of file generation	One file, created at receiver start (time tagging possible)
Number of files	One file per navigation module with activated output
Data output rate into the file	Continuously, defined by overall measurement rate
External Tool	\\WorkingData\MatlabTools\+MuSNAT\plotSpplog.m (MATLAB plotting tool) \\WorkingData\MatlabTools\+MuSNAT\GetKmlFileFromSppFile.m (MATLAB data conversion tool)

Field Description:

Field	Field Name	Description	Unit
1	GPS Week	GPS time week number	-
2	GPS second	GPS time second	s
3	Latitude	Ellipsoidal latitude WGS84	deg
4	Longitude	Ellipsoidal longitude WGS84	deg
5	Height	Ellipsoidal height WGS84	m
6	WGS84 Position x	WGS84 X-coordinate of receiver	m
7	WGS84 Position y	WGS84 Y-coordinate of receiver	m
8	WGS84 Position z	WGS84 Z-coordinate of receiver	m
9	Accuracy position	Estimated 3-dimensional accuracy of the calculated receiver position	m
10	WGS84 Velocity x	WGS84 X-direction velocity of receiver	m/s

11	WGS84 Velocity y	WGS84 Y-direction velocity of receiver	m/s
12	WGS84 Velocity z	WGS84 Z-direction velocity of receiver	m/s
13	Accuracy velocity	Estimated 3-dimensional accuracy of the calculated receiver velocity	m/s
14	Measured Clock Error	One part of the overall receiver clock error, this part is determined during the receiver position calculation and is not included within the pseudo range calculation, typically <10ms	s
15	Std.Dev. Measured Clock Error	Standard deviation of the measured clock error	s
16	Used Clock Error	One part of the overall receiver clock error, this part is used when the pseudoranges are calculated and therefore is included within the pseudoranges, typically constant for plenty of consecutive log file entries	s
17	Std.Dev Used Clock Error	Standard deviation of the used clock error, typically 0.0	s
18	Clock Drift	Estimated receiver clock drift	s/s
19	Std.Dev Clock Drift	Standard deviation of estimated receiver clock drift	s/s
20	Number of signals	Number of GNSS signals used to calculate the receiver position	-
21	PDOP	Position (3D) Dilution of Precision	m
22	HDOP	Horizontal Dilution of Precision	m
23	VDOP	Vertical Dilution of Precision	m
24	EDOP	Easting Dilution of Precision	m
25	NDOP	Northing Dilution of Precision	m

Navigation Data File (.YYn)

This file contains navigation data for GPS satellites. The settings for the configuration can be defined in *musnat.-[Section NavigationData](#)

File Extension	*.n or *.YYn or *.nav
File Sort	MuSNAT specific
File Type	Output
Output activation setting:	NavigationData.RINEXGPSNavCreateFile
Output path setting:	NavigationData.OutputFileName
Rate of file generation:	MuSNAT not running permanently: <ul style="list-style-type: none"> • One file per MuSNAT shut down • New file created in case of a day change MuSNAT running permanently: <ul style="list-style-type: none"> • One daily file
Number of files:	One file per creating event
Data output rate into the file:	Once at receiver shut-down and once per day change

Following is a screenshot of the .nav file obtained as an output from the MuSNAT processing of GPS L1, L5 and Galileo E1,E5 dataset.

```
#Main
NumberOfRecords,i,72
#Record1
Record_Issue,i,0
Record_Type,i,43
GlonassSatellites,vi,0,
Record_SatId,vi,3,0,0,-123
Record_RefTime,pt,2,0,0.0000000000000000
Record_RangeRefTime,ptb,6,0,0.0000000000000000,0,0,0.0000000000000000,0
#Record10
Record_Issue,i,0
Record_Type,i,7
Almanac_Af0,d,-4.57763671875000e-05
Almanac_Af1,d,2.54658516496420e-11
Record_SatId,vi,3,2,3,-123
Record_RefTime,pt,2,1382892288,0.0000000000000000
Record_RangeRefTime,ptb,6,259200,0.0000000000000000,1,0,0.0000000000000000,0
#Record11
Record_Issue,i,0
Record_Type,i,6
Almanac_DeltaI,d,0.0207327090862669
Almanac_Eccentricity,d,0.00265598297119141
Almanac_MeanAnomaly,d,2.46959147328900
Almanac_Omega0,d,1.62341643680512
Almanac_OmegaDot,d,-7.71460705863541e-09
Almanac_Perigee,d,-3.05836595669237
Almanac_SquareRootSemiMajorAxis,d,5153.55761718750
Record_SatId,vi,3,2,4,-123
Record_RefTime,pt,2,1382978304,0.0000000000000000
Record_RangeRefTime,ptb,6,259200,0.0000000000000000,1,0,0.0000000000000000,0
```

Field Description:

Field Name	Description	Notes	Unit
NumberOfRecords	Number of records to read/write		

Record_Issue	Number of record identifier		
Record_Type	Type of record	Possible types of navigation record:	
		<ul style="list-style-type: none"> • Unknown record - 0 • Satellite health - 1 • C/A orbit - 2 • C/A clock - 3 • C/A ionosphere - 4 • C/A UTC - 5 • C/A almanac orbit - 6 • C/A almanac clock - 7 • Static position - 8 • Kepler orbit - 9 • Static clock - 10 • Tropospheric parameters - 11 • Precise orbit data (for one satellite) - 12 • Precise clock data (for one satellite) - 13 • EGNOS Message Type 1 - 14 • EGNOS Message Type 2 to 5 - 15 • EGNOS Message Type 7 - 16 • EGNOS Message Type 10 - 17 • EGNOS Message Type 25 - 18 • RTCM Message Type 1 - 19 • CNAV ionosphere - 20 • CNAV UTC - 21 • CNAV almanac orbit - 22 • CNAV almanac clock - 23 • CNAV Health - 24 • CNAV clock - 25 • CNAV orbit - 26 • CNAV Reduced Almanac Orbit - 27 • Giove-A Orbit - 28 • Giove-A Clock - 29 • Giove-A Clock Almanac - 30 • Giove-A Orbit Almanac - 31 • Galileo Orbit - 32 • Galileo Clock - 33 • Copy of the received navigation data - 34 • EGNOS satellite coordinates - 35 • EGNOS clock corrections - 36 • EGNOS almanac - 37 • EGNOS Ionospheric Delay Corrections - 38 • GLONASS ephemeris orbit - 39 • GLONASS clock - 40 • GLONASS almanac orbit - 41 • Data from a reference station - 42 • GLONASS satellites identifiers - 43 • Galileo Iono - 44 • Beidou orbit - 45 • Beidou clock - 46 • Beidou ionosphere - 47 • Beidou almanac orbit - 48 • Beidou almanac clock - 49 • Beidou almanac clock - 50 • (Crypto) Key data - 51 • IRNSS orbit - 52 • IRNSS clock - 53 • IRNSS ionosphere - 54 • IRNSS almanac orbit - 55 • IRNSS almanac clock - 56 • Galileo High Accuracy Service (HAS) orbit correction - 57 • Galileo High Accuracy Service (HAS) clock correction - 58 • Galileo High Accuracy Service (HAS) code bias correction - 59 • Galileo High Accuracy Service (HAS) phase bias correction - 60 	
GlionassSatellites			
Record_SatId	Transmitter identifier		
Record_RefTime	Reference time (epoch) of navigation record (for example, that's the Toe+GPSWeek of orbit ephemeris, Toc+GPSWeek of clock ephemeris,...)	If using Bits, RefTime refers to middle of Symbol/Bit Sequence	
Record_RangeRefTime	Validity range of reference time		s
Almanac_Af0	Polynomial coefficient		s
Almanac_Af1	Polynomial coefficient		s/s
Almanac_DeltaI	Correction of inclination angle		rad

Bits	String containing the raw navigation data stream as bits or symbols (depending on the original signal data) in a hexadecimal coding.	Activated when NumberAssistanceBits in Section Receiver* is set to a positive, non zero number (f.e. 300). The following translation table has to be applied:																					
		<table border="1"> <thead> <tr> <th>Symbol</th> <th>Translation</th> </tr> </thead> <tbody> <tr> <td>f</td> <td>-1, -1</td> </tr> <tr> <td>d</td> <td>-1, 1</td> </tr> <tr> <td>c</td> <td>-1, 0</td> </tr> <tr> <td>7</td> <td>1,-1</td> </tr> <tr> <td>5</td> <td>1, 1</td> </tr> <tr> <td>4</td> <td>1, 0</td> </tr> <tr> <td>3</td> <td>0,-1</td> </tr> <tr> <td>2</td> <td>0, 1</td> </tr> <tr> <td>1</td> <td>0, 0</td> </tr> </tbody> </table>	Symbol	Translation	f	-1, -1	d	-1, 1	c	-1, 0	7	1,-1	5	1, 1	4	1, 0	3	0,-1	2	0, 1	1	0, 0	
Symbol	Translation																						
f	-1, -1																						
d	-1, 1																						
c	-1, 0																						
7	1,-1																						
5	1, 1																						
4	1, 0																						
3	0,-1																						
2	0, 1																						
1	0, 0																						
Ephemeris_FitInterval	GPS specific fit interval																						
Ephemeris_Health	Ephemeris health information																						
Ephemeris_L2Codes	GPS specific L2 codes																						
Ephemeris_L2PFlag	GPS specific L2P data flag																						
Ephemeris_Urldx	Ephemeris user range accuracy																						
Ephemeris_Cic	Amplitude of the cosine harmonic correction term to the angle of inclination		rad																				
Ephemeris_Cis	Amplitude of the sine harmonic correction term to the angle of inclination		rad																				
Ephemeris_Crc	Amplitude of the cosine harmonic correction term to the orbit radius		m																				
Ephemeris_Crs	Amplitude of the sine harmonic correction term to the orbit radius		m																				
Ephemeris_Cuc	Amplitude of the cosine harmonic correction term to the argument of latitude		rad																				
Ephemeris_Cus	Amplitude of the sine harmonic correction term to the argument of latitude		rad																				
Ephemeris_DeltaN	Mean motion difference from computed value		rad/s																				
Almanac_Eccentricity	Eccentricity, it is dimensionless		rad																				
Ephemeris_Eccentricity																							
Ephemeris_I0	Inclination angle at reference time		rad																				
Ephemeris_IDOT	Rate of Inclination Angle		rad/s																				
Almanac_MeanAnomaly	Mean anomaly at reference time		rad																				
Ephemeris_MeanAnomaly																							
Almanac_Omega0	Longitude of ascending node of orbital plane at weekly epoch		rad																				
Ephemeris_Omega0																							
Almanac_OmegaDot	Rate of right ascension		rad/s																				
Ephemeris_OmegaDot																							
Almanac_Perigee	Argument of perigee		rad																				
Ephemeris_Perigee																							

Almanac_Square RootSemiMajorA xis	Square root of semi-major axis		$m^{(1/2)}$
Ephemeris_Squa reRootSemiMajo rAxis			

RINEX Observation Data File (.YYo)

This document compiles observation data in the Receiver-INdependent EXchange format, accommodating both standard observations and the option to store code-phase measurements as non-standard observations. It also allows for multiple independently configured RINEX output writers. The settings for the configuration can be defined in *musnat.-Section [RINEX_Obs](#)*

File Extension	*.O or *.YYO or *.obs
File Sort	Standard
File Type	Output
File Version:	2.11 or 3.01
Output activation setting:	RINEX_Obs*.GenerateRINEX_Output
Output path setting:	RINEX_Obs*.FileName
Rate of file generation:	File created at receiver start, in case a new day begins the current file is finalized, and a new daily output file is opened
Number of files:	One file per creating event, but <ul style="list-style-type: none">• New file created in case of a day change
Data output rate into the file:	Continuously defined by the RINEX output interval and the overall measurement rate
Applicable Reference Documents	RINEX 2.11 or RINEX 3.01

The field descriptions and data examples are integral components of the RINEX reference documentation. Consequently, they are not provided here.

Tracking Dump Log (.dumplog)

This file details the status of the satellite signal tracking channels following specific intervals known as primary integrate and dump periods. These periods align with the primary PRN code length considering a possibly present division of the coherent integration time defined by the parameter Receiver.CoherentFraction used for long spreading codes like L2CL.

File Extension	*.dumplog
File Sort	MuSNAT specific
File Type	Output
File Version:	1.4
Output activation setting:	Receiver*.PRNsToLog
Output path setting:	Receiver*.LogFileTemplateName
Rate of file generation:	Depending on configuration: <ul style="list-style-type: none"> • One file per acquired satellite signal • One file per satellite signal per re-acquisition
Number of files:	One file per acquired and tracked satellite signal (satellite has to be configured for output)
Data output rate into the file:	One output per primary integrate and dump period (typically the PRN code length)
External Tools	\\WorkingData\MatlabTools\+MuSNAT\plotDumplog.m (MATLAB Tool)

Field Description:

Field	Field Name	Description
1	Week	Receiver time GPS week number
2	GPSsec(rec.)	Receiver time GPS second
3	Clock Error	Receiver estimated clock error
4	Week	Sent time signal GPS week number
5	GPSsec(sent)	Sent time signal GPS second
6	Code PR	Code pseudorange
7	Phase PR	Carrier pseudorange
8	C	Master channel status in tracking (0=idle, 1=tracking)
9	L	Loop mode (0=PLL, 1=FLL, 2-4=FLL-assisted PLL modes)
10	U	Indicates if the channel's work of undersampling (=1) or not (=0)
11	C/N0	Estimated signal power
12	Dcyc	NCO code cycles for data channel
13	Dcode	phase NCO code phase for data channel
14	Pcyc	NCO code cycles for pilot channel
15	Pcode	phase NCO phase values for pilot channel
16	Code rate	NCO code rate

17	Carrier phase	NCO values for slave (dual component) channel
18	Doppler	NCO carrier phase rate
19	C	NCO cycle slip flag (0 = phase lock maintained, 1=cycle slip)
20	DLL disc	Code phase error (Code tracking error including std. dev.)
21	PLL disc	Carrier phase error (Carrier phase tracking error)
22	FLL disc	Frequency tracking error
23	Week	Week number of used receiver clock error
24	Sec(clock er)	Used receiver clock error in seconds
25	N	Channel size (1=data, 2=data+pilot)
26	Vd	If 1 then data correlator valid
27	Dred	Real part of complex output of early-late correlator (data)
28	Dimd	Imaginary part of complex output of early-late correlator (data)
29	P1red	Real part of complex output of the first punctual correlator (data)
30	P1imd	Imaginary part of complex output of the first punctual correlator (data)
31	P2red	Real part of complex output of the second punctual correlator (data)
32	P2imd	Imaginary part of complex output of the second punctual correlator (data)
33	Vp	If 1 then pilot correlator valid
34	Drep	Real part of complex output of early-late correlator (pilot)
35	Dimp	Imaginary part of complex output of early-late correlator (pilot)
36	P1rep	Real part of complex output of the first punctual correlator (pilot)
37	P1imp	Imaginary part of complex output of the first punctual correlator (pilot)
38	P2rep	Real part of complex output of the second punctual correlator (pilot)
38	P2imp	Imaginary part of complex output of the second punctual correlator (pilot)
40	Ncd	Number of multi-correlator values in data channel
41	MCd1re	Real part of first data multi-correlator
42	MCd1im	Imaginary part of first data multi-correlator
	...	(Multiple repetitions)
	MCd(Ncd)re	Real part of last (Ncd) data multi-correlator
	MCd(Ncd)im	Imaginary part of last (Ncd) data multi-correlator
40+ 2*Ncd	Ncp	Number of multi-correlator values in pilot channel
	MCp1re	Real part of first pilot multi-correlator
	MCp1im	Imaginary part of first pilot multi-correlator
	...	(Multiple repetitions)
	MCp(Ncp)re	Real part of last (Ncp) pilot multi-correlator

40+ 2*Ncd+ 2*Ncp	MCp(Ncp)im	Imaginary part of last (Ncp) pilot multi-correlator
------------------------	------------	---

Week	GPSsec(rec.)	Clk. err. Week	GPSsec(sent)	Code PR	Phase PR	C L U	C/N0	DCyc	DCode phase	PCyc	PCode phase
Code rate	Carrier phase	Doppler	C DLL disc PLL disc	FL disc Week	Sec(clock err.)	N Vd		Dred	Dimd	Pired	Piimd
P2red	P2imd	Ncd MCdlre	MCdlim ... MCd(Ncd)re MCd(Ncd)im	[m]	[m]	[dBHz]		[subchip]		[subchip]	
	[s]	[s]	[s]	[subchip]	[cyc]	[Hz]					
	[subchip/s]	[cyc]	[Hz]								
	[s]										
2286	212067.6720910500	0.0000000000	0	0.0000000000	0.000	0.00000	1 1 0 -100.000	2	0.1012109799	0	0.0000000000
20460065.637835	7287.2572	5054.11431	1	0.0000	0.0000	0.0000	0.0000000000 1 1	1	531.1	-1663.7	10249.6 306.5
10849.9	-2670.2	0									
2286	212067.6730910500	0.0000000000	0	0.0000000000	0.000	0.00000	1 1 0 -100.000	3	0.1668488151	0	0.0000000000
20460065.637835	12292.7572	5054.11431	1	0.0000	0.0000	0.0000	0.0000000000 1 1	1	468.2	361.5	5955.5 673.7
6668.5	-3597.2	0									
2286	212067.6740910500	0.0000000000	0	0.0000000000	0.000	0.00000	1 1 0 -100.000	4	0.2324866503	0	0.0000000000
20460062.795901	17298.2572	5054.14225	1	-1.0634	0.0863	29.3531	0.0000000000 1 1	1	469.9	-1089.2	6426.4 -3457.5
6815.3	-4522.1	0									
2286	212067.6750910500	0.0000000000	0	0.0000000000	0.000	0.00000	1 1 0 -100.000	5	0.2952825510	0	0.0000000000
20460060.709393	22303.7572	5054.36111	1	-1.8427	0.2271	221.7887	0.0000000000 1 1	1	76.3	-1457.2	3376.9 -6287.8
-1450.1	-7012.3	0									

Multi-Correlator log (.mclog)

These files comprise a grid of correlation function values (usually called multi-correlator values or Doppler/delay maps) derived from GNSS signals tracked by MuSNAT. The grid is determined by intentionally chosen PRN-code delays and Doppler frequency offsets, representing the disparities between the incoming and generated reference signals. The computation of the values is described in <https://www.ion.org/publications/abstract.cfm?articleID=9756>. Multicorrelator values are generated at user defined rate for the tracked signals. It should be noted that this results usually in large number of files depending on the considered duration, number of tracked signals and multi-correlator integration time.

File Extension	.mclog
File Sort	MuSNAT
File Type	Output
File Version:	1.0
Output activation setting:	Receiver*.DumpMultiCorrValues
Output path setting:	Receiver*.DumpMultiCorrValuesFileTemplate
Rate of file generation:	File generation for every coherent integration epoch of the multi-correlator, usage of a continuously counter to distinguish between integration epochs
Number of files:	Five files per coherent integration epoch: - Info file containing meta-data (...Info.mclog) - Time domain I-data (...IT.mclog) - Time domain Q-data (...QT.mclog) - Frequency domain I-data (...IF.mclog) - Frequency domain Q-data (...QF.mclog)
Data output rate into the file:	Once immediately after file creation
Connected GUI widget:	Corr-3D
External Tools	\\WorkingData\MatlabTools\+MuSNAT\plotDumplogWithMC.m (MATLAB plotting tool)

Field	Field Name	Description
1	Nominal rec. time	Receiver time GPS week number and second
2	Applied clock error	Receiver estimated clock error [s]
3	Sent time	Sent time signal GPS week number and second
4	Related RINEX epoch	Week and second of the RINEX observation file record simultaneously recorded
5	Position age	Time difference between time of multi-correlator calculation and time of position calculation [s] (PositionAge=MCTime-TimeOfPositon)
6	Latitude	Latitude (WGS84) of current receiver position [deg]
7	Longitude	Longitude (WGS84) of current receiver position [deg]
8	Height	Height (WGS84) of current receiver position [m]
9	Velocity	Current receiver velocity WGS84 [m/s]
10	Signal component	Definition of the GNSS signal component - "0" for data component - "1" for pilot component
11	Satellite	System number, service number and PRN number of the logged satellite, or instance 2/1/13 represent GPS L1 C/A satellite PRN 13
12	Sat. position	Satellite position X,Y,Z WGS84 [m] at time of multi-correlator calculation
13	Sat. velocity	Satellite velocity X,Y,Z WGS84 [m/s] at time of multicorrelator calculation
14	Sat. clock error	Satellite clock error [s]

15	Tropo delay	Tropospheric delay alongside signal path [m]
16	Iono delay	Ionospheric delay alongside signal path [m]
17	Sample rate	Sample rate of the GNSS signal [Hz]
18	Code rate	Code rate of the GNSS signal [chip/s]
19	Reference pseudorange	Pseudorange [m] at time of multi-correlator calculation
20	Number of code bins	Number of PRN code bins used to define the 2-dimensional grid for the multi-correlator calculation
21	Code offsets	Vector of PRN code offsets for the 2-dimensional grid points with respect to the punctual correlator used for signal tracking, all value [m]
22	Reference Doppler	Carrier Doppler offset of the currently tracked GNSS signal [Hz]
23	Number of Doppler bins	Number of Doppler bins used to define the 2-dimensional grid for the multi-correlator calculation
24	Doppler offsets	Vector of relative Doppler offsets for the 2-dimensional grid points with respect to the punctual correlator used for signal tracking, all values [Hz] (Hint: the punctual correlator used for signal tracking differs from the nominal carrier frequency by "Reference Doppler")

These files contain a 2-dimensional grid of correlation function values from GNSS signal which are tracked by the MuSNAT. The grid points are defined by intentionally used PRN-code delays and Doppler frequency offsets between the incoming signal and the generated reference signal.

```

OrbitFilter.m  SimpleCpsd.L1SGalileoE1ES.musnat  SimpleGPS.L1.musnat  GPS_L5_MCDump_sys2_sat30_chan0_cnt1_typeIF.mclcg  MuSNAT.defs.xml
1  8101.96 4558.52 17371.17 1563.45 -31976.45 -14139.34 13823.74 3571.23 -13910.99 4610.59 22146.41 20620.70 6938.08 13164.44 4634.90 -14763.60 -41848.73 -20441.56 20316.02 6617.38 2666
2  11286.77 11048.57 6295.45 15776.25 -13451.96 -49107.81 -9541.46 23561.13 -5895.04 -19495.30 270.96 19991.57 10778.26 -10009.25 -14561.03 -12910.39 -9693.04 -4931.35 -9661.58 10525.00
3  7504.21 -10324.86 -37471.04 -11452.56 6339.05 772.43 15369.40 4692.91 -10520.57 -1328.15 10212.95 -3704.42 4770.13 7437.72 8952.12 6574.04 -2924.81 12925.95 45553.52 22322.01 2259.35
4  19083.22 4492.43 -33779.26 -21275.23 17681.54 30198.39 12827.01 10771.94 22734.79 -13127.15 -26764.62 -19585.68 -33952.19 -12621.82 -5558.88 1371.63 4654.23 10196.00 4452.96 -16103.8
5  11637.27 -8597.85 9172.26 22991.56 299.28 2731.83 23218.70 1753.57 -13847.29 -19099.88 -15087.94 11467.17 14480.76 24938.32 17331.51 -4735.78 -2429.09 -511.33 -7704.97 -10793.30 -120
6  -7526.69 -16131.08 -22369.70 -9892.53 -5213.29 -3653.97 -29716.49 -35374.63 -16802.63 -25783.10 -20455.08 -18517.14 -28470.62 -16121.85 -5528.88 -19654.60 4661.25 5205.27 -23270.21 -
7  -17935.28 -29632.62 -15327.79 11338.71 12421.45 -4487.78 -31728.11 -9557.89 12971.30 -6437.27 -7826.12 -14109.90 4228.97 18243.22 -5865.73 3239.26 17659.43 -266.18 -632.92 4824.85 -6
8  25729.38 10960.38 200.98 -10043.31 -9428.99 975.08 -3563.25 -11980.30 -12859.70 1170.41 13216.74 10190.15 21882.03 2956.88 2506.64 -15895.03 -42442.34 -32410.17 -12137.22 -12256.07 -
9  -20881.18 -9955.26 -1384.25 -37358.06 -9761.91 14595.31 -4879.83 -11755.84 -13693.02 -2082.20 1204.77 -5180.64 -14306.01 -19404.81 -18988.64 -29302.34 -15888.62 -16011.71 -25360.38 98
10 -5286.35 2084.18 -14190.25 -912.53 8936.23 -14713.38 -7031.82 -6258.53 2487.93 -17066.97 -12094.80 -767.52 3913.01 -8240.27 -16529.34 -17287.18 -2469.94 1785.01 -21674.77 -28039.59 -
11  2365.56 -25631.75 -20911.23 18865.81 36579.14 -8010.89 -6460.61 42836.48 27078.09 -8468.86 -12301.52 -3831.67 6237.51 1668.16 26680.07 14889.76 -403.03 -22423.80 -28555.67 -19094.73
12  25519.25 4826.41 8050.29 2734.56 -15080.09 -10175.51 6736.28 -9476.07 -33737.49 -13559.60 729.43 16119.97 19477.67 26301.78 14575.95 -18336.67 -4630.13 447.59 -37446.64 -46898.27 854
13 -25048.95 -5901.84 16248.70 -1864.68 3252.58 -3145.59 -8297.79 4072.76 -4457.18 -9636.67 -3809.10 -1586.83 -1876.60 -6025.71 -43603.96 -40956.37 -29006.12 203.12 -8413.85 -17346.48 -
14 -5086.96 22428.32 5898.80 -20279.06 -34161.93 -5029.70 499.85 -25024.21 -9355.26 4004.05 12514.63 -4108.21 -2663.42 22748.03 18670.41 7516.69 -25603.31 -19262.36 92.46 10833.90 8897.
15  9047.86 336.75 -15339.07 4274.24 4878.35 -27867.56 -22172.11 7356.60 -2317.04 -9551.92 -25210.44 -14405.37 13551.89 7265.39 -24314.63 -25863.70 -24656.00 -18611.71 13604.51 27088.11
16  13694.40 24247.85 558.71 1616.10 2672.23 -11820.80 -18913.09 -16606.79 18149.63 26863.47 13431.29 10599.34 23837.81 12107.89 51.14 5486.36 4493.92 -13668.38 -16189.24 -8638.21 -21945
17 -12226.22 1166.85 23404.20 19381.87 5834.81 -2511.42 -2930.11 3498.87 15237.16 -2455.82 -21343.02 -15013.09 3143.58 3843.14 -17807.67 -53664.46 -16747.06 4729.90 -10761.35 -16954.83
18  21064.16 13253.77 -3422.99 8473.33 24654.04 11717.09 -13.88 5257.59 14979.47 -9600.31 -6568.62 -16065.94 -16337.19 -13083.99 -25805.31 -20742.11 -24655.51 -31730.26 -1725.86 8925.05
19  22272.14 14523.92 -1804.97 -5094.59 -2723.83 -14121.38 6710.29 13152.56 33419.99 32690.63 10399.24 9946.69 -2783.50 -17721.87 -25096.47 -25933.97 24358.18 38063.83 -8342.21 -20360.93

