Errata and Updates, 2-3Q 2022

1 Overview **1**

1.1 Ratings and Module Types 1

1.2 Source Code Notation 1

2 Errata **2**

2.1 Errata Ratings 2

2.2 Errata Table 2

3 Updates **5**

3.1 Update Ratings 5

3.2 Update Table 5

4 Current Software Versions **10**

1 Overview

Signalogic software errata and updates are published twice per year (semi-annual). Errata include bug fixes, workarounds, and notes for any known issues that remain unresolved or scheduled to be resolved. Updates include improvements, new features, and notes for related features scheduled to be implemented.

*Note – this document is a summary of important and key changes. For example, small changes to API definitions that expand usage but are backwards compatible with prior releases may not be documented here. Definitive and detailed errata and update information about **all changes** are given by comments in source code files located on the signalogic/SigSRF_SDK github page.*

Note – some errata and updates earlier than 2-3Q 2022 may be included for reference, due to their importance, even if previously published.

1.1 Ratings and Module Types

Signalogic software errata are rated with Severity Levels of Low, Medium, High, and Critical, and software updates are rated with Functional Levels of Low, Medium, and High.

For example, errata rated with a Severity Level of Critical indicates that if encountered by your application, then without installing updated software your application is likely to crash.

As another example, an update rated with a Functional Level of High indicates that if utilized by your application, then the update is likely to have a major, positive impact on your application and installing updated software is necessary to take advantage.

In the tables below, Module specifies the build-time binary name, and Type specifies the module type, either library (L) or application (A). For codecs Module may describe the codec functionality, and the actual binary name may be given in the Description.

1.2 Source Code Notation

In the itemized table Descriptions below, function / API names are denoted as name(); i.e. with empty parentheses. Constant, type, struct, and flag definitions are given in upper case, for example TERMINATION_INFO. In some cases, source code filenames are given, for example .c and .cpp for C/C++ files, .h for header files, or no extension for a Makefile.

2 Errata

Errata Severity Level rating explanations and itemized errata list are given below.

2.1 Errata Ratings

Errata Severity Level ratings are explained in the table below.

Severity Level	Description
Critical	Bugs rated with a Severity Level of Critical, when encountered by Signalogic modules or your application using Signalogic modules, can possibly cause a crash. Possible crash types include segfaults, illegal instructions, and illegal memory or I/O addresses
High	Bugs rated with a Severity Level of High, when encountered by Signalogic modules or your application using Signalogic modules, are likely to cause an incorrect result but cannot cause a crash
Medium	A Severity Level of Medium is similar to High, but applies to infrequently used functionality and is less likely to be encountered
Low	A Severity Level of Low is similar to Medium, but has a workaround straightforward to implement

2.2 Errata Table

Errata are listed in the table below.

Module	Type	Severity Level	Date	Description	Workaround / Notes
pktlib	L	High	Feb 2021	Fix bug in DSGetPacketInfo() when calculating total packet length for IPv6 packets with extension headers	No workaround exists for IPv6 packets containing extension headers
pktlib	L	Low	Apr 2021	Fix bug in DSReadPcapRecord() when pcap has VLAN headers	Use Wireshark to strip or replace VLAN headers in the pcap beforehand

pktlib	L	High	Apr 2021	Fix bug in DSGetPacketInfo() when calculating total packet length for IPv6 packets with extension headers	None
alglib	L	Medium	Mar 2022	Fix bug in DSConvertFs() sampling rate conversion for multichannel input data	Call DSConvertFs() on each channel separately
EVS encoder	L	Critical	Sep 2022	Fix bug with encoder initialization of AMR-WB IO mode bitrates. Add function get_encoder_mode() in evsenc_sig_ialg.c to correctly initialize flag “Opt_AMR_WB” based on user-supplied bitrate. The correction is in the encoder library libevsenc_sig.x86_gcc4.6.so or libevsenc_sig.x86_gcc11.3.so	Initialize EVS encoder to a non AMR-WB IO mode bitrate, then switch bitrates after at least one encoder frame *Note - the bug may or may not cause a segfault (SIGSEGV). A segfault occurred in about 65% of lab tests*
EVS decoder	L	Critical	Sep 2022	Fix segfault when decoder encountered an AMR-WB IO mode bitrate after stream start (i.e. bitrate switching) The segfault was in the common library libevscom_sig.x86_gcc4.6.so or libevscom_sig.x86_gcc11.3.so in get_evs_bitrate() in bitstream.c	No workaround exists for coded bitstreams containing AMR-WB IO mode packets *Note – a segfault (SIGSEGV) is highly likely. There were no lab tests where a segfault did not occur*
voplib	L	Medium	Sep 2022	Fix bug in DSGetPayloadInfo() when handling payload sizes 6 and 7, including ambiguous case between EVS primary mode 2.8 kbps payload and AMR-WB IO mode SID packets (CMR + ToC + 5-byte SID)	None