```

Receiver Status Output File (*.pvt)

The following file contains the output comprising the last receiver status in a .pvt file.

```
#Main
NominalReceiverTime,pt,2,1382785174,0.546834470153259
ClockError,ptb,6,-5,0.550000000000000,1,0,0.00000000000000,1
MeasuredClockError,ptb,6,-1,0.996834471126619,1,0,6.14020823740305e-09,1
ClockDrift,db,4,-1.13067996010223e-06,1,2.10242655251724e-09,1
Position,pos,16,4182061.96296047,860621.143948746,4723115.44210594,1,-0.219104230597665,0.0960072552906451,-1.64544170303469,1,3.21479327449029,1,0.00000000000000,0,1.08073023199114,1,0.00000000000000,0
```

Field Description:

Name	Field Description	Unit
Nominal Receiver Time		
ClockError	This part of the receiver clock error is used when the pseudorange values are calculated, and this is included in the pseudorange values (Even if the total receiver clock error is very large, the calculated pseudorange values are in a reasonable range (about 20000-30000km). The standard deviation of this clock error is usually set to 0.0)	
MeasuredClockError	A posteriori receiver clock error; This part of the clock error is determined during the single point positioning and is NOT used to calculate the pseudorange values.	
ClockDrift	Clock Drift	s/s
Position		

Structure:

#Main

NominalReceiverTime,pt,2,GPSSeconds(*Integer*), GPSSeconds(*Double*, for decimal spaces)

ClockError,ptb,6,Clockerror(*Integer*),Clockerror(*Double*, for decimal spaces),Valid Flag for Clockerror(*Bool*),Clockerror standard deviation(*Integer*), Clockerror standard deviation(*Bool*, for decimal spaces),Valid Flag for Clockerror standard deviation(*Bool*)

MeasuredClockError,ptb,6,MeasuredClockerror(*Integer*),MeasuredClockerror(*Double*, for decimal spaces),Valid Flag for MeasuredClockerror(*Bool*), MeasuredClockerror standard deviation(*Integer*),MeasuredClockerror standard deviation(*Bool*, for decimal spaces),Valid Flag for MeasuredClockerror standard deviation(*Bool*)

ClockDrift,db,4,Drift in s/s (*Double*),Valid Flag for Drift (*Bool*),Standard deviation of Drift (*Double*),Valid Flag for Std of Drift (*Bool*)

Position,pos,16, X in m (*Double*),Y in m (*Double*), Z in m (*Double*), Valid Flag for Position (*Bool*),Velocity of X in m (*Double*),Velocity of Y in m (*Double*), Velocity of Z in m (*Double*),Valid Flag for Velocity (*Bool*), Position accuracy in m (*Double*),Valid Flag for position accuracy (*Bool*),Standard deviation for position accuracy in m (*Double*),Valid Flag for position accuracy standard deviation (*Bool*),Velocity accuracy in m (*Double*),Valid Flag for velocity accuracy (*Bool*),Standard deviation for velocity accuracy in m (*Double*),Valid Flag for velocity accuracy standard deviation (*Bool*)

Configuration Structure

Some of the basic building blocks of a .musnat configuration file can be seen in the sample configuration file attached here. The basic sections involved in the receiver configuration file to ensure an optimal functionality of the receiver are as follows:

	Section	Purpose
1	Acquisition	Basic settings for the acquisition of signal before doing an FFT Acquisition
2	CentralizedPhaseTracking	To exchange NCO values between channels. It is essential for vector tracking or master/slave operations
3	FFTAcquisitionLevel*	GNSS signal acquisition settings
4	Main	Basic musnat settings including the path to IF stream file
5	Navigation	Basic settings for the management of the navigation modules of the receiver.
6	NavigationData	Settings related to managing and utilizing navigation data, such as ephemeris data. Furthermore, it covers the configuration of navigation data file generation
7	RINEX_Obs	Basic settings for the output of RINEX 2.11 or 3.01 observation data files
8	Receiver*	GNSS signal processing settings for a received signal
9	ReceiverStatus	Settings to maintain the latest receiver position estimate and store data from the most recent observed epoch. (Also used to initiate acquisition modules to emulate a warm or hot start of the receiver in post-processing).
10	SPP	Settings for the receiver position calculation using a single point positioning least squares algorithm
11	SpectrumAnalyzer	Basic settings for the estimation of the power spectral density of incoming IF samples

Each parameter mentioned in the configuration file shall be explained in the dedicated section.

[As an example, take a look at the *.musnat file in Appendix](#)

Description of the User Interface

The complete MuSNAT software receiver package comprises **three distinct user interfaces** that present visually rich information to the user, generated from the input data set and configured settings.

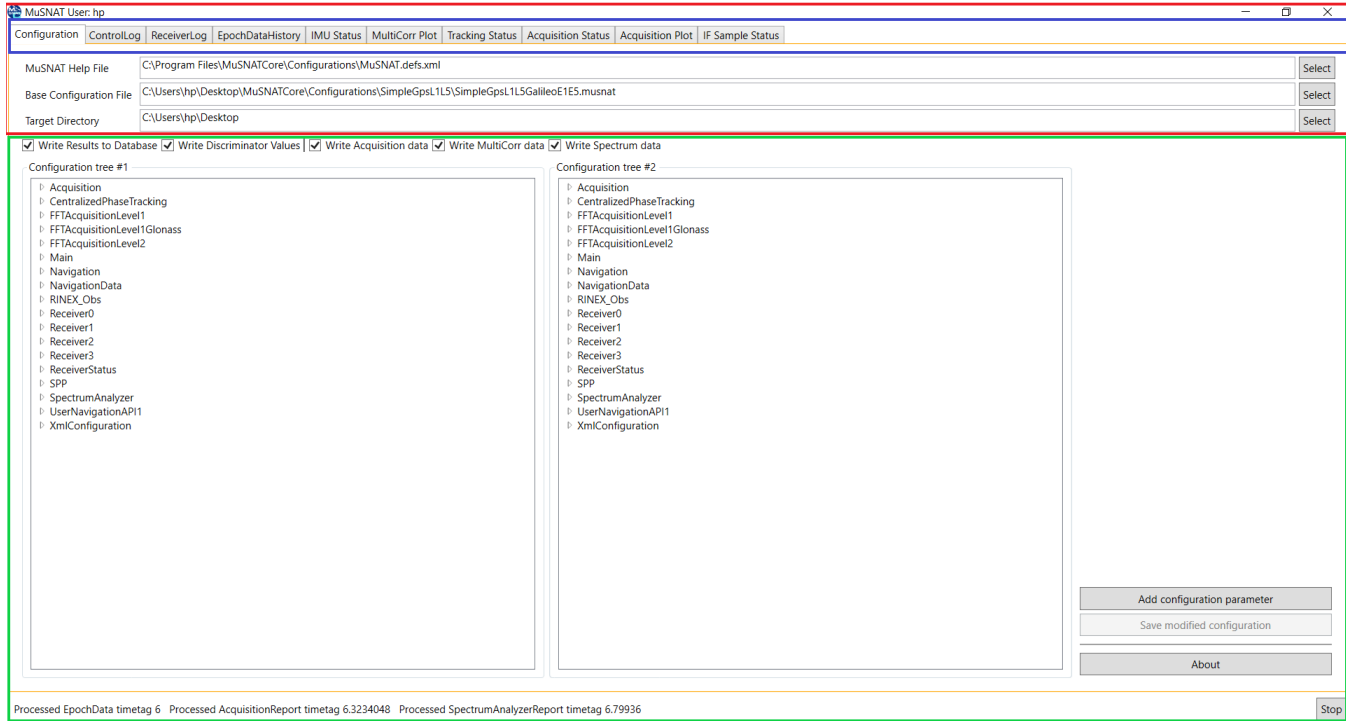
1. MuSNAT-Core
2. MuSNAT-Analyzer
3. MuSNAT-ASCII

MuSNAT-Core with Graphical User Interface

The MuSNAT Core GUI serves as the primary User Interface that should be launched to analyze various parameters associated with the input dataset or specific cases and provides feedback while processing the data.

After starting the receiver, the user can see the three main components of the MuSNAT Core User Interface, which is divided as follows:

1. **Menu bar** (framed red)
2. **Settings and configuration window** (framed green)
3. **Tool and plot window** (framed blue)



As seen in the settings and configuration window (framed green), checking the **'Write results to Database'** creates an SQL-based database file (.db), which the **MuSNAT Analyzer** can read for a more detailed analysis. Each of these checkboxes corresponds to a specific type of data to be written onto the SQL database file, just as the name of these checkboxes indicates.

However, in addition to **SQL db file**, it is possible to generate **ASCII log files** (such as .acqlog for logging the acquisition data) separately through the .musnat configuration file. More information about the activation settings to generate these ASCII logfiles has been explained in the [dedicated section](#). It is important to note that the SQL file is equivalent to the ASCII log files. A distinct advantage of having SQL file is consolidating various types of information (acquisition data, discriminator data etc), into a single file rather than dispersing them across multiple files.

To initiate the receiver manually, please adhere to the following instructions on the Menu bar (framed red):

1. Choose the location of the **MuSNAT Help File** (MuSNATdefs.xml).
2. Specify the location of the appropriate **base configuration file** (.musnat) based on the specific scenario.
3. Indicate the **target directory** where the output files should be saved. By default, these files are typically written to the desktop, with the folder being named according to the date and start time of the receiver.
 - a. Please note that in addition to the target directory, some output files are written to the directory containing the base configuration file. This is due to some legacy code inside MuSNAT that does not recognize the target directory settings. In future all output files shall reside in the target directory.
4. Click on **Start**
5. Carefully watch the processing results in the **output windows** until processing stops
6. During or after the processing use the MATLAB tools to visualize processing results and/or use MuSNAT-Analyzer.

Menu Bar

The 'Menu Bar' grants access to the core functionalities of MuSNAT. It allows for direct actions, such as launching the software receiver, as well as providing access to additional GUI windows for further functionality. As depicted in the figure, the menu bar has 3 main items and the table below provides an overview of each of these items.

Item	Description	Format
MuSNAT Help File	File consisting of definitions for all the configuration parameters	.xml
Base Configuration File	File consisting of all the necessary settings required to process the desired input dataset	.musnat
Target Directory	Folder consisting of all the output files. Please note that in addition to the target directory, some output files are written to the directory containing the base configuration file. This is due to some legacy code inside MuSNAT that does not recognize the target directory settings. In future all output files shall reside in the target directory.	Folder named based on the start date and time of the data processing

Settings and Configuration Window

Users can toggle specific settings related to the output data through this window, deactivating or activating them as needed. The configuration tree also offers a comprehensive view of all user-defined parameters within the configuration file. It is possible to modify any parameter directly or add parameters within this interface and save these changes to the configuration file.

Item	Description
Write results to Database	All the tracking and acquisition data is written to one single SQL-based database (*.db) file, which can further be examined through the MuSNAT Analyzer
Write Discriminator Data	Orders the receiver to store the code and carrier discriminator information to be able to be further examined by the MuSNAT Analyzer.
Write Acquisition Data	Orders the receiver to store the signal acquisition-related information such as the carrier frequency, code-phase, doppler shift, satellite PRN etc
Write MultiCorr Data	Stores the multi-correlator data such as the doppler, code phase, and the complex correlator values
Write Spectrum Data	Stores information related to the spectral characteristics of the acquired signal.

When the specified options are enabled, the results are saved at the SQL database level as a Data Base File (*.db). This file can then be used for further analysis with the MuSNAT Analyzer. Kindly note that opting for all the options might lead to a minor reduction in the receiver's operational speed, and this is a typical occurrence.

Each parameter mentioned in the configuration tree shall be explained in detail in the [configuration parameters](#) chapter.

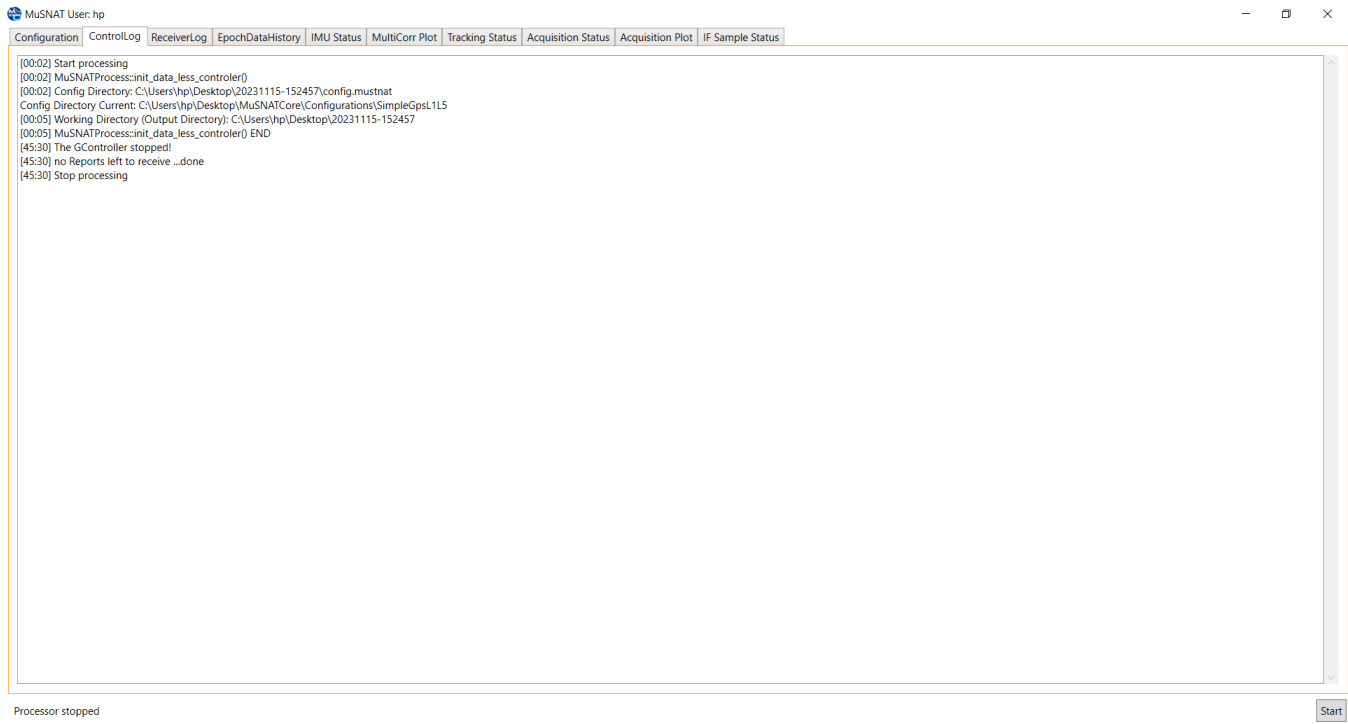
Once the user has configured the parameters and files in these two windows, initiating the receiver can be accomplished by clicking the 'start' button. Once running, the receiver will generate a log containing essential output data, including various plots, which can be viewed in the 'Tool and Plot' window.

Tool and Plot Window

This window displays all the information about the data processing held within the receiver.

Control Log Window

This section of the User Interface offers details regarding the time duration of the receiver's start and endpoints. Furthermore, it provides information about the current directory, which is the source location for reading IF sample data. It also indicates the working directory where the collected output data is being stored.



Receiver Log Window

This window provides a chronological record of the receiver's core operations. These logged messages play a pivotal role in detecting and tracking missing input files and revealing the factors contributing to unexpected receiver process terminations.

```

[00:04] Error reading navigation data from 'SimpleGpsGal_L1E1.nav'
[00:04] Sample Info: stream 0, FMT=iF, IF=5.000446 MHz, fs=20.000000 MHz
[00:04] Sample Info: stream 1, FMT=iF, IF=5.000446 MHz, fs=20.000000 MHz
[00:04] Sample Info: stream 2, FMT=iF, IF=14.998706 MHz, fs=20.000000 MHz
[00:04] Sample Info: stream 3, FMT=iF, IF=14.998706 MHz, fs=20.000000 MHz
[00:04] SampleFileInpustart -> use local found sensor file "SwRxData.la" (do not use sensor file defined within stream file configuration)
[00:04] Initializing master receiver
[00:04] CPU Type: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
[00:04] MuSNAT_core build date/time: Jul 24 2023, 22:00:23
[00:04] NO CUDA support
[00:04] NO QT support
[00:04] IF sample packet length = 0.033996800 s
[00:04] Rec. 0 (GPS_C/A_Code_Receiver): stream=0, RF=1575.420000 MHz, IF=5.000446 MHz, fs=20.000000 MHz.
[00:05] Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CNO: 0000
[00:05] Freq. Offset = 0.000 Hz
[00:05] Rec. 1 (Galileo_E1_OS_B+C_INAV V12): stream=0, RF=1575.420000 MHz, IF=5.000446 MHz, fs=20.000000 MHz.
[00:05] Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CNO: 0000
[00:05] Freq. Offset = 0.000 Hz
[00:05] Rec. 2 (GPS_L1_S+Q): stream=2, RF=1176.450000 MHz, IF=14.998706 MHz, fs=20.000000 MHz.
[00:05] Code rate =40.920000 Msubchip/s, Subchip length = 7.326 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CNO: 0000
[00:05] Freq. Offset = 0.000 Hz
[00:05] Rec. 3 (Galileo_E5a_OS_+Q): stream=2, RF=1176.450000 MHz, IF=14.998706 MHz, fs=20.000000 MHz.
[00:05] Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CNO: 0000
[00:05] Freq. Offset = 0.000 Hz
[00:05] Receiver 4 (Dual_SpectrumAnalyzer_Config) works on multiple streams.
[00:05] Number of threads for all receivers+1(9) including OpenMP workers is exceeding number of cores(4). This may lead to excessive overhead!
[00:05] Acquisition type 4 running for Receiver0 on stream 0 and RF=1575420000.0000.
[00:05] Acquisition type 4 running for Receiver1 on stream 0 and RF=1575420000.0000.
[00:05] Acquisition type 4 running for Receiver2 on stream 2 and RF=1176450000.0000.
[00:05] Acquisition type 4 running for Receiver3 on stream 2 and RF=1176450000.0000.
[00:05] Acquisition engine "SecCodeNonCohFtAcq" running on CPU successfully initialized.
[00:05] SPP weight scheme: Unity variance
[00:05] Backend started
[00:05] Acquisition timing, preprocessing(level1): 0.060
[00:10] Acquisition timing, total(level1): 5.100 s for 50921472 correlator values
[00:10] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:13] Acquisition timing, preprocessing(level1): 0.045
[00:18] Acquisition timing, total(level1): 5.079 s for 41484288 correlator values
[00:18] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:20] Receiver2: switching PRN30 from FLL to PLL.
[00:21] Acquisition timing, preprocessing(level1): 0.050
[00:26] Acquisition timing, total(level1): 5.068 s for 48529408 correlator values
[00:26] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:32] Acquisition timing, preprocessing(level1): 0.092
[00:36] Acquisition timing, total(level1): 4.411 s for 38404096 correlator values

```

```

[00:05] Backend started
[00:05] Acquisition timing, preprocessing(level1): 0.060
[00:10] Acquisition timing, total(level1): 5.100 s for 50921472 correlator values
[00:10] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:13] Acquisition timing, preprocessing(level1): 0.045
[00:18] Acquisition timing, total(level1): 5.079 s for 41484288 correlator values
[00:18] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:20] Receiver2: switching PRN30 from FLL to PLL.
[00:21] Acquisition timing, preprocessing(level1): 0.050
[00:26] Acquisition timing, total(level1): 5.068 s for 48529408 correlator values
[00:26] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:32] Acquisition timing, preprocessing(level1): 0.092
[00:36] Acquisition timing, total(level1): 4.411 s for 38404096 correlator values
[00:36] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[00:52] Acquisition timing, preprocessing(level1): 0.113
[00:57] Acquisition timing, total(level1): 5.262 s for 35258368 correlator values
[00:57] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[01:21] Acquisition timing, preprocessing(level1): 0.156
[01:28] Acquisition timing, total(level1): 7.801 s for 22315008 correlator values
[01:28] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[01:35] Receiver2: switching PRN6 from FLL to PLL.
[01:44] Acquisition timing, preprocessing(level1): 0.222
[01:50] Acquisition timing, total(level1): 5.052 s for 17006592 correlator values
[01:50] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[02:05] Acquisition timing, preprocessing(level1): 0.168
[02:10] Acquisition timing, total(level1): 5.335 s for 32473088 correlator values
[02:10] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[02:13] Receiver0: switching PRN30 from FLL to PLL.
[02:13] Receiver0: switching PRN9 from FLL to PLL.
[02:13] Receiver1: switching PRN21 from FLL to PLL.
[02:13] Receiver1: switching PRN10 from FLL to PLL.
[02:13] Receiver3: switching PRN36 from FLL to PLL.
[02:18] Receiver2: switching PRN9 from FLL to PLL.
[02:30] Acquisition timing, preprocessing(level1): 0.128
[02:35] Acquisition timing, total(level1): 5.240 s for 24838144 correlator values
[02:35] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[02:36] Receiver0: switching PRN6 from FLL to PLL.
[02:37] Receiver1: switching PRN9 from FLL to PLL.
[02:37] Receiver1: switching PRN36 from FLL to PLL.
[02:42] Receiver3: switching PRN10 from FLL to PLL.
[02:42] Receiver3: switching PRN21 from FLL to PLL.
[02:47] Acquisition timing, preprocessing(level1): 0.174
[02:53] Acquisition timing, total(level1): 5.699 s for 21561344 correlator values
[02:53] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[02:56] Receiver1: switching PRN11 from FLL to PLL.

```

MuSNAT User: hp

Configuration | ControlLog | ReceiverLog | EpochDataHistory | IMU Status | MultiCorr Plot | Tracking Status | Acquisition Status | Acquisition Plot | IF Sample Status

```

[43:20] Receiver1: switching PRN19 from FLL to PLL
[43:23] Acquisition timing, preprocessing(level1): 0.100
[43:26] Acquisition timing, total(level1): 2.651 s for 18874368 correlator values
[43:26] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[43:27] Receiver1: switching back PRN19 to FLL due to low signal power.
[43:35] Acquisition timing, preprocessing(level1): 0.091
[43:40] Acquisition timing, total(level1): 5.030 s for 44302336 correlator values
[43:40] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[43:47] Acquisition timing, preprocessing(level1): 0.093
[43:53] Acquisition timing, total(level1): 5.167 s for 46366720 correlator values
[43:53] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[43:58] Receiver2: switching back PRN10 to FLL due to low signal power.
[44:01] Acquisition timing, preprocessing(level1): 0.092
[44:06] Acquisition timing, total(level1): 5.075 s for 48726016 correlator values
[44:06] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[44:14] Acquisition timing, preprocessing(level1): 0.090
[44:19] Acquisition timing, total(level1): 5.103 s for 57344000 correlator values
[44:19] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[44:27] Acquisition timing, preprocessing(level1): 0.131
[44:31] Acquisition timing, total(level1): 4.244 s for 47775744 correlator values
[44:31] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[44:38] Acquisition timing, preprocessing(level1): 0.096
[44:43] Acquisition timing, total(level1): 5.100 s for 64143360 correlator values
[44:43] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[44:52] Acquisition timing, preprocessing(level1): 0.086
[44:57] Acquisition timing, total(level1): 5.019 s for 59604992 correlator values
[44:57] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[45:04] Acquisition timing, preprocessing(level1): 0.099
[45:08] Acquisition timing, total(level1): 3.218 s for 29917184 correlator values
[45:08] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[45:12] Receiver1: switching PRN19 from FLL to PLL.
[45:15] Acquisition timing, preprocessing(level1): 0.086
[45:20] Acquisition timing, total(level1): 5.057 s for 68452352 correlator values
[45:20] Number of FFTs performed (CPU only): 0, time in [ms]: 0
[45:25] Sample stream ended
[45:25] Process command: 'Controller.stop'
[45:25] command successfully executed
[45:25] GMasterReceiver working thread ended in a regular way
[45:25] Controller thread tries gracefully stopping. Now stopping GMasterReceiver and NavProcessor
[45:25] Controller now sleeping for 5 s to allow other threads to end
[45:26] NavProcessor working thread ended in a regular way
[45:30] Controller thread gracefully stopped
[45:30] Backend stopped
[45:30] Backend stopped, API/Controller was already down, errors could have occurred.

```

Processor stopped Start

Upon completing the sample stream, the receiver will halt automatically and notify the user ("Sample stream ended") through the Receiver Log. By clicking the 'Stop' button, the receiver operations can deliberately be ceased at any time. When this is done, the receiver log will record the message 'Backend stopped' without referencing the termination of the sample stream.

Epoch Data History Window

MuSNAT User: hp

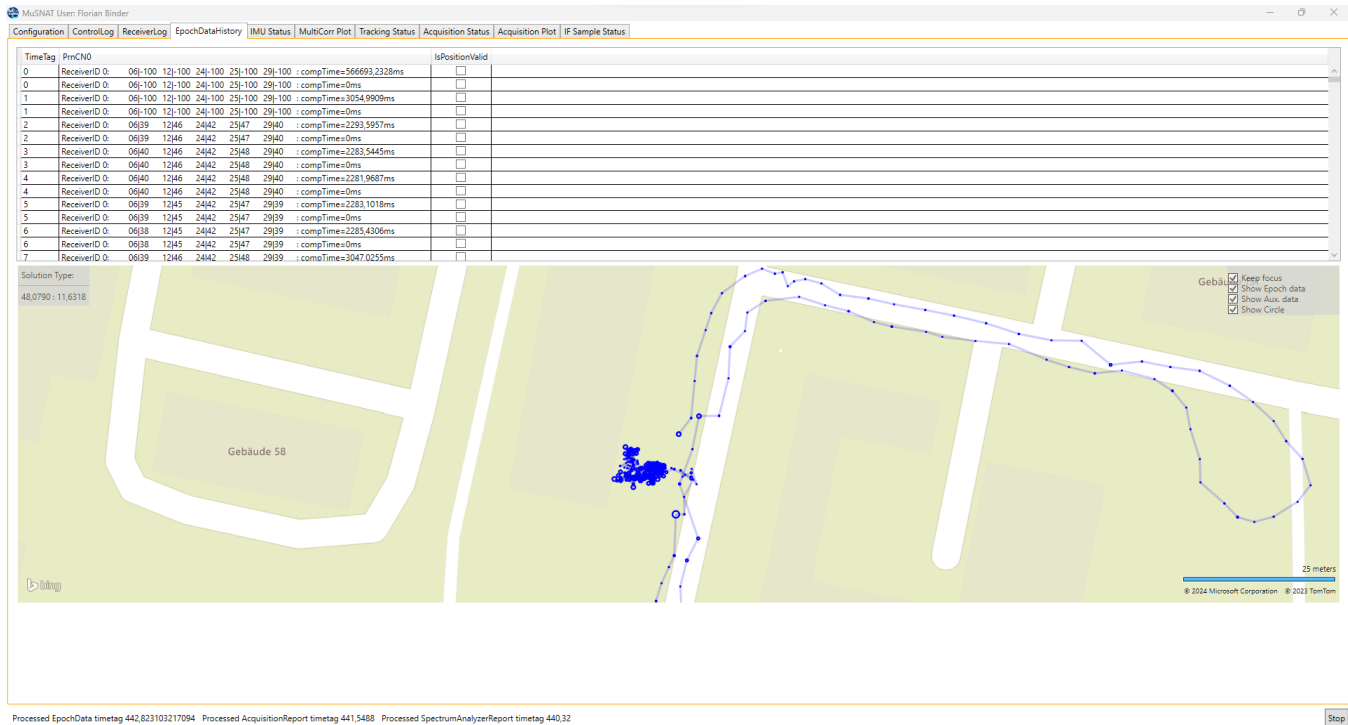
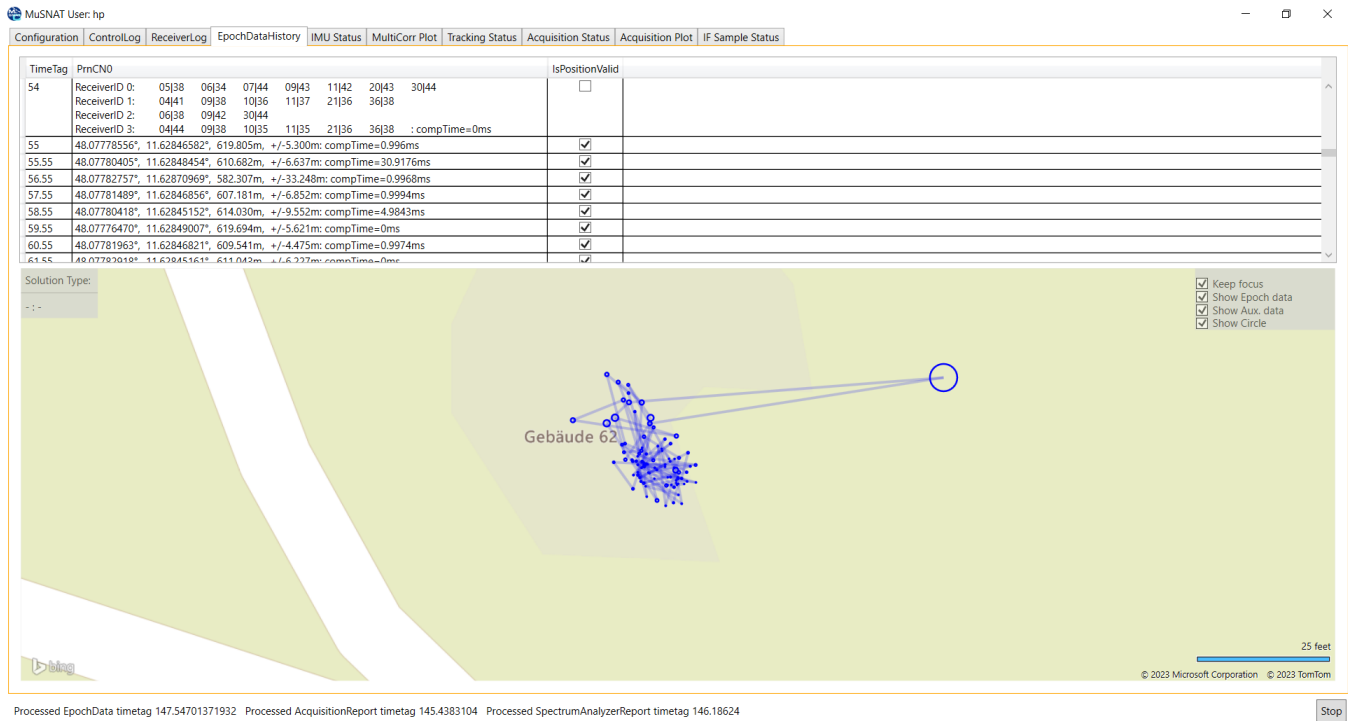
Configuration | ControlLog | ReceiverLog | EpochDataHistory | IMU Status | MultiCorr Plot | Tracking Status | Acquisition Status | Acquisition Plot | IF Sample Status

TimeTag	PrnCNO	IsPositionValid
0	ReceiverID 0: 09j-100 30j-100 ReceiverID 1: 10j-100 21j-100 ReceiverID 2: 30j-100 ReceiverID 3: 36j-100	<input type="checkbox"/>
1	ReceiverID 0: 09j-100 30j-100 ReceiverID 1: 10j-100 21j-100 ReceiverID 2: 30j-100 ReceiverID 3: 36j42 : compTime=784.9023ms	<input type="checkbox"/>
2	ReceiverID 0: 09j-100 09j-100 30j-100 ReceiverID 1: 09j-100 10j40 21j41 36j-100 ReceiverID 2: 09j-100 30j-100 ReceiverID 3: 36j42 : compTime=6001.2719ms	<input type="checkbox"/>
3	ReceiverID 0: 09j-100 09j-100 30j-100	<input type="checkbox"/>

Solution Type: -:-

Processed EpochData timetag 14 Processed AcquisitionReport timetag 14.7546112 Processed SpectrumAnalyzerReport timetag 16.9984 Stop

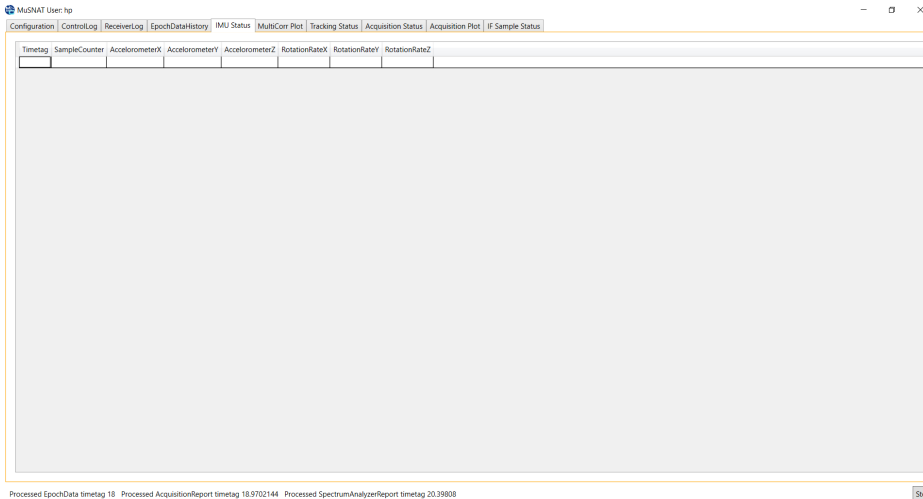
The receiver's performance in terms of the measured signal strength and position fixing over time shall be monitored and analyzed through this window. Usually, this phase serves as the point to verify the successful signal acquisition and tracking through position fixes. MuSNAT receive continuously recorded observations of satellite signals at predefined time intervals, typically epochs. The receiver saves this data to provide a historical record of signal strength, position fixes, and other relevant information. In the context of the initial receiver operation, the figure presented below is accompanied by timestamps in the **TimeTag** column. Notably, no position fixes were achieved during the initial few seconds, as confirmed by the absence of a checkmark in the **IsPositionValid** column."



In the scenario where satellites exhibit high visibility and strong signal strength, a valid position fix becomes available after approximately 30 seconds of processing IF samples. The **IsPositionValid** column is then marked, signifying a successful fix that encompasses **'Longitude,' 'Latitude,' 'Altitude,' 'PositionAccuracy,'** and **'ComputationTime.'** Furthermore, the receiver generates a **web map** that displays these achieved position fixes at every epoch.

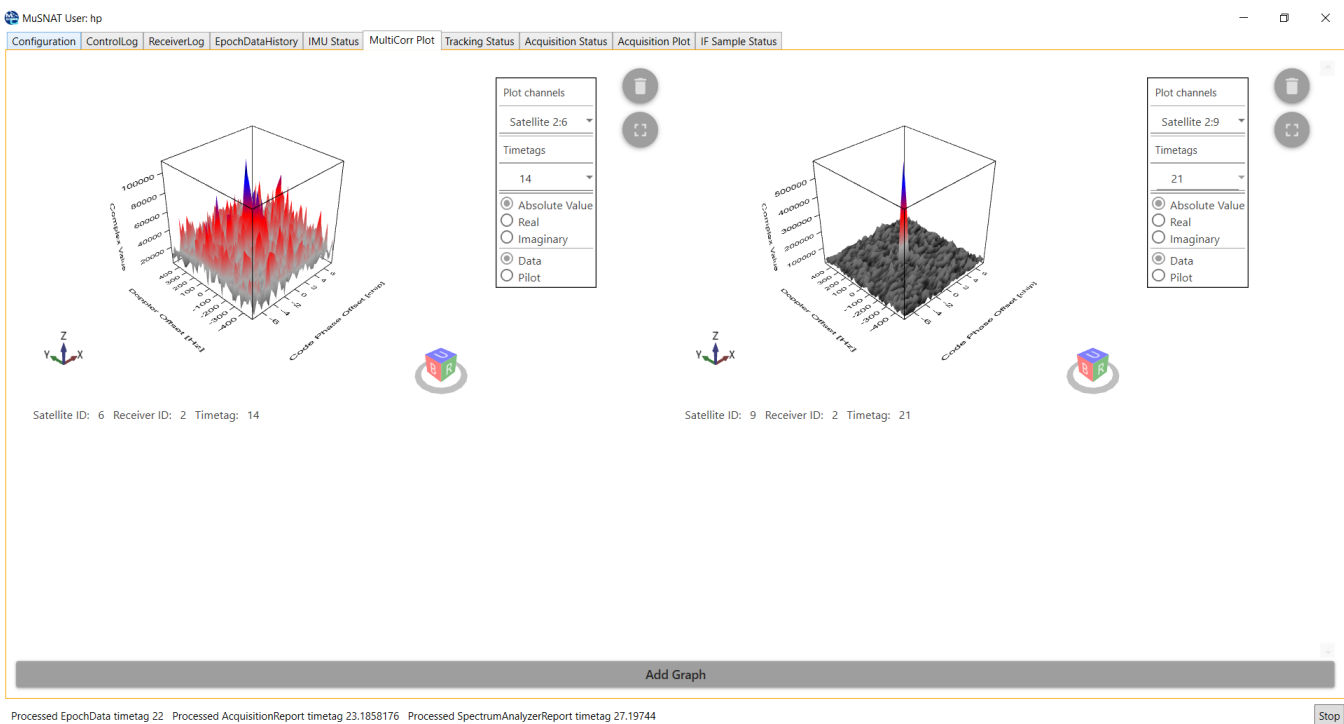
In the initial web map, a dataset containing fixed measurements was analyzed. The cluster of points visible represents the Single Point Positioning (SPP) achieved at each epoch, determined by the configured measurement rate in the *.musnat file. Conversely, in the subsequent web map, a trajectory path is depicted, derived from a dataset of Real-Time Kinematic (RTK) measurements.

IMU Status Window



Due to the absence of IMU data in the provided sample, essential components like accelerometer values and rotational rates along the X, Y, and Z axes have not been filled in. A more comprehensive depiction of this window should be presented within the [Enhanced Processing](#) section.

Multi-Correlator Plot Window



A multi-correlator plot shows the correlation results for multiple correlators operating simultaneously during the signal tracking process. Each correlator corresponds to a different replica code, and the plot illustrates how well each correlator aligns with the incoming signal over time.

The x-axis represents the code phase offset [chips] of the incoming signal. The y-axis represents the doppler offset [Hz] and the z axis typically represents the correlation amplitude or strength. Each correlator is associated with a specific replica code, which corresponds to a specific satellite signal. As the incoming signal is processed, the correlators compare the incoming signal with their respective replica codes. Peaks or spikes in the plot indicate instances where the correlator has successfully aligned with the incoming signal, indicating a potential correlation which could be seen in the right plot of the above snippet.

Within MuSNAT, users can explore the multi-correlator functionality by specifying a particular receiver ID and PRNs corresponding to a selected time tag. Additionally, users have the convenience of incorporating multiple plots simultaneously through the **'Add Graph'** feature, streamlining the analysis process.

The content of the multi-correlator plots is identical to the [.mclg files](#).

Tracking Status Window

MuSNAT User: hp

Configuration ControlLog ReceiverLog EpochDataHistory IMU Status MultiCorr Plot Tracking Status Acquisition Status Acquisition Plot IF Sample Status

SatId	ReceiverId	Service	AntennaId	Range	RangeRate	Phase	Power	CmcDoppler	SendTime	CycleSlip	PhaseLock	DataFree	CarrierPhaseSign	Frequency
20	0	GPS C/A	0	19981490.5618545	-637.7	-605286.04	44.47	89.65	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
30	0	GPS C/A	0	20448094.9626455	-740.78	-700252.48	43.75	138.01	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
7	0	GPS C/A	0	19873085.4562193	-351.68	-337110.47	43.6	-52.87	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
11	0	GPS C/A	0	21388378.6080847	17.5	7177.06	43.06	74.92	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
9	0	GPS C/A	0	20794500.4045826	107.86	96631.49	42.81	10.23	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
5	0	GPS C/A	0	22285170.3261004	-963.3	-904915.77	39.78	-164.25	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
6	0	GPS C/A	0	22561603.8876638	318.1		33.63	-44.45	1382785105.92	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1575420000
30	2	GPS L5	0	20448099.425576	-740.73	-522916.44	44.43	7.3	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
4	3	Galileo OS E5a	0	22452010.3372305	-365.03	-257706.09	44.14	7.16	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
4	1	Galileo OS E1	0	22452000.5856585	-365.08	-345099.1	42.05	670.67	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
9	2	GPS L5	0	20794503.0499345	108.14	72158.69	41.9	-6.83	1382785105.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
36	3	Galileo OS E5a	0	24845426.6281373	-395.98	-282492.45	39.44	330.13	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
9	1	Galileo OS E1	0	24830905.325861	-732.97	-687342.7	39.41	174.76	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
10	1	Galileo OS E1	0	25877504.7886709	252.16	233516.9	38.05	225.94	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
36	1	Galileo OS E1	0	24845422.9604423	-395.33	-378291.78	37.88	262.64	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
9	3	Galileo OS E5a	0	24830909.742598	-732.75	-513278.16	37.34	-263.26	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
6	2	GPS L5	0	22561615.7964786	318.45	220117.05	37.28	1.27	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
21	1	Galileo OS E1	0	25112134.1754028	-369.82	-339758.32	37.21	1029.69	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
21	3	Galileo OS E5a	0	25112140.4043572	-369.62	-264334.83	36.9	249.88	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
11	1	Galileo OS E1	0	23609277.8983515	90.21	78839.65	36.06	-1002.29	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000
11	3	Galileo OS E5a	0	23609279.6020738	90.3	58873.26	35.9	750.18	1382785105.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
10	3	Galileo OS E5a	0	25877510.0199434	251.97	174379.54	34.48	-1535.27	1382785105.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	1176450000
19	1	Galileo OS E1	0	25288382.485607	64.85		33.83	-743.83	1382785105.91	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1575420000

Processed EpochData timetag 233.546916755036 Processed AcquisitionReport timetag 233.9659776 Processed SpectrumAnalyzerReport timetag 231.17824

MuSNAT User: hp

Configuration ControlLog ReceiverLog EpochDataHistory IMU Status MultiCorr Plot Tracking Status Acquisition Status Acquisition Plot IF Sample Status

Time	CycleSlip	PhaseLock	DataFree	CarrierPhaseSign	Frequency	IsPosSatValid	SatPosX	SatPosY	SatPosZ	SatPosAcc	Elevation	Azimuth	TropDelay	IonDelay	IonModelDelay	RangeResiduum	RangeRateResiduum	LockTime	Api	Vector	Slaving
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	14607832.76	-7944965.18	20717551.74	3.6	56.52	292.61	2.69	0	0	0.3		218.55	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	24683471.41	927786.66	10229704.36	3.6	55.63	199.93	2.72	0	0	0.3		224.88	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	18296195.54	8252285.77	18102444.55	3.6	76	118.52	2.31	0	0	1.89		218.55	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	21368077.73	-121110087.11	10060717.82	3.6	37.39	246.68	3.69	0	0	-1.55		218.55	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	8756188.77	17706521.63	17642031.48	3.6	43.23	79.74	3.27	0	0	-0.02		224.88	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	3770998.11	-15769920.13	20838249.66	3.6	24.17	309.99	5.46	0	0	0.42		218.55	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	25898166.08	-5834959.34	-1809273.99	3.6	20.94	209.72	6.24	0	0	-1.34		67.76	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	24683471.41	927786.66	10229704.36	3.6	55.63	199.93	2.72	0	0	-0.99		235.88	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	16968296.68	1969628.27	24172417.09	3.6	80.58	337.43	2.28	0	0	-0.79		220.66	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	16968296.68	1969628.27	24172417.09	3.6	80.58	337.43	2.28	0	0	2.4		218.55	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.93	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	8756188.77	17706521.63	17642031.48	3.6	43.23	79.74	3.27	0	0	-3.2		224.28	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	9136543.71	-17463602.81	22090190.48	3.6	32.16	299.65	4.21	0	0	0.39		224.88	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	16118166.02	-18735940.84	16275698.1	3.6	32.41	275.3	4.18	0	0	-0.58		222.77	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	9136543.71	-17463602.81	22090190.48	3.6	32.16	299.65	4.21	0	0	1.94		222.77	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	28667219.77	-7007999.62	-2195949.1	3.6	21.63	210.75	6.05	0	0	-1.42		224.88	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	2387328.47	21532261.3	20170122.3	3.6	30.43	67.42	4.42	0	0	-3.5		162.69	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	16118166.02	-18735940.84	16275698.1	3.6	32.41	275.3	4.18	0	0	-0.22		219.61	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	25898166.08	-5834959.34	-1809273.99	3.6	20.94	209.72	6.24	0	0	4.19		228.5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	2387328.47	21532261.3	20170122.3	3.6	30.43	67.42	4.42	0	0	-0.39		221.71	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	25301638.34	-12952724.25	8261743.39	3.6	34.94	238.48	3.91	0	0	-1.17		217.5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.92	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	25301638.34	-12952724.25	8261743.39	3.6	34.94	238.48	3.91	0	0	-3.38		220.66	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1176450000	<input checked="" type="checkbox"/>	28667219.77	-7007999.62	-2195949.1	3.6	21.63	210.75	6.05	0	0	2.17		221.71	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
785111.91	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1575420000	<input checked="" type="checkbox"/>	6108689.88	23641470.94	16726595.1	3.6	30.9	81.59	4.36	0	0	4.53		7.95	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Processed EpochData timetag 239.546910005242 Processed AcquisitionReport timetag 238.1815808 Processed SpectrumAnalyzerReport timetag 237.9776

Stop

This user interface segment is designed to offer real-time insights into the tracking and acquisition process, allowing users to monitor their GNSS receiver's performance. The window provides a comprehensive overview of the receiver's operation, covering both signal processing and positioning aspects. It's important to note that no data is recorded or displayed in this window until the signal-tracking process is initiated. As tracking commences, the receiver provides the following information:

Parameter	Description	Unit
SatID	PRN number of the satellite	-
ReceiverID	Receiver number as designated in the receiver .musnat configuration file	-
Service	Service of the GNSS signal	-
AntennaID	RF Input ID	-
Range	Code pseudorange observation	meter
RangeRate	Doppler shift for this signal expressed as a range rate. Note, that the Doppler frequency in Hertz equals to $-f_{RF} / c * RangeRate$.	meter /second
Phase	Carrier phase observation	degree
CmcDoppler	Difference between Code Doppler and Carrier Doppler (is used to detect gross tracking errors which manifest as very large CmcDoppler values)	Hertz
SendTime	Send Time in the nominal satellite time scale as measured in the receiver (since 6th January 1980)	second
Cycleslip	Discontinuity in the phase measurements of the carrier signal of a satellite as received by the GNSS receiver	Y/N
PhaseLock	Ability for the receiver to lock on to the phase of the signal (PLL), in contrast to operation in frequency lock (FLL) mode only	Y/N
DataFree	Indicates that the signal contains a data free component (=pilot component)	Y/N
CarrierPhaseSign	Indicates for signal without a pilot component, if the carrier phase sign (180° ambiguity) was resolved by the message decoder	Y/N
Frequency	Nominal center frequency of the GNSS service (=f _{RF})	Hz
IsPosSatValid	Indication of the valid position of the GNSS Satelliter	Y/N
SatPosX	Satellite Position in X-coordinate (usually in WGS84)	meter
SatPosY	Satellite Position in Y-coordinate (usually in WGS84)	meter
SatPosZ	Satellite Position in Z-coordinate (usually in WGS84)	meter
SatPosAcc	Accuracy of the satellite position (3DRMS)	meter
Elevation	Angle between the horizon and a line connecting a GNSS receiver to a specific satellite	degree
Azimuth	Horizontal angle measured clockwise from the North direction (0 degrees) to the direction of a specific satellite	degree
TropoDelay	Amount of delay of GNSS signals as they pass through the Earth's troposphere as calculated from a model (MuSNAT implements a simplified Hopfield model)	meter
IonDelay	Amount of delay and bending of GNSS signals as they pass through the Earth's ionosphere (either estimated from dual-frequency measurement or from a model)	meter
IonModelDelay	as above but only the modelled value. MuSNAT currently implements only the GPS C/A Klobuchar model.	meter
RangeResiduum	Difference between the observed pseudorange and the computed or predicted pseudorange based on the receiver's estimated position	meter
RangeRateResiduum	Difference between the observed range rate and the predicted range rate based on the receiver's estimated position and velocity	meter /second
LockTime	Time since the last status change of the channel (usually time since signal acquisition)	second
Api	Not used	
Vector	Indicates that this channel is controlled via vector tracking	Y/N
Slaving	Indicates that this channel is slaved to a master channel	Y/N