hwlib (DirectCore)	L	Critical	Sep 2022	Fix bug in detection of rdtscp instruction support. For VMs in particular, depending on their configuration, the underlying hardware may support rdtscp but the VM configuration may not *Note – the bug may cause an Illegal Instruction (SIGILL) fault*	None without changing either (i) platform / CPU type or (ii) VM configuration
mediaTest	A	Medium	Apr 2021	Fix bug in AMR test sequence when encoding audio to intermediate .amr or .awb file, then decoding the intermediate file to audio	This bug did not affect back-to-back encode/decode (i.e. audio to audio without intermediate file) test sequences
mediaTest	A	Medium	Apr 2021	Fix bug in G726 uncompressed vs. compressed mode	All bitrates retested

3 Updates

Update Functional Level ratings explanation and itemized update list are given below.

3.1 Update Ratings

Update Functional Level ratings are explained in the table below.

Functional Level	Description
High	Updates with a Functional Level of High are likely to make a major, positive impact on your application in terms of performance, energy savings, or readability and maintenance, and installing updated software is necessary to take advantage
Medium	A Functional Level of Medium is similar to High, but applies to functionality less likely to be encountered
Low	Updates with a Functional Level of Low have no functional impact, such as removing compiler warnings, event log information or warning messages, or statistics improvements

3.2 Update Table

Updates are listed in the table below.

Module	Type	Functional Level	Date	Description
derlib	L	Medium	Apr 2021	Create TCP/IP encapsulated media packet decoder *Note – currently supports DER and BER encapsulated data formats*
pktlib	L	High	Apr 2021	Implement input .pcapng file format support *Note – write to pcapng format not currently supported*
pktlib	L	Medium	Dec 2021	Add TCP/IP capability to DSFormatPacket(), enabled by DS_FMT_PKT_TCPIP flag in pktlib.h

pktlib	L	Medium	Dec 2021	Add DS_OPEN_PCAP_RESET flag to instruct DSOpenPcap() to reset an existing (already open) pcap to start of first packet record. This improves performance slightly for high capacity packet impairment testing using repeating pcaps
inferlib	L	Medium	Jan 2021	Integrate inferlib (inference library) into mediaMin and mediaTest release packages, publish inferlib API *Note – inferlib is currently used by ASR (Automatic Speech Recognition) functionality supported by mediaMin*
alglib	L	Medium	Feb 2022	Add user-defined filter params (pFilt and filt_len), add uFlags param and DS_FSCONV_xxx flags, change DSConvertFs() pData and pDelay params to void* to support both fixed-point and floating-point filter coefficients and data
hwlib (DirectCore)	L	Medium	Feb 2022	Add optional HFILE* pFilelibHandle param to DSLoadDataFile() API. If the user supplies this param (non-NULL) the file is kept open after open + write operations and the file handle is returned in pFilelibHandle. This allows filelib DSxx filelib API calls to be used directly by user apps, and mixed/matched with DSLoadDataFile() calls
pktlib	L	Medium	Mar 2022	Support ARP and LLC frames and ICMPv6 extension type
voplib	L	High	Mar 2022	Modify DSCodecEncode() and DSCodecDecode() APIs to accept pointers to arrays of codec handles, to support multichannel processing; e.g. N-channel RTP payloads and N-channel wav files
codecs	L	High	Apr 2022	Improve performance approximately 20% using gcc/g++ 11.x vs 4.6 previously *Note – separate shared object libs are provided for v4.6 and v11.3*