Acquisition Status Window

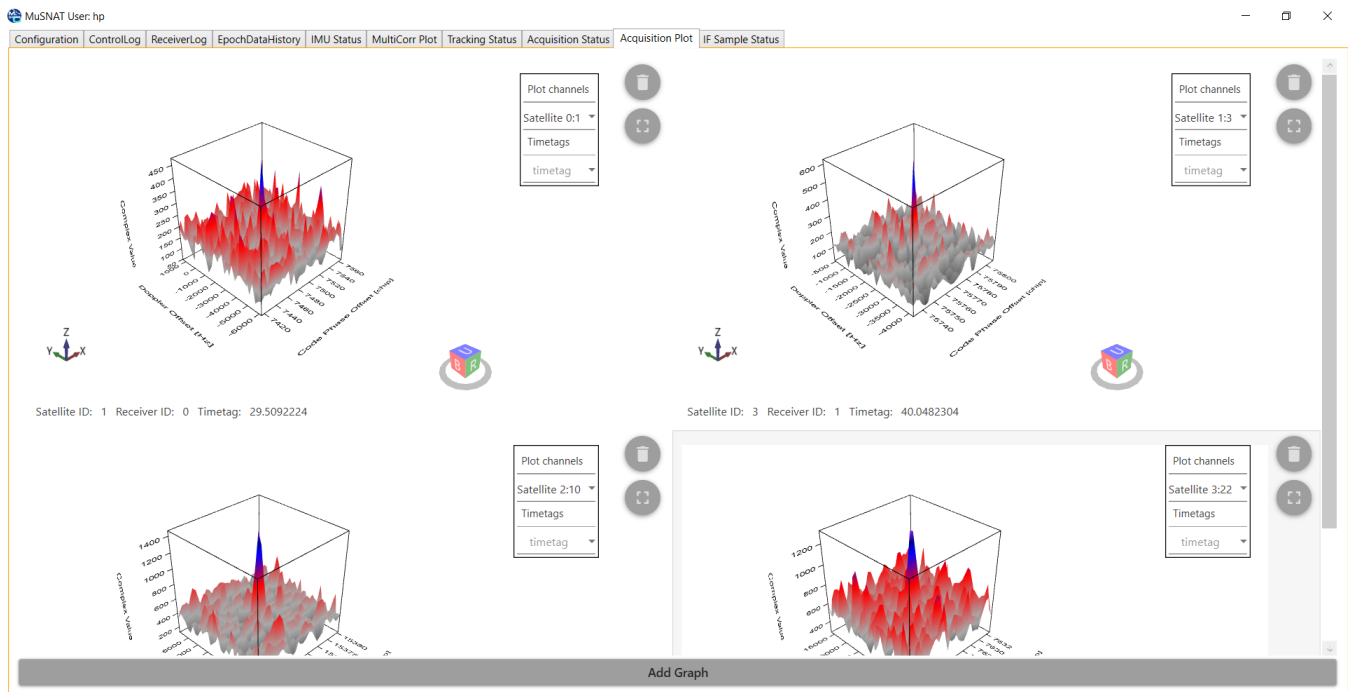
Satellite	System	ReceiverId	Service	AntennaId	RefTime	EstimatedPower	PowerThreshold	EstimatedCodePhase	EstimatedDoppler	AcqLevel	Elevation	ElevationStdDiv	Azimuth	AzimuthStdDiv
3	Gps	0	1	0	1382784920.33	27.24	38	1715.82	4804.11	1				
23	Gps	0	1	0	1382784920.33	27.09	38	7545.12	304.11	1				
18	Galileo	1	5	0	1382784920.33	26.91	33	40032.12	1804.11	1				
10	Gps	2	4	0	1382784920.33	30.21	34	38123.36	-4205.88	1				
29	Galileo	3	6	0	1382784920.33	31.53	34	10697.98	4294.12	1				
1	Galileo	1	5	0	1382784920.33	27.14	33	6884.52	-3820.89	1				
35	Galileo	1	5	0	1382784920.33	26.96	33	60384.72	2929.11	1				
26	Galileo	1	5	0	1382784920.33	27.09	33	15001.59	-2945.89	1				
3	Galileo	1	5	0	1382784920.33	27.48	33	9909.06	4304.11	1				
37	Galileo	3	6	0	1382784920.33	31.38	34	6347.86	5294.12	1				
5	Galileo	1	5	0	1382784920.33	27.26	33	68544.25	1179.11	1				
30	Galileo	3	6	0	1382784920.33	30.88	34	3069.19	-3705.88	1				
43	Galileo	3	6	0	1382784920.33	31.16	34	5242.06	4294.12	1				
17	Galileo	1	5	0	1382784920.33	26.77	33	13115.93	4054.11	1				
32	Gps	2	4	0	1382784920.33	30.52	34	28023.23	794.12	1				
13	Gps	0	1	0	1382784920.33	27.96	38	3164.41	3554.11	1				
24	Galileo	1	5	0	1382784920.33	26.59	33	61218.91	-2945.89	1				
26	Gps	2	4	0	1382784920.33	30.24	34	26346.12	3794.12	1				
5	Galileo	3	6	0	1382784920.33	31.05	34	10508.17	-705.88	1				
32	Galileo	3	6	0	1382784920.33	31.43	34	16590.35	-705.88	1				
27	Gps	2	4	0	1382784920.33	30.73	34	24485.44	2794.12	1				
32	Gps	0	1	0	1382784920.33	27.32	38	9043.66	4304.11	1				
45	Galileo	3	6	0	1382784920.33	31.15	34	12761.59	294.12	1				
16	Gps	0	1	0	1382784920.33	27.73	38	18924	-195.89	1				
24	Galileo	3	6	0	1382784920.33	31.48	34	15295.36	-705.88	1				
25	Galileo	3	6	0	1382784920.33	31.6	34	5565.5	-3705.88	1				
41	Galileo	3	6	0	1382784920.33	31.37	34	17523.18	-2705.88	1				
15	Gps	0	1	0	1382784920.33	27.5	38	10941.8	1054.11	1				
26	Gps	0	1	0	1382784920.33	27.26	38	127.38	-4195.89	1				
17	Galileo	3	6	0	1382784920.33	31.28	34	14285.72	294.12	1				
40	Galileo	3	6	0	1382784920.33	31.36	34	1148.56	5294.12	1				
29	Galileo	1	5	0	1382784920.33	27.06	33	24212.58	3679.11	1				
33	Galileo	1	5	0	1382784920.33	27.41	33	12511.52	-5070.89	1				
43	Galileo	1	5	0	1382784920.33	27.73	33	10363.62	-3695.89	1				
22	Gps	0	1	0	1382784920.33	27.83	38	16686.19	1804.11	1				

Processed EpochData timetag 63.5471084608401 Processed AcquisitionReport timetag 52.69504 Processed SpectrumAnalyzerReport timetag 61.19424

Stop

Parameter	Description	Unit
Satellite	PRN Number of the satellite	-
System	Corresponding constellation	-
Receiver ID	Receiver ID (No.) as configured in the .musnat file	-
Service	Identifier of the GNSS service of the tracked signal	-
Antenna ID	Antenna ID of the main antenna	-
RefTime	Reference time (epoch) of navigation record (since 6th January 1980)	s
Estimated Power	Calculated C/N0 of the received GNSS signals based on height of the correlation peak	dB-Hz
Power Threshold	C/N0 threshold to declare successful signal acquisition as set by the user (Receiver*.AcqCnoThreshold)	dB-Hz
EstimatedCodePhase	Code phase value corresponding to the peak of the acquisition function	subchip
EstimatedDoppler	Doppler value corresponding to the peak of the acquisition function	Hz
AcqLevel	Type of acquisition unit (1 ... cold start unit, 2 ... warm start unit)	1/2
Elevation	Angle between the horizon and a line connecting a GNSS receiver to a specific satellite	degree
ElevationStdDiv	not used	
Azimuth	Horizontal angle measured clockwise from the North direction (0 degrees) to the direction of a specific satellite	degree
AzimuthStdDiv	not used	

Acquisition Plot Window

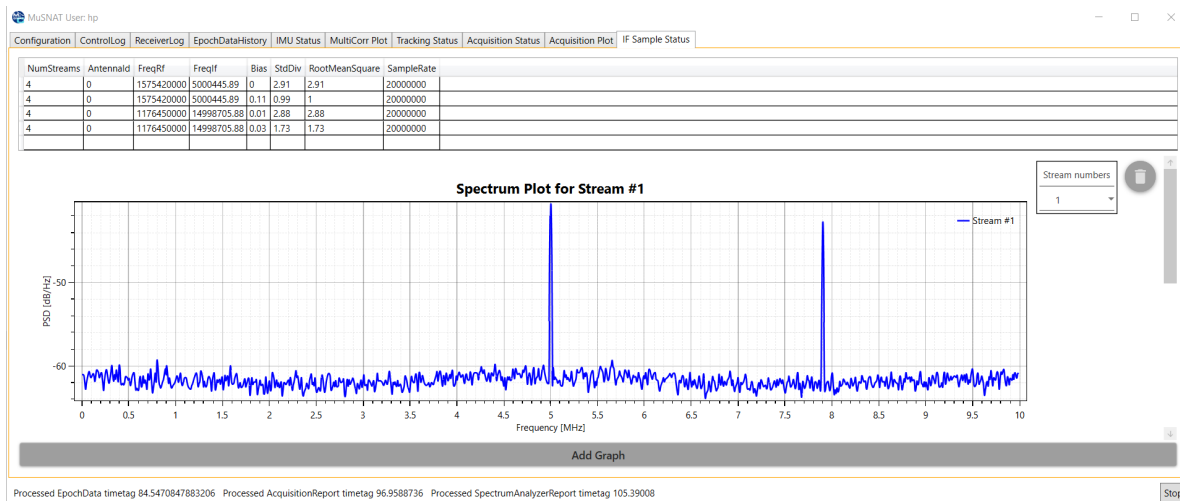


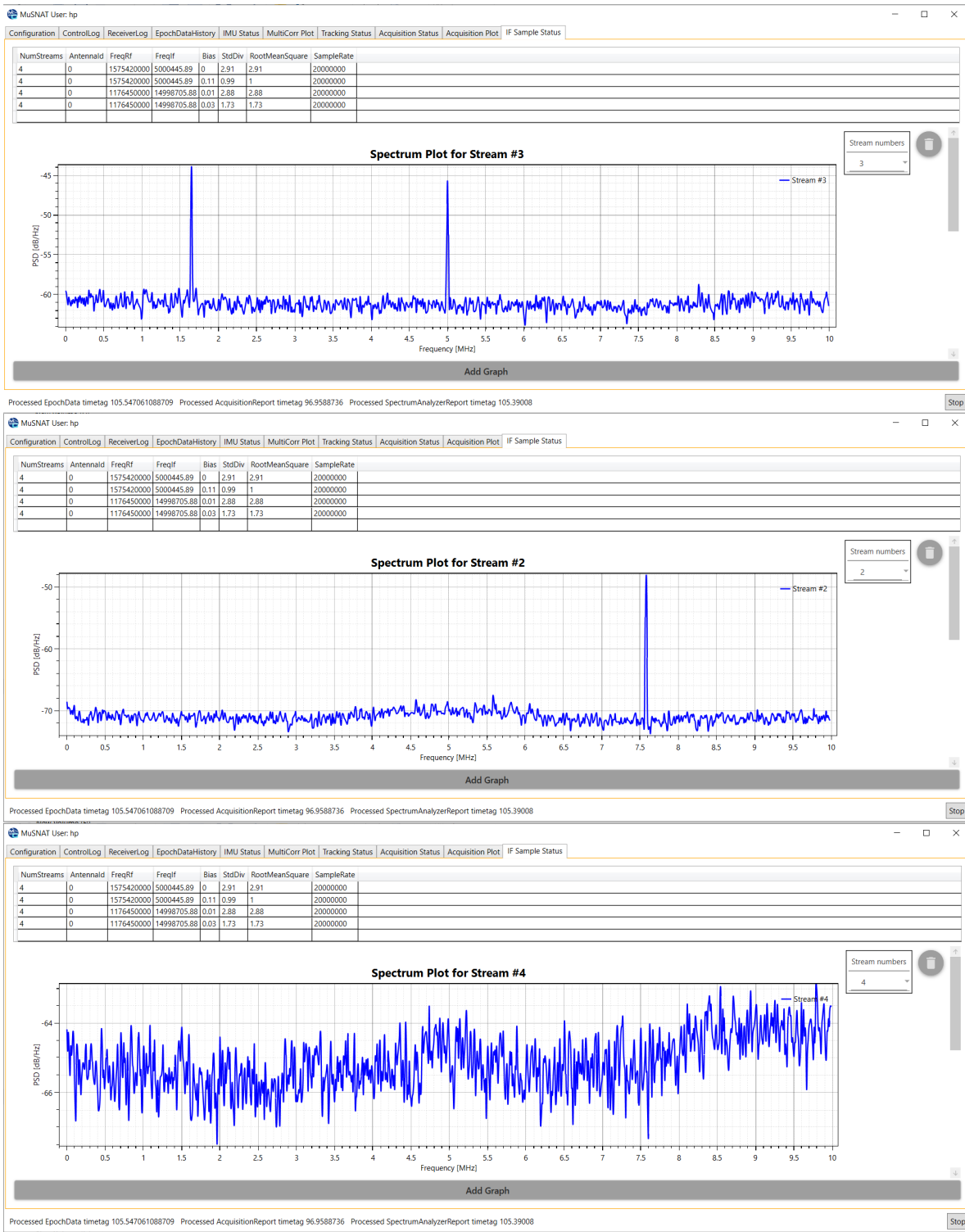
Produce acquisition plots by following the below mentioned steps:

Prerequisite: Enable Write Acquisition Data in the **Configuration** Window and configure generation of [acquisition logs](#).

1. Click on **Add Graph**
2. Select the **Satellite**
 - a. First index is the Receiver ID
 - b. Second index is the Satellite PRN
3. Select the desired **Timetag** at which the acquisition plot is of interest
4. Repeat Step 1-3 to see the acquisition plot for other PRNs

IF Sample Status Window





This window features information about the recorded Intermediate Frequency samples and also offers a visual representation of the Power Spectral Density of the selected stream.

Parameter	Description	Unit
NumStreams	Number of IF signal streams, typically equals to the number of frequency bands of the frontend	-
Antennald	Antenna ID of the main antenna.	-
FreqRF	Nominal (i.e. without any Doppler shift) center frequency of the signal	Hz

FreqIF	Nominal Frequency at which down-converted satellite signals are processed for navigation data extraction and tracking	Hz
Bias	Arithmetic mean value of the IF sample representation values Note: For typical sample representation values the bias "ideally" is 0.0 indicating that the ADC is well balanced	-
StdDiv	Estimated standard deviation of the inter-frequency bias of the selected stream	-
RootMeanSquare	Root mean square of the incoming IF sample representation values Note 1: The "ideal" RMS value for a data stream depends on the sample representation, for 2-bit representation it should be around 1.7-2.0	-
SampleRate	Sampling Rate of the recorded GNSS signal	Hz

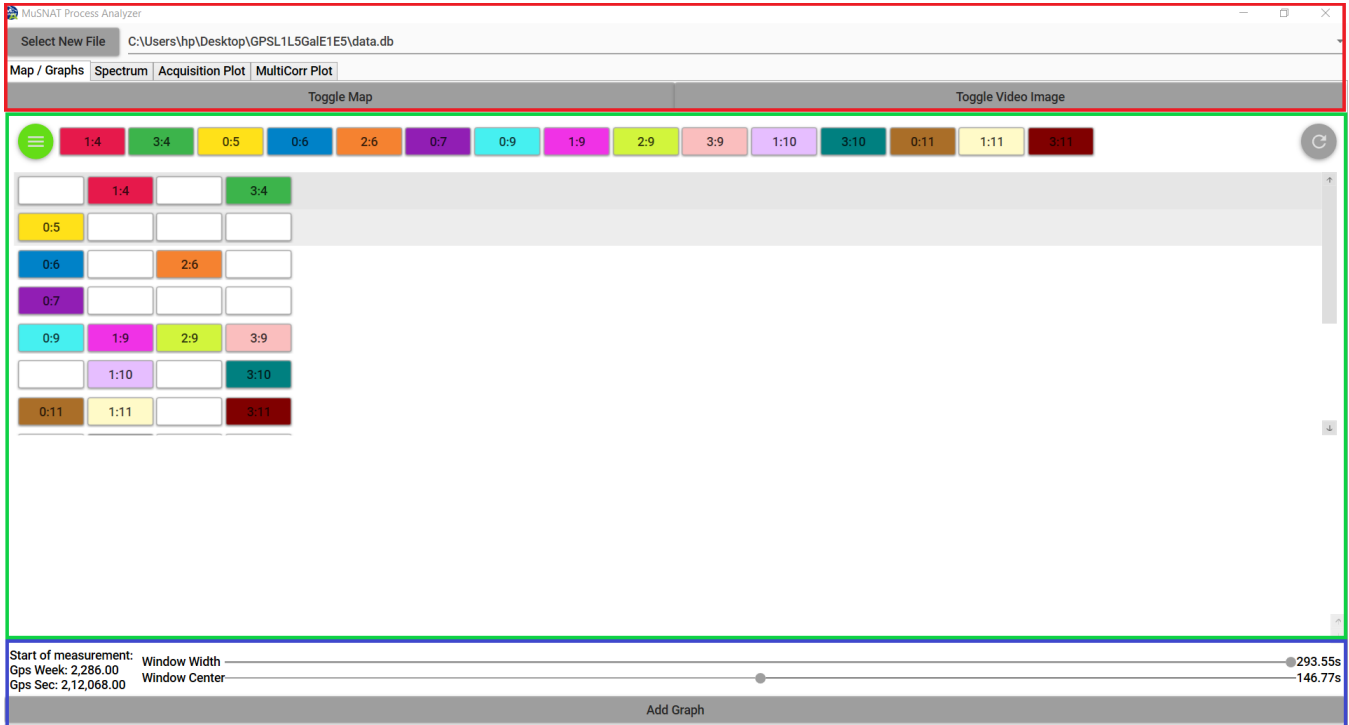
The distribution of power or energy across different frequencies within a signal is analyzed through the visualization of 'Power Spectral Density' plot This could be done by clicking on the **Add Graph** option and selecting the desired **stream number** (frequency band).

In the case of dealing with a dataset with more than one frequency band recorded at the front end, it is possible to view the power spectral density of them too by using the **Add Graph** option and by choosing the Stream Number.

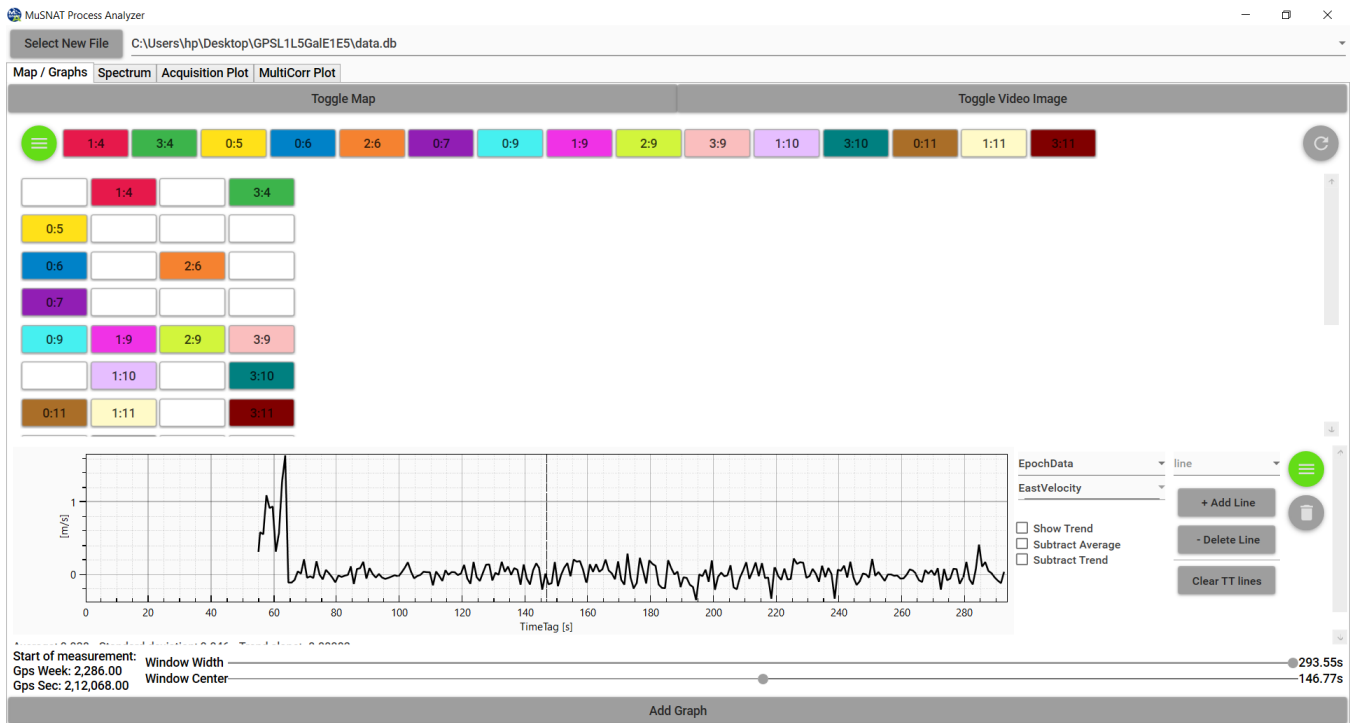
MuSNAT-Analyzer

The MuSNAT-Analyzer is a post processing tool, allowing the detailed presentation and analysis of the results of MuSNAT. It uses database file that is generated after processing the IF samples, containing all results otherwise found in the [Output Files](#).

Upon selecting the .db file (as seen in the **red frame**), users can access plots and data specifically about their chosen channels (**green frame**). As depicted in the figure, the channels selected comprise two indices. The first index designates the receiver ID, while the second indicates the satellite PRN. For instance, in the case of "0:5," 0 signifies the receiver ID and 5 stands for the satellite PRN. The toggling of maps and videos based on the obtained PVT solution is also possible.



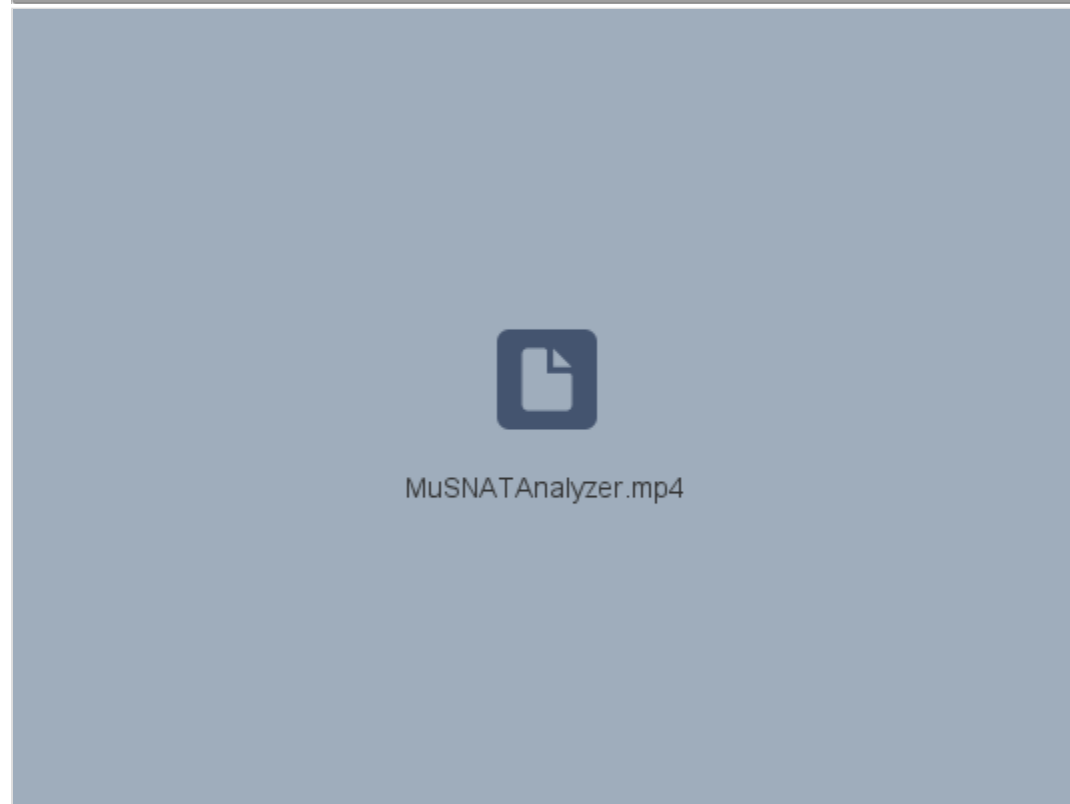
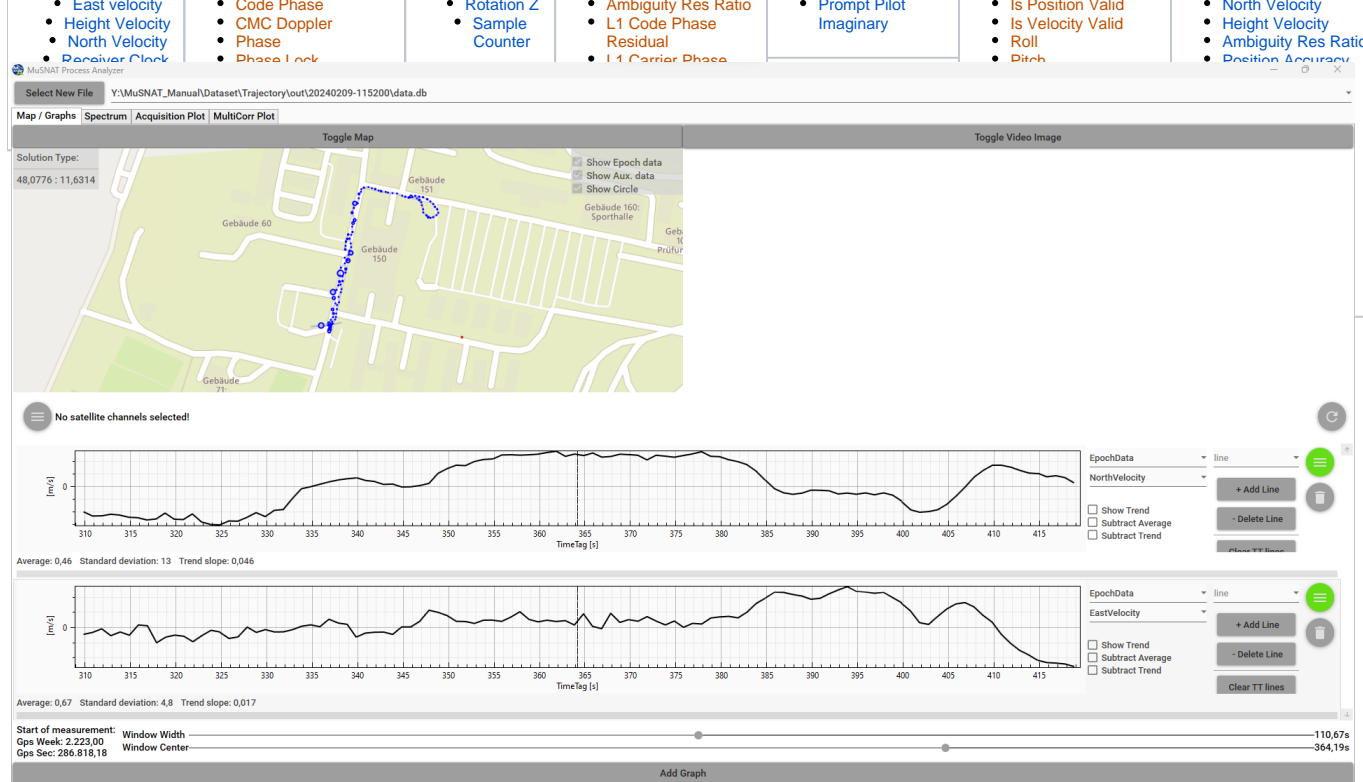
Further, the following graphical representations can be generated by selecting the "Add Graph" option. The table below gives an overview of available plotting options concerning the channel data, epoch data or other parameters.



The table below outlines various plots accessible through the MuSNAT Analyzer. The **table header** designates the **X-axis**, and the associated items listed below can be plotted on the **Y-axis**.

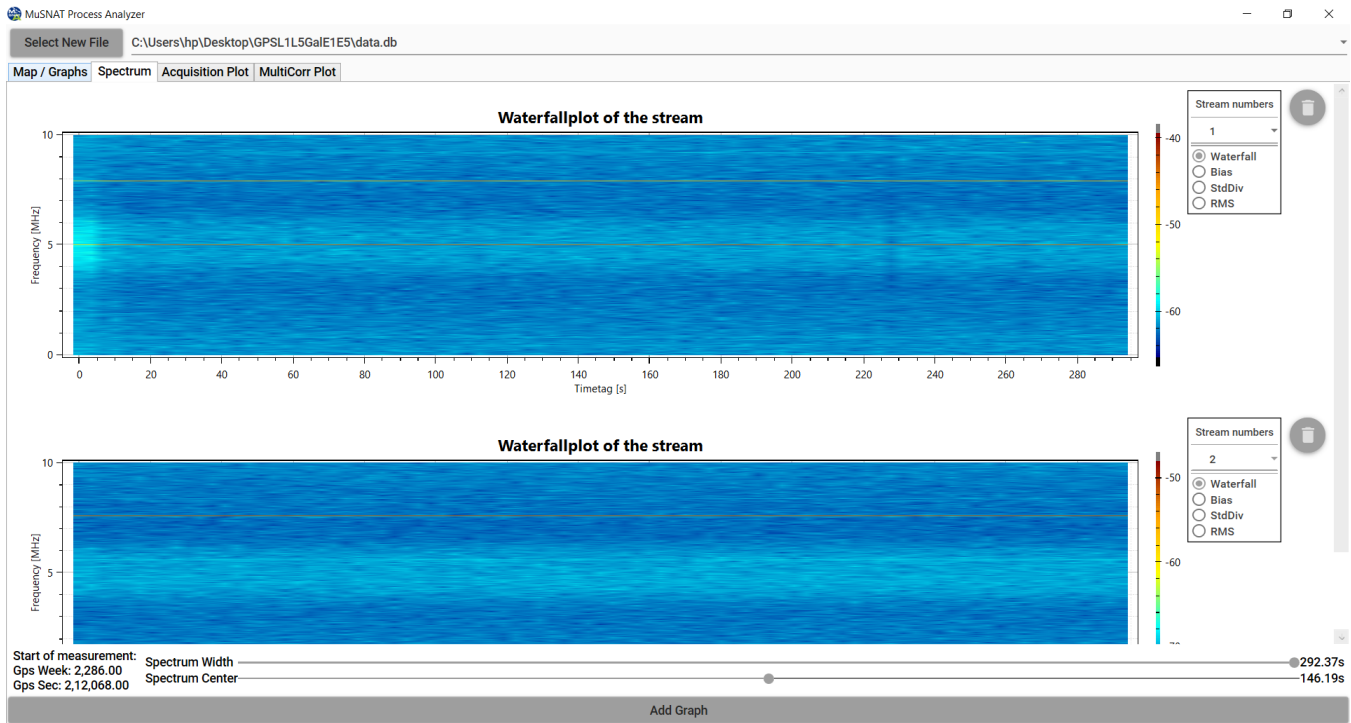
Epoch data	Channel Data	Sensor Record	Auxiliary Position	Discriminator Data	GNSS/INS	RTKLib
<ul style="list-style-type: none"> GPS sec GPS week 	<ul style="list-style-type: none"> Azimuth Elevation 	<ul style="list-style-type: none"> Acceleration X 	<ul style="list-style-type: none"> Latitude Longitude 	<ul style="list-style-type: none"> DLL Error PLL Error 	<ul style="list-style-type: none"> Latitude Longitude 	<ul style="list-style-type: none"> N Source Is Position Valid

The Graphs added are time aligned with each other and the map view and can be controlled with the sliders on the bottom. With the 'window width' option, the observed data window around the 'window center' can be chosen on the time-axis to be wider or more narrow. The 'center' determines the precise observation time on the x-axis. This allows for example the investigation of tracking issues due to signal blocking.

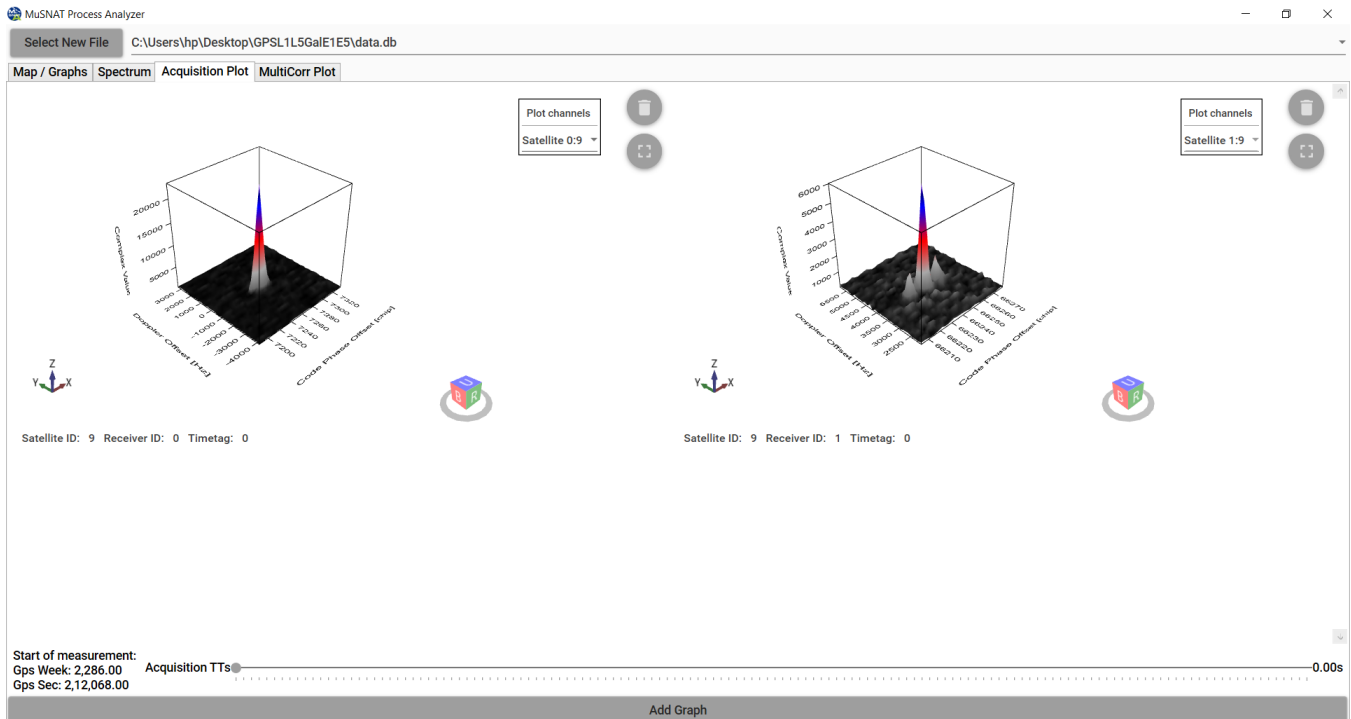


As visible in the red-framed menu bar, users can perform further comprehensive analyses by generating graphical data for 'Spectrum data,' 'Acquisition Plot,' and 'Multi Correlator Plot.'

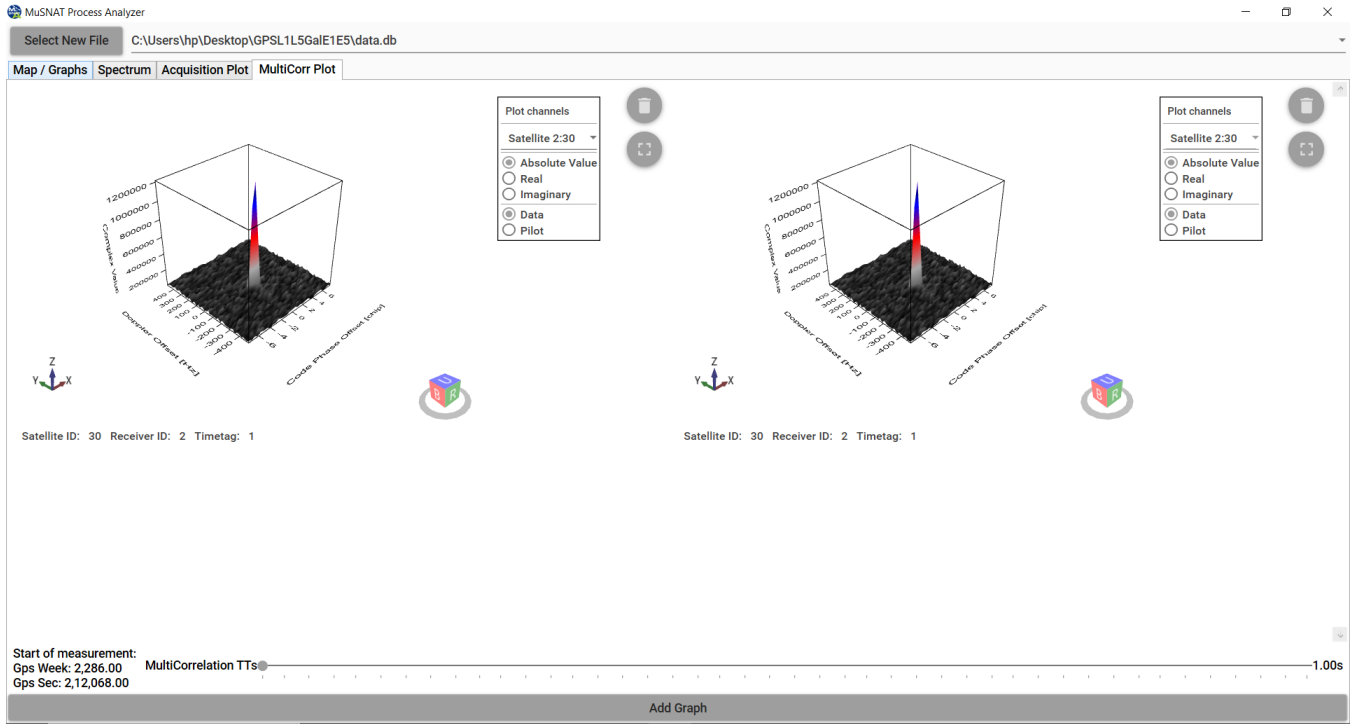
A spectrum plot is a visual representation illustrating the variations in frequency components of received signals over time. This tool proves essential for comprehending the dynamic characteristics of GPS signals and detecting distinctive patterns or irregularities within the signal spectrum indicating jamming and other interference sources. Users can specify the stream number through the user interface, and various options are available for analyzing the waterfall plot of the spectrum, including verifying bias, standard deviation, and root mean square (RMS).



By selecting the **'Write Acquisition Data'** option within the Settings and Configuration window of the MuSNAT-Core GUI, users can record acquisition data into the database file. This functionality serves as a prerequisite for generating these corresponding plots in the MuSNAT Analyzer. Based on the dataset, different receiver IDs and PRNs of interest can be chosen for further analysis.



To create multi-correlator plots for various receiver IDs and PRNs at a specified time tag, users can enable the 'Write MultiCorr Data' option within the Settings and Configuration window of the MuSNAT Core GUI. This feature allows the recording of multi-correlator data in the database, facilitating subsequent generation of relevant plots as been below.



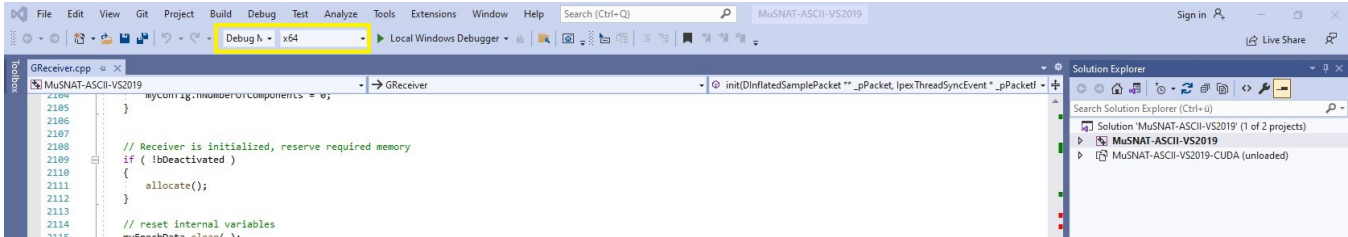
MuSNAT-ASCII

How to run/debug MuSNAT-ASCII on Microsoft Visual Studio VS-2019?

The MuSNAT-ASCII version employs the identical signal processing algorithm as the MuSNAT-Core GUI. The primary benefit of utilizing the GUI for processing IF samples is its ability to present data visually, offering an efficient means of interpretation. Conversely, the graphical representation is absent when employing the ASCII version, necessitating interaction via the command window / Microsoft Visual Studio Debug Console. Nevertheless, utilizing the ASCII version for debugging purposes and when working with other functionalities incompatible with the GUI is strongly advised.

After successfully loading the MuSNAT ASCII VS-2019(follow the steps outlined in the Installation guide), the IF sample files can be accessed by the ASCII version of the receiver through the steps mentioned below.

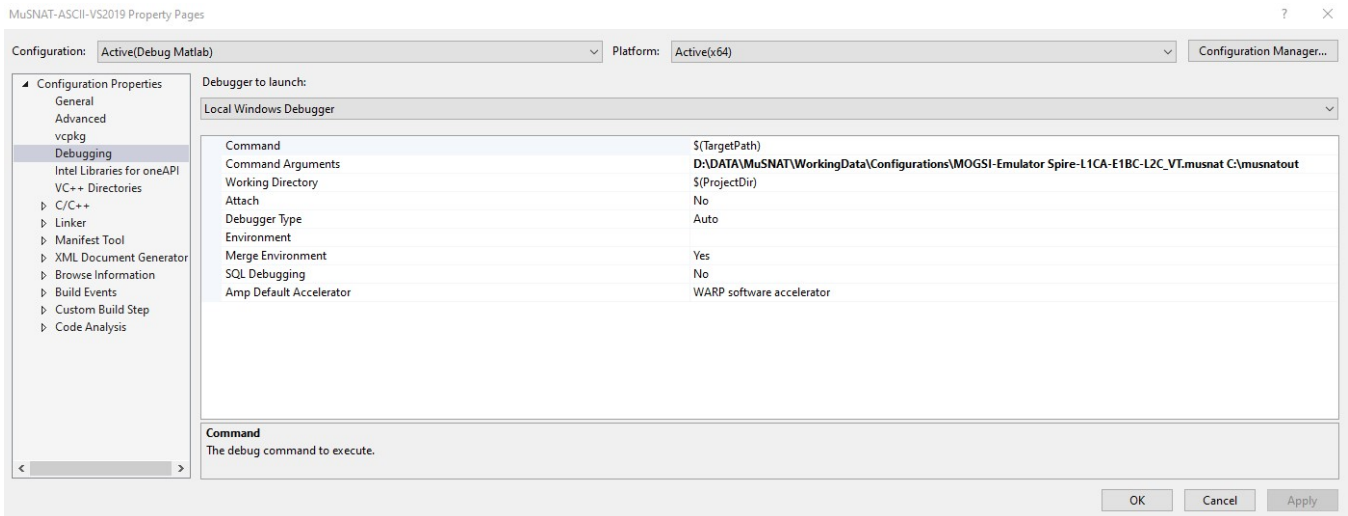
1. Right Click on the MuSNAT ASCII VS2019 displayed in the Solution Explorer and Select Properties



1. Click on the Debugging option, Go to Command Arguments

- Specify the configuration file (with input IF samples) and the output directory
- Three different arguments have to be set:
 1. Path to the configuration file
 2. Name of the configuration file
 3. Path to the output folder

Ensure the receiver runs under the **Active (Debug Matlab)** configuration and on the **Active (x64)** platform. Click on Apply.



1. Initiate the receiver's operation by selecting either 'Start Debugging (F5)' or 'Start without Debugging (Ctrl+F5)' from the Debug tab. Please be aware that the receiver may require a few seconds to initialize and read the configuration file.

Once the receiver has successfully read the configuration file, it will present the following messages within the command prompt: Highlighted in yellow, these messages will indicate the initialization directory, configuration data file, and output directory. These details are determined by the command arguments specified in the Debugging section of the properties.

Upon activation, the receiver will present a comprehensive report detailing the status of the IF samples, signal frequency, and other relevant information about the signal.

```

Microsoft Visual Studio Debug Console
MUSNAT_ASCII
(c) 2005-2021 Universitaet der Bundeswehr Muenchen
-----
Setting initialization directory ...
Initialization directory is:D:\DATA\MuSNAT\WorkingData\Configurations\WOGSI-Emulator
Loading configuration data from "Spire-L1CA-E1BC-L2C_VT_musnat" ...
Error reading navigation data from "WorkNavData.nav"
Successfully read SP3c file: COD22002.EPH_M
Initialization complete.
Output directory is:C:\musnatout
Receiver started
Press 'c' for commanding and 'x' to stop ...
Sample Info: stream 0, FMT-IF, IF=1.588750 MHz, fs=6.200000 MHz
Sample Info: stream 1, FMT-IF, IF=1.646875 MHz, fs=6.200000 MHz
Initializing master receiver
CPU Type: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
MUSNAT core build date/time: Sep 20 2023, 17:18:40
NO CUDA support
NO QT support
IF sample packet length = 0.109006452 s
Rec. 0 (GPS_C/A_Code_Receiver): stream=0, RF=1575.420000 MHz, IF=1.588750 MHz, fs=6.200000 MHz.
Code rate =20.460000 Msbchip/s, Subchip length = 14.653 m, Vector Tracking = Yes, PolyFit Code/Carrier/Doppler/CN0: 1110
Freq. Offset = 0.000 Hz
Rec. 1 (GPS_L2CM): stream=1, RF=1227.600000 MHz, IF=1.646875 MHz, fs=6.200000 MHz.
Code rate =20.460000 Msbchip/s, Subchip length = 14.653 m, Vector Tracking = Yes, PolyFit Code/Carrier/Doppler/CN0: 1110
Freq. Offset = 0.000 Hz
Rec. 2 (Galileo_E1_05_B1C_INAV_V12): stream=0, RF=1575.420000 MHz, IF=1.588750 MHz, fs=6.200000 MHz.
Code rate =20.460000 Msbchip/s, Subchip length = 14.653 m, Vector Tracking = Yes, PolyFit Code/Carrier/Doppler/CN0: 1110
Freq. Offset = 0.000 Hz
Receiver 3 (Default_SpectrumAnalyzer_Config) works on multiple streams.
Number of threads for all receivers+1(38) including OpenMP workers is exceeding number of cores(8)! This may lead to excessive overhead!
Acquisition type 4 running for Receiver0 on stream 0 and RF=1575.420000.0000.
Acquisition type 4 running for Receiver2 on stream 0 and RF=1575.420000.0000.
Acquisition engine 'SecCodeNonCohFftAcq running on CPU' successfully initialized.
Acquisition engine 'SecCodeNonCohFftAcq running on CPU' successfully initialized.
SPP weight scheme: Unity variance
Backend started
Acquisition timing, preprocessing(level1): 0.577
Acquisition timing, total(level1): 170.024 s for 2400124928 correlator values
Number of FFTs performed (CPU only): 0, time in [ms]: 0
Bump jump on rec. 2, PRN10 of 68.22 subchip
Bump jump on rec. 2, PRN2 of 81.09 subchip
Bump jump on rec. 2, PRN33 of 75.18 subchip
Bump jump on rec. 2, PRN8 of 75.41 subchip
Bump jump on rec. 2, PRN45 of 73.42 subchip
DeepCoupling active in Matlab no. sats: 0

Header contains GPS week and seconds, footer receiver id with max. CPU load factor
ACQ: signals tested, above acquisition threshold
TRK: no. of signals tracked, with pseudorange, phase locked, with cycle slip, below 40 dBHz, used in SPP
PVT: latitude, longitude, height, 3D position accuracy (DRMS), 3D velocity accuracy (DRMS)

-----
--2206--202830.889-----
| ACQ: 82, 33 | TRK: 13 ch., 0 sync., 0 PLL, 0 CycS, 13 <40dBHz, 0 in SPP |
-----
Bump jump on rec. 2, PRN33 of 4.85 subchip
DeepCoupling active in Matlab no. sats: 0
--2206--202831.889-----
| no ACQ | TRK: 13 ch., 0 sync., 0 PLL, 0 CycS, 13 <40dBHz, 0 in SPP |
-----

```

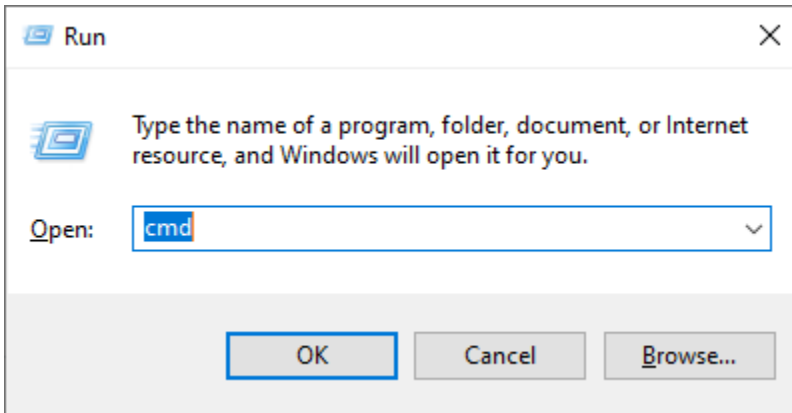
Following activation, the receiver initiates the process of recording all data related to signal acquisition, signal strength, tracking, and other pertinent receiver operations. The measurement rate specified in the configuration file determines the logging interval. Throughout the logging process, the system displays the recorded time at each step in the format of GPS time (highlighted in Green).

Once the sample stream ends, the receiver stops on its own with a statement at the end of the log file mentioning 'sample stream has ended'. Click X to stop the Receiver at any instant during the operation.

How to run MuSNAT-ASCII from command prompt?

The MuSNAT-ASCII can also be controlled using the command prompt following the steps below.

- Open Command Prompt (**Win + R**) and search **cmd**



- Navigate to the directory where MuSNAT-ASCII is installed by entering the right commands

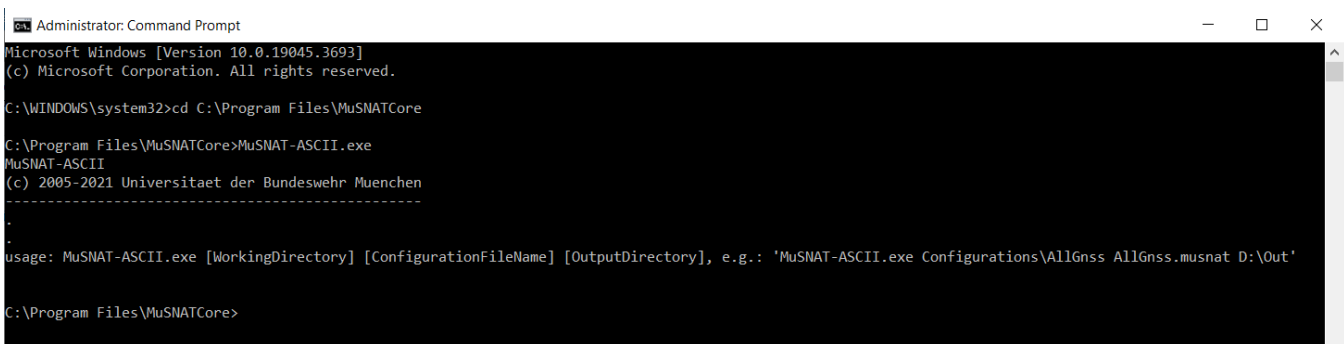
Example: `C:\`

`cd C:\Program Files\MuSNATCore`

- Once you've navigated to the MuSNATCore directory, initiate MuSNAT-ASCII.exe using the following command:

Example: `C:\Program Files\MuSNATCore>MuSNAT-ASCII.exe`

- Executing this command will launch MuSNAT-ASCII, and you will receive instructions on using the application. Refer to the example command provided in the screenshot below for guidance.
- Please note that contrary to this example MuSNAT should be installed in C:\Program Files, as this is a Windows controlled directory. See also [here](#)