pktlib	L	Medium	Sep 2022	1) Implement flag to disable dormant session flush due to customer need for sessions using identical SSRCs (see TERM_DISABLE_DORMANT_SESSION_DETECTION flag in shared_include/session.h) 2) Implement dormant session minimum time threshold *Note – packet logging, including drops and timestamp matching, not yet updated to handle duplicated SSRCs*
pktlib	L	Low	Sep 2022	Adjust event log warning in DSOpenPcap() for link layer lengths other than supported lengths. Supported lengths currently include standard Ethernet (14 or 12 bytes, depending on OS type), Linux “cooked capture” encapsulation (16 bytes), and Raw, Raw IPv4, and Raw IPv6 (0 bytes)
pktlib	L	High	Oct 2022	Improve jitter buffer auto-adjusting dynamic delay
pktlib	L	Low	Oct 2022	Enhance session stats to show all detected bitrates. As an example: Sessions (hSession/ch/codec/bitrate[,ch...]) 0(grp owner)/0/EVS/13200(1750,2400,6600,8850,12650,24400) 1/2/EVS/24400(2400,13200),8(1750,6600,8850,12650) 2/4/EVS/13200(1750,2400,6600,8850,12650,24400) 3/6/EVS/24400(2400,13200) Detected bitrates include SID rates
pktlib	L	Low	Oct 2022	Enhance session stats to itemize SID and non-SID packet counts. As an example: Input (ch/pkts) 0/1103 2/285 8/854 4/1105 6/283 9/855, SIDs 0/112 2/21 8/110 4/112 6/21 9/110, RFC7198 duplicates 0/0 2/0 8/0 4/0 6/0 9/0, bursts 0/0 2/0 8/0 4/0 6/0 9/0

voplib	L	Medium	Sep 2022	Consolidate several separate APIs into DSGetCodecInfo() and define the following flags for the uFlags param: DS_CODEC_INFO_NAME DS_CODEC_INFO_RAW_FRAMESIZE DS_CODEC_INFO_CODED_FRAMESIZE DS_CODEC_INFO_BITRATE DS_CODEC_INFO_SAMPLERATE DS_CODEC_INFO_PTIME DS_CODEC_INFO_VOICE_ATTR_SAMPLERATE DS_CODEC_INFO_BITRATE_TO_INDEX DS_CODEC_INFO_INDEX_TO_BITRATE DS_CODEC_INFO_PAYLOAD_SHIFT DS_CODEC_INFO_BITRATE_CODE Detailed explanations for these flags are in voplib.h
voplib	L	Medium	Sep 2022	Implement "payload shift" handling in DSCodecDecode(), with maximum codec payload shift +/- 7 bits and controllable by “filter flags” that specify which payloads to shift (defined for the TERMINATION_INFO struct in shared_include/session.h)
mediaMin	A	Medium	Mar 2021	Implement SDP info handling to modify and/or override parameters for dynamically created sessions. Both .sdp file command line input and TCP/IP SIP Invite packets supported *Note – SAP/SDP packet support scheduled for 1Q23*
mediaMin	A	Medium	Oct 2022	Accept command line config file input for dynamic sessions (was previously supported only for static sessions). This allows per-session and per-codec configuration for dynamically created sessions
mediaTest	A	Medium	Jan 2022	Enable -Rn and -Dn command line options with same usage and functionality as mediaMin
mediaTest	A	Medium	Aug 2022	Add support for .cod to .pcap pass-thru case (i.e. converting encoded bitstream files to pcaps)
hello_codec	A	Low	Jul 2022	Basic reference example and source code showing how to use SigSRF codecs. The example encodes and decodes an audio data array and saves the result to wav file

All	N/A	Low	Apr-Oct 2022	Build and test all functionality with gcc/g++ versions 7 thru 11, remove compiler warnings
All	N/A	Medium	Apr-Oct 2022	Test all functionality on CentOS 7 and 8, Debian 12

4 Current Software Versions

Below are current software module version numbers.

Module	Type	Version
hwlib (DirectCore)	L	4.1.2
pktlib	L	3.3.0
streamlib	L	1.9.1
voplib	L	1.5.1
derlib	L	1.2.0
inferlib	L	1.2.0
alglib	L	1.2.5
diaglib	L	1.5.0
mediaMin	A	3.2.1
mediaTest	A	2.9.6