- By employing the correct commands, MuSNAT-ASCII initializes and searches for the specified working directory. It also seeks the configuration file pointing to the required IF sample files.

```

Select Administrator: Command Prompt - MuSNAT-ASCII.exe Configurations\SimpleGpsL1L5 SimpleGpsL1L5GalileoE1E5.musnat C:\Out
C:\Program Files\MuSNATCore\MuSNAT-ASCII.exe Configurations\SimpleGpsL1L5 SimpleGpsL1L5GalileoE1E5.musnat C:\Out
MuSNAT-ASCII
(c) 2005-2021 Universitaet der Bundeswehr Muenchen
-----
Setting initialization directory ...
Initialization directory is: C:\Program Files\MuSNATCore\Configurations\SimpleGpsL1L5
Reading configuration data from 'SimpleGpsL1L5GalileoE1E5.musnat'...
Error reading navigation data from 'SimpleGpsGalL1E1.nav'
:Sample Info: stream 0, FMT=IF, IF=5.000446 MHz, fs=20.000000 MHz
:Sample Info: stream 1, FMT=IF, IF=5.000446 MHz, fs=20.000000 MHz
:Sample Info: stream 2, FMT=IF, IF=14.998706 MHz, fs=20.000000 MHz
:Sample Info: stream 3, FMT=IF, IF=14.998706 MHz, fs=20.000000 MHz
:SampleFileInput::start -> use local found sensor file 'SwRxData.laf' (do not use sensor file defined within stream file configuration)
:Initializing master receiver
: CPU Type: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
: MuSNAT core build date/time: Jul 24 2023, 22:05:12
: NO CUDA support
: NO QT support
:IF sample packet length = 0.033996800 s
:Rec. 0 (GPS_C/A_Code_Receiver): stream=0, RF=1575.420000 MHz, IF=5.000446 MHz, fs=20.000000 MHz.
Initialization complete.
Output directory is: C:\Out
Receiver started ...
Press 'c' for commanding and 'x' to stop ...
: Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CN0: 0000
: Freq. Offset = 0.000 Hz
:Rec. 1 (Galileo_E1_OS_B+C_INAV V12): stream=0, RF=1575.420000 MHz, IF=5.000446 MHz, fs=20.000000 MHz.
: Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CN0: 0000
: Freq. Offset = 0.000 Hz
:Rec. 2 (GPS_L5_I+Q): stream=2, RF=1176.450000 MHz, IF=14.998706 MHz, fs=20.000000 MHz.
: Code rate =40.920000 Msubchip/s, Subchip length = 7.326 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CN0: 0000
: Freq. Offset = 0.000 Hz
:Rec. 3 (Galileo_E5a_OS_I+Q): stream=2, RF=1176.450000 MHz, IF=14.998706 MHz, fs=20.000000 MHz.
: Code rate =20.460000 Msubchip/s, Subchip length = 14.653 m, Vector Tracking = No, PolyFit Code/Carrier/Doppler/CN0: 0000
: Freq. Offset = 0.000 Hz
:Receiver 4 (Default_SpectrumAnalyzer_Config) works on multiple streams.
: Number of threads for all receivers+1(9) including OpenMP workers is exceeding number of cores(4)! This may lead to excessive overhead!
: Acquisition type 4 running for Receiver0 on stream 0 and RF=1575420000.0000.
: Acquisition type 4 running for Receiver1 on stream 0 and RF=1575420000.0000.
: Acquisition type 4 running for Receiver2 on stream 2 and RF=1176450000.0000.
: Acquisition type 4 running for Receiver3 on stream 2 and RF=1176450000.0000.
: Acquisition engine 'SecCodeNonCohFFtAcq running on CPU' successfully initialized.
: Acquisition engine 'SecCodeNonCohFFtAcq running on CPU' successfully initialized.
: Acquisition engine 'SecCodeNonCohFFtAcq running on CPU' successfully initialized.
: Acquisition engine 'SecCodeNonCohFFtAcq running on CPU' successfully initialized.
: SPP weight scheme: Unity variance
: Backend started

```

- In the initialization phase, just before the receiver begins operation, it provides details about the frequency specifications configured within the receiver configuration file (*.musnat) as seen in the screenshot above.

```

:Backend started
:Acquisition timing, preprocessing(level1): 0.052
:Acquisition timing, total(level1): 5.062 s for 81526784 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0

Header contains GPS week and seconds, footer receiver id with max. CPU load factor
ACQ: signals tested, above acquisition threshold
TRK: no. of signals tracked, with pseudorange, phase locked, with cycle slip, below 40 dBHz, used in SPP
PVT: latitude, longitude, height, 3D position accuracy (DRMS), 3D velocity accuracy (DRMS)

+---2286---212068.000-----+
| ACQ: 62, 7 | TRK: 7 ch., 0 sync., 0 PLL, 0 CycS, 7 <40dBHz, 0 in SPP |
+-----+
| no ACQ | TRK: 7 ch., 0 sync., 0 PLL, 1 CycS, 6 <40dBHz, 0 in SPP |
+-----+
:Acquisition timing, preprocessing(level1): 0.040
:Acquisition timing, total(level1): 50.166 s for 486113280 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+---2286---212070.000-----+
| ACQ: 20, 6 | TRK: 13 ch., 0 sync., 0 PLL, 3 CycS, 10 <40dBHz, 0 in SPP |
+-----+
| no ACQ | TRK: 14 ch., 0 sync., 0 PLL, 6 CycS, 8 <40dBHz, 0 in SPP |
+-----+
:Receiver2: switching PRN30 from FLL to PLL.
:Acquisition timing, preprocessing(level1): 0.052

```

- After initialization, the receiver continuously acquires and tracks signals. Successfully tracked satellites contribute to the Position, Velocity, and Time (PVT) computation, displayed as "N" in SPP (N=Number of satellites).

```

| no ACQ          | TRK: 20 ch., 13 sync., 20 PLL, 0 CycS, 11 <40dBHz, 0 in SPP |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:Acquisition timing, preprocessing(level1): 0.074
:Acquisition timing, total(level1): 2.119 s for 27197440 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212121.000-----+
| ACQ: 25, 0      | TRK: 20 ch., 13 sync., 20 PLL, 0 CycS, 11 <40dBHz, 0 in SPP |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+--2286---212122.000-----+
| no ACQ          | TRK: 20 ch., 13 sync., 20 PLL, 0 CycS, 11 <40dBHz, 0 in SPP |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:Acquisition timing, preprocessing(level1): 0.079
:Acquisition timing, total(level1): 5.085 s for 64028672 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212123.000-----+
| ACQ: 49, 0      | TRK: 20 ch., 15 sync., 20 PLL, 0 CycS, 11 <40dBHz, 14 in SPP |
|                 | PVT: 48.077798, 11.628456, 616.099, 5.32, 0.104 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+--2286---212124.547-----+
| no ACQ          | TRK: 20 ch., 19 sync., 20 PLL, 1 CycS, 11 <40dBHz, 16 in SPP |
|                 | PVT: 48.077848, 11.628720, 583.984, 33.60, 0.472 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:Acquisition timing, preprocessing(level1): 0.104
:Acquisition timing, total(level1): 5.055 s for 58359808 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
:Receiver2: resolving code ambiguity for PRN30
+--2286---212125.547-----+
| ACQ: 42, 0      | TRK: 19 ch., 19 sync., 19 PLL, 0 CycS, 10 <40dBHz, 15 in SPP |
|                 | PVT: 48.077816, 11.628465, 607.276, 6.71, 0.425 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+--2286---212126.547-----+
| no ACQ          | TRK: 20 ch., 19 sync., 19 PLL, 0 CycS, 11 <40dBHz, 15 in SPP |
|                 | PVT: 48.077802, 11.628444, 611.833, 9.61, 0.290 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:Acquisition timing, preprocessing(level1): 0.098
:Acquisition timing, total(level1): 5.065 s for 77889536 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212127.547-----+
| ACQ: 53, 0      | TRK: 20 ch., 19 sync., 19 PLL, 0 CycS, 11 <40dBHz, 15 in SPP |
|                 | PVT: 48.077769, 11.628482, 616.393, 6.12, 0.090 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+--2286---212128.547-----+
| no ACQ          | TRK: 20 ch., 19 sync., 19 PLL, 0 CycS, 11 <40dBHz, 15 in SPP |
|                 | PVT: 48.077824, 11.628462, 608.580, 4.12, 0.403 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:Acquisition timing, preprocessing(level1): 0.071
:Acquisition timing, total(level1): 2.418 s for 26607616 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0

```

Quick troubleshooting tip: While tracking, the receiver may enter sleep mode if you hit a key, as it then waits for further inputs. If this happens, simply pressing the **space bar** can wake the receiver up.

```

+--2286---212331.547-----+
| ACQ: 56, 0 | TRK: 21 ch., 21 sync., 20 PLL, 1 CycS, 11 <40dBHz, 21 in SPP |
| | PVT: 48.077783, 11.628464, 610.608, 3.82, 0.097 |
+-----3>295.1%+
+--2286---212332.547-----+
| no ACQ | TRK: 21 ch., 21 sync., 20 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077774, 11.628502, 614.391, 2.98, 0.085 |
+-----3>299.1%+
:Acquisition timing, preprocessing(level1): 0.076
:Acquisition timing, total(level1): 5.071 s for 73039872 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212333.547-----+
| ACQ: 54, 0 | TRK: 21 ch., 21 sync., 20 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077798, 11.628491, 608.431, 5.88, 0.394 |
+-----3>297.5%+
+--2286---212334.547-----+
| no ACQ | TRK: 21 ch., 21 sync., 20 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077799, 11.628471, 613.581, 4.62, 0.422 |
+-----3>302.4%+
:Acquisition timing, preprocessing(level1): 0.072
:Acquisition timing, total(level1): 0.380 s for 6520832 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212335.547-----+
| ACQ: 3, 0 | TRK: 21 ch., 21 sync., 20 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077772, 11.628492, 612.514, 3.42, 0.109 |
+-----3>302.5%+
:Receiver1: switching PRN19 from FLL to PLL.
+--2286---212336.547-----+
| no ACQ | TRK: 21 ch., 21 sync., 21 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077779, 11.628506, 609.989, 2.74, 0.095 |
+-----3>303.5%+
:Acquisition timing, preprocessing(level1): 0.076
:Acquisition timing, total(level1): 5.058 s for 74940416 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
+--2286---212337.547-----+
| ACQ: 52, 0 | TRK: 21 ch., 21 sync., 21 PLL, 1 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077768, 11.628490, 607.397, 4.18, 0.171 |
+-----3>293.4%+
+--2286---212338.547-----+
| no ACQ | TRK: 21 ch., 21 sync., 21 PLL, 0 CycS, 11 <40dBHz, 21 in SPP |
| | PVT: 48.077771, 11.628504, 607.956, 2.92, 0.248 |
+-----3>314.4%+

```

```

+----- 3>296.1%-+
:Acquisition timing, preprocessing(level1): 0.073
:Acquisition timing, total(level1): 5.092 s for 69812224 correlator values
:Number of FFTs performed (CPU only): 0, time in [ms]: 0
:Receiver1: switching PRN19 from FLL to PLL.
+--2286--212373.547-----+
| ACQ: 54, 0 | TRK: 21 ch., 21 sync., 20 PLL, 2 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077748, 11.628470, 611.243, 5.24, 1.054 |
+----- 3>299.0%-+
+--2286--212374.547-----+
| no ACQ | TRK: 21 ch., 21 sync., 20 PLL, 2 CycS, 12 <40dBHz, 21 in SPP |
| | PVT: 48.077780, 11.628486, 613.626, 3.20, 1.696 |
+----- 3>301.4%-+
:Sample stream ended
:Process command: 'Controller.stop'
: command successfully executed
:GMasterReceiver working thread ended in a regular way
:Controller thread tries gracefully stopping. Now stopping GMasterReceiver and NavProcessor
:Controller now sleeping for 5 s to allow other threads to end
:NavProcessor working thread ended in a regular way
:Controller thread gracefully stopped
The GController ended as it should. Bye bye.

...done

```

- The receiver can be stopped at any time by pressing 'X'
- By pressing 'C' some minor interactions can be performed
- Once the sample stream ends, the receiver displays it on the command window mentioning '**Sample stream ended**' and stops the receiver automatically.

Configuration Settings

General Information

The MuSNAT offers an extensive array of configurable parameters that can be adjusted. While this wide range of configuration options provides significant flexibility, it also introduces the potential for suboptimal or conflicting configurations. Therefore, validating these parameters meticulously by comparing them with the original configuration file and referencing the MuSNATdefs.xml document is strongly recommended to ensure consistency and adherence to valid parameter values.

Configuration Parameters

The following sections and parameters are required for **standard GNSS processing** (i.e., without having any enhanced modules in the receiver) in the [AIG nss.musnat](#) configuration file. It is also possible to modify/create your configuration file for different receivers further based on this configuration file.

Section Acquisition

This section contains some timing parameters relevant for both FFT based acquisition modules.

Parameter	Description	Unit	Default / Related Values	Notes
MaxNavProcAcqBeforeFft	Maximum number of acquisitions via navigation processing which are to be done before an acquisition via signal processing defined in the "FFTAcquisitionLevel"-units is started. Before doing an FFT acquisition, the receiver tries 'MaxNavProcAcqBeforeFft' times to acquire the signals by computing the signal's code phase and Doppler from the last position fix (so-called vector acquisition). If the accuracy of those values is precise enough, the values are used to directly start tracking.	-	-	To deactivate acquisitions via navigation processing, set this parameter to 0.
NoLevel1TotalFailuresBeforeFullOpenSkyLevel2	Number of unsuccessful level 1 acquisition runs before a full open sky (full Doppler range) level 2 acquisition is initiated. Used when cold starting in an indoor environment.	-	-	Use with care, as a level 2 acquisition with the full Doppler range can be computationally demanding.
RangeRateDifferenceThreshold	Acquisition for objects within expected range rates being closer than this threshold + 3 x standard deviation will be removed in that acquisition step.	m/s	0	If 0 or not set, it will be deactivated
ServicesWithDopplerAmbiguitySolution	List of services for which RangeRateDifferenceThreshold will be applied. Each signal is just tested against signals of the same service.	-	Necessary only when RangeRateDifferenceThreshold is set to value other than 0	

Section CentralizedPhaseTracking

Parameter	Description	Unit	Default / Related Values	Note
MaxChannel	Maximum number of channels that can be used within vector tracking or master/slave operation.	-	200	Just increase in case MuSNAT reports in the receiver log an error message.

Section FFTAcquisitionLevel*

The section refers to FFTAcquisitionLevel1 and FFTAcquisitionLevel2.

FFTAcquisitionLevel1 controls the cold start acquisition.

FFTAcquisitionLevel2 controls the warm start acquisition.

Due to the abundance of processing power in modern PCs, it is common practice to configure only a cold start acquisition unit and set FFTAcquisitionLevel2.AcquisitionType == 0.

Parameter	Description	Unit	Default / Related Values	Notes
AcquisitionType	Type of acquisition unit	-		Example: 0 ... deactivated, 1 ... single coherent acquisition (deprecated), 2 ... coherent+noncoherent acquisition with squaring or differential detector (deprecated), 3 ... invalid, 4 ... coherent + noncoherent on either CPU or GPU (default), 5 ... Glonass single coherent acquisition
CancelStrongGnssSignals	Parameter used for acquisition type 4. If true, then interference cancellation will be activated. Interference cancellation is a process that generates replica signals of all tracked GNSS signals and subtracts them from the incoming signal before the acquisition of other (weak) signals takes place.	-		Use for indoor and mixed indoor/outdoor scenarios
GenerateReports	If true, the acquisition will generate reports, which allow displaying the correlation function.	-		Generating reports slows down acquisition and may cause memory defragmentation if the software receiver runs for more than several hours.
GpuDeviceId	Device id of GPU if 'GpuUsage==1'	-		
GpuUsage	GPU acceleration mode (0 = use CPU and do not use GPU acceleration, 1 = GPU acceleration with CUDA) Note: GPU acceleration requires MuSNAT versions with CUDA support.	-		
LogFileAcqMode	Set to 'true' if an acquisition log file shall be generated. Works for type 1, 2 and 4.	-		Log file writing significantly slows down the acquisition process. Needs GenerateReports set to true.
MaxTime	Maximum time, an FFT unit is allowed to run. If the run time exceeds this value, then the unit skips the remaining signals. Those are acquired in the next steps. Set to 0 to processing all signals within the acquisition search list.	s		Should be set to 0 in post-processing to ensure that all signals are acquired. Set to 1 ... 3 seconds in case real-time processing is done.
PlotClipSizeCode	Number of points in code phase Dimension around the maximum peak of the 3D acquisition plot.	-		Large values will slow down GUI considerable. 0 = no clipping and all values are displayed
PlotClipSizeDoppler	Number of points in Doppler Dimension around the maximum peak of the 3D acquisition plot.	-		Large values will slow down GUI considerable. 0 = no clipping and all values are displayed
PlotClipSize	(deprecated: PlotClipSizeCode/PlotClipSizeDoppler instead) use Number of points on each side of the maximum peak of the 3D acquisition plot.	-		Large values will slow down GUI considerable. 0 = no clipping and all values are displayed
SatSignalStatistics	File name of the acquisition log file.	-		Example form (type 1/2): 'AcqLogFile_System% s_ReclId%d_PRN%d_DataBit%d_Week%d_WeekSec%.2f.dat'. Example form (type 4): 'sxnrsr_acq-id=sys%d_prn%d_recid%d-time=week%d_sec%.5f.acqlog'.
DopplerMaxRangeRate	Maximum assumed range rate used to predict the Doppler search range. A list of two values is required. The first element is the min. range rate (typically a negative value), the second element is the max. range rate.	m/s		
ReceiverId	List of Receivers to identify signals to be acquired. If receiver support two components, the second signal component (typically the pilot) will be used for acquisition.	-		First Receiver starts at 0

Section Main

Parameter	Description	Unit	Default / Related Values	Notes
AcquisitionRate	Maximum acquisition rate. The receiver waits 'AcquisitionRate' seconds after each acquisition step, before it starts a new acquisition. It applies to FFT based and vector acquisition.	s		
BlockTrackingDuringAcquisition	If true, then tracking waits for the acquisition to complete. This is the standard setting. If set to false tracking runs during acquisition, which can be more inaccurate and lead to burn jump errors for Galileo BOC-signals.	-		Set to 'true' for post-processing analysis and to 'false' for real-time operation
Description	This string is used to describe the current MuSNAT configuration file	-		Has no influence on the processing.
InitialSkipTime	If the sample input is given by a file, then this value allows to skip the first seconds of the IF sample stream.	-		Only if *.smf file is used, not for SDR ION Metadat afile (*.sdrx)
InputFile	Path of the file containing the meta information about the sample file. If the configuration of the sample file is written in an XML format using the SDR metadata standard, 'xml_ion:' must be added before the path. MuSNAT also tries an automatic identification of the metadata file type.	-		e.g. the file with .smf ending
MeasurementRate	Code and phase pseudorange measurement rate. Must exceed the IF sample input packet size divided by sample rate, i.e. at maximum one measurement record per packet is possible.	s		
RandomCodeConfigFile	Filename of the random code file (.rcf) containing the information about the PRN codes. As default the file 'AllGnss\CodeFamilyAllGnss.rcf' shall used. There exist various special purpose *.rcf files, e.g. for the CBOC signals 'TestbedAnalyse\CodeFamilyE1_100andMBOC.rcf' can be used.	-		
ReceiverClockJumpInterval_ms	Maximum clock error (absolute value), which does not cause a receiver clock jump. If the measured receiver clock error exceeds this value, then the receiver clock is shifted by adjusting the applied clock error value (see Position window). This causes a (consistent) jump in the code and phase pseudoranges, but keeps these values in a reasonable range. If you use '0.0' to deactivate clock jumps, then the receiver clock error is steered by software towards 0.0 s (only if position fix is possible).	ms		
SignalProcessingWaitsForNavigationProcessing	If true, GNSS signal processing waits for the navigation processing (PVT solution(s) and navigation modules) to finish. Default value is true. Attention: This option is intended for developers and must be true for vector tracking applications.	-		
SimulateRealTime	If true, then real-time behavior is simulated in post-processing. The receiver reads the IF sample packets from the hard disk with a rate not faster than real-time.	-		
SpecialConfiguration	Used to define a special receiver configuration, e.g. to use MuSNAT as software GNSS signal simulator	-		0 = Normal; 1 = Signal generation; 2 = Write LOS parameter into *.txt file (not Working in Rev. 592)

UseFftMixer	If true, then complex IQ to real conversion is done with FFT method instead of fs/4 mixer. Should be used with care and shall be set per default to 'false'.	-		Supported sample formats: IQ_2Comp8Bit Further information see: https://athene-forschung.unibw.de/85049?query=hepzibah&show_id=139125
UseFftMixerAmplitudeScaling	If bUseFftMixer is true, apply post-FFT amplitude scaling to prevent quantization losses (check status window output)	-	1.0	
LinkTypeLEO	Used to type of link: 0...Uplink between LEO zenith antenna and GNSS, 1...Downlink between LEO nadir antenna and ground-based spoofer	-		Used within MuSNAT signal generator
ReceiverSectionNames	This list defines the actual 'receivers' used to process the signal samples. For each entry in this list, a corresponding section must exist in the *.musnat file.	-		

Section Navigation

Parameter	Description	Unit	Default / Related Values	Notes
Description	A string describing the navigation processor configuration.	-		The string has no influence on the processing.
LvpPositionFeedbackThreshold	Position accuracy threshold of an LVP position to be considered as a feedback to vector tracking.	m, 3D RMS		
MaxPvtAge	Maximum age of a PVT solution to be considered as a valid feedback within the code/Doppler vector tracking algorithm.	s		
MonitorSppResiduals	If true, then the receiver monitors the residuals of the single point positioning (SPP) navigation module and stops channels with large residuals.	-		
ProvidePositionForVectorTracking	Bool value. Define if calculated Position of navigation modules are provided for vector tracking.	-		
StopWeakSignalCrossCorrelations	If true, then the receiver detects cross-correlation tracking (only relevant for GPS C/A code signals) and stops the respective channel. Useful in an indoor scenario where the tracking channels are configured to a high sensitivity. Then it may happen that a channel actually tracks a cross-correlation peak of a strong signal.	-		
SectionNames	A list containing the section names data of the used navigation modules. The software receiver produces a record of raw measurements (code/phase pseudoranges + auxiliary data) every e.g. full GPS second and passes them to the navigation processor. The record is then sequentially processed by the listed navigation modules. A navigation module may modify the content of the record.	-		
ServicesThatNeedSppOk	Applies if 'MonitorSppResiduals' == true. Signals from those services need to get a SppOK flag at least one time within 5 minutes. Otherwise, they corresponding channel is stopped. Used to avoid false lock on interfering signals.	-		

Section Navigation Data

Parameter	Description	Unit	Default / Related Values Optionality	Notes
InputFileName	Filename where to read from the initial navigation data in the internal software receiver format (.nav). This file is read at receiver start-up. Can be used to read assistance data (ephemeris, clock, almanac, data bits ...) to aid processing. Use string '-' to disable reading pre-recorded navigation data (this leads to a receiver cold-start).	-		Note 1: Use the value of 'OutputFileName' plus '.nav' to accumulate all navigation data during multiple post-processing runs. Note 2: Use an initial .pvt file plus an initial .nav file to mimic a receiver warm start (or if a precise .pvt file is provided plus ephemeris data, a hot start).
DeleteOldEgnosRecords	Variable to set if the EGNOS messages contained in the container have to be deleted if they are too old (value = 1) or have to be kept in the container (value = 0).	-	0	Set to 0 when extracting HAS-Data to *.nav
OutputFileName	Filename where to store the navigation data. This file is written when the receiver stops in NAV format. Currently, only data for GPS ephemeris are written into the NAV file. Additionally, a '.nav' suffix is added to 'OutputFileName' for an additional file in the MuSNAT internal format. The .nav file contains all navigation data read from 'InputFileName' plus the ones decoded from the GNSS signals plus any other data from SP3 files of generated tropospheric models.	-		
RINEXGPSNavCreateFile	To generate navigation message plus the internal .nav format select true.	-		
RINEXGPSNavAgency	Name of the agency creating current NAV file (19 characters usable).	-		
RINEXGPSNavComment	Comment line for the NAV file (59 characters usable).	-		
RINEXGPSNavProgram	Name of the program creating current NAV file (19 characters usable).	-		
SP3InputFileName	Name of first SP3 to read at receiver startup. Use '-' if no SP3 file is available.	-		
SP3InputFileNameGlonass	Name of a second SP3 file to read at receiver startup (Note: the word 'Glonass' has no meaning here, it is just an second SP3 file to read). Use '-' if no second SP3 file is available.	-		
SP3InputFileTimeOffset	Offset of Timetags of SP3 File to GPS Time in seconds	s	0	positive adds seconds to Timetags of the SP3 File, negativ subtract
UseTroposphereModel	If set to 1, then one NavRecord entry with tropospheric model parameters is automatically generated at receiver startup.	-		Usually set to 1 to ensure the tropospheric correction can be applied during SPP.

Section RINEX_Obs

This is module to write RINEX observation files.

Parameter	Description	Unit	Default / Related Values	Notes
Agency	Name of observing agency (20 characters usable).	-		
AntennaNumber	Antenna number (20 characters usable).	-		
AntennaOffsetE	Antenna offset East.	m		
AntennaOffsetH	Antenna offset Height.	m		
AntennaOffsetN	Antenna offset North.	m		
AntennaType	Antenna type (20 characters usable).	-		
ApplyMeasuredClockErrorToMeasurements	Determines if the measurement epochs are adjusted by the SPP estimated receiver clock error before writing to the RINEX file (the receiver time then is also adjusted by the measured clock error). This produces RINEX records, whose nominal times coincide with the true GPS times. Works only if a position fix can be obtained.	-		
Approx_X_CoordinateMarkerWGS84	Approximate WGS84 X-Coordinate of the marker (see ValidApproxMarkerPosition).	m		
Approx_Y_CoordinateMarkerWGS84	Approximate WGS84 Y-Coordinate of the marker (see ValidApproxMarkerPosition).	m		
Approx_Z_CoordinateMarkerWGS84	Approximate WGS84 Z-Coordinate of the marker (see ValidApproxMarkerPosition).	m		
FileName	Path to and name of the RINEX file. Filename can be generated using substitutes.	-		Explanation: put a '#' as the first character of this string to enable substitute usage. Then <code>_YYYY_</code> will be replaced for year 4 digits, <code>_DDD_</code> for day, <code>_YY_</code> for year 2 digits and <code>_R_</code> for revision number e.g.: 'o: _YYYY_DDD_GPS_DDD_R_..yy_o'
GenerateRINEX_Output	Set to true to generate RINEX output files.	-		
Interval	RINEX output interval, a value of 0.0 results in output of all valid epochs.	s		
MarkerName	Name of RINEX marker (4 characters usable).	-		
MarkerNumber	Number of RINEX marker (20 characters usable).	-		
Observer	Name of observer (40 characters usable).	-		
OutputOnlyWhenValidClockErrorDifference	If true, then RINEX data is only generated, when the receiver was able to estimate the receiver clock error.	-		Note: set OutputOnlyWhenValidClockErrorDifference and OutputOnlyWhenValidReceiverPosition to false to output all data.
OutputOnlyWhenValidReceiverPosition	If true, then a RINEX record is only generated when the receiver was able to compute a position fix.	-		Note: set OutputOnlyWhenValidClockErrorDifference and OutputOnlyWhenValidReceiverPosition to false to output all data.
RINEX_OutputDataCarrierPhase	Set true to output carrier phase measurements.	-		
RINEX_OutputDataCodePhase	Set true to output code phase measurements. (not standard compliant)	-		Note: this is a non-standard(!) RINEX extension, identified with character H within RINEX file.
RINEX_OutputDataDoppler	Set true to output Doppler frequency.	-		
RINEX_OutputDataPseudorange	Set true to output code pseudorange measurements.	-		

RINEX_OutputDataSignalStrength	Set true to output the ratio of the carrier power to the noise spectral density (C/N0).	-		
RINEX_OutputTimeDifferenceEpoch	Set true to output the difference in time of first observable in that epoch in regards to epoch	ns		
RINEX_OutputFormat3	Select RINEX output format, 0 = format 2.11, 1 = format 3.00.	-		
RINEX_OutputGPS_L1C/A	If true observations from GPS satellites on L1 C/A are stored within the RINEX file.	-	Service Id = 1	
RINEX_OutputGPS_L2C (M+L)	If true observations from GPS satellites on L2C are stored within the RINEX file.	-	Service Id = 2	
RINEX_OutputGPS_L2P (Y)	If true observations from GPS satellites on L2P(Y) obtained via cross-correlation are stored within the RINEX file."	-	Service Id = 3	
RINEX_OutputGPS_L5IQ	If true observations from GPS satellites on L5 are stored within the RINEX file.	-	Service Id = 4	
RINEX_OutputGlonass_L1	If true observations from GLONASS L1C/A are stored within the RINEX file.	-	Service Id = 8	
RINEX_OutputGlonass_L2	If true observations from GLONASS L2C/A are stored within the RINEX file.	-	Service Id = 9	
RINEX_OutputGalileo_E5aIQ	If true observations from Galileo satellites on E5a are stored within the RINEX file.	-	Service Id = 6	
RINEX_OutputGalileo_E5ab	If true observations from Galileo satellites on E5ab (AltBOC) are stored within the RINEX file.	-	Service Id = 14	
RINEX_OutputGalileo_E5b	If true observations from Galileo satellites on E5b are stored within the RINEX file.	-	Service Id = 13	
RINEX_OutputGalileo_E6	If true observations from Galileo satellites on E6 are stored within the RINEX file.	-	Service Id = 15	
RINEX_OutputGalileo_L1 (B+C)	If true observations from Galileo satellites on E1 are stored within the RINEX file.	-	Service Id = 5	Note: in the beginning of Galileo E1 was denoted as L1. This explains this odd name for this entry.
RINEXOutputBeidou_B1C	If true observations from Beidou B1C are stored within the RINEX file.	-	Service Id = 35	
RINEXOutputBeidou_B1	If true observations from Beidou satellites on B1 are stored within the RINEX file.	-	Service Id = 26	
RINEXOutputBeidou_B2	If true observations from Beidou satellites on B2 are stored within the RINEX file.	-	Service Id = 27	
RINEXOutputBeidou_B3	If true observations from Beidou satellites on B3 are stored within the RINEX file.	-	Service Id = 28	
RINEX_OutputSBAS_L1C/A	If true observations from SBAS satellites (e.g. EGNOS) on L1 are stored within the RINEX file.	-	Service Id = 7	
RINEXOutputLTE_PCI	If true then observations from processing communications signals (LTE, 5G, etc.) are written into the RINEX file	-	Service Id = 36	Special 3-digit satellite ids are used to identify the transmitters within the RINEX OBS file
RINEX_OutputStarlink_C1	If true observations from Starlink satellites on Channel 1 are stored within the RINEX file.	-	Service Id = 47 (Centerfreq 10.825117 1875 GHz)	
RINEX_OutputStarlink_C2	If true observations from Starlink satellites on Channel 2 are stored within the RINEX file.	-	Service Id = 48 (Centerfreq 11.075117 1875 GHz)	
RINEX_OutputStarlink_C3	If true observations from Starlink satellites on Channel 3 are stored within the RINEX file.	-	Service Id = 37 (Centerfreq 11.325117 1875 GHz)	

RINEX_OutputStarlink_C4	If true observations from Starlink satellites on Channel 4 are stored within the RINEX file.	-	Service Id = 38 (Centerfreq 11.575117 1875 GHz)	
RINEX_OutputStarlink_C5	If true observations from Starlink satellites on Channel 5 are stored within the RINEX file.	-	Service Id = 49 (Centerfreq 11.825117 1875 GHz)	
RINEX_OutputStarlink_C6	If true observations from Starlink satellites on Channel 6 are stored within the RINEX file.	-	Service Id = 50 (Centerfreq 12.075117 1875 GHz)	
RINEX_OutputStarlink_C7	If true observations from Starlink satellites on Channel 7 are stored within the RINEX file.	-	Service Id = 51 (Centerfreq 12.325117 1875 GHz)	
RINEX_OutputStarlink_C8	If true observations from Starlink satellites on Channel 8 are stored within the RINEX file.	-	Service Id = 52 (Centerfreq 12.575117 1875 GHz)	
RINEX_OutputIRNSS_L1	If true observations from IRNSS satellites on L1 are stored within the RINEX file.	-		
RINEX_OutputIRNSS_L5	If true observations from IRNSS satellites on L5 are stored within the RINEX file.	-		
RINEX_OutputQRZSS_L1C	If true observations from QZSS satellites on L1C are stored within the RINEX file.	-		
RINEX_OutputIridium_RingAlert	If true observations from Iridium satellites of Simplex Channel 7 are stored within the RINEX file. (Ring Alert)	-		
RINEX_OutputXona_L5	If true observations from Xona satellites on L5 are stored within the RINEX file.	-		
RINEX_Output5G_NTN	If true observations from 5G satellites off the NTN are stored within the RINEX file.	-		
RINEX_Output5G_TN	If true observations from 5G stations off the TN are stored within the RINEX file.	-		
RINEX_OutputIris2	If true observations from Iris ² satellites are stored within the RINEX file.	-		
ReceiverNumber	Receiver number (20 characters usable).	-		
ReceiverType	Receiver type (20 characters usable).	-		
ValidApproxMarkerPosition	Determines if a known approximate marker position is written into the RINEX header or the a value determined by SPP (if available)	-		

Section Receiver*

The receiver section is responsible for tracking one GNSS service, like Galileo E1BC (Open Service on E1) or GPS L5 just to name a few examples. Typically multiple receiver sections are present and they are named as Receiver0, Receiver1, ... , Receiver10, Receiver11 and so on. The receiver section comprises a number of parameters for replica generation, tracking loop configuration, navigation message decoding and other features. The table below lists the ones most relevant for standard tracking, like it is found in reference stations and described in most text books. At the end of this page some special remarks are given including configuration of the correlation engine.

Parameter	Description	Unit	Default / Related Values	Notes
AntennaId	The antenna ID is used to distinguish different GNSS antennas in a multi-antennae system. The first antenna has the id = 0. This entry actually defines the id (it is not a selector).	-		The IF sample data selection is done by the field 'StreamNo'.
BumpJump	If true, a bump jump algorithm is performed to avoid false locks with BOC signals (or for any navigation signal exhibiting a correlation function with multiple peaks). The bump jump algorithm is carried out with the data signal component.	-		Note, that you need to configure a proper number of multi-correlators to identify false locks.
BumpJumpIntegrationTime	Time period used to determine the position of the correlation function peak. The bump jump algorithm shifts the peak back to the prompt correlator, if a significant deviation is detected.	s		
BumpJumpMinDistance	If the bump jumper does not attempt to jump for more subchips than this value, the bump jump is cancelled (to avoid frequent jumps of low distance)	m		
BumpJumpThreshold	If the ratio of the largest correlator value divided by the prompt correlator value exceeds this threshold, the bump jumper is triggered.	-		
CarrierPhaseAlways	If true, then the carrier phase is always measured, even if the channel is not in phase lock.	-		Use mostly for debugging applications. Should be set to 'false' during normal receiver operation.
CodeMultiPathStdDev	Code pseudorange accuracy offset, which is added to the code discriminator based standard deviation. The output code pseudorange accuracy follows the formula $\sqrt{\text{CodeMultiPathStdDev}^2 + \text{DiscNoiseStdDev}^2}$. Applies only if 'AddDiscriminatorToRawMeasurements' is true.	m		Normals not used; see also SPP.StdDevCodePseudorangeAutomatic
CoherentFraction	Defines primary coherent integration period for the data channel. The primary coherent integration period is determined as the PRN code period divided by 'CoherentFraction'.	-		Normally set to 1, with the notable exception of L2CL
CoherentFractionSlave	Same as CoherentFraction, but for pilot channel	-		The primary coherent integration period is determined as the PRN code period divided by 'CoherentFractionSlave'.
CoherentIntegrator	Modul to do secondary coherent integration and bit-sync for the data channel.	-	0 = bypass (default if signal has no secondary code); 1 = histogram for GPS C/A; 2 = Kalman Filter+Viterbi (default method for all signals with secondary codes); 3 = cross-correlation with predefined secondary code (default for signals with long secondary codes, e.g. L1CP).	
CoherentIntegratorSlave	Same as CoherentIntegrator but for pilot channel	-		
CorrelatorMode	Flag to indicate if numerical approximations are applied when generating the replica signals for the data channel. To obtain an utmost precision, CorrelatorMode must be set to 0. Only applicable if the parameter CorrelatorType = 1. CorrelatorType 4 always works in precise mode.	-	0 precise correlation 1 fast correlation	Use 0 per default, as speed gain by using 1 is low.
CorrelatorModeSlave	Same as CorrelatorMode but for pilot channel.			
CorrelatorTrackingMode	If CorrelatorTrackingMode == 1 then waveform based tracking is used (P- and D-correlator are employed), if CorrelatorTrackingMode==0 then correlator based tracking is used (early, prompt, late, ... correlators are used). Applies only for 'CorrelatorType == 1' CorrelatorType 4 always works in correlator based mode.	-		Note 1: waveform based tracking requires D-correlator entries within the *.rcf file and is insensitive to the used IF sample rate. Note 2: correlator based tracking requires calculation of the EarlyLateScale based on the used correlator configuration and sampling rate.

CorrelatorType	Indicates which code base is used for signal correlation CorrelatorType should be set to 1 for standard GNSS tracking. CorrelatorType==1 realizes a full 16-bit hardware NCO plus multi-correlator on the CPU without any approximations CorrelatorType==4 is a GPU-based correlator exploiting FP16 values for IF samples and needs a nVidia graphics card of at least Turing architecture. CorrelatorType==5 is a true AltBOC correlator following the same principles as type 1. The other types are for special use only and should be used with care: CorrelatorType==0 is CPU based waveform based tracking only and quite slow (legacy code) and CorrelatorType==2 is an 8-bit CPU based correlator with less memory bandwidth demands but introducing some extra noise. CorrelatorType==3 is a special type to be used if this section defines a GNSS signal generator. CorrelatorType==6 is an GPU-based correlator exploiting FP16 values and shared memory for IF samples and needs a nVidia graphics card of at least Turing architecture	-		Use 1 as default
CycleSlipBadPoints	Number of bad points added to the cycle slip stability counter each time when the PLL exceeds a stability threshold (0.125 cyc for data channels, 0.25 cyc for data+pilot channels). If no cycle slip is detected the stability counter is decreased by 1.	-		
CycleSlipThreshold	If the number of bad points exceeds this threshold, a cycle slip flag will be set and the number of bad points will be reset.	-		
DataWipeOff	If set to true, then the data channel tries to perform a data wipe-off to improve carrier tracking performance.	-		requires that data bits are provide as .nav file to MuSNAT
DataWipeOffSlave	Same as DataWipeOff but for pilot channel	-		
Description	A string describing the current receiver configuration. The string has no influence on the processing.	-		
DllBandwidthNarrow	The bandwidth used in the DLL; applies for FilterMode 1 and 2 (after filter convergence)	Hz		
DllBandwidthWide	For 'FilterMode' = 2. Defines the DLL bandwidth, used immediately after acquisition (and thus before convergence).	Hz		
DllFilterOrder	DLL filter order for 'FilterMode' = 1. Choose 1-3 for a 1st to 3rd order filter. Choose 0 for open loop tracking.	-		
DoHardSync	If set to true then an artificial satellite time stamp 0 is used to calculate pseudoranges. This can be used when the navigation signal does not carry any time stamp. Useful when pseudorange differences are used in a specific test measurement scenario.	-		Set to false during normal operation
NavDataBitPresent	Indicates if a Costas phase discriminator is going to be used or not and if navigation data is used or ignored. The Costas phase discriminator will cope with data bits or with a secondary code on the data channel.	-		
DumpMultiCorrValues	If true, then multi-correlator values are logged into a files with the *.mlog format. Note: Dumping of multi-correlator values is recommended only in post-processing mode. If set to true then multi-correlator positions are defined via 'MinMCIndex/MaxMCIndex'. See also parameter: DumpMultiCorrValuesFileTemplate	-		true = log MC Values, false = no logging
DumpMultiCorrValuesFileTemplate	Filename template for multi-correlator dump files. Must be of the form 'MCDump_service%d_sat%d_chan%d_cnt%d_type%d.s.mlog'. See also parameter: DumpMultiCorrValues	-		e.g. Out\Galileo_E1_MCDump_sys%d_sat%d_chan%d_cnt%d_type%d.s.mlog
EarlyIndex	Defines early correlator for data channel as index to the list of CorrelatorPositions. Used if correlator based tracking is used (see CorrelatorType)	-		
EarlyIndexSlave	Same as EarlyIndex but for pilot channel.	-		
EarlyLateScale	Factor to multiply with the (E-L)/P difference for code tracking (data channel). Used if correlator based tracking is used (see CorrelatorType)	-		The YASST tool calculates this value in the function doPerfAnalysis (the parameter doPerfAnalysis must be set to true for that). The value to be used is called Code correlator gain and is given in µs. The Code_correlator_gain value represents the slope of the discriminator (DLL) S-curve. The formula to calculate this value is given below: EarlyLateScale = Code_correlator_gain [s] * Oversampling_factor [Subchip/Chip] * Nominal_code_rate [Chip/s]
EarlyLateScaleSlave	Same as EarlyLateScale but for pilot channel.	-		
ExtendedCoherentIntegrationDuration	Duration of the extended coherent integration time of the data channel in terms of number of primary coherent integrations (e.g. 1, 5, 25). The secondary coherent integration time is the product of this number with the primary code length divided by the 'CoherentFraction'.			Usually set to a value to achieve an extended coherent integration time of 4 ... 10 ms.
ExtendedCoherentIntegrationDurationSlave	Same as ExtendedCoherentIntegrationDuration for pilot channel.	-		
FilterMode	Tracking loop filter mode. 1 ... standard DLL/PLL tracking where PLL switches to FLL if signal power is below ThresholdFLL 2 ... aided DLL/FLL/PLL (see Kaplan book)	-		Use 1 for default operation
FllBandwidthNarrow	The bandwidth used in the FLL	Hz		
FllBandwidthWide	For 'FilterMode' = 2. Defines the FLL bandwidth, used immediately after acquisition.	Hz		
FllCycleSlipThreshold	Applies for 'FilterMode' = 1 and PLL tracking only: If the FLL discriminator exceeds this value in [Hz], then a cycle slip is inserted and the carrier tracking loop switches back to FLL tracking. If you choose 0, then no check is employed.	-		
FllFilterOrder	FLL filter order for 'FilterMode' = 1. Choose 1-3 for a 1st to 3rd order filter. Choose 0 for open loop tracking.	-		
FllCodePhaseOffset	Offset for LEO-SVs in handover between acqtracking	m		Determined by multicorrelator observation
FllErrorPulseLength	Energy containing Pulselength of replica, used in FLL error computation. Used f.e. with 5G PRS Tracking	s		
FrameSize	Number of IF samples within one frame. A frame is a batch of samples used in signal tracking. An integer number of frames must fit into packet of IF samples. The frame size needs to be smaller than the smallest primary coherent integration period.	-		Choose '0' for automatic frame size calculation.

FrequencyOffset	A frequency offset to the carrier frequency can be set, for example 750000 Hz. Gets added upon IF. This was for example used in experiments to not disturb real GPS-users. Can be used to identify the E5a and E5b center frequency within a IF sample stream capturing the whole E5 band. If used, increase AcqMinReqSampleRate accordingly avoid bandlimitations in Acquisition.	Hz		for example: 750000 Hz. Gets added onto IF. If this Parameter is used, make sure to increase
GpsWeekOffset	This value is added to the decoded GPS NAV week number.	-	Should be 0, 1024 or 2048	The first rollover (1024) took place midnight August 21 to 22, 1999. The second rollover (2048) occurred on the night of April 6 to 7, 2019.
HardwareDelay	Receiver hardware delay. This value is subtracted from the code phase, the pseudorange and the carrier phase measurements when the pseudoranges are generated in the receiver. If you want to steer the interfrequency biases displayed in the Position window to 0, then add the IFBs to the hardware delay values.	m		
HistogramLength	Number of primary coherent integrations (i.e. PRN code periods) within one secondary code period of the data channel. If the secondary coherent integrator supports extended coherent integration times (e.g. 20 ms for GPS C/A), then you may use the negative value for 'HistogramLengthSlave' to stick normal coherent integration (e.g. 1 ms). Applies for CoherentIntegrator type 1 and 2 (but not for 3)			
HistogramLengthSlave	Same as HistogramLength but for pilot channel.			
HistogramRunTime	How many primary coherent integration values are used for bit/symbol-synchronization of the data channel. Applies for the CoherentIntegrator type 1, 2 and 3.			
HistogramRunTimeSlave	Same as HistogramRunTime but for pilot channel			
HistogramTrials	Number of identical histogram results in a row before bit sync.		3	Used with CoherentIntegratorSlave 3
LateIndex	See EarlyIndex	-		
LateIndexSlave	See EarlyIndex	-		
LogFileHighRate	Activate if high rate .dumplog files should be written (huge file sizes possible) High rate logging results in a dump every primary coherent integration interval (i.e. at the PRN period rate). Normal rate logging follows the extended coherent integration time.			
LogFileNameMode	File mode when the dump-log file is opened. A dump-log file is opened, every time the corresponding signal has been (re-)acquired. The file is closed after a loss-of-lock event. Choose 'wt' to always create a new file or 'at' to append to a possibly existing file (this option is useful if the channel experiences loss-of-locks and subsequent re-acquisitions).	-		
LogFileNameTemplate	File name template for dump logging. Must be of the form 'GPS_dump_PRN%d.dumplog', where '%d' will receive the PRN number. See also parameter: PRNsToLog	-		
MasterForSlave	Channels of this receiver are providing NCO values to other receivers to allow channel slaving	-	true or false	Set to false for normal receiver operation
MasterControlsAlsoCode	If set to true, then channel aiding between master and slave channels also includes code phase and code rate (carrier phase and Doppler is always slaved)	-	true or false	
MasterCarrierScaling	If true, then carrier phase/Doppler is scaled from master to slave to achieve identical phase / Doppler values expressed in [m] (resp.[m/s]). If false, carrier phase in [cyc] and Doppler in [Hz] is identical for master and slave.	-	true or false	
MasterId	If different from -1, identifies the ReceiverId that acts as master for channel slaving	-	0 for Receiver. -1 to deactivate	
MaxMCIndex	Use to configure the multi-correlator in case MultiCorrelatorConfiguration == true. It is the multi-correlator maximum index for data channel. Position in [chip] = 'MinMCIndex/sample_rate*code_rate. Note: working with 'CorrelatorType' = 2 implies that half the sample rate is used. If for example set to -100 then (for GPS L1/C/A or Galileo E1B and sampling rate of 100MHz) correlators from 100*-1.023*10^6/(100*10^6) - ca. 1 chip till the prompt correlator=0 are used. If MaxMCIndex set to 100 in this case, the correlation function for +/- 1 chip can be seen.	-		
MaxMCIndexSlave	Same as MaxMCIndex but for pilot channel.	-		
MaxMultiCorrOrder	Maximum multi-correlator order; if 'MultiCorrOrder' exceeds this value, then timely successive correlator values will be averaged.	-		
MeasurementDopplerNoiseCov, MeasurementDopplerNoiseCovSlave	For bit-sync. with CoherentIntegrator type 2. Assumed Doppler measurement noise covariance	-		Those values are very specific and require trial and error to set them properly.
MeasurementPhaseNoiseCov, MeasurementPhaseNoiseCovSlave	For bit-sync. with CoherentIntegrator type 2. Assumed Phase measurement noise covariance	-		Those values are very specific and require trial and error to set them properly.
MinMCIndex	See MaxMCIndex	-		
MinMCIndexSlave	See MaxMCIndex	-		
MultiCorrDataFraction	The fraction (0.0-1.0) of the FFT integration time that is actually filled with correlation values. The remaining part is filled with zeros. This value is used to increase the Doppler resolution of the multi-correlator; e.g. a value of 0.25 increases the Doppler resolution by a factor of 4.			
MultiCorrNavBitsRequired	If true, then navigation data bits need to be available to compute multi-correlator values. If they are not available, then multi-correlator values are not computed. Note: if data bits/symbols are available, they will be wiped off before increasing the coherent integration time according to 'MaxCorrOrder'.			
MultiCorrNcoMode	To select the NCO operating mode during the (long multi-correlator) coherent integration time. 0 = use NCO of internal tracking, 1 = constant Doppler, 2 = directly estimate and reduce carrier phase from small batches.	-		

MultiCorrOrder	Defines the coherent integration time in [µs] for the multi-correlator, $T_{coh} = 2^{\wedge}MultiCorrOrder * PRN_code_period / CoherentFraction$. Note, if you give a 'MultiCorrOrder' > 'MaxMultiCorrOrder', then the receivers starts to average successive primary integrate and dump values to achieve an total number of 'MaxMultiCorrOrder' integrate and dumps, which are then used to form the frequency domain values. This effectively reduces the Doppler span by $2^{\wedge}(MultiCorrOrder - MaxMultiCorrOrder)$. In this case no time domain values are visualized or written to a file.	-		
MultiCorrRecoveryTime	Minimum time between two multi-correlator calculations in 'MultiCorrelatorUsage' = 2 or 3.	s		
MultiCorrRefSignal	0 = infinite bandwidth PRN code, 1 = P-correlator reference signal, 2 = D-correlator reference signal, 3 = W-correlator reference signal. The index refers to the reference signal types of the file specified in 'Main.RandomCodesFile'.	-		
MultiCorrelatorConfiguration	If true, then multi-correlator positions are defined via 'MinMCIndex/MaxMCIndex', if false then positions are defined via 'CorrelatorPositions' or 'CorrelatorPositionsSlave'. Applies only for CorrelatorType 1 and 4	-		It is recommended to use MultiCorrelatorConfiguration == false to maximize compatibility with older configurations.
MultiCorrelatorForTracking	If true, then multi-correlator are computed continuously, if false then depending on 'MultiCorrelatorUsage'. Applies only for 'CorrelatorType == 1'.	-		Needs to be set to true for correlator based tracking; if waveform based tracking is used, then true or false can be selected depending on the specific use case.
MultiCorrelatorUsage	Defines if the multi-correlator is used.	-	0 = Off 1 = All channels 2 = Fixed PRN 3 = Time sequencing 4 = Dynamic on the fly from GUI	Typical cases: 0 or 1
MultiCorrelator_PRN	Applies if 'MultiCorrelatorUsage' = 2 and defines PRN for which the multi-correlator is computed.	-		
NavDataBitsPresent	If set to true a Costas phase discriminator is used to cope with data bits or with a secondary code on the data channel.	-		
NavDataBitsPresentSlave	If set to true a Costas phase discriminator is used to cope with data or with a secondary code on the pilot signal.	-		
NavMsgDecoder	Module to do navigation message decoding on the data channel	-	0 = deactivated 1 = GPS C/A NAV decoder 2 = GPS CNAV decoder 3 = GIOVE-A decoder 4 = Galileo INAV V12 decoder 5 = Egnos /Waas decoder 6 = GATE XI /NAV V9 on E1-B 7 = Pseudolite decoder 8 = GATE XI /NAV V12 on E1-B 9 = IMES 10 = QZSS 11 = GLONASS L1 decoder 12 = BeiDou decoder 13 = GALILEO E5a FNAV 14 = GALILEO E6 CNAV 15 = BeiDou D1 16= BeiDou D2	
NavMsgDecoderSlave	Same as NavMsgDecoder but for the pilot channel			Note: it is possible to configure message decoding for the pilot channel, in case a message is broadcast on this component (which normally is not the case)

NavigationMessageType	Defines the type of navigation message bit stream used for configuration of the navigation message decoder.	-	-1 Not applicable 0 Unknown 1 GPS NAV 2 GPS CNAV on L2 3 GIOVE A1 INAV on E1 4 Galileo INAV on E1 5 Galileo INAV on E5 6 SBAS 7 GLONASS 8 Reserved 9 Galileo FNAV on E5 10 Galileo CNAV on E6 11 GPS CNAV on L5 12 BeiDou D1 13 BeiDou D2 14 IRNSS L5 15 IRNSS S	
NavigationMessageSlaveType	An identifier used to distinguish different navigation message bit streams used in the data wipe-off process (-	0 = unknown 1 = GPS NAV 2 = GPS CNAV 3 = GIOVE A1 INAV on E1 4 = Galileo INAV on E1 5 = Galileo INAV on E5 6 = SBAS 7 = GLONASS.	
NumberAssistanceBits	This is a parameter used to store navigation data (from the data channel) bits/symbols at a reference station in the .nav file. Bits/symbols are stored in the form of packets in the navigation data container and are written to file at receiver stop. This value defines the number of bits /symbols per packet. Use 0 to deactivate navigation bit/symbol storage.	-		Note 1: It is actually mostly irrelevant which value is used to activate bit logging as subsequent packets are concatenated within the .nav file. Use e.g. 300 as a default value to activate bit logging. Note 2: should be set to zero during normal receiver operation.
NumberAssistanceBitsSlave	Same as NumberAssistanceBits but for pilot channel	-		Note: makes only sense if - for whatever reason - a message is modulated onto the pilot channel. Otherwise only '1's will be logged.
NumberOfMasterChannels	Number of channels in this receiver	-		
NumberOfSignalComponents	1 = only the data component is tracked, 2 = data and pilot are tracked.	-		
ParallelizingMasterChannels	Indicates if the processing of the MasterChannels is parallelized by OpenMP: 0 ... no parallelization, 1 ... parallelized. If more than one receiver is used adapt the number of OpenMP threads, otherwise the overhead is too big (see config entry ParallelizingMasterChannelsThreadNumber).	-		Normally parallelization is recommended for faster processing, i.e. set ParallelizingMasterChannels to 1
ParallelizingMasterChannelsThreadNumber	Number of parallel OpenMP threads when parallelizing MasterChannels (see config entry ParallelizingMasterChannels)	-		Depending on the CPU type, a good value is 2 to 4.
PhaseDifference	Carrier phase difference between data and pilot signal.	deg		To be set according to the signal ICD to ensure that data and pilot component are properly combined.
PIIBandWidthNarrow	PLL bandwidth for FilterMode 1 and 2 (after convergence)	Hz		
PIIBandWidthWide	For 'FilterMode' = 2. Defines the wide PLL bandwidth, used immediately after acquisition (and thus before convergence).	Hz		
PIIFilterOrder	PLL filter order for 'FilterMode' = 1. Choose 1-3 for a 1st to 3rd order filter. Choose 0 for open loop tracking.	-		
PIIWeightFactorBig	Applicable only for 'FilterMode' = 2. Defines the phase discriminator contribution to the carrier tracking filter. The frequency discriminator contributes (1-PIIWeightFactorBig) to the carrier tracking filter. The 'big' value is used in final tracking mode (narrow bandwidths). Usually this factor is big (i.e. near to 1).	-		
PIIWeightFactorSmall	Applicable only for 'FilterMode' = 2. Defines the phase discriminator contribution to the carrier tracking filter. The frequency discriminator contributes (1-PIIWeightFactorSmall) to the carrier tracking filter. The 'small' value is used after acquisition. Usually this factor is small (i.e. near to 0).	-		
PolyFitCarrier	If true, then apply a poly fit of order 'PolyFitOrderCarrier' based on the carrier discriminator plus the carrier pseudorange is performed during one measurement interval to extract the carrier measurement.	-		Only used for vector tracking. Set to false for normal receiver operation.
PolyFitCarrierMaxAge	Maximum age of the carrier discriminator values to be considered for 'PolyFitCarrier' or 'PolyPreciseDoppler'.	s		Only used for vector tracking.

PolyFitCarrierQualityCheck	If true, then code/carrier pseudoranges plus Doppler measurements are only generated, if the polyfit to the carrier phase did show a maximum residual below 'PolyFitCarrierThreshold'.	-		Only used for vector tracking.
PolyFitCarrierThreshold	If the maximum residual of the carrier-polyfit is below this threshold, then the carrier polyfit method is declared to be successful and used for further processing.	cyc		Only used for vector tracking.
PolyFitCn0	If true, then apply a poly fit of order 'PolyFitOrderCn0' based on the integrate-and-dump interval C/N0 values is performed during one measurement interval to extract the C/N0 estimate.	-		Only used for vector tracking. Set to false for normal receiver operation.
PolyFitCode	If true, then apply a poly fit of order 'PolyFitOrderCode' based on the code discriminator plus the code pseudorange is performed during one measurement interval to extract the code measurement.	-		Only used for vector tracking. Set to false for normal receiver operation.
PolyFitDoppler	If true, then apply a poly fit of order 'PolyFitOrderDoppler' based on the Doppler discriminator plus the NCO Doppler value is performed during one measurement interval to extract the Doppler measurement.	-		Only used for vector tracking. Set to false for normal receiver operation.
PolyFitOrderCarrier	Order of the polynomial to fit carrier discriminator values over the measurement interval (used if 'AddDiscriminatorToRawMeasurements' == true).	-		Only used for vector tracking.
PolyFitOrderCn0	Order of the polynomial to fit CN0 values over the measurement interval (used if 'AddDiscriminatorToRawMeasurements' == true).	-		Only used for vector tracking.
PolyFitOrderCode	Order of the polynomial to fit code discriminator over the measurement interval (used if 'AddDiscriminatorToRawMeasurements' == true).	-		Only used for vector tracking.
PolyFitOrderDoppler	Order of the polynomial to fit Doppler discriminator over the measurement interval (used if 'AddDiscriminatorToRawMeasurements' == true).	-		Only used for vector tracking.
PolyPreciseDoppler	If true, then precise Doppler estimates are output based on a polyfit of the carrier phase during the measurement interval. The maximum possible polynomial order is determined by 'PolyFitOrderCarrier'. If polyfit residuals are too high (see 'PolyFitCarrierThreshold'), then conventional or 'PolyFitDoppler' based measurements are generated.	-		Only used for vector tracking.
PowerAveragingTime	Averaging period for C/N0 estimation of the data channel.	s		Note: the output C/N0 values is the sum of the data and pilot contribution.
PowerAveragingTimeSlave	Same as PowerAveragingTime but for pilot channel.	s		
PrecomputedCorrelationCode	PrecomputedCorrelationCode needs to be set to 2 (correlation dot-product is computed via Intel IPP if CPU-based correlation is used). Other values are not supported anymore.	-		Always set to 2!
ProcessAcceleratorNoiseCovSlave	For bit-sync. with CoherentIntegrator type 2. Assumed Doppler rate process noise covariance	rad ² /s ⁵		Those values are very specific and require trial and error to set them properly.
ProcessFrequencyNoiseCovSlave	For bit-sync. with CoherentIntegrator type 2. Assumed Doppler process noise covariance	rad ² /s ³		Those values are very specific and require trial and error to set them properly.
ProcessPhaseNoiseCovSlave	For bit-sync. with CoherentIntegrator type 2. Assumed phase process noise covariance	rad ² /s		Those values are very specific and require trial and error to set them properly.
PromptIndex	See EarlyIndex	-		
PromptIndexSlave	See EarlyIndex	-		

SatelliteSystem	Satellite system of the signals to be tracked by the receiver	-	<ul style="list-style-type: none"> 0 Unknown system 1 All satellites of a system (no specific satellite number) 2 GPS system 3 Galileo system 4 GLONASS system 5 GEO/WAAS system 6 Signal from Pseudolite 7 IMES signal (PRN 173-182) 8 QZSS signal (PRN 182-187, 193-197) 9 BeiDou system 10 BeiDou system (Compass) 11 LTE system 12 Starlink 13 IRNSS system 14 Iridium system 15 Xona system 16 3GPP NR-like 17 5G NTN system (Non Terrestrial Network) 18 5G TN system (Terrestrial Network) 	Note: this value is attached to the raw observations to label them properly.
SecondaryCode	Secondary code sequence for the data channel, to be specified in chips. Use a ',' to separate PRN-specific secondary codes. If only one sequences is given, it will be used for all PRNs.	-		Note: the secondary code is used by the CoherentIntegrator type 2 and 3 to perform secondary code synchronization and to wipe-off the secondary code chip after synchronization has been achieved.
SecondaryCode	Same as SecondaryCode for the pilot channel.	-		
Service	Label to identify the GNSS service tracked by the receiver	-	<ul style="list-style-type: none"> 0 Unknown service, for test purposes 1 GPS C/A code on L1 2 GPS Civil signal on L2 3 GPS P(Y)-code code on L2 4 GPS L5 5 Galileo E1 OS 6 Galileo E5a OS 7 EGNOS (or WAAS, ...) on L1 8 GLONASS L1 C/A 9 GLONASS L2 C/A 10 IMES on L1 11 QZSS on L1 12 Pseudolite C/A code on L1 	Note: this value is attached to the raw observations to label them properly.

13 Galileo E5b OS
14 Galileo E5a+b OS
15 Galileo E6
16 GPS L1C
17 GPS L1 P (Y)
18 GLONASS L1 P
19 GLONASS L2 P
20 EGNOS (or WAAS, ...) on L5
21 AUX1
22 AUX2
23 AUX3
24 AUX 4
25 AUX 5
26 COMPASS B1
27 COMPASS B2
28 COMPASS B3
29 Galileo E1 PRS Wideband
30 Galileo E1 PRS Upper
31 Galileo E1 PRS Lower
32 Galileo E6 PRS Wideband
33 Galileo E6 PRS Upper
34 Galileo E6 PRS Lower
35 Beidou B1C
36 LTE
37 Starlink Channel 3
38 Starlink Channel 4
39 IRNSS NAVIC L1
40 IRNSS NAVIC L5 SPS
41 QZSS L1C
42 Iridium Ring Alert (Simplex Channel 7)
43 Xona-lite (not Xona, mimics L5)
44 5G NTN
45 5G TN
46 3GPP NR-like
47 Starlink Channel 1
48 Starlink Channel 2
49 Starlink Channel 5
50 Starlink Channel 6

			51 Starlink Channel 7 52 Starlink Channel 8	
SignalPowerEstimatorType	Estimator type for C/N0 estimator. Intended to be changed for signals with uneven energy distribution.	-	1 Differential Estimator (classic, default) 2 Squaring Estimator	described in doi: 10.1109/PLANS.2006.1650689.
StreamNo	Selection of the IF band to be processed by this receiver unit. Either IF sample stream id (0, 1, 2, ...) or nominal RF of respective frequency band (e.g. 1575420000).	-		
SubdividePacketsIntoFrames	Set to true to enable subdivision of IF sample packets into smaller frames (this is required for all 'GNSS receiver' modules but not for the spectrum analyzer)	-		
ThresholdExpo	C/N0 threshold for a raw pseudorange measurement record. The record's estimated C/N0 value must exceed this threshold in order that the record is exported to the navigation processor.	dBHz		
ThresholdFLL	For 'FilterMode' 1 = minimum C/N0 required to switch from FLL into PLL mode and to stay there. For 'FilterMode' 2 = minimum C/N0 required to narrow down tracking loop bandwidths.	dBHz		
ThresholdLossOfLock	Loss-of-lock threshold, if C/N0 falls below this value, tracking is stopped.	dBHz		
TrackingTimeLimitFLL	For 'FilterMode' 1 = minimum time in [s] to work in FLL mode before a switch into the PLL mode is considered. For 'FilterMode' 2 = minimum time to work in wide loop bandwidth mode.	-		
UnderSamplingFactor	IF sample decimation factor when channel is in undersampling mode (see UnderSamplingThreshold). Undersampling reduces computational demands at the cost of a reduced C/N0.	-		
UnderSamplingThreshold	If the estimated C/N0 value exceeds the 'UnderSamplingThreshold', then the channels switches to the under sampling mode.	dBHz		Note: Undersampling is a technique to speed up processing at the cost of higher measurement noise. Should be used with care. Set to 99 for normal receiver operations (to deactivate undersampling)
Viterbi	Type of used Viterbi decoder for the data channel	-	0 = Viterbi decoding deactivated 1 = GPS CNAV mode 2 = Galileo mode 3 = Beidou B1C mode 4 = Galileo HAS.	
ViterbiSlave	Same as Viterbi but for pilot channel	-		
AcqCnoThresho	C/N0 threshold to exceed to declare acquisition. If acquisition estimates a C/N0 above this value, then it activates tracking for this channel. This parameter reflects the threshold γ of chapter 5 of [1]. You should choose is slightly larger than the reported C/N0 values under H0 conditions. The first entry is for level 1 (cold start), the second entry is for level 2 (warm start). The parameter is used only for FFTAcquisitionLevel".AcquisitionType = 4.	dBHz		
AcqMinReqSampleRate	Minimum required sample rate to perform signal acquisition. This value is used to determine the down sampling factor from the native sample rate to the acquisition sample rate (cf. section 5.3 of [1]). The first entry is for level 1 (cold start), the second entry is for level 2 (warm start). The parameter is used only for FFTAcquisitionLevel".AcquisitionType = 4.	MSamples/s		
AcqNumberCoh	Signal acquisition coherent integration time in multiples of the primary PRN code length. The first entry is for level 1 (cold start), the second entry is for level 2 (warm start). The parameter is used only for FFTAcquisitionLevel".AcquisitionType = 4.	-		
AcqNumberNonCoh	Number of non-coherent integrations for signal acquisition. The first entry is for level 1 (cold start), the second entry is for level 2 (warm start). The parameter is used only for FFTAcquisitionLevel".AcquisitionType = 4.	-		
AcqPrimaryFitOrder	FFT order used to represent the primary PRN code sequence; this value determines the code phase resolution = $\#chips/2^{\wedge}AcqPrimaryFitOrder$. The first entry is for level 1 (cold start), the second entry is for level 2 (warm start). The parameter is used only for FFTAcquisitionLevel".AcquisitionType = 4.	-		
CodeFamilyIndex	Index to the values to be extracted from the RandomCodesConfigFile defined in the Main section. The first entry defines the PRN spreading codes of the data channel, the second entry for the pilot channel (if present).	-		
CorrelatorPositions	Defines a number of correlator positions for the data channel respect to the NCO value (i.e. defines the code phase offsets of the various correlators). Several values can be defined. The corresponding index to the values will be used in the following parameters: 'EarlyIndex', 'PromptIndex', 'LateIndex' and so on ...	-		Use this MATLAB script to generate proper entries automatically: <pre>cnt=0;for k=-20:1:-20, fprintf(' <- %d --><item>%d</item>\n',cnt,k), cnt=cnt+1;, end</pre>
CorrelatorPositionsSlave	Same as CorrelatorPositions but for pilot channel	-		
CodeDiscCoefficients	If at least one element differs from zero, then this lists overrides 'EarlyIndex' and 'LateIndex'. The entries of 'CodeDiscCoefficients' are used to obtain the Dcorrelator for the data channel (i.e. early-late correlator for code tracking) from a linear combination of several correlators shifted in the code phase. List needs to have the same length as CorrelatorPositions. The linear combination is then multiplied by EarlyLateScale and divided by the prompt correlator to obtain the code tracking error in [subchip].			Should be used with care to realize e.g. a double-delta correlator to mitigate multipath. Shall be absent for normal receiver operation realizing standard early-late tracking.

PRNSToLog	List of PRNs to log in the dumplog files.	-	-1 : Logging of all tracked satellites 0: No logging is done Otherwise: The PRN number(s) to be logged	
TrackedPrn	The PRNs to be tracked.	-	empty or -1 All PRNs are tracked	

[1] ... PANY, Thomas. *Navigation signal processing for GNSS software receivers*. Artech House, 2010.

Notes on the correlator settings in the .musnat configuration file:

Main correlator setting	Code tracking mode	Replica signal for code tracking	Replica signal for phase tracking	Replica signal for multi-correlator	Note and example configuration
CorrelatorType=1, CorrelatorTrackingMode=true	Waveform based tracking	vsCodeFileNames	vsCarrierFileNames	If MultiCorrelatorForTracking == true then vsCarrierFileNames, otherwise see MultiCorrRefSignal	This should be the default configuration for most applications. Waveform based tracking also to reuse the same parameter set for different sampling rates. The correlation runs on the CPU making use of the Intel IPP library. Note, the Intel IPP supports also most processors from AMD. Example: Configurations\AllGnss\AllGnss.musnat (all receivers there are configured for waveform based tracking and tailored for reference station use)
CorrelatorType=1 or 4, CorrelatorTrackingMode=false	Correlator based tracking	vsCarrierFileNames	vsCarrierFileNames	vsCarrierFileNames	Having access to dedicated correlators including early, late and prompt is the basis for most analysis in navigation signal processing. This mode requires some care in configuring the correlators and in setting the EarlyLateScale as this does depend on the sampling rate. Both CPU and GPU based computation are supported (GPU only if CUDA is enabled during compile time). Example: Configurations\CUDA_Correlator\SQM_CUDA.musnat (this is a Galileo E1BC+E5aIQ plus GPS C/A+L5IQ receiver config tailored for signal quality monitoring). Set CorrelatorType=1 or 4 depending on your preferred processing mode (1=CPU, 4 = GPU) and all other parameters remain identical.
Obsolete Configuration:					
CorrelatorType=0	Waveform based tracking	vsCodeFileNames	vsCarrierFileNames	n/a	
CorrelatorType=2	Correlator based tracking	vsCarrierFileNames	vsCarrierFileNames	vsCarrierFileNames	

Notes on baseband logging

Settings necessary to activate baseband logging. Baseband logging writes short snippets of the received GNSS signal to a file after carrier wipe-off has been performed. Mostly used for debugging and spreading code authentication.

BasebandLoggingTime	Time to log the baseband signal e.g. 10. Logging time is calculated by trying to achieve an SNR value of abs(BasebandLoggingTime) e.g. 16 [dB]	ms		If the value is negative, than the logging time is calculated depending on the MasterReceiver signal power, high C/N0 short logging, low C/N0 long logging
TrueBasebandLoggingFile	Logs the first 100 ms after PLL phase lock has been achieved and after carrier wipe-off in the defined .tbb file	-		
RepeatBasebandLogging	If >0, the baseband logging is repeated with a time interval according to its value in [s]	-		0, 1, 2, 2.5

Section Receiver Status

The ReceiverStatus module is responsible for maintaining the most recent internal PVT solution. This PVT solution triggers a number of operations within the receiver and is e.g. used to compute satellite visibility to coordinate signal acquisition or for vector tracking.

Parameter	Description	Unit	Default / Related Values	Notes
AllowSnapShotQuickFix	Use true if a coarse estimation of the receiver position and clock error by a code ambiguity resolution algorithm shall be used. This includes a set of algorithms summarized as QuickFix and decreases the time-to-first-fix.	-		Set to false (=0) for default operation.
Cutoff	Cutoff angle for signal acquisition.	deg		The receiver does not try to acquire signals below this cutoff angle.
EstimatedMinimumReceiverVelocity	This value is used to extrapolate the uncertainty of the receiver position for QuickFix. It is assumed that the receiver moves at least with this velocity.	m/s		Only required if AllowSnapShotQuickFix == true
LinearClockDriftErrorParameter	This value is used to linearly extrapolate the uncertainty of the receiver clock error for QuickFix.	s/s		Only required if AllowSnapShotQuickFix == true
MaximumNominalTimeDeviation	This value defines the time limit, how long the receiver position can be precisely extrapolated from the last verified position solution for QuickFix.	s		Only required if AllowSnapShotQuickFix == true
QuadraticClockDriftErrorParameter	This value is used to quadratically extrapolate the uncertainty of the receiver clock error for QuickFix.	s/s ²		Only required if AllowSnapShotQuickFix == true
ReceiverStatusInputFileName	File (*.pvt) containing the initial receiver position, velocity and time estimate.	-		Use an initial .pvt file plus an initial .nav file to mimic a receiver warm start (or if a precise .pvt file is provided plus ephemeris data a hot start).
ReceiverStatusOutputFileName	Filename (*.pvt) where receiver will store its last position, velocity, and time estimate before closing.	-		
RequiredHealthLevel	Required minimum health level for satellites. If the health level of the satellite is below this value, it will never be acquired.	-		Possible values: 0 (health non-existent), 1 (health unusable), 2 (health bad), 3 (health good).
StaticMeasurement	This value is used to extrapolate the uncertainty of the receiver position for QuickFix. If set to true, the uncertainty of the receiver position does not increase with time (when extrapolating).	-		Only required if AllowSnapShotQuickFix == true

Section SPP*

The SPP module computes an epoch-per-epoch position, velocity and time (PVT) solution making use of code pseudorange and Doppler observations. The SPP modules validates each PVT solution by a simplified RAIM algorithm thus requiring as a minimum pseudorange observations from 5 (not 4) satellites. In case pseudorange observations from only 4 satellites are present, then the SPP module outputs only the a position data but no receiver clock error.

Multiple SPP section can be present in the .musnat file, and each section can be responsible for a dedicated SPP configuration (e.g. GPS only, Galileo only, all GNSS, ...). The sections are then named SPP1, SPP2, SPP3,

As default only on SPP section is present calculating the PVT solution from all available signals. In this case the SPP section shall be named 'SPP' only without any digit.

Parameter	Description	Unit	Default / Related Values	Notes
AntennaSelector	If different from 0, then only data from the selected antenna is processed by this module. Records from other antennas are left untouched.	-		
ElevationCutoff	Elevation cutoff for single point positioning	deg		
ElevationDependentWeighting	Boolean value to indicate if elevation dependent weighting shall be used.	-	true/false	If true, the weight is reduced by the following factor (e = elevation in [rad]): $\text{pow}(1.0 - \text{pow}(1.0 - \sin(e), 2.0), 2.0)$.
SppWeightingScheme	Defines weighting scheme employed within SPP. 0 UnityVariance (all measurements same predefined variance value); 1 Unmodified (no extra weighting scheme); 2 WeightingElevDefault (Default weighting scheme, depending on sat elevation); 3 ElevCosSquared (depending on sat elevation and \cos^2 function)	-		
EstimateIonosphericDelay	If true, single point positioning will estimate (unmodeled) ionospheric delays for all multi-frequency pseudorange measurements.	-		Note 1: Ionospheric models are always applied and delays are estimated with respect to the applied model. Note 2: Use this option with care, it is mostly intended for ionospheric delay monitoring and will in general NOT improve PVT accuracy
EstimateGalileonDelay	If true, single point positioning will estimate (unmodeled) ionospheric delays for Galileo multi-frequency pseudorange measurements.	-		Note 1: ionospheric models are always applied and delays are estimated with respect to the applied model. Note 2: Use this option with care, it is mostly intended for ionospheric delay monitoring and will in general NOT improve PVT accuracy
ServicesUsedInSPP	A list of services that is used in SPP.	-	[-1 or 0] = default selection of GPS[C/A, L2C, P(Y), L5], Galileo (E1, E5a, E6), BeiDou (B1C), Glonass (L1C/A, L2C/A). 1 = GPS C/A, 2 = GPS L2C/S, 3 = GPS L2P(Y), 4 = GPS L5, 5 = Galileo E1 OS, 6 = Galileo E5a OS, 7 = SBAS on L1, 8 = GLONASS L1C/A, 9 = GLONASS L2C/A, 10 = IMES, 11 = QZSS, 12 = Pseudolite C/A code on L1, 13 = Galileo E5b OS, 14 = Galileo E5a+b OS, 15 = Galileo E6, 16 = GPS L1C, 17 = GPS L1P (Y), 18 = GLONASS L1P, 19 = GLONASS L2P, 20 = SBAS L5, 21-25 = AUX1-5, 26 = Beidou B1, 27 = Beidou B2, 28 = Beidou B3 (see List of Services)	
StdDevCodePseudorangeAutomatic	Boolean value (true/false) to indicate if the standard deviation for code pseudorange and range rate is taken from signal processing (=true) or defined by the user (= false). In the later case, the value are taken from 'RangeStdDev' and 'RangeRateStdDev'.	-		Use 'false' as default option. Provision of accuracy values from signal processing requires well written .rcf files and is more an experimental feature.
RangeStdDev	Default value for the standard deviation of a code pseudorange measurement.	m		
RangeRateStdDev	Default value for the standard deviation of a code range rate measurement.	m/s		

MaxFirstAdjustmentVUW	Maximum a posteriori variance of unit weight in first (coarse) adjustment step. If the value is exceeded, SPP is aborted. The first step computes a coarse position with signals from the five strongest transmitters. It neglects IFBs and atmospheric delays.	-		
MaxSecondAdjustmentVUW	Maximum a posteriori variance of unit weight in second (fine) adjustment step. If the value is exceeded, SPP is aborted. The check is performed for position and velocity estimation. In case of velocity estimation, the value is divided by 10 ⁶ to achieve a more stringent limit.	-		
MaxRangeResiduum	If a SPP residuum of a pseudorange measurement exceeds this value in the coarse adjustment, then the measurement will be flagged as 'ResiduumExceeded' and it will not be considered in the fine adjustment	m		
ServicesWithIFBs	A list of services that require IFB estimation.	-	1 = GPS C/A, 2 = GPS L2C/S, 3 = GPS L2P(Y), 4 = GPS L5, 5 = Galileo E1 OS, 6 = Galileo E5a OS, 7 = SBAS on L1, 8 = GLONASS L1C/A, 9 = GLONASS L2C/A, 10 = IMES, 11 = QZSS, 12 = Pseudolite C/A code on L1, 13 = Galileo E5b OS, 14 = Galileo E5a+b OS, 15 = Galileo E6, 16 = GPS L1C, 17 = GPS L1P(Y), 18 = GLONASS L1P, 19 = GLONASS L2P, 20 = SBAS L5, 21-25 = AUX1-5, 26 = Beidou B1, 27 = Beidou B2, 28 = Beidou B3 (see List of Services)	
SingleFrequencyStdDev	Default value for the standard deviation of single-frequency ionospheric delays.	m		Only relevant if ionospheric delay estimation is activated.
CreateSppLogFile	If true, then single point positioning creates a log file in the *.poslog format.	-		
CreateIfbLogFile	If true, then an interfrequency bias (IFB) log file is created with the same filename as the SPP (*.poslog) file but using the extension *.ifblog.	-		
SppLogFile	Filename of single point positioning log file.	-		
CreateAdditionalWeightSppLogFile	If value CreateSppLogFile is true and this also is set true, single point positioning creates an additional log file for the weighting factors used within the least squares adjustment. Filename is defined by entry SppLogFile, where '..._LSWeights...' is inserted	-		
UseAtmosphericModels	If true, applies atmospheric models to SPP solution	-	true, false (default: true)	
UseRelativisticEffect	If true, applies relativistic effect on time for SPP solution	-	true, false (default: true)	

Section SpectrumAnalyzer

The SpectrumAnalyzer computes the signal spectrum by applying a FFT transform to a number sample batches (each of size of 2^{FFTOrder}) and averaging their magnitude. A Hanning window function is applied and adjacent batches overlap by 50 %. A maximum number of 'MaxSegments' batches is considered. If the length of the incoming IF sample packets, defined by the .smf file, is shorter, then the number of batches is smaller than 'MaxSegments'. All streams are considered. Furthermore, the SpectrumAnalyzer computes bias and RMS values of the sample streams.

The SpectrumAnalyzer can be used to monitor the spectrum to detect e.g. jamming. This feature is not yet documented here.

Parameter	Description	Unit	Default / Related Values	Notes
Description	A string describing the current configuration for the spectrum analyzer. The string has no influence on the processing.	-		
FFTOrder	FFT order is used to estimate the signal power spectral density (PSD). The number of points of the FFT is given by 2^{FFTOrder} .	-		
MaxSegments	Number of FFTs which are used to calculate the averaged spectrum for Welch's method.	-		
MonitoringAtAll	Not implemented; choose 0.	-		
SpectrumCalculationRate	Each 'SpectrumCalculationRate' packets of IF samples, the spectrum is calculated. The higher the value, the lower the processing load is.	-